

Sascha Kersken

openSUSE 11.2

Das umfassende Handbuch

- Installation, Anwendung, Administration
- KDE, GNOME, OpenOffice.org, Shell, Netzwerke, Dateisysteme
- System-Automatisierung, NFS, Samba, OpenLDAP u.v.m.



openSUSE 11.2 auf DVD
Für 32-/64-Bit-Systeme

Galileo Computing



11.2

openSUSE
Das umfassende Handbuch

inkl. DVD
mit openSUSE 11.2

Liebe Leserin, lieber Leser,

openSUSE wird nicht nur auf vielen privaten PCs erfolgreich eingesetzt, sondern auch in vielen Behörden und Firmen. Es ist einfach zu installieren und zu administrieren, bietet aber dennoch alle Möglichkeiten eines professionellen Betriebssystems. Wenn Sie ein vollständiges Handbuch zu allen Fragen im Umgang mit openSUSE suchen, sind Sie mit dem Buch von Sascha Kersken auf der sicheren Seite. Er ist Anwender der ersten Stunde, überzeugter Anhänger von Open-Source-Software, Fachbuchautor und erfahren in Schulungen. Er sagt von sich selbst, dass openSUSE sein favorisiertes Betriebssystem ist.

Ob Sie nun einen umfassenden Einstieg suchen oder sich mit fortgeschrittenen Themen befassen möchten – unser Autor Sascha Kersken vermittelt Ihnen alles Wissenswerte zu openSUSE 11. So dient Ihnen das Buch als erste Orientierung beim Installationsvorgang, als detailliertes Handbuch für Konfiguration und Nutzerverwaltung, als umfassender Ratgeber bei den zahlreichen Anwendungspaketen, wie z.B. OpenOffice.org 3 sowie als sachkundiges Profi-Buch für Themen wie Netzwerk-Konfiguration, Internet-Serverdienste, Kernel-Kompilierung und Shell-Skripting. Um Ihnen den sofortigen Ein- oder Umstieg zu ermöglichen, finden Sie in diesem Buch die vollständige openSUSE-Distribution als 32- und 64-Bit-Version auf 2 DVDs.

Dieses Buch wurde mit großer Sorgfalt geschrieben, begutachtet, lektoriert und produziert. Sollte dennoch etwas nicht so funktionieren, wie Sie es erwarten, dann scheuen Sie sich nicht, sich mit mir in Verbindung zu setzen. Ihre freundlichen Anregungen und Fragen sind jederzeit willkommen.

Viel Vergnügen beim Lesen!

Stephan Mattescheck

Lektorat Galileo Computing

stephan.mattescheck@galileo-press.de

www.galileocomputing.de

Galileo Press · Rheinwerkallee 4 · 53227 Bonn

Auf einen Blick

1	Einführung	21
2	openSUSE installieren	53
3	Systemrundgang und Grundkonfiguration	103
4	Mit der Shell arbeiten	149
5	Grafische Oberflächen	207
6	Wichtige Hardware konfigurieren	253
7	OpenOffice.org	273
8	Desktop-Software	323
9	Datenträger und Dateisysteme	369
10	Klassische Texteditoren	405
11	Textbasierte Auszeichnungssprachen	439
12	Der Datenbankserver MySQL	525
13	Netzwerkgrundlagen	575
14	Datei- und Verzeichnisdienstserver	655
15	LAMP (Linux, Apache, MySQL und PHP)	691
16	Weitere Internet-Serverdienste	761
17	System-Automatisierung	783
18	Hinter den Kulissen	839
19	Referenz der Shell- und Systembefehle	889
A	Glossar	939
B	Literaturtipps	953

Der Name Galileo Press geht auf den italienischen Mathematiker und Philosophen Galileo Galilei (1564–1642) zurück. Er gilt als Gründungsfigur der neuzeitlichen Wissenschaft und wurde berühmt als Verfechter des modernen, heliozentrischen Weltbilds. Legendär ist sein Ausspruch *Eppur se muove* (Und sie bewegt sich doch). Das Emblem von Galileo Press ist der Jupiter, umkreist von den vier Galileischen Monden. Galilei entdeckte die nach ihm benannten Monde 1610.

Gerne stehen wir Ihnen mit Rat und Tat zur Seite:

stephan.mattesccheck@galileo-press.de bei Fragen und Anmerkungen zum Inhalt des Buches
service@galileo-press.de für versandkostenfreie Bestellungen und Reklamationen
britta.behrens@galileo-press.de für Rezensionen- und Schulungsexemplare

Lektorat Stephan Mattesccheck, Sebastian Kestel

Korrektorat Barbara Decker

Typografie und Layout Vera Brauner

Einbandgestaltung Barbara Thoben, Köln

Herstellung Lissy Hamann

Satz III-satz, Husby

Druck und Bindung Bercker Graphischer Betrieb, Kevelaer

Dieses Buch wurde gesetzt aus der Linotype Syntax Serif (9,25/13,25 pt) in FrameMaker.
Gedruckt wurde es auf chlorfrei gebleichtem Offsetpapier.

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

ISBN 978-3-8362-1497-1

© Galileo Press, Bonn 2010

3., aktualisierte und erweiterte Auflage 2010

Das vorliegende Werk ist in all seinen Teilen urheberrechtlich geschützt. Alle Rechte vorbehalten, insbesondere das Recht der Übersetzung, des Vortrags, der Reproduktion, der Vervielfältigung auf fotomechanischem oder anderen Wegen und der Speicherung in elektronischen Medien. Ungeachtet der Sorgfalt, die auf die Erstellung von Text, Abbildungen und Programmen verwendet wurde, können weder Verlag noch Autor, Herausgeber oder Übersetzer für mögliche Fehler und deren Folgen eine juristische Verantwortung oder irgendeine Haftung übernehmen. Die in diesem Werk wiedergegebenen Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. können auch ohne besondere Kennzeichnung Marken sein und als solche den gesetzlichen Bestimmungen unterliegen.

Inhalt

Vorwort	15
---------------	----

1 Einführung 21

1.1 Die Entwicklung von UNIX und Linux	23
1.1.1 Die Vorgeschichte der Betriebssysteme	25
1.1.2 Die Geschichte von UNIX	27
1.1.3 Freie Software und Open-Source-Software	28
1.1.4 Die Entwicklung von Linux	32
1.1.5 Streit und Schwierigkeiten um Linux	42
1.2 openSUSE	47
1.2.1 Die Geschichte von SUSE Linux	47
1.2.2 openSUSE	48
1.3 Zusammenfassung	51

2 openSUSE installieren 53

2.1 Vorbehandlung eines bereits installierten Windows-Systems	54
2.2 Vorbereitung der Installation	58
2.2.1 PC und Boot-Datenträger aufeinander abstimmen	59
2.2.2 Die Installation einleiten	59
2.2.3 Lizenzvereinbarung	63
2.2.4 Neuinstallation oder Aktualisierung?	64
2.2.5 Zeitzone, Datum und Uhrzeit einstellen	64
2.2.6 Desktop auswählen	65
2.2.7 Die Partitionierung	66
2.2.8 Einrichten des ersten Standardbenutzers	80
2.2.9 Das root-Passwort festlegen	83
2.3 Softwareauswahl und Installation	84
2.3.1 Software-Auswahl	84
2.3.2 Länderspezifische Einstellungen	92
2.3.3 Weitere Einstellungen	94
2.4 Erstkonfiguration nach der Installation	94
2.4.1 Host- und Domainname	95
2.4.2 Netzwerkkonfiguration	96
2.4.3 Abschluss der Installation	101
2.5 Zusammenfassung	102

3	Systemrundgang und Grundkonfiguration	103
3.1	Das System kennenlernen	103
3.1.1	Booten und Anmeldung	103
3.1.2	Erste Schritte mit KDE	106
3.1.3	Erste Schritte mit GNOME	113
3.2	Hilfe und Online-Handbücher	118
3.2.1	Manpages	118
3.2.2	GNU info	121
3.3	Grundkonfiguration	123
3.3.1	YaST starten und einsetzen	123
3.3.2	Benutzer- und Gruppenverwaltung mit YaST	127
3.3.3	Software nachinstallieren, deinstallieren und aktualisieren	136
3.3.4	Programme automatisch starten	142
3.4	Zusammenfassung	147
4	Mit der Shell arbeiten	149
4.1	Wichtige Grundbegriffe	150
4.1.1	Dateisysteme und Verzeichnisbaum	150
4.1.2	Zugriffsrechte	155
4.2	Die Shell aufrufen und anwenden	156
4.2.1	Virtuelle Terminals und Terminalfenster	156
4.2.2	Bedeutung der Shell	159
4.2.3	Grundfunktionen der Shell	160
4.2.4	Ein-/Ausgabeumleitung und Pipes	166
4.3	Die wichtigen Systembefehle	176
4.3.1	Vorbemerkungen zu Shell-Kommandos	176
4.3.2	Arbeiten mit Dateien und Verzeichnissen	178
4.3.3	Benutzerkonten und Zugriffsrechte modifizieren	187
4.3.4	Kommandos zur Textanzeige und -manipulation	193
4.3.5	Systeminformation und -verwaltung	200
4.4	Zusammenfassung	205
5	Grafische Oberflächen	207
5.1	Das X Window System	207
5.1.1	Über Grafikhardware	208
5.1.2	Den X-Server konfigurieren	212
5.2	KDE und GNOME – die Desktops	219

5.3	KDE	220
5.3.1	KDE 4 konfigurieren	221
5.3.2	Der Dateimanager Dolphin	234
5.3.3	Die Terminalemulation KDE-Konsole	235
5.3.4	Weitere KDE-Anwendungen	237
5.4	GNOME	238
5.4.1	GNOME konfigurieren	239
5.4.2	Der GNOME-Dateimanager Nautilus	244
5.4.3	Das GNOME Terminal	246
5.4.4	GNOME-Anwendungen	247
5.5	Weitere Window-Manager	247
5.5.1	FVWM – der Klassiker	247
5.5.2	WindowMaker	248
5.5.3	IceWM	249
5.6	Zusammenfassung	250

6 Wichtige Hardware konfigurieren 253

6.1	Kernel-Module	253
6.1.1	Module laden und entladen	253
6.1.2	Einen eigenen Kernel kompilieren	255
6.2	Drucker	259
6.2.1	Druckerarten	259
6.2.2	CUPS	260
6.2.3	Drucker mit YaST konfigurieren	262
6.2.4	Drucken	264
6.3	Weitere Hardware mit YaST einrichten	265
6.3.1	Soundkarten	266
6.3.2	Scanner	268
6.4	Zusammenfassung	270

7 OpenOffice.org 273

7.1	Textverarbeitung mit Writer	274
7.1.1	Grundlagen der Typografie	275
7.1.2	Grundlegende Textformatierung	294
7.1.3	Formatvorlagen verwenden	303
7.1.4	Weitere Features	304
7.2	Tabellenkalkulation mit Calc	312
7.2.1	Eine Tabelle erstellen	313
7.2.2	Ein Diagramm erstellen	317

7.2.3	Formeln und Funktionen	318
7.3	Zusammenfassung	322

8 Desktop-Software 323

8.1	Bildbearbeitung und Grafik	323
8.1.1	Theoretische Grundlagen	323
8.1.2	Bildbearbeitung mit GIMP	336
8.1.3	Das Vektorzeichenprogramm Inkscape	359
8.2	Der Webbrowser Firefox	364
8.2.1	Firefox im Überblick	364
8.2.2	Firefox-Add-ons	367
8.3	Zusammenfassung	367

9 Datenträger und Dateisysteme 369

9.1	Festplatten	369
9.1.1	Partitionierung und Dateisysteme	371
9.1.2	mount, umount und /etc/fstab	379
9.1.3	Weitere Dateisystem-Kommandos	382
9.1.4	SoftRAID	383
9.1.5	LVM	384
9.2	CDs, DVDs und Blu-ray Discs	384
9.2.1	Technische Grundlagen	384
9.2.2	CDs und DVDs brennen	389
9.3	Archivdateien und virtuelle Dateisysteme	392
9.3.1	Archivdateien verwenden	393
9.3.2	Loop-Devices	396
9.3.3	Das /proc-Dateisystem	397
9.4	Die Boot-Konfiguration	398
9.4.1	Den Bootloader konfigurieren	398
9.4.2	Die Rettungskonsole	401
9.5	Zusammenfassung	403

10 Klassische Texteditoren 405

10.1	vi und Vim	405
10.1.1	Navigation im Text	407
10.1.2	Textsuchbefehle	409
10.1.3	Sprungmarken	411

10.1.4	Löschen, Kopieren und Einfügen	412
10.1.5	Textobjekte	413
10.1.6	Der visuelle Modus	414
10.1.7	Befehle wiederholen, rückgängig machen und wiederherstellen	415
10.1.8	Datei- und Fensterverwaltung	416
10.1.9	Weitere Features	419
10.1.10	Vim-Befehle im Überblick	420
10.2	Emacs	425
10.2.1	Dateien, Buffer und Fenster	427
10.2.2	Die Emacs-Modi	428
10.2.3	Navigation und einfache Textbearbeitung	430
10.2.4	Spezielle Funktionen	432
10.2.5	Emacs-Befehle im Überblick	435
10.3	Zusammenfassung	438

11 Textbasierte Auszeichnungssprachen 439

11.1	XML	439
11.1.1	Der Aufbau von XML-Dokumenten	441
11.1.2	DTDs und XML Schema	450
11.1.3	XSLT	467
11.2	HTML, XHTML und CSS	472
11.2.1	HTML und XHTML	472
11.2.2	Cascading Stylesheets (CSS)	499
11.2.3	Der Webseiten-Editor Quanta	510
11.3	LaTeX	511
11.3.1	Erstes Beispiel	512
11.3.2	LaTeX-Basisformate	514
11.4	Zusammenfassung	524

12 Der Datenbankserver MySQL 525

12.1	Relationale Datenbanken	525
12.2	Installation und Inbetriebnahme	530
12.3	Eigenschaften von MySQL	531
12.3.1	MySQL 5.4 manuell installieren	532
12.4	Der Kommandozeilenclient mysql	534
12.5	MySQL-Datenbanken erstellen und verwalten	538
12.5.1	Datenbanken und Tabellen erstellen	539

12.5.2	Daten einfügen	544
12.5.3	MySQL-Datentypen	545
12.5.4	Auswahlabfragen	549
12.5.5	Änderungsabfragen	555
12.5.6	Löschabfragen	557
12.6	MySQL-Administration	557
12.6.1	mysqladmin	557
12.6.2	Benutzerverwaltung	558
12.6.3	Import und Export von Daten, Backups	564
12.6.4	Konfigurationsdateien	567
12.6.5	Logdateien	568
12.6.6	Replikation	570
12.7	Zusammenfassung	572

13 Netzwerkgrundlagen 575

13.1	TCP/IP-Grundlagen	575
13.1.1	Das Internet-Schichtenmodell	576
13.1.2	Das Internet Protocol (IP)	578
13.1.3	Transportprotokolle	588
13.2	Lokale Netzwerkschnittstellen konfigurieren	592
13.2.1	Ethernet	592
13.2.2	Wireless LAN	602
13.3	Wählverbindungen ins Internet	606
13.3.1	PPP	607
13.3.2	Internetzugang über DSL	608
13.3.3	Internetzugang per Modem (analoge Telefonleitung)	612
13.3.4	ISDN	616
13.4	Das Domain Name System (DNS)	621
13.4.1	Das DNS-Konzept	622
13.4.2	Der DNS-Server BIND	624
13.5	TCP/IP-Diagnose und -Fehlersuche	630
13.5.1	ping	630
13.5.2	traceroute	631
13.5.3	netstat	632
13.5.4	nslookup	633
13.5.5	telnet	634
13.5.6	tcpdump	635
13.6	Grundlagen der Netzwerksicherheit	637
13.6.1	Sicherheitsrisiken	637

13.6.2	Allgemeine Gegenmaßnahmen	640
13.6.3	Grundbegriffe der Kryptografie	643
13.6.4	Die openSUSE-Firewall konfigurieren	646
13.7	Zusammenfassung	652

14 Datei- und Verzeichnisdienstserver 655

14.1	Dateiserver	655
14.1.1	NFS	655
14.1.2	Samba	661
14.2	Verzeichnisdienste	666
14.2.1	NIS	666
14.2.2	OpenLDAP	673
14.3	Weitere Dienste	679
14.3.1	DHCP-Server	679
14.3.2	Proxyserver	682
14.3.3	SSH	685
14.3.4	Remote-Verwaltung (VNC)	687
14.4	Zusammenfassung	689

15 LAMP (Linux, Apache, MySQL und PHP) 691

15.1	HTTP-Grundlagen	691
15.1.1	Die HTTP-Kommunikation	691
15.1.2	HTTP-Statuscodes	696
15.1.3	HTTP-Header	701
15.2	Der Webserver Apache 2	706
15.2.1	Über Apache 2	707
15.2.2	Installation mit YaST	708
15.2.3	Apache-Konfigurationsmöglichkeiten in openSUSE	709
15.2.4	Apache manuell installieren	718
15.2.5	Apache steuern	729
15.2.6	Apache-Konfiguration	730
15.2.7	PHP manuell installieren	739
15.2.8	Die PHP-Konfigurationsdatei php.ini	741
15.2.9	php/MyAdmin installieren	744
15.3	LAMP-Webanwendungen	745
15.3.1	Formulardaten auslesen	747
15.3.2	Datei-Uploads	748
15.3.3	Sessions	750

15.3.4	Cookies	750
15.3.5	Zugriff auf MySQL-Datenbanken	752
15.4	Zusammenfassung	759

16 Weitere Internet-Serverdienste 761

16.1	xinetd	761
16.1.1	Der klassische inetd	762
16.1.2	xinetd-Konfigurationsdateien	763
16.2	FTP	767
16.2.1	FTP-Clients	767
16.2.2	Den FTP-Server vsftpd einrichten und verwenden	771
16.3	E-Mail	772
16.3.1	Theoretische Grundlagen	773
16.3.2	Lokale Mail-Konfiguration mit Postfix	779
16.4	Zusammenfassung	782

17 System-Automatisierung 783

17.1	Nützliche Helfer	783
17.1.1	Aliasse	783
17.1.2	Cronjobs	785
17.1.3	Logdateien und syslog	788
17.1.4	sudo – Einzelne Kommandos als root ausführen	789
17.2	Shell-Skripte	790
17.2.1	Eigenschaften von Shell-Skripten	790
17.2.2	Das erste Beispiel	791
17.2.3	Variablen und Ausdrücke	794
17.2.4	Die Arithmetik-Sprache bc	796
17.2.5	Fallentscheidungen	797
17.2.6	Schleifen	803
17.2.7	Funktionen	808
17.2.8	sed und awk	810
17.3	Die Alternative Perl	812
17.3.1	Das erste Beispiel	813
17.3.2	Elemente der Sprache Perl	815
17.3.3	In Perl mit regulären Ausdrücken arbeiten	826
17.3.4	Administrationsbeispiel: Webserver-Logdateien auswerten	832
17.4	Zusammenfassung	837

18	Hinter den Kulissen	839
18.1	Die zugrundeliegende Hardware	839
18.1.1	Aufbau und Aufgaben des Prozessors	841
18.1.2	Der Arbeitsspeicher	849
18.1.3	Bus- und Anschlussysteme	851
18.2	Die Linux-Systemarchitektur	861
18.2.1	Allgemeiner Aufbau des Systems	861
18.2.2	Prozessverwaltung	866
18.2.3	Threads	874
18.2.4	Speicherverwaltung	875
18.2.5	Die Implementierung des TCP/IP-Stacks	876
18.2.6	Netzwerkprogrammierung – ein praktisches Beispiel	884
18.3	Zusammenfassung	888
19	Referenz der Shell- und Systembefehle	889
19.1	ls – ein Kommando im Detail	889
19.2	Alphabetische Kommandoreferenz	894
	Anhang	939
A	Glossar	939
B	Literaturtipps	953
B.1	Allgemeines zu Linux (openSUSE und allgemein), UNIX & Co.	953
B.2	Shell, Scripting und Programmierung	953
B.3	OpenOffice.org und andere GUI-Anwendungen	955
B.4	Editoren und Auszeichnungssprachen	955
B.5	Netzwerk- und Systemadministration	956
B.6	LAMP	956
	Index	959

*Lass den Anfang mit dem Ende
Sich in Eins zusammenziehn!
– Johann Wolfgang von Goethe*

Vorwort

Als das Konzept für die erste Auflage dieses Buches entstand, war SUSE Linux noch eine kommerzielle Linux-Distribution. Zweifellos meine Lieblingsdistribution, die ich zu dem Zeitpunkt bereits seit acht Jahren erfolgreich einsetzte – ansonsten hätte ich mich niemals auf dieses Projekt eingelassen. Aber eben trotzdem nur eine Distribution unter vielen. Umso erfreulicher war es, dass SUSEs Muttergesellschaft Novell sich im August 2005 entschloss, SUSE Linux weiterzuführen und vollständig als Open-Source-Projekt unter dem Namen openSUSE freizugeben.

Mit openSUSE 11 stand Ende Juni 2008 eine neue Hauptversion des Klassikers bereit, und im November 2009 erschien Version 11.2. Sie bietet nach wie vor alles, was man sich von einer modernen Linux-Distribution wünschen kann. Sie enthält Unmengen an Software für beinahe jedes erdenkliche Einsatzgebiet und ist für fast alle Arten von Usern geeignet: von Computereinsteigern über Windows-Umsteiger bis hin zu erfahrenen Entwicklern, Administratoren und anderen Linux-Hackern. Ein besonderes Highlight der 11er-Versionen ist die erheblich beschleunigte Installation; daneben können Sie sich natürlich wie immer über Unmengen von Software in der jeweils neuesten Version freuen.

Zu Linux selbst braucht man eigentlich nicht mehr viel zu sagen (falls es Sie dennoch interessiert, erfahren Sie das Wichtigste in Kapitel 1 dieses Buches, »Einführung«). Es handelt sich um das beliebteste freie Betriebssystem der Welt, und nachdem es die Rechenzentren der Universitäten und die Server der Internet-Provider erobert hat, ist es Zeit, dass es endlich auch auf den Desktops der Büro- und Heimanwender eine größere Verbreitung findet.

openSUSE enthält bereits ab Werk alles, was Sie dazu brauchen: ein Office-Paket, das locker mit dem Platzhirsch Microsoft Office mithalten kann, eine recht gut ausgebaut Multimedia-Unterstützung und sogar einige unterhaltsame Spiele. Hinzu kommt traditionelle, leistungsfähige Software aus der UNIX-Welt, bei-

spielsweise die mächtigen Texteditoren Vim und Emacs oder das professionelle Textsatzsystem LaTeX. Auch die Serverseite wird erstklassig unterstützt: Die Dateiserver NFS und Samba, der Webserver Apache 2 und die Datenbank MySQL sind ebenfalls mit an Bord. All diese Software und vieles mehr wird in diesem Buch ausführlich beschrieben.

Das Buch im Überblick

Dieses Buch ist eine vollständige gedruckte Einführung und Referenz zu openSUSE 11.2. Auf den beiden beiliegenden DVDs ist zudem die komplette Distribution selbst enthalten, einmal für 32- und einmal für 64-Bit-Systeme. Damit erhalten Sie eine echte Alternative zum Kauf der Original-openSUSE-Box oder auch zum langwierigen Download der Installations-Images.

Die 19 Kapitel des Buches decken ein sehr breites Spektrum ab – von den ersten Schritten mit einem Linux-System über zahlreiche mitgelieferte Anwendungen bis hin zur System- und Netzwerkadministration. Im Einzelnen geht es um Folgendes:

- ▶ Kapitel 1, »Einführung«, beschreibt die Entwicklungsgeschichte der Betriebssysteme UNIX und Linux sowie der Distribution SUSE Linux mitsamt ihrer Open-Source-Weiterentwicklung openSUSE.
- ▶ In Kapitel 2, »openSUSE installieren«, wird Schritt für Schritt ausführlich erläutert, wie Sie das Betriebssystem auf Ihrem Rechner zum Leben erwecken. Dabei werden viele verschiedene Szenarien berücksichtigt, unter anderem der verbreitete Fall einer Parallelinstallation neben einem bestehenden Windows-System oder sonstigen Betriebssystemen.
- ▶ Das Thema von Kapitel 3, »Systemrundgang und Grundkonfiguration«, ist die Inbetriebnahme des fertig installierten Systems: Nach einer ersten kleinen, aber realistischen Praxisaufgabe erfahren Sie etwas über die verschiedenen umfangreichen Hilfesysteme eines gut ausgestatteten Linux-Systems. Anschließend geht es um die nachträgliche Installation von Software und andere grundlegende Konfigurationsaufgaben.
- ▶ Kapitel 4, »Mit der Shell arbeiten«, schildert den Umgang mit der zu Unrecht gefürchteten Kommandozeile. Neben allgemeinen Konzepten lernen Sie viele der wichtigsten Kommandos kennen, sodass Sie bald überaus produktiv damit arbeiten können.
- ▶ In Kapitel 5, »Grafische Oberflächen«, werden die verschiedenen Aspekte der Arbeit mit grafischen Benutzeroberflächen erläutert: Zunächst geht es um die Konfiguration des X Window Servers, der die grundlegende Grafikfunktiona-

lität bereitstellt. Anschließend wird ausführlich beschrieben, wie Sie mit den beiden wichtigsten Linux-Desktops KDE und GNOME arbeiten und sie an Ihre Bedürfnisse anpassen können. Zum Schluss gibt es einen Ausblick auf einige alternative Desktops und Window-Manager.

- ▶ Kapitel 6, »Wichtige Hardware konfigurieren«, kümmert sich um die Inbetriebnahme widerspenstiger Geräte, die während der Systeminstallation nicht automatisch erkannt wurden. Als Erstes lernen Sie das Konzept der Kernel-Module und den praktischen Umgang mit ihnen kennen. Danach erfahren Sie, wie Sie den Kernel bei Bedarf selbst neu kompilieren und installieren können. Zum Schluss geht es um einige konkrete Gerätearten wie Drucker und Soundkarten.
- ▶ Kapitel 7, »OpenOffice.org«, beschäftigt sich mit dem gleichnamigen leistungsfähigen Office-Paket. Besprochen werden die Textverarbeitung Writer und die Tabellenkalkulation Calc.
- ▶ Der Gegenstand von Kapitel 8, »Desktop-Software«, sind diverse weitere Anwendungen für die grafische Benutzeroberfläche. Zuerst erhalten Sie eine Einführung in die theoretischen Grundlagen der Computergrafik sowie die praktische Arbeit mit dem freien Bildbearbeitungsprogramm GIMP und dem Vektorzeichenprogramm Inkscape. Da Webanwendungen immer wichtiger werden und langsam anfangen, den Desktop-Anwendungen den Rang abzulaufen, wird danach kurz der Webbrowser Firefox vorgestellt.
- ▶ In Kapitel 9, »Datenträger und Dateisysteme«, wird eine etwas komplexere Administrationsaufgabe beschrieben: die Verwaltung von Festplatten und anderen Datenträgern. Es geht unter anderem um die Partitionierung, um das Brennen von CDs und DVDs sowie um den Umgang mit Archivdateien und anderen virtuellen Dateisystemen.
- ▶ In Kapitel 10, »Klassische Texteditoren«, wird die Arbeit mit den beiden leistungsfähigen Editoren Vim und Emacs beschrieben. Beide verwenden (völlig verschiedene) Tastaturkürzel für ihre zahlreichen Komfortfunktionen; nach einer gewissen Einarbeitungszeit lässt es sich mit jedem von ihnen schneller und produktiver arbeiten als mit mausbasierten Editoren.
- ▶ Kapitel 11, »Textbasierte Auszeichnungssprachen«, führt Sie in die Welt der unter UNIX/Linux weitverbreiteten Sprachen ein, die die Struktur und das Layout von Dokumenten durch Klartextbefehle formatieren. Im Einzelnen werden XML, (X)HTML und LaTeX besprochen.
- ▶ Kapitel 12, »Der Datenbankserver MySQL«, beschreibt den Aufbau relationaler Datenbanken sowie die Anwendung und Administration des beliebtesten Open-Source-Datenbanksystems.

- ▶ In Kapitel 13, »Netzwerkgrundlagen«, wird nach einigen theoretischen Grundlagen ausführlich erläutert, wie Sie Ihr openSUSE-System in ein Netzwerk einbinden – per Netzwerkkarte, Wireless LAN sowie über Internet-Wählverbindungen. Zum Schluss geht es um das wichtige Thema Netzwerksicherheit.
- ▶ Die in Kapitel 14 beschriebenen »Datei- und Verzeichnisdienstserver« bilden eine wichtige Grundlage moderner Arbeitsumgebungen. Behandelt werden die Dateiserver NFS (Klassiker für UNIX-Netzwerke) und Samba (freier Server für Windows-Netze) sowie die Verzeichnisdienste NIS und OpenLDAP. Abgerundet wird das Kapitel durch einen Blick auf einige andere Serverdienste wie DHCP, Proxy und SSH.
- ▶ Kapitel 15, »LAMP«, behandelt die beliebteste Softwarekombination für den Betrieb von Webanwendungen: ein Linux-System mit dem Webserver Apache 2, der Datenbank MySQL und der Skriptsprache PHP. Neben der ausführlich erläuterten Einrichtung und Konfiguration werden einige einführende Programmierbeispiele gezeigt.
- ▶ Kapitel 16, »Weitere Internet-Serverdienste«, kümmert sich um einige weitere wichtige Server aus dem Internet-Umfeld: `xinetd` als Starter für allerlei »kleinere« Serverdienste, FTP und E-Mail.
- ▶ In Kapitel 17, »System-Automatisierung«, werden einige Wege zur automatisierten Systemadministration beschrieben. Nach Shell-Aliassen als Befehlskürzel und Cronjobs zur regelmäßigen automatischen Ausführung von Aufgaben werden insbesondere Shell-Skripte und die mächtige Alternative Perl beschrieben.
- ▶ Kapitel 18, »Hinter den Kulissen«, beschreibt einige Aspekte der inneren Funktionsweise des Linux-Kernels und seiner Bestandteile, wobei zunächst die Grundlagen der Hardware erläutert werden, auf der das System läuft.
- ▶ Zum Schluss bietet Kapitel 19, »Referenz der Shell- und Systembefehle«, einen schnellen, alphabetischen Überblick über die wichtigsten Kommandos für alle Lebenslagen.

Zusätzlich besitzt das Buch zwei Anhänge:

- ▶ ein Glossar mit kurzen Erläuterungen der wichtigsten Fachbegriffe
- ▶ ein kommentiertes Literaturverzeichnis

Für wen ist dieses Buch geeignet?

Im Prinzip können sowohl Linux-Einsteiger als auch Fortgeschrittene mit diesem Buch produktiv arbeiten. Selbst gestandene Entwickler oder Admins mögen es nützlich finden, um zwischendurch das eine oder andere nachzuschlagen. In der Regel wird keine der genannten Zielgruppen das Buch von vorn bis hinten durchlesen, sondern jeweils gezielt auf die Suche nach bestimmten interessanten Themen gehen.

Ganz ohne Vorkenntnisse geht es allerdings nicht: Sie sollten wissen, was Dateien und Verzeichnisse sind, wie Maus und Tastatur verwendet werden oder wie man einfache (zumindest externe) Peripheriegeräte anschließt. Ob Sie diese Vorkenntnisse unter Linux oder einem anderen Betriebssystem wie Windows oder Mac OS X erworben haben, ist dabei gleichgültig.

Danksagungen

Dieses Buch wäre nicht möglich gewesen ohne die Leistungen und die Mithilfe der folgenden Personen und Gruppen, denen ich von Herzen dafür danke:

- ▶ Linus Torvalds für den ursprünglichen Linux-Kernel und dessen unermüdliche Weiterentwicklung bis heute.
- ▶ Den Tausenden von Freiwilligen, die Linux und den unzähligen Anwendungsprogramme, die unter diesem System laufen, zu ihrer heutigen Leistungsfähigkeit verholfen haben.
- ▶ Der Firma SUSE beziehungsweise Novell für die (meiner Ansicht nach) beste Linux-Distribution der Welt und den Mut, diese als vollständiges Open-Source-Projekt weiterzuführen.
- ▶ Meinem Lektor Stephan Mattescheck, mit dem ich bereits einige erfolgreiche Buchprojekte verwirklicht habe, für den ursprünglichen Vorschlag, dieses Buch zu schreiben, und für die vielfältige Unterstützung.
- ▶ Den Leserinnen und Lesern der ersten beiden Auflagen, die mich sowohl durch den Kauf als auch durch diverse E-Mails mit konstruktiver Kritik und Verbesserungsvorschlägen unterstützt haben.
- ▶ Meiner Familie für alle Formen der Unterstützung, Ermutigung und vor allem für ihre Geduld.

If I have seen further it is by standing on the shoulders of giants.
– Sir Isaac Newton

1 Einführung

Mit openSUSE haben Sie sich für ein stabiles, sicheres und freies (nicht nur kostenloses) Betriebssystem entschieden – herzlichen Glückwunsch! In diesem Kapitel erhalten Sie einen historischen Überblick über die Entwicklung des Konzepts der freien Software sowie der Betriebssysteme UNIX und Linux. Danach geht es konkret um die Distribution openSUSE; hier erfahren Sie auch, wie Sie gegebenenfalls eine neuere Version als die auf der Buch-DVD herunterladen können.

Linux Survival Guide

Dieses Buch legt einigen Wert darauf, nicht nur praktische Anleitungen zu geben, sondern auch grundlegende Konzepte eingehend zu erläutern. Auf diese Weise kommen Sie auch mit den Teilen Ihres openSUSE-Systems besser zurecht, die hier nicht behandelt werden können. Zudem können Sie viele der in diesem Buch vermittelten Kenntnisse auch auf anderen Systemen der großen UNIX-Familie anwenden.

Wenngleich alle wichtigen Linux-Funktionen und -Konzepte in den jeweils passenden Kapiteln ausführlich diskutiert werden, sollten Sie über einige Dinge doch vorab Bescheid wissen. Dieser kleine Survival Guide richtet sich vornehmlich an bisherige Windows-Benutzer, die beschlossen haben, Linux zu ihrem neuen (zusätzlichen) Betriebssystem zu machen, und beantwortet in Form einer FAQ (Frequently Asked Questions) drei wichtige Fragen.

► *Wo findet man unter Linux die verschiedenen Laufwerke?*

Das ist wahrscheinlich die wichtigste Frage, die sich Windows-Benutzer stellen. Unter Windows wird jedes Laufwerk (genauer gesagt jede Partition) durch einen Buchstaben wie A: (Diskettenlaufwerk), C: (erste Festplatte) und so weiter bezeichnet. Ein Dateipfad beginnt mit diesem Buchstaben und enthält dann eine Verzeichnishierarchie wie *C:\Dokumente und Einstellungen\Sascha\Eigene Dateien\openSUSE\Einfuehrung.txt*.

UNIX-Systeme kennen keine Laufwerksbuchstaben. Hier werden alle Laufwerke und Datenträger an benutzerdefinierten Stellen in einen einzigen, gro-

ßen Verzeichnisbaum eingehängt (der Fachausdruck lautet »gemountet«). Jeder Dateipfad beginnt daher mit der Wurzel /; darauf folgt ebenfalls eine Verzeichnishierarchie. Hier ein Beispiel, das analog zu dem Windows-Pfad wäre: `/home/sascha/opensuse/einfuehrung.txt`.

Die Laufwerke selbst (und alle anderen Hardwaregeräte) werden ebenfalls im Verzeichnisbaum abgebildet, und zwar im Verzeichnis `/dev` (Abkürzung für »devices«) – nicht umsonst heißt es:

Unter UNIX ist alles eine Datei.

Die erste Festplatte heißt beispielsweise `/dev/hda`. An dieser Stelle im Verzeichnisbaum können Sie allerdings nicht darauf zugreifen – als Geräte liefern sie keine geordneten Dateien und Verzeichnisse, sondern einfache Datenströme. Erst das besagte Mounten macht sie als sogenannte Dateisysteme verfügbar. Häufig wird eine Festplatte als Wurzel / gemountet und eine zweite unter `/home`. Das Verzeichnis `/home` besitzt eine besondere Bedeutung, weil es die Home-Verzeichnisse der einzelnen Benutzer enthält – standardmäßig ist `/home/Benutzername` der einzige Ort, wo Sie als gewöhnlicher Benutzer Dateien speichern und verändern dürfen.

► *Muss man unter Linux mit der Konsole arbeiten?*

Mit der Konsole ist die von vielen Einsteigern (zu Unrecht!) gefürchtete Kommandozeilenoberfläche gemeint. Und, gleich vorweg: Nein, Sie *müssen* unter Linux nicht damit arbeiten, auch wenn dies eines der am schwersten auszurrottenden Vorurteile gegenüber Unix-artigen Betriebssystemen ist. Genau wie Windows oder Mac OS X sind auch moderne Linux-Distributionen mit einer leistungsfähigen grafischen Benutzeroberfläche ausgestattet. In gewisser Weise haben Sie es unter Linux sogar noch besser: Sie sind nicht auf die eine grafische Oberfläche angewiesen, die der Hersteller Ihres Systems mitzuliefern geruht, sondern können sich zwischen mehreren entscheiden, die unterschiedliche Stärken besitzen (zum Beispiel besonders viele Features gegenüber Ressourcenschonung für ältere Rechner). openSUSE ist ab Werk mit den beiden »großen« Desktops KDE und GNOME ausgestattet; daneben werden einige unbekanntere, größtenteils schlankere grafische Oberflächen mitgeliefert. Sie werden in Kapitel 5 ausführlich behandelt.

Andererseits: Es schadet überhaupt nichts, neben der Arbeit mit einer mächtigen grafischen Oberfläche auch den Umgang mit der Kommandozeile zu erlernen. Vielleicht haben Sie in den 80er- oder frühen 90er-Jahren einmal mit MS-DOS arbeiten müssen und sind daher nicht sonderlich von dieser Idee angetan. Nun, die Linux-»Kommandozeile« ist kaum damit vergleichbar. Sie besteht aus der Shell, einem sehr leistungsfähigen Kommandointerpreter mit zahllosen bequemen Mechanismen und Tastenkürzeln, sowie unzähligen klei-

nen, spezialisierten Dienstprogrammen. Beides zusammen stellt Ihnen ein hochpräzises Instrumentarium zur Datei- und Systemverwaltung zur Verfügung. Und anders als unter Windows können Sie wirklich *alle* Systemverwaltungsaufgaben über die Kommandozeile durchführen (einige sehr spezielle sogar ausschließlich). Ein besonders überzeugendes Argument ist zudem, dass Sie die Konsole auch über ein Netzwerk aufrufen können, ohne an dem Rechner zu sitzen, mit dem Sie gerade arbeiten. In Kapitel 4 erhalten Sie eine ausführliche Einführung in die wunderbare Welt der Linux-Shell.

► *Linux ist ein Mehrbenutzersystem. Was bedeutet das praktisch?*

Zu der Zeit, als UNIX entwickelt wurde (siehe unten), waren Computer selten, groß und teuer. Es war unvorstellbar, dass sie nur von je einer Person zur gleichen Zeit genutzt werden sollten. Also entwickelte man Konzepte, um mehreren Usern gleichzeitig die Verwendung des Computers zu ermöglichen; dazu wurden auch tatsächlich mehrere Terminals (Ein-/Ausgabestationen) angeschlossen.

Viele Funktionen von UNIX-Betriebssystemen basieren bis heute auf diesem Erbe. So ist es selbstverständlich, dass beliebig viele Benutzerkonten eingerichtet werden können; jeder Benutzer darf standardmäßig nur seine eigenen Dateien modifizieren. Zwar sind zumindest die meisten Personal Computer heutzutage nicht mit mehreren Bildschirmen und Tastaturen ausgestattet, aber Sie können sich in mehreren sogenannten virtuellen Terminals unter verschiedenen Benutzernamen anmelden. Zudem können Sie, wie oben erwähnt, über ein lokales Netzwerk oder das Internet auf den Rechner zugreifen und darauf arbeiten, als säßen Sie unmittelbar davor.

Neben den gewöhnlichen Benutzern mit ihren eingeschränkten Rechten gibt es den sogenannten Superuser *root*, der alles darf – auf jede Datei und jedes Verzeichnis zugreifen sowie jedes Programm starten und beenden. Wenn Sie openSUSE auf Ihrem PC installieren, sind Sie selbst *root* – allerdings sollten Sie im Alltag nie unter dieser Benutzerkennung arbeiten, weil Sie sonst versehentlich wichtige Systemdateien löschen oder sonstigen Schaden anrichten können. Für gelegentliche administrative Aufgaben besteht die Möglichkeit, vorübergehend *root*-Rechte zu erlangen.

1.1 Die Entwicklung von UNIX und Linux

Linux, und jedes andere *Betriebssystem*, ist *das* grundlegende Computerprogramm. Es steuert die Hardware, koordiniert Ressourcenzugriffe der Anwendungsprogramme und stellt dem Benutzer Steuerungs- und Interaktionsmöglich-

keiten zur Verfügung. Im Einzelnen erfüllen Betriebssysteme vor allem die folgenden Aufgaben:

▶ *Prozessmanagement*

Die Ressourcen des Computersystems müssen zwischen den verschiedenen laufenden Programmen und Systemaufgaben verteilt werden. Zu diesem Zweck werden die einzelnen Aufgaben als sogenannte Prozesse ausgeführt, die vom Betriebssystem als übergeordnetem Steuerprozess verwaltet werden.

▶ *Speichermanagement*

Obwohl Computersysteme heutzutage über einen vergleichsweise großen Arbeitsspeicher verfügen (der mit jedem Jahr und jeder neuen Produktreihe noch größer wird), finden dennoch oftmals nicht alle Programme und Daten auf einmal Platz darin. Das Speichermanagement sorgt dafür, dass immer die gerade benötigten Speicherinhalte zur Verfügung stehen, ohne dass die Programmierer der Anwendungssoftware sich sonderlich darum kümmern müssten.

▶ *Steuerung und Abstraktion der Hardware*

Computersysteme sind hochgradig modular aufgebaut; jede Aufgabe kann durch viele verschiedene Geräte unterschiedlicher Hersteller erledigt werden. Betriebssysteme lösen dieses Problem durch den Einsatz der sogenannten *Gerätetreiber*, die die Steuerlogik für bestimmte Hardware enthalten. Vor dem Programmierer werden so die konkreten Einzelheiten bestimmter Geräte verborgen, weil es vollkommen unzumutbar wäre, beim Schreiben eines Anwendungsprogramms auf die Besonderheiten Hunderter möglicher Geräte einzugehen.

▶ *Ein- und Ausgabesteuerung*

Computerprogramme sind auf die Eingabe von Daten angewiesen, und ihre Benutzer erwarten die Ausgabe von Ergebnissen. Betriebssysteme steuern die Zusammenarbeit mit vielen verschiedenen Ein- und Ausgabekanälen wie Tastatur und Bildschirm, Datenträgern oder Netzwerken.

▶ *Dateiverwaltung*

Programme und Daten müssen auf einem Computer dauerhaft gespeichert werden, weil der Arbeitsspeicher zu klein ist, und vor allem deswegen, weil dessen Inhalt beim Ausschalten verloren geht. Aus diesem Grund werden Daten in Form von *Dateien* auf Datenträgern wie Festplatten, CDs oder Disketten gespeichert. Die Logik der Dateiverwaltung wird in Form sogenannter Dateisysteme vom Betriebssystem zur Verfügung gestellt, damit alle Programme auf dieselbe Art und Weise darauf zugreifen.

▶ *Bereitstellen der Benutzeroberfläche*

Ein Spezialfall der Ein- und Ausgabesteuerung besteht im Bereitstellen der Benutzeroberfläche. Das Betriebssystem selbst und alle Programme müssen

mit dem Benutzer kommunizieren, um Aufgaben nach dessen Wünschen zu erledigen. Wie bereits erwähnt, existieren zwei verschiedene Arten von Benutzeroberflächen: Die dialogorientierte *Konsole* ermöglicht das Arbeiten durch Kommandoeingabe; Benutzer geben per Tastatur Befehle ein und erhalten Antworten. Die *grafische Benutzeroberfläche* ermöglicht dagegen die intuitive Erledigung von Aufgaben über das Aktivieren von Schaltflächen, Menüs und Symbolen mit der Maus. Als moderne Linux-Distribution besitzt openSUSE jeweils mehrere unterschiedlich leistungsfähige Varianten beider Arten von Benutzeroberflächen.

1.1.1 Die Vorgeschichte der Betriebssysteme

Betriebssysteme im heutigen Sinn wurden erst erforderlich, als Menschen begannen, direkt mit Computern zu kommunizieren.

Die allerersten Computersysteme der 40er- und 50er-Jahre wurden über Schalter und Steckverbindungen programmiert. Dem einzigen Programm, das zu einer bestimmten Zeit auf dem Computer lief, standen sämtliche Ressourcen ungeteilt zur Verfügung. Diese Ressourcen waren den Programmierern unmittelbar bekannt, es musste keine Abstraktion der Hardwareressourcen stattfinden. Solche Rechner besaßen überhaupt kein Betriebssystem.

Als die Lochkarten Einzug in die Rechenzentren hielten, war es üblich, dass ein Programmierer dem Operator den mithilfe eines mechanischen Geräts gestanzten Kartenstapel übergab. Die Programmierer selbst bekamen den eigentlichen Computer möglicherweise nicht einmal zu Gesicht, weil nur die mit weißen Kitteln bekleideten Operatoren das »Allerheiligste« betreten durften. Der Operator war dafür verantwortlich, den Inhalt des Lochkartenstapels in den Computer einzulesen, und händigte dem Programmierer einen Stapel Endlospapier mit den Ergebnissen aus, falls das Programm keine Fehler enthielt.

Erst in den 60er-Jahren wurden bestimmte häufig verwendete Programmteile oder Hilfsmittel wie Interpreter für höhere Programmiersprachen auf Magnetbändern statt auf Lochkarten abgespeichert. In der ersten Zeit waren wieder die Operatoren dafür zuständig, das richtige Band einzulesen, um die Programmlochkarten eines bestimmten Programmierers korrekt zu verarbeiten.

Allmählich wurden übergeordnete Steuerprogramme eingeführt, die in der Lage waren, auf Lochkarten mit speziellen Meta-Befehlen zu reagieren, die nicht zum Programm selbst gehörten, sondern organisatorische Informationen enthielten. Auf diese Weise konnten einige der Aufgaben von Operatoren automatisiert werden, sodass die Steuerprogramme als *Operating Systems* bezeichnet wurden – das englische Wort für »Betriebssystem«. Diese allerersten Systeme ermöglich-

ten die automatisierte Abarbeitung mehrerer Lochkartenstapel; diese Form der Verarbeitung wird deshalb als *Stapelverarbeitung* (englisch »batch processing«) bezeichnet.

»Richtige« Betriebssysteme, wie sie bis heute verwendet werden, wurden allerdings erst eingeführt, als die *Dialogverarbeitung* entwickelt wurde. Seit Mitte der 60er-Jahre wurden immer mehr Rechenzentren mit *Terminals* ausgestattet. Ein Terminal (wegen seiner fehlenden eigenen Rechenfähigkeiten auch »dummes Terminal« genannt) ist eine Ein- und Ausgabeeinheit, die direkt am Arbeitsplatz eines Programmierers steht und mit dem eigentlichen Computer verbunden ist.

Das Betriebssystem muss die Eingabe des Benutzers in den Computer transportieren und die Antwort des Computers an das Terminal zurückmelden. Die besondere Herausforderung für die Entwickler der frühen Betriebssysteme bestand darin, die Benutzer mehrerer Terminals zeitgleich zu bedienen. Die damaligen Computersysteme waren nämlich viel zu selten und zu teuer, um einem einzelnen Benutzer sämtliche Ressourcen zu überlassen.

Aus dieser Anforderung entwickelte sich das sogenannte *Timesharing*-Verfahren, das die Anfragen mehrerer Benutzer scheinbar gleichzeitig verarbeitet, indem es die Rechenzeit in kleine Einheiten (Zeitscheiben oder »time slices«) unterteilt und den Anforderungen der einzelnen Benutzer der Reihe nach zuweist. Aus diesem grundsätzlichen Prinzip entwickelte sich beispielsweise das moderne Multitasking, das die Ausführung mehrerer Programme auf demselben Rechner ermöglicht. Außerdem wurden allmählich Mechanismen eingeführt, um die Arbeit verschiedener Benutzer im Speicher des Rechners voneinander zu trennen – die Grundlagen der Zugriffsrechte, der persönlichen Anmeldung und des Speicher- und Ressourcenmanagements wurden entwickelt.

Die ersten Timesharing-Betriebssysteme wurden für einzelne Computer und ihre Anwender in Universitäten und anderen großen Institutionen entworfen. Eines der wenigen frühen Beispiele, die relativ bekannt geworden sind, ist das am MIT (Massachusetts Institute of Technology) entwickelte *ITS (Incompatible Time-sharing System)* – es wurde von *Richard M. Stallman* mitentwickelt, dem Begründer der Idee der freien Software.¹ Außerdem wurden einige der Fähigkeiten von ITS später in UNIX aufgenommen.

Der erste Versuch, ein kommerzielles Timesharing-System zu schreiben, erfolgte Ende der 60er-Jahre: Die Bell Laboratories, das Entwicklungszentrum der US-Telefongesellschaft AT&T, arbeitete mit einigen anderen Firmen an einem Sys-

1 Zuvor hatte es ein – längst in Vergessenheit geratenes – System namens Compatible Time Sharing System (CTSS) gegeben, auf das sich die Namenswahl bezog.

tem namens MULTICS. Die Idee war, viele Hundert Terminals an einem für damalige Verhältnisse recht mächtigen Rechner anzuschließen. Leider stieg AT&T aus dem zunächst erfolglosen Projekt aus, und MULTICS wurde erst viel später fertiggestellt, als es kaum noch jemanden interessierte.

1.1.2 Die Geschichte von UNIX

Einige Ideen von MULTICS inspirierten jedoch *Ken Thompson* von den Bell Labs dazu, mit der Arbeit an einem eigenen Betriebssystem zu beginnen – anfangs war es als Einzelplatzsystem geplant und wurde auf einem PDP-7-Kleincomputer von Digital Equipment (DEC) entwickelt, der verglichen mit der für MULTICS verwendeten Maschine nicht besonders leistungsfähig war. Thompson und sein späterer Mitstreiter *Dennis Ritchie* nannten ihr Betriebssystem in einer Anspielung auf MULTICS »Unics«; später wurde daraus UNIX. Die ursprünglich nicht geplante Mehrbenutzer-Fähigkeit wurde übrigens sehr schnell nachgerüstet, nachdem die ersten Versionen von UNIX gut funktionierten.

Eines der wichtigsten Designmerkmale des Betriebssystems UNIX ist seine Modularität: Jeder einzelne Befehl und jeder Bestandteil des Systems kann einzeln ausgetauscht werden, ohne das Gesamtsystem ändern zu müssen. UNIX wurde zunächst in der Maschinensprache der PDP-7 programmiert. 1971 entwickelten Dennis Ritchie und *Brian Kernighan* dann die Programmiersprache C, in der UNIX schließlich neu geschrieben wurde. Da C-Compiler bald für verschiedene Computersysteme verfügbar waren, fand UNIX schnell Verbreitung.

Eine kommerzielle Verbreitung von UNIX war zu Anfang nicht möglich. Da AT&T in den 70er-Jahren noch das US-Telekommunikationsmonopol innehatte, durfte das Unternehmen keine Geschäfte in anderen Branchen wie etwa dem Computerbereich tätigen. Deshalb wurden Lizenzen für den UNIX-Quellcode unter sehr lockeren Bedingungen an Universitäten vergeben. Das ist der Grund, warum die Grundlagen des Betriebssystemaufbaus bis heute vornehmlich am Beispiel von UNIX und seinen Nachfolgesystemen gelehrt werden.

Die aktivste aller Universitäten, die den UNIX-Quellcode erhielten, war die University of California in Berkeley. Die dortigen Programmierer erweiterten das ursprüngliche UNIX und brachten schließlich eine eigene Version heraus, die *Berkeley System Distribution (BSD)*. Als AT&T schließlich in den 80er-Jahren doch noch die Lizenz erhielt, UNIX kommerziell zu vertreiben, hatten sich die AT&T-Version, genannt System V, und die BSD bereits erheblich auseinanderentwickelt. Zudem wurde die Berkeley University durch einen gerichtlichen Vergleich verpflichtet, sämtliche Bestandteile des AT&T-UNIX aus der BSD zu entfernen.

Seitdem ist jede kommerzielle und jede freie UNIX-Version eine mehr oder weniger starke Mischung aus BSD- und System-V-Features. Allerdings haben System V und BSD wieder zahlreiche Fähigkeiten voneinander übernommen, sodass es nicht immer ganz leicht ist, sie auseinanderzuhalten.

Heute existiert eine Vielzahl unterschiedlicher UNIX-kompatibler Betriebssysteme. Dazu gehören kommerzielle Systeme wie *Sun Solaris*, *IBM AIX*, *HP UX* und freie Varianten wie *Linux*, das Thema dieses Buches, oder *FreeBSD*. Eine Sonderstellung nimmt Apples neues Betriebssystem für Macintosh-Rechner ein: *Mac OS X* löst das technisch vollkommen veraltete *Mac OS 9* ab. Es basiert auf einer Version der BSD; der System-Unterbau *Darwin* (der seinerseits einen Mach-Mikrokernel enthält) ist Open-Source-Software und läuft auf verschiedenen Plattformen. Die grafische Benutzeroberfläche Aqua ist dagegen eine kommerzielle Eigenproduktion von Apple und funktioniert bisher nur auf Macs mit PowerPC-Prozessor sowie auf der neuen Mac-Produktlinie mit Intel-Prozessoren.

Die Mindestanforderung dessen, was ein UNIX-System leisten muss, ist in einem Standard namens *POSIX* (Portable Operating System Interface) festgeschrieben. Allerdings werden von einem »richtigen« UNIX heute auch einige weitere Quasi-Standards verlangt, die nicht im POSIX-Standard festgelegt sind. Am wichtigsten ist dieser Standard für Programmierer. Wenn Sie ein Programm POSIX-konform schreiben, können Sie davon ausgehen, dass es sich unter jeder beliebigen UNIX-Version kompilieren lässt.

Anfang der 90er-Jahre sah es übrigens bereits fast so aus, als würde UNIX nicht mehr lange überleben: Die verschiedenen Varianten entwickelten sich zunehmend auseinander; auf dem Desktop dominierte Microsoft Windows, während Server für die immer häufiger eingesetzten PC-Netzwerke vor allem unter Novell Netware betrieben wurden.

Zwei wichtige Umstände haben UNIX gerettet und machen es heute, über 30 Jahre nach seiner Entwicklung, zu einem der gefragtesten Betriebssystemkonzepte: Der eine Grund ist die immense Ausbreitung des Internets, dessen wichtigste Konzepte unter UNIX entwickelt wurden. Der zweite Anlass für die Verbreitung von UNIX ist die Erfolgsgeschichte des freien Betriebssystems Linux.

1.1.3 Freie Software und Open-Source-Software

Der Vertrieb von UNIX durch AT&T war nur ein Aspekt der zunehmenden Kommerzialisierung von Software, denn parallel begann der Aufstieg der heute großen Softwareunternehmen wie Microsoft. Dies bereitete einigen wissenschaftlich orientierten IT-Spezialisten große Sorgen; sie fürchteten, dass das Verstecken von Quellcodes als Betriebsgeheimnis den freien Gedankenaustausch zwischen

Entwicklern behindern und in der Konsequenz die Qualität der Software verschlechtern würde. Vergleicht man die durchschnittliche Sicherheit von freier und kommerzieller Software, muss man zu dem Schluss kommen, dass diese Befürchtung nicht unbegründet ist. Hinzu kommen ärgerliche Entwicklungen wie Softwarepatente, die der im Grunde mathematischen Natur von Computerprogrammen nicht im Geringsten gerecht werden und Programmierer ohne jede Not an der Verwendung naheliegender Ansätze hindern.

Die Free Software Foundation und das GNU-Projekt

Einer der wichtigsten Kritiker der Software-Kommerzialisierung war der bereits genannte Richard Stallman. Deshalb gründete er 1984 die *Free Software Foundation* (FSF), um von Grund auf neu programmierte und daher freie UNIX-Standard-Tools bereitzustellen. Das betreffende Projekt heißt *GNU*. Dies steht für »GNU's not Unix«; es handelt sich um ein *rekursives Akronym*, in dem der erste Buchstabe immer wieder für den Namen des Ganzen steht – eine beliebte Form der Unterhaltung in der UNIX-Szene.

Ein erstes und bis heute eines der wichtigsten GNU-Tools ist der GNU C Compiler (GCC); zudem wurden von Anfang an GNU-Versionen der UNIX-Standard-Dienstprogramme geschrieben. Die meisten von ihnen sind inzwischen sogar leistungsfähiger als die Versionen aus dem ursprünglichen UNIX. Lediglich der seit vielen Jahren in Entwicklung befindliche GNU-Kernel Hurd ist bis heute nicht in einer finalen Version erschienen, was bis zu einem gewissen Grad sicherlich auch daran liegt, dass die Entwicklung von Linux den Druck gemildert hat, diesen Kernel fertigzustellen.

Freie Software hat nicht nur etwas damit zu tun, dass die Programme kostenlos erhältlich sind, sondern auch damit, dass Sie den Quellcode erhalten und damit fast alles machen dürfen, was Sie möchten. Richard Stallman bringt es gern auf die folgende Formel: »Free as in freedom, not as in free beer.« (»Frei wie in Freiheit, nicht wie in Freibier.«) Kommerzielle Softwarelizenzen enthalten dagegen in der Regel eine Reihe von Einschränkungen und erlauben im Grunde nichts weiter, als die Software für ihren offiziellen Anwendungszweck einzusetzen. Die FSF hat dafür eine eigene Softwarelizenz ausgearbeitet, die vor allem verhindern soll, dass kommerzielle Softwareentwickler freie Softwareprojekte an sich binden und die ursprüngliche Freiheit beeinträchtigen. Die Lizenz trägt den Namen *GNU General Public License* (GPL). Die GPL gewährt insbesondere vier Freiheiten:

1. Die Software darf für jeden beliebigen Verwendungszweck eingesetzt werden (sogar dieses eigentlich selbstverständliche Recht schränken kommerzielle Softwarelizenzen oft ein).

2. Die Software darf von jedermann in beliebiger Anzahl frei weiterverbreitet werden; es ist sogar erlaubt, für diese Dienstleistung (nicht jedoch für die Software selbst oder ein Nutzungsrecht!) Geld zu kassieren. Der Quellcode muss mitgeliefert oder auf Anforderung zur Verfügung gestellt werden.
3. Jeder darf die Funktionsweise der Software studieren und an eigene Bedürfnisse anpassen.
4. Es ist erlaubt, auch geänderte Fassungen der Software frei zu verbreiten, solange deren Quellcode wiederum freigegeben und unter die GPL gestellt wird.

Trotz dieser Freiheiten ist die GPL eine echte, unter der Aufsicht von Anwälten geschriebene, rechtswirksame Softwarelizenz. Sie basiert auf dem Prinzip des *Copyleft* – ein Gegenbegriff zum bekannten Copyright, der besagt, dass die einmal gewährten Rechte nicht wieder aufgehoben werden dürfen.

Dies verhindert insbesondere den Einsatz GPL-basierter Software in rein kommerziellen Projekten. Zum Problem wird dies vor allem dann, wenn Programm-bibliotheken eingebunden werden sollen, die unter der GPL stehen. Zu diesem Zweck existiert die abgemilderte Library oder Lesser GPL (LGPL), die dies für kommerzielle Software in Grenzen auch dann gestattet, wenn diese nicht unter die GPL gestellt wird.

Inzwischen wurde Version 3.0 der GPL eingeführt. Sie soll aktuelle Entwicklungen der letzten Jahre einbeziehen, etwa der Bedrohung durch Softwarepatente etwas entgegensetzen. Andere geplante Neuerungen – beispielsweise ein generelles Verbot, Software für DRM (Digital Rights Management, technische Einschränkungen von Mediendateien zur Durchsetzung des Urheberrechts) unter die GPL zu stellen – gehen einigen Kritikern zu weit. Zu diesen Kritikern gehört auch Linus Torvalds, der befürchtet, dass solche Beschränkungen die universelle Einsetzbarkeit von Linux behindern könnten. Daher kündigte er bereits an, dass Linux möglicherweise nicht unter die GPL 3.0 gestellt würde, wenn diese endgültig verabschiedet wird. Dies ist problemlos möglich, weil Linux nicht – wie viele andere GPL-Programme – den Hinweis enthält, dass automatisch die neueste Version der Lizenz zur Geltung kommt.

Neben Linux stehen zahlreiche weitere namhafte Programme unter dieser Lizenz. Dazu gehört zum einen die Software des GNU-Projekts selbst, beispielsweise die Compiler-Suite GCC, der Editor Emacs, das Bildbearbeitungsprogramm GIMP oder die Desktop-Umgebung GNOME. Zum anderen gibt es auch zahlreiche nicht aus dem GNU-Projekt stammende Programme, die die GPL anwenden; dazu gehören beispielsweise der Datenbankserver MySQL, das Office-Paket OpenOffice.org und der Desktop KDE.

Manchen freien Softwareprojekten sind die GPL-Bedingungen dagegen zu streng oder sie haben andere Probleme damit. Deshalb verwenden sie eigene Lizenzen. Eine bekannte Variante ist etwa die BSD-Lizenz, die auf ein Copyleft verzichtet und das Einbinden entsprechender Software und ihrer Bestandteile in beliebige Programme gestattet, ob diese nun ihrerseits freigegeben werden oder nicht. Ebenfalls sehr weit verbreitet ist die Apache-Lizenz der Apache Software Foundation, die früher BSD-ähnlich war und sich mit der aktuellen Version 2.0 eher den Gedanken der GPL angenähert hat.

Freie Software oder Open-Source-Software?

Mit der im nächsten Unterabschnitt skizzierten Entwicklung von Linux und anderen neueren Softwareprojekten etablierte sich eine neue Generation von Entwicklern freier Software, die den traditionellen Hackern um Richard Stallman irgendwann nicht mehr technisch unterlegen waren. Der Unterschied bestand eher darin, dass die neue Generation in einer Umgebung mit kommerzieller Software aufgewachsen war und diese daher nicht so sehr als Bedrohung empfand. Dies führte zu einer pragmatischeren und weniger ideologischen Herangehensweise an die Freiheit in der Software.

Die beiden Kulturen prallten 1996 aufeinander, als Richard Stallman und der weiter unten vorgestellte Linux-Erfinder Linus Torvalds als gleichberechtigte Keynote-Sprecher einer Konferenz über freie Software auftraten. Torvalds überzeugte als eloquenter und geistreicher Redner und fand so auch Zustimmung in den Reihen der klassischen Hacker, die allerdings später etwas misstrauischer wurden, weil er sich als Fan des Microsoft-Präsentationsprogramms PowerPoint outete. Aus der Sicht der neuen Generation war es allerdings völlig normal, auch kommerzielle Software einzusetzen, zumal dann, wenn es keine gleichwertigen freien Alternativen gab.

Eine recht genaue und weitgehend unparteiische Sicht auf die damaligen Ereignisse liefert das Buch »The Cathedral & The Bazaar« von *Eric S. Raymond*, der zwar zur ursprünglichen Hacker-Generation gehörte, aber die mitunter allzu ideologische Sicht Stallmans nicht teilte und stattdessen Verständnis für die pragmatische Sicht Stallmans nicht teilte und stattdessen Verständnis für die pragmatische orientierten Linux-Hacker entwickelte.

Zu den Meinungsverschiedenheiten zwischen den beiden Gruppen gehörte auch der Streit um den Namen »Linux«. Dieser wurde zunehmend für das Gesamtsystem, also den Linux-Kernel und die GNU-Dienstprogramme, eingesetzt. Die Free Software Foundation missbilligte dies mit dem Argument, dass beide Komponenten gleichwertig seien. Nach dem unglücklichen Konstrukt »lignux« wurde der akzeptablere Vorschlag »GNU/Linux« für eine solche Betriebssystemumgebung

geprägt. Allerdings hielten sich von Anfang nicht allzu viele Distributoren und Linux-User daran, lediglich Debian heißt bis heute so.

Als Ausdruck des neuen, sachlicher orientierten Selbstverständnisses des Linux-Umfeldes wurde 1998 auf Vorschlag von *Christine Peterson*, der Vorsitzenden eines Forschungsinstituts, der Begriff *Open-Source-Software* geprägt. Dies geschah anlässlich der unten erwähnten Entscheidung von Netscape, den Quellcode des Navigators freizugeben. Die Free Software Foundation legt seither Wert darauf, mit Open-Source-Software nichts zu tun zu haben. Dennoch muss man bei nüchterner Betrachtung zu dem Schluss kommen, dass freie Software und Open-Source-Software sich nicht sonderlich stark voneinander unterscheiden, zumal es schon lange vor der Entwicklung des neuen Begriffs auch freie Software gegeben hatte, die nicht von der FSF stammte und unter anderen Lizenzen verfügbar gemacht wurde.

Letztendlich hat der Pragmatismus der Open-Source-Bewegung Vor- und Nachteile: Einerseits ist es bedauerlich, dass sie sich von den ursprünglichen, umfassenden, auch politisch zu verstehenden »Weltverbesserungsideen« der Free Software Foundation verabschiedet hat, aber andererseits konnte nur dieser rein lösungsorientierte Geist große Unternehmen wie Sun oder IBM sowie Regierungen, Behörden und nicht zuletzt zahllose Privatanwender von dieser neuen Art der Softwareverbreitung überzeugen. Ist sie auf diese Weise erst einmal in unpolitischen oder gar politisch anders orientierten Kreisen angekommen, wird sie auf Dauer vielleicht auch dort zu einem neuen Denken beitragen, das letztlich wiederum gesellschaftliche Veränderungen in Gang setzen könnte.

Mit anderen Worten: Auch wenn der berühmte »Marsch durch die Institutionen« der 68er-Generation leider nicht die Institutionen, sondern vielmehr die Marschierenden verändert hat, kann es nichts schaden, Mut für einen neuen Versuch aufzubringen.

1.1.4 Die Entwicklung von Linux

Der Linux-Kernel wurde ab 1991 von dem finnischen Informatikstudenten *Linus Torvalds* entwickelt. Torvalds hatte sich soeben einen PC mit Intel-80386-Prozessor gekauft und suchte nach einem Betriebssystem, das dessen im Zusammenhang mit der 32-Bit-Architektur neu entwickelte Fähigkeiten wie Task-Switching, Speicherschutz und mehrere Prozessorbetriebsmodi ausnutzen konnte. Die damals hauptsächlich verbreitete Kombination aus MS-DOS und Windows 3.x konnte dies nicht im Geringsten, und die einzige halbwegs flächendeckend für PCs verfügbare UNIX-Variante war das zu Lehrzwecken geschriebene und daher stark abgespeckte *Minix* von *Andrew S. Tanenbaum*.

Die Anfänge

Mit den Optionen Windows oder Minix zu Recht unzufrieden, beschloss Linus Torvalds, seinen eigenen UNIX-ähnlichen Kernel zu programmieren (wenngleich er ursprünglich eigentlich nur eine Terminalemulation für den Zugriff auf das Universitätsrechenzentrum schreiben wollte). Die Ankündigung veröffentlichte er am 25. August 1991 in der Minix-Newsgroup *comp.os.minix*:

»Hello everybody out there using minix – I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones. This has been brewing since april, and is starting to get ready. I'd like any feedback on things people like/dislike in minix, as my OS resembles it somewhat (same physical layout of the file-system (due to practical reasons) among other things).

I've currently ported bash(1.08) and gcc(1.40), and things seem to work. This implies that I'll get something practical within a few months, and I'd like to know what features most people would want. Any suggestions are welcome, but I won't promise I'll implement them :-)

Linus (torvalds@kruuna.helsinki.fi)

PS: Yes – it's free of any minix code, and it has a multi-threaded fs. It is NOT portable (uses 386 task switching etc), and it probably never will support anything other than AT-harddisks, as that's all I have :-(.«

[»Hallo an alle da draußen, die minix verwenden – ich arbeite an einem (freien) Betriebssystem (nur ein Hobby, wird nicht groß und professionell wie gnu) für 386(486)-AT-Kompatible. Das läuft schon seit April und wird allmählich fertig. Ich hätte gern jegliches Feedback darüber, was Leute an minix mögen/nicht mögen, da mein OS ihm etwas ähnelt (das gleiche physische Layout des Dateisystems (aus praktischen Gründen), unter anderem).

Ich habe bereits bash(1.08) und gcc(1.40) portiert, und alles scheint zu funktionieren. Es zeichnet sich ab, dass ich in ein paar Monaten etwas Brauchbares haben werde, und deshalb wüsste ich gern, welche Features die meisten Leute sich wünschen. Vorschläge aller Art sind willkommen, aber ich verspreche nicht, sie zu implementieren :-)

Linus (torvalds@kruuna.helsinki.fi)

PS. : Ja – es ist frei von sämtlichem minix-Code und hat ein Multithread-Dateisystem. Es ist NICHT portierbar (verwendet 386-Task-Switching usw.) und wird wahrscheinlich nie etwas anderes als AT-Festplatten unterstützen, da ich nur solche habe :-(.«]

Wie das Posting zeigt, waren die ersten unter dem neuen Kernel lauffähigen Programme die GNU-Shell bash und der GNU C Compiler gcc – man konnte also

bereits von Anfang an auf der Kommandozeile mit dem System arbeiten, und durch die Verfügbarkeit eines leistungsfähigen Compilers konnte schnell weitere Software portiert werden. Auffallend ist zudem, dass Linus Torvalds die Software aus dem GNU-Projekt wählte. Trotz seiner Bescheidenheit (»just a hobby, won't be big and professional like gnu«) zeichnete sich hier – zumindest im Rückblick – schon ab, dass GNU hier seinen fehlenden Kernel finden würde.

Der besagte Professor Tanenbaum war davon anfangs überhaupt nicht begeistert. In seinem einige Zeit später verfassten Antwortposting »Linux is obsolete« machte er sich über den Linux-Ansatz lustig, weil es zu der seiner Ansicht nach aussterbenden Spezies der Systeme gehörte, bei denen ein sogenannter *monolithischer Kernel* für zahlreiche, mit der Zeit gewachsene Aufgaben zuständig ist. Er fand, dass die Zukunft dem *Mikrokernel* gehörte, der nur die allerwichtigsten Aufgaben selbst erledigt und die restlichen typischen Systemaufgaben nichtprivilegierten Prozessen überlässt, deren Ausführungszeit mit den Anwendungsprogrammen konkurriert. Da ein großer Teil des Linux-Kernels heute aus dynamisch ladbaren (und wieder entladbaren!) Modulen besteht, ist nicht Linux, sondern Tanenbaums Kritik obsolet – dies hat er übrigens auch längst selbst eingesehen; in neueren Auflagen seines empfehlenswerten Lehrbuches »Moderne Betriebssysteme« (siehe Anhang B) dient Linux als eines der konkreten Beispiele.

Der Name Linux (eine Abkürzung für »Linus' Unix«) stammte nicht von Torvalds selbst; er bevorzugte ursprünglich die Bezeichnung Freax (aus »free« und »freak« sowie dem UNIX-typischen X). Allerdings nannte ein anderer Entwickler das Verzeichnis für die Entwicklung des Kernels auf dem FTP-Server der Universität Helsinki *linux*, was sich schließlich als Name des Systems durchsetzte, weil es sich schlicht besser anhört als »Freax«.

Linux war von Anfang an freie Software, anfangs unter einer eigenen Lizenz; 1993 stellte Linus Torvalds den Kernel dann offiziell unter die GNU GPL. Erst 1994 erachtete er den Kernel als stabil genug, um die erste offizielle Final Release (1.0) herauszubringen. Inzwischen besaß der Kernel volle Netzwerkfähigkeit in Form eines TCP/IP-Stacks; zudem war die gesamte GNU-Software unter dem neuen System lauffähig.

Im Juli 1996, kurz nach Veröffentlichung der Kernel-Version 2.0, kündigte Linus Torvalds an, er sei auf der Suche nach einem Maskottchen oder Wappentier für das Betriebssystem. Da bekannt war, dass er seit einem Besuch im Aquarium von Canberra (Australien) eine Vorliebe für Zwergpinguine (*Eudyptula minor*) hatte, schlug jemand vor, einen fetten, glücklichen Pinguin zu verwenden. Das Rennen machten schließlich die Entwürfe von *Larry Ewing*, die dieser mit dem damals ganz neuen, freien Bildbearbeitungsprogramm GIMP (siehe Kapitel 8, »Desktop-

Software«) erstellt hatte. Den Namen *Tux* steuerte *James Hughes* bei; er lässt sich als Abkürzung sowohl für »Torvalds' Unix« als auch für »tuxedo« (Smoking, die vermeintliche »Bekleidung« der Pinguine) deuten.

In Newsgroups oder Foren wird manchmal halb scherzhaft die Kritik laut, Pinguine seien flugunfähig und daher keine vollwertigen Vögel. Logisch unzulässig schlussfolgern User daraus, dass Linux kein vollwertiges Betriebssystem sei. Dem begegnen Linux-Befürworter mit einem wichtigen Hinweis auf Stabilität: Wer nicht fliegen kann, kann auch nicht abstürzen. In Abbildung 1.1 sehen Sie einen der ursprünglichen Tux-Entwürfe von Larry Ewing.

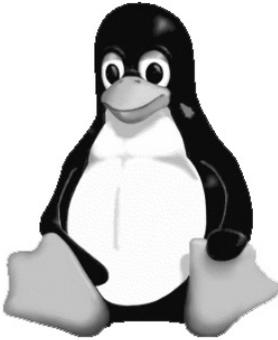


Abbildung 1.1 Tux, der Linux-Pinguin

Kernel-Versionen

Der Kernel besteht inzwischen aus mehreren Tausend Quelldateien und etlichen Millionen Codezeilen und ist so natürlich viel zu umfangreich, als dass eine einzelne Person ihn komplett analysieren oder auch nur lesen könnte. Dennoch wird die Endfassung des jeweils aktuellsten Linux-Kernels bis heute von Linus Torvalds selbst abgenommen. Der bei Weitem größte Teil des Codes besteht ohnehin aus Treibern für unzählige Hardwarekomponenten, die jeweils von eigenen Spezialisten betreut und weiterentwickelt werden. Die meisten dieser Treiber werden bei Bedarf als dynamische Module in den bereits laufenden Kernel geladen und können nach Gebrauch auch wieder entfernt werden.

Die Kernelversionsnummern bestehen aus drei bis vier durch Punkte getrennten Ziffern:

- ▶ Die erste Ziffer ist die Hauptversionsnummer; sie wird nur bei grundlegender Neugestaltung des Kernels gewechselt. Dies geschah bisher nur zweimal: 1994 erfolgte der Sprung von 0.x auf Version 1.0; 1996 begannen die 2.x-Versionen.
- ▶ Mit der zweiten Ziffer wird die sogenannte Major-Release-Nummer gekennzeichnet. Traditionell stehen gerade Zahlen wie 1.2, 2.0 oder 2.6 für eine sta-

bile Release, während ungerade Zahlen wie etwa 2.3 oder 2.5 den aktuellen Entwickler oder »Hacker«-Kernel bezeichnen, der dann eines Tages unter der nächsthöheren geraden Major-Release-Nummer als stabil erscheinen wird. Diverse Softwareprojekte wie der Apache-Webserver oder die Programmiersprache Perl haben dieses Nummernschema übernommen, die Linux-Kernel-Entwickler verwenden sie aber zurzeit nicht mehr: Obwohl die aktuelle Major-Release 2.6 bereits Ende 2003 veröffentlicht wurde, gibt es bis heute keine Entwicklerversion 2.7.

- ▶ Die dritte Ziffer ist die Minor-Release-Nummer, mit der jede einzelne offizielle Release der Kernel-Entwickler bezeichnet wird. Die aktuellste Version ist 2.6.31.
- ▶ Eine vierte Ziffer, das sogenannte Patch Level, wird immer dann verwendet, wenn Fehler oder Sicherheitslücken sofortige Teiländerungen (Patches) nötig machen. Dies war bei der aktuellen stabilen Version noch nicht der Fall, sodass sie (Ende Oktober 2009) 2.6.31 ohne weitere Ziffer ist.

Tabelle 1.1 zeigt eine Übersicht über die Veröffentlichungstermine der diversen Kernel-Versionen; genaue Informationen über die Funktionsweise des Kernels erhalten Sie implizit im Verlauf dieses Buches und noch einmal gründlich in Kapitel 18, »Hinter den Kulissen«.

Version	Veröffentlichungstermin
0.0.1	September 1991
0.0.2	Oktober 1991
1.0.0	März 1994
1.1.0 (Entwicklerversion)	April 1994
1.2.0	März 1995
1.3.0 (Entwicklerversion)	Juni 1995
2.0.0	Juni 1996
2.1.0 (Entwicklerversion)	September 1996
2.2.0	Januar 1999
2.3.0 (Entwicklerversion)	Mai 1999
2.4.0	Januar 2001
2.5.0 (Entwicklerversion)	November 2001
2.6.0	Dezember 2003

Tabelle 1.1 Die Versionsgeschichte des Linux-Kernels

Ob die nächste stabile Version unter der Nummer 2.7, 2.8 oder gar 3.0 erscheinen wird, steht zurzeit noch nicht fest. Die Erhöhung der Hauptversionsnummer ist allerdings nicht so abwegig, da sie bereits für die letztendlich unter der Nummer 2.6 veröffentlichte Release diskutiert worden war.

Die Kernel 2.0, 2.2 und 2.4 werden übrigens noch in Maßen weiterentwickelt, da manche ältere Hardware mit neueren Versionen nicht zurechtkommt. Echte Neuerungen werden in diese Versionen zwar nicht mehr eingebaut, aber bisweilen müssen Fehler behoben und Treiber angepasst werden.

Unterstützte Plattformen

Linux ist heute ein voll ausgestattetes, POSIX-konformes Betriebssystem der UNIX-Familie, dessen Kernel allerdings vollkommen neu geschrieben wurde. Seit seiner ursprünglichen Entwicklung wurde der Linux-Kernel durch die Mitarbeit zahlloser Freiwilliger immer weiter ausgebaut. Auch die ursprünglichen Architekturbeschränkungen – »It is NOT portable (uses 386 task switching etc.), and it probably never will support anything other than AT-harddisks« – wurden nach und nach überwunden. Der Linux-Kernel läuft nicht etwa nur auf der Intel-PC-Architektur, sondern wurde auf viele verschiedene Plattformen portiert. Dazu gehören insbesondere folgende Prozessorarchitekturen:

- ▶ Alle Intel-32-Bit-Prozessoren und Kompatible – das heißt, ab *Intel 80386* werden alle gängigen PC-Prozessoren unterstützt; dazu gehört auch die 32-Bit-Entwicklungslinie von AMD (bis zum Athlon).
- ▶ Ein spezielles Projekt namens ELKS (<http://elks.sourceforge.net/>) arbeitet sogar erfolgreich an einer abgespeckten Variante des Kernels für die 16-Bit-Generation ab *Intel 8086*; dies könnte die (bereits heute verfügbare) Linux-Verbreitung in Embedded Systems, das heißt in Steuer- und Regelungseinheiten, fördern, da diese oft solche einfachen und sehr kostengünstigen Prozessoren verwenden.
- ▶ *IA-64*, die von Intel selbst entwickelte, nur für Server und Workstations verwendete 64-Bit-Architektur
- ▶ *AMD64*, die 64-Bit-Architektur von AMD (Athlon 64, Athlon X2 und so weiter). Die 64-Bit-Desktop-Prozessoren von Intel (Core2Duo und ähnliche) basieren ebenfalls auf dieser in diesem Fall Intel 64 genannten Erweiterung.
- ▶ *PowerPC* – von etwa 1996 bis 2005 das Herzstück aller Macintosh-Rechner
- ▶ *Alpha*, der erste verbreitete 64-Bit-Prozessor (wurde von der später durch Compaq aufgekauften Traditionsfirma Digital Equipment entwickelt, auf deren früheren PDP-Maschinen UNIX erfunden wurde)

- ▶ *Sun SPARC* und *UltraSparc*, die Prozessoren der klassischen Sun-Workstations
- ▶ *Motorola 68020* und Nachfolger (spätere Atari-ST- und Commodore-Amiga-Modelle sowie ältere Macintoshs)
- ▶ *MIPS*, die RISC-Prozessoren der Workstations von Silicon Graphics
- ▶ *ARM*, *StrongARM* und kompatible
- ▶ Hewlett Packard *PA-RISC*
- ▶ *IBM S/390* und *Zseries* (moderne IBM-Großrechner)
- ▶ Axis Communications *CRIS*
- ▶ Hitachi *H8/300*
- ▶ Hitachi *SuperH*, der Prozessor der Sega-Dreamcast-Spielkonsolen
- ▶ *NEC v850e*

Linux-Distributionen

Zu Beginn seiner Entstehung war Linux vor allem im Kreis seiner Entwickler, unter einigen Studenten der Universität Helsinki sowie unter ein paar Teilnehmern der Minix-Newsgroup verbreitet. Es war schwierig, den Linux-Kernel zu installieren und die GNU-Versionen aller erforderlichen Systemprogramme zu beschaffen und miteinander zu koordinieren. In den ersten Jahren kam Linux somit nicht über den Kreis von Informatikern und interessierten Studenten hinaus, zumal die Universitäten bereits über Internetanbindungen verfügten, als der Rest der Welt noch nichts damit zu tun hatte.

Einige der Studenten, die mit Linux arbeiteten, begannen allmählich, den Kernel und die Systemprogramme zusammenzustellen und Installationsprogramme für diese Betriebssystempakete zu schreiben. Aus diesen ersten Bemühungen entstanden allmählich verschiedene Linux-Distributionen, die auf CD-ROMs verbreitet wurden, meist zusammen mit mehr oder weniger ausführlichen gedruckten Dokumentationen. Firmen wurden gegründet, die solche Distributionen erstellten und zu vergleichsweise günstigen Preisen verkauften.

Beachten Sie, dass der Kauf einer Distribution nichts mit dem Erwerb kommerzieller Software gemeinsam hat: Sie dürfen die Software, die Sie erhalten, auf beliebig vielen Rechnern installieren und an alle Ihre Bekannten weitergeben. Die Distributoren erhalten das Geld nicht für die Software selbst oder für ein Nutzungsrecht daran, sondern lediglich für ihre Arbeit an Installationsprogrammen und Dokumentation. Entsprechend können Sie die meisten Distributionen alternativ aus dem Internet herunterladen. Eine Distribution kann allerdings einzelne kommerzielle Programme enthalten, für die andere Bedingungen gelten – beachten Sie die Beschreibung, die der Distributor mitliefert.

Heute ist eine Reihe verschiedener Distributionen erhältlich, die sich bezüglich ihres Umfangs, ihres Anwendungsschwerpunkts und ihres Preises unterscheiden. Die wichtigsten Distributionen sind folgende:

- ▶ *Slackware* wurde 1993 von *Patrick Volkerding* begründet und ist die älteste bis heute gepflegte Distribution. Sie basierte auf Fehlerkorrekturen der einzigen noch älteren, aber längst eingestellten Distribution SLS. Die aktuelle Version ist 10.2; sie kann auf der Website des Projekts – <http://www.slackware.org> – heruntergeladen werden. Die Installation gilt als schwierig und die Ausstattung als spartanisch, wenngleich beides im Laufe der Jahre viel besser geworden ist.
- ▶ *openSUSE*, das Thema dieses Buches, ist die verbreitetste Distribution in Europa, insbesondere in Deutschland. Sie wurde 1994 unter dem Namen *SUSE Linux* eingeführt, basierte ursprünglich auf Slackware und gewann rasch eine große Nutzergemeinde. Mittlerweile sind alle Teile der Distribution frei verfügbar, sodass eine offene Entwickler-Community sie weiterentwickeln kann. Novell, nach einer Übernahme die Muttergesellschaft von SUSE, verwendet diese Entwicklungen in seinen kommerziellen Linux-Distributionen. Die Geschichte von SUSE Linux beziehungsweise openSUSE wird im folgenden Abschnitt genauer beschrieben.
- ▶ *RedHat Linux* stammt von dem gleichnamigen amerikanischen Unternehmen und ist die beliebteste Distribution in den USA. Eine Zeitlang wurden eine Personal und eine Professional Edition angeboten; inzwischen wurde die Personal Edition als eigenständige, völlig freie Distribution namens *Fedora Core* ausgelagert und stark erweitert. Von RedHat haben SUSE und andere Distributionen den RedHat Package Manager (RPM) als Tool zur Installation von Softwarepaketen übernommen.
- ▶ *Debian GNU/Linux* besteht voll und ganz aus freier Software unter der GPL – auch das Installationsprogramm. Vor der Einführung von Fedora Core und der ersten freien Version von openSUSE (SUSE Linux 10.0) war dies eine echte Besonderheit. Debian gilt als besonders stabil und zuverlässig, allerdings um den Preis, dass die offizielle Release meist keine besonders aktuellen Versionen wichtiger Softwarepakete enthält. Auf Debian basieren zahlreiche andere Distributionen, etwa die von CD startende Live-Distribution *Knoppix*.
- ▶ *Ubuntu GNU/Linux* wurde ursprünglich auf der Basis von Debian entwickelt, ist aber erheblich einsteigerfreundlicher. Die Distribution wird von dem südafrikanischen Unternehmer Mark Shuttleworth gesponsert; der Name »Ubuntu« stammt aus der Zulu-Sprache und bedeutet etwa »Menschlichkeit« oder »Solidarität«. Das Original-Ubuntu verwendet den Desktop GNOME,

daneben gibt es beispielsweise Kubuntu mit KDE, Xubuntu mit dem besonders ressourcenschonenden Desktop XFCE oder das für Schul- und Ausbildungszwecke optimierte Edubuntu.

- ▶ *Mandriva Linux* stammt von dem Zusammenschluss des französischen Distributors *Mandrake* und der brasilianischen Firma *Conectiva*. Ähnlich wie SUSE besitzt die Distribution einen großen Lieferumfang und ein gut durchdachtes Installationsprogramm.

Neben diesen häufigsten Distributionen werden unzählige weitere angeboten, jede von ihnen besitzt ihre besonderen Eigenschaften, Vor- und Nachteile. Die Unterschiede treten nicht so sehr beim normalen Arbeiten mit dem System zutage, sondern vor allem bei der Installation des Systems und neuer Programme sowie bei Konfigurationsarbeiten.

Linux-Unterstützung durch Hard- und Softwarehersteller

Im Lauf der Linux-Entwicklung zeichnete sich irgendwann ab, dass das System immer mehr Nutzer für sich gewinnen konnte – zunächst vor allem im Server-Umfeld, später auch in zunehmendem Maß im Desktop-Bereich. Viele kommerzielle Softwarehersteller begannen, Linux nicht mehr nur als gefährliche Konkurrenz, sondern zunehmend auch als Markt für ihre Produkte zu sehen. Um das Jahr 2000 begannen namhafte Unternehmen wie IBM, Sun, Oracle oder SAP nicht nur, ihre eigene Software auf Linux zu portieren, sondern unterstützten seine Entwicklung zusätzlich, indem sie großzügige Spenden leisteten oder gezielt Linux-Entwickler anstellten.

Einige Kritiker befürchteten daraufhin, dass der Einfluss dieser Großunternehmen auf Linux zu dominant werden könnte, sodass es seinen ursprünglichen freien Geist verlieren würde. Bis heute ist dies glücklicherweise nicht der Fall. Einige Unternehmen haben im Gegenteil erkannt, dass der Weg der freien Software keine Bedrohung, sondern eine Bereicherung für sie darstellt. So haben sie nicht nur die Weiterentwicklung und Verbreitung vorhandener freier Softwareprojekte gefördert, sondern auch eigene kommerzielle und zum Teil sehr teure Produkte frei verfügbar gemacht. Hier einige Beispiele:

- ▶ Sun gab das Office-Paket Star Office der aufgekauften Hamburger Firma Star-Division unter dem Namen OpenOffice.org frei (siehe das gleichnamige Kapitel 7).
- ▶ Netscape veröffentlichte den Quellcode seiner Internet-Suite, bestehend aus dem Browser Navigator und anderen Elementen. Daraus entstand zunächst die Mozilla-Suite; später wurden spezialisiertere Einzelprogramme wie der Browser Firefox oder der E-Mail-Client Thunderbird entwickelt.

- ▶ IBM gab seine Entwicklungsumgebung Visual Age for Java frei; unter dem Namen Eclipse erfreut sie sich inzwischen großer Beliebtheit unter Java-Entwicklern.
- ▶ SAP übergab die Distributionsrechte seiner Datenbankkomponente SAP DB an die Firma MySQL AB, die auch den gleichnamigen freien Datenbankserver (siehe Kapitel 12) vertreibt. Heute ist SAP DB unter dem Namen MaxDB unter der GNU GPL verfügbar.
- ▶ Ich selbst arbeite hauptberuflich für die Kölner papaya Software GmbH. Ihr Hauptprodukt, das PHP-basierte Web-Content-Management-System papayaCMS, ist Open-Source-Software unter der GPL, und die Firma lebt von kommerziellen Anpassungen und Erweiterungen, die für bestimmte Kunden erstellt werden.

Die berechtigte Frage lautet natürlich, was profitorientierte Unternehmen davon haben, ihre von teuren Spezialistenteams entwickelte Software einfach zu verschenken. Nun, dafür gibt es eine Reihe von Gründen.

Zunächst einmal ist eine über die ganze Welt verteilte freie Entwicklergemeinde kritischer und offener für Neuerungen als ein in sich geschlossenes Entwicklungslabor; neue Trends werden früher erkannt und integriert, Fehler schneller behoben. Das gilt selbst für solche Projekte, bei denen nach wie vor ein Kernteam innerhalb des ursprünglichen Unternehmens die Hauptentwicklungsarbeit leistet (etwa das OpenOffice.org-Team bei Sun) – denn die Chancen sind groß, dass interessierte Dritte sich Teile des frei verfügbaren Quellcodes ansehen und Fehler finden oder Verbesserungsvorschläge machen.

Hinzu kommt, dass viele der genannten Unternehmen neben der freien Version auch eine kommerzielle Version weiterpflegen. Solche Versionen enthalten zusätzliche Leistungen, die besonders für Firmenkunden interessant sind – beispielsweise offiziellen Support und gedruckte Handbücher. In solchen Fällen beinhaltet die Lizenz der freien Version in der Regel eine Klausel, die es dem Unternehmen erlaubt, die Neuerungen der freien Version wieder in die kommerzielle Fassung zu übernehmen.

Nicht zuletzt gewinnen Unternehmen, die freie Software unterstützen, auch an Renommee. Sie werden in Zeitschriften oder in Büchern wie diesem wohlwollender besprochen als gewisse Konkurrenzfirmen, die sich diesen Gedanken völlig verschließen. Ihre technischen Mitarbeiter sind gern gesehene Sprecher auf Softwarekonferenzen, weil nicht jedes zweite Wort als Betriebsgeheimnis verschwiegen werden muss. Kurz: Diese moderne, souveränere Art des Umgangs mit »geistigem Eigentum« kommt gerade in der IT-Gemeinde erheblich besser an als ein ewig gestriges Beharren auf Gesetzen, die für physische Waren geschrieben wurden und das Wesen von Software nicht erfassen können.

Dass einige der genannten Unternehmen andererseits zu den Befürwortern von Softwarepatenten gehören, die der Idee und Praxis der freien Software schaden wie nichts anderes, wird weiter unten diskutiert.

Was bis heute ein wenig fehlt, sind Linux-Versionen bekannter kommerzieller Desktop-Software – zu nennen wären hier etwa die diversen Produkte von Adobe, die nach wie vor nur für Windows und Mac OS X angeboten werden. Während es zu Photoshop die eingeführte und leistungsfähige Alternative GIMP gibt, sind für Produkte wie die DTP-Software InDesign oder das Web-Multimedia-Entwicklungstool Flash bis heute keine passenden Ersatzlösungen verfügbar. Einige von ihnen laufen allerdings eingeschränkt unter der Windows-Emulation *wine*, die zum Lieferumfang von openSUSE gehört.

Schwerer als viele Softwarefirmen tut sich so mancher Hardwarehersteller mit Linux. Hier geht es vor allem um die Frage der Bereitstellung von Treibern. Im Grunde gibt es bezüglich der Linux-Unterstützung von Hardware drei Möglichkeiten:

1. Der Hersteller stellt selbst einen optimierten Linux-Treiber bereit – dies gewährleistet meist ein reibungsloses Funktionieren des entsprechenden Geräts.
2. Der Hersteller hat keine Möglichkeit, selbst einen Treiber zu entwickeln, veröffentlicht aber detaillierte Schnittstellenbeschreibungen. Auf diese Weise können interessierte Entwickler ohne Schwierigkeiten selbst einen gut funktionierenden Treiber schreiben.
3. Der Hersteller stellt selbst keinen Treiber zur Verfügung und weigert sich zudem, die Schnittstellenbeschreibung herauszugeben. In solchen Fällen kann ein Treiber nur durch sogenanntes Reverse-Engineering (Neuentwicklung aufgrund einer Untersuchung der Funktionsweise, in der Regel anhand des Windows-Treibers) geschrieben werden und weist dann meist Fehler oder Funktionsmängel auf.

SUSE stellt seit Jahren eine sehr ausführliche und ständig aktualisierte Hardware-Kompatibilitätsdatenbank zur Verfügung. Hier können Sie prüfen, ob ein vorhandenes Gerät oder (noch besser) eines, dessen Anschaffung Sie planen, Linux-kompatibel ist – dies gilt in der Regel nicht einmal nur für (open)SUSE, sondern für Linux allgemein. Sie erreichen diese Datenbank unter <http://hardwaredb.suse.de>.

1.1.5 Streit und Schwierigkeiten um Linux

Diverse Gruppen sahen und sehen es bis heute nicht gern, dass mit Linux ein freies Projekt, das auf dem Engagement begeisterter Entwickler basiert, auch

kommerzielle Märkte erobert. Deshalb ist Linux fast von Anfang an immer wieder mit diversen Gegnern konfrontiert worden, konnte sich aber bis heute gegen alle Streitigkeiten behaupten.

Allein der Name »Linux« war bereits Anlass diverser Markenrechtsstreitigkeiten und Gerichtsprozesse. Verschiedene kommerzielle Unternehmen versuchten, sich die Bezeichnung als Marke schützen zu lassen. Die Linux-Entwickler konnten dies oft nur unter massiven Schwierigkeiten unterbinden. Inzwischen ist Linux aber in vielen Ländern – auch in Deutschland – eine eingetragene Marke von Linus Torvalds selbst; für die Lizenzierung an Distributoren und andere Organisationen ist das *Linux Mark Institute* (<http://www.linuxmark.org>) zuständig. Lizenzen für die kommerzielle Nutzung sind umsatzabhängig und jährlich zu erneuern, während Non-Profit-Organisationen mit maximal 200 US-Dollar (etwa 140 Euro) einmalig belastet werden.

Zu den Kuriositäten des Markenrechts gehört aber, dass eine Markeneintragung immer nur für bestimmte Kategorien oder Branchen gilt. Deshalb stellt eine Schweizer Firma völlig legal ein Vollwaschmittel namens Linux her (siehe Abbildung 1.2); es wurde in den letzten Jahren unter anderem bei Plus und Wal-Mart angeboten.



Abbildung 1.2 Vollwaschmittel »Linux« (Foto: Björn Lange)

Streit mit Microsoft

In der zweiten Hälfte der 90er-Jahre sah es eine Zeit lang so aus, als würde sich Windows NT nach Novell Netware auf breiter Front als Server-Betriebssystem für PC-Netzwerke durchsetzen; selbst IBM gab nach langem Ringen sein eigenes fortgeschrittenes PC-Betriebssystem OS/2 auf und begann, NT zu unterstützen. Relativ hohe Kosten, eine versprochene, aber nicht verwirklichte Multiplattformunterstützung und die nicht immer optimale Sicherheit der Microsoft-Produkte ließen jedoch immer mehr Netzwerkbetreiber nach Alternativen suchen.

Zur gleichen Zeit begann sich das Internet als allgegenwärtiges Massenmedium durchzusetzen. Da die Standards dieses Netzes auf UNIX-Systemen gewachsen waren, bot sich der Einsatz UNIX-kompatibler Betriebssysteme für Internetverbindungen an, etwa als Webserver, Proxyserver oder Router. Kommerzielle UNIX-Varianten waren allerdings sehr teuer und nur selten auf der Intel-Hardware gewöhnlicher PCs lauffähig. So geriet Linux zunehmend ins Blickfeld von Serverbetreibern, insbesondere bei den neu gegründeten Internet Providern und Webhostern. Mit Samba verfügte Linux sogar über einen freien Dateiserver für Windows-Netzwerke, sodass selbst dem umfassenden Einsatz von Linux als Firmenserver-System nichts mehr im Wege stand.

Microsoft indessen zeigte sich vom plötzlichen Siegeszug dieses Systems böse überrascht – hatte man das klassische UNIX gerade erst totgesagt oder zumindest für stark geschwächt gehalten, stand es nun im neuen Gewand des »Anarchisten-Systems« Linux wieder an vorderster Front. Dass Microsoft diese Bedrohung seiner Marktmacht ernst nahm, zeigten zuerst die »Halloween-Dokumente« (benannt nach ihrem Erscheinungstermin Ende Oktober 1998). In diesen eigentlich internen Memos zeigten sich führende Microsoft-Mitarbeiter äußerst besorgt über Open Source im Allgemeinen und Linux im Besonderen. Sie bescheinigten dieser Software, technisch mindestens ebenbürtig zu sein, und dachten über Strategien nach, sie zu besiegen. Anders als für einige andere große Softwareunternehmen stand es für Microsoft allerdings nie zur Debatte, auch nur eine Zeile des Windows-Quellcodes freizugeben.

In den letzten Jahren hat Microsoft zahlreiche »völlig unabhängige und streng wissenschaftliche« Studien finanziert, deren wenig überraschendes Ergebnis lautete, dass Windows in vielerlei Hinsicht besser als Linux sei und – gemessen am Gesamtaufwand (»Total Cost of Ownership« oder TCO) – sogar kostengünstiger. Aggressive Werbung und öffentlich geäußerte Zweifel an der Rechtssicherheit von Open-Source-Lizenzen sind weitere Strategien von Microsoft gegen die freie Software. Viele Beobachter sehen auch den als Nächstes diskutierten Rechtsstreit zwischen der SCO Group und diversen Linux unterstützenden Firmen als heimliches Werk Microsofts an.

Linus Torvalds macht sich umgekehrt mitunter einen Spaß daraus, Microsoft und besonders deren als cholerisch geltenden Vorstandsvorsitzenden Steve Ballmer zu provozieren – er redet bisweilen gern ironisch über die »Linux World Domination« oder behauptet: »Really, I'm not out to destroy Microsoft. That will just be a completely unintentional side effect.«²

Zumindest auf dem Desktop bleibt es allerdings vorerst bei der »Windows World Domination« mit über 90 % Marktanteil, wenngleich seit etwa 2001 jedes Jahr der »Durchbruch für Linux im Desktop-Geschäft« angekündigt wurde. In diesem Buch lernen Sie jedenfalls beide Linux-Einsatzmöglichkeiten – auf dem Server sowie auf dem Desktop – kennen.

Der Streit mit der SCO Group

Eines vorweg: Die heute unter dem Namen SCO auftretende Firma hat mit dem ursprünglichen Unternehmen dieses Namens – der Santa Cruz Operation, die eines der ersten (kommerziellen) UNIX-Systeme für die PC-Plattform anbot – nichts zu tun. Es handelt sich vielmehr um die umfirmierten Überreste der Firma Caldera, die einst selbst eine Linux-Distribution veröffentlichte. Dieses Unternehmen behauptet, die Rechte am ursprünglichen AT&T-UNIX zu besitzen; Linux enthalte Code aus diesem System und verletze diese Rechte daher.

Unter dieser Prämisse versuchte die neue SCO Group zunächst, diversen Unternehmen eine teure und – außerhalb der merkwürdigen Logik von SCO – nutzlose »Antidot-Lizenz« für den unbedenklichen Einsatz von Linux zu verkaufen. Als sich abzeichnete, dass so gut wie keine Firma darauf einging, verklagte SCO im Jahr 2003 IBM und einige andere Unternehmen, die sie mit Linux in Verbindung brachte. Bis heute hat sie allerdings keine von einem Gericht anerkannten Beweise für ihre Behauptungen vorlegen können, sondern die schwebenden Verfahren durch immer neue Gutachten und anwaltliche Tricks immer wieder aufgeschoben.

Im Laufe der Auseinandersetzungen verstieg sich der damalige SCO-Vorstandsvorsitzende Darl McBride zudem zu diversen extrem unsachlichen Äußerungen – als einer der Höhepunkte gilt, dass er die gesamte Open-Source-Szene in bester McCarthy-Tradition als »unamerikanisch« und »kommunistisch« beschimpfte (<http://www.heise.de/newsticker/meldung/43886>).

2 »Wirklich, ich habe nicht die Absicht, Microsoft zu zerstören. Das wird nur ein völlig unbeabsichtigter Nebeneffekt sein.«

Seit 2008 ist SCO pleite, weil wohl selbst der letzte Geldgeber gemerkt hat, dass das Verklagen der Konkurrenz auf die Dauer kein tragfähiges Geschäftsmodell ist. Zudem hat ein Gericht festgestellt, dass die Rechte an UNIX letztlich gar nicht der SCO Group, sondern Novell gehören. Dennoch hat das inzwischen durch Insolvenzverwalter geleitete Unternehmen noch nicht alle Hoffnungen aufgegeben, einen seiner zahlreichen Prozesse zu gewinnen.

Eine vollständige Chronik aller Ereignisse um den Streit mit SCO können Sie unter <http://www.heise.de/ct/aktuell/meldung/44492> in Form von Links auf zahlreiche Einzelartikel abrufen.

Softwarepatente

Eine der schädlichsten Entwicklungen für die freie Software ist das Unwesen der *Softwarepatente*. In den USA wird es bereits seit Jahren praktiziert und treibt die merkwürdigsten Blüten – denken Sie nur an den jahrelangen patentbedingten Bann von GIF-Bildern, der erst vor Kurzem endete, oder an Amazons Trivialpatent auf die »Ein-Klick-Bestellung«.

Zu den rückhaltlosen Befürwortern von Softwarepatenten gehören neben Microsoft auch einige der oben genannten Unternehmen, die ansonsten großzügig die Entwicklung der freien Software unterstützen – allen voran IBM. Diese Unternehmen üben seit Jahren massiven Druck aus, um Softwarepatente auch in der EU einzuführen. Nur einer hellwachen Gegenbewegung ist es zu verdanken, dass dies gegen diesen Lobbydruck, gegen ignorante Medien und gegen (mindestens) indifferente Politiker bis heute nicht geschehen ist – aber die Gefahr ist nach wie vor nicht gebannt.

Softwarepatente können in der Konsequenz das vollständige Ende der freien Software bedeuten, weil Unternehmen sich mit ihrer Hilfe die exklusive Nutzung allgemeingültiger Softwarebausteine, Programmierstrategien oder gar Algorithmen und Datenstrukturen aneignen können – in manchen Fällen sogar dann, wenn sie gar nicht deren Urheber sind.

Jeder, der ernsthaft an der Nutzung von Open-Source-Software interessiert ist, sollte sich zumindest gründlich über Softwarepatente informieren und nach Möglichkeit auch dagegen engagieren. Ein guter Einstiegspunkt ist die Website des FFII e. V. (Förderverein für eine Freie Informationelle Infrastruktur), die Sie unter <http://www.ffii.org> finden. Hier erfahren Sie die Anschriften Ihrer lokalen EU-Abgeordneten, können an Online-Petitionen teilnehmen und erhalten umfangreiche Informationen zu Softwarepatenten und verwandten Themen.

1.2 openSUSE

Nachdem Sie nun allgemein über freie Software und über Linux Bescheid wissen, geht es in diesem Abschnitt spezieller um das eigentliche Thema dieses Buches, nämlich um die Linux-Distribution openSUSE und ihren Vorläufer SUSE Linux.

1.2.1 Die Geschichte von SUSE Linux

Bereits 1992 gründeten Roland Dyroff, Thomas Fehr, Hubert Mantel und Burchard Steinbild in Nürnberg die Firma Software und Systementwicklung GmbH (abgekürzt S.u.S.E., anfangs mit Pünktchen). Ihr erstes Produkt war eine deutsch lokalisierte und leicht erweiterte Fassung der Slackware-Distribution auf anfangs 50, später sogar über 70 Disketten. Erst 1994 gab es die Software auch auf CD-ROM, anfangs wahlweise und bald ausschließlich. Im gleichen Jahr erschien auch die erste selbstständig entwickelte, Slackware-unabhängige Distribution unter der von Douglas Adams' »Per Anhalter durch die Galaxis« inspirierten Versionssumme 4.2.

S.u.S.E. Linux, das bald in SuSE Linux, einige Jahre später in SUSE Linux und zu guter Letzt in openSUSE umbenannt wurde, erfreute sich in Deutschland sofort großer Beliebtheit. Die wichtigsten Pluspunkte waren das von Anfang an gut durchdachte und – für damalige Verhältnisse – komfortable Installationsprogramm sowie die ausführlichen deutschsprachigen Handbücher, die besonders wertvoll waren, weil es zu Beginn so gut wie gar keine gedruckte und erst recht keine deutschsprachige Literatur zu Linux gab.

Eine weitere Besonderheit der Distribution ist das seit Langem existierende Installations- und Konfigurationsprogramm YaST (inzwischen in der gleichberechtigt grafik- oder textmodus-orientierten Version 2). Es erleichtert den Umgang mit der Installation und dem System selbst erheblich und wurde, nachdem es 2004 als Open Source freigegeben worden war, sogar auf die eher bei Fortgeschrittenen beliebte Distribution Debian portiert. Ähnlich angenehm ist das seit einigen Jahren verfügbare Tool SaX (SuSE Advanced X Configuration Utility), das die Konfiguration des sogenannten X Window Servers als Grundlage grafischer Oberflächen dialogorientiert und zum Teil automatisiert durchführt.

Da in Europa und besonders in Deutschland vor dem Siegeszug der DSL-Angebote eine gewisse Verbreitung von ISDN bestand, war SuSE die erste Distribution, die funktionierende ISDN-Software für Linux bereitstellte. Auch sonst war der angepasste SuSE-Kernel stets großzügiger mit Treibern ausgestattet als viele andere Distributionen. Version 6.1 vom April 1999 wurde erstmals auch in einer Fassung für eine andere Prozessorarchitektur angeboten, den 64-Bit-Klassiker

DEC Alpha. Während diese Unterstützung durch SUSE Linux inzwischen wieder eingeschlafen ist, wurde die seit Version 6.3 (November 1999) verfügbare PowerPC-Version bis vor kurzem weitergepflegt, sodass SUSE Linux auch als Zweitbetriebssystem auf alten PowerMacs eingesetzt werden kann. Auch die neuen 64-Bit-Prozessoren von AMD und Intel werden seit Längerem unterstützt. Die Versionen 7.0 (September 2000) bis 9.1 (April 2004) wurden jeweils in einer Personal und einer Professional Edition angeboten. Die Personal Edition enthielt vorwiegend Software für Privatanwender. Einige Netzwerk-, Server- und Sicherheitstools fehlten, aber dafür war die Version viel billiger. Im August 2001 erschien zudem die erste Version des SUSE Linux Enterprise Servers (SLES), einer speziell für Firmennetzwerke angepassten Version der Distribution.

Im November 2003 wurde die SUSE Linux AG von dem traditionsreichen US-Netzwerkspezialisten Novell aufgekauft, dessen Serversystem Netware einst den Markt der PC-Netzwerke dominiert hatte. Novell reagierte damit auf den Wandel im Betriebssystemmarkt, in dem Linux und andere UNIX-Varianten eine entscheidende Rolle spielen. Die SUSE-Distribution und SLES wurden beibehalten, hinzukommen Pakete unter der Marke Novell, die mit zusätzlicher Software und speziellem Support ausgestattet sind.

Alle Varianten von SUSE und Novell Linux sind seit einigen Jahren von der Linux Standard Base (LSB) zertifiziert und entsprechen somit einem anerkannten gemeinsamen Standard für Linux-Distributionen. Dies betrifft etwa die Unterteilung des Verzeichnisbaums, den Aufbau von Konfigurationsdateien und andere Konventionen.

1.2.2 openSUSE

Seit August 2005 wird SUSE Linux von dem Community-Projekt *openSUSE* weitergeführt. Anfangs hieß die Distribution selbst noch SUSE Linux; erst mit der Version 10.2 vom Dezember 2006 wurde auch diese in openSUSE umbenannt. Damit schlägt die Distribution einen ähnlichen Weg ein wie RedHat mit dem freien Fedora-Projekt – mit dem Unterschied, dass es von openSUSE auch weiterhin Kaufboxen mit Datenträgern und Handbuch geben wird, die auf dem jeweils aktuellen Stand der Download-Version basieren.

Die Projektwebsite von openSUSE finden Sie unter <http://www.opensuse.org> (siehe Abbildung 1.3). Auf der Startseite finden Sie drei Hauptbereiche:

- ▶ Get it (Download), Abbildung 1.4. Durch Klick auf diesen Link oder durch die direkte Eingabe der Adresse <http://software.opensuse.org> können Sie sich die jeweils aktuelle stabile Version herunterladen. Dies ist zum Zeitpunkt der Veröffentlichung dieses Buches Version 11.2.

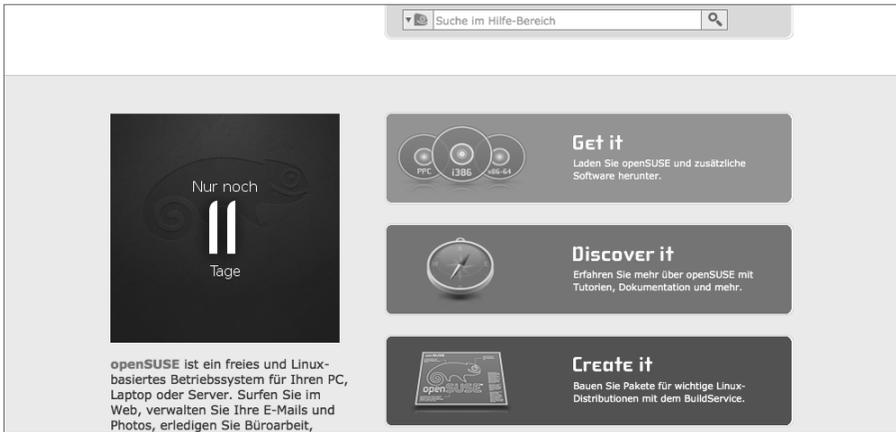


Abbildung 1.3 Die Projektwebsite von openSUSE

Auf der Seite können Sie in einem Formular genau bestimmen, welche Version Sie haben möchten. Zuerst wählen Sie die Prozessorarchitektur: 32-Bit-PC oder 64-Bit-PC. Auf den DVDs zu diesem Buch sind beide Versionen von openSUSE 11.2 enthalten; eine künftige neuere Version müssen Sie sich hier herunterladen.

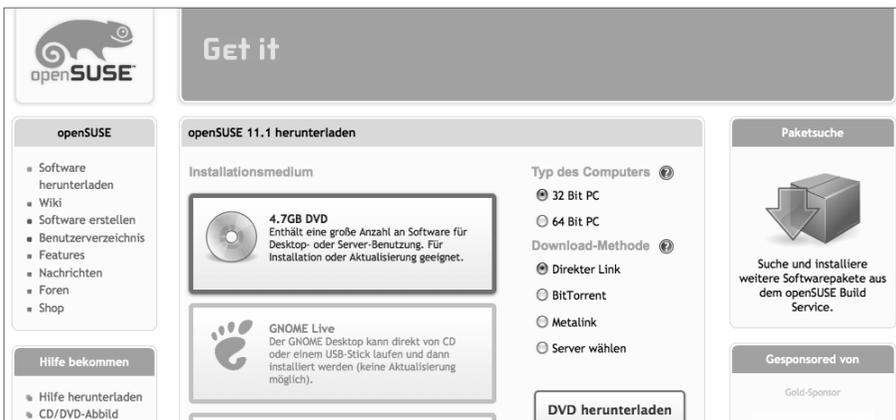


Abbildung 1.4 Die openSUSE-Download-Seite

Als Zweites wird das Installationsmedium gewählt: Die Installations-DVD mit allen Paketen, die KDE- oder GNOME-LiveCD, von der Sie openSUSE direkt booten können, sowie Netzwerk für eine Minimal-Boot-CD (nur knapp 90 Megabyte), die alle Installationspakete aus dem Internet bezieht. Alle drei Installationsvarianten werden im nächsten Kapitel erläutert.

Zuletzt können Sie sich bei der DVD und bei der LiveCD für eine der Download-Optionen BitTorrent und Standard (FTP oder HTTP) entscheiden; bei der Network-CD steht nur Standard zur Verfügung. Für BitTorrent (ein Protokoll für dezentrale Downloads aus mehreren Quellen) brauchen Sie einen speziellen Client – für Windows können Sie sich beispielsweise den Original-BitTorrent-Client herunterladen; bei openSUSE beziehungsweise SUSE Linux gehört seit Version 10.0 KTorrent zum Lieferumfang.

Nachdem Sie diese Auswahlen getroffen haben, beginnt der eigentliche Download. Weiter unten auf der Seite finden Sie noch zwei Zusatz-CDs: eine mit exotischeren Sprachen (»Weitere Sprachen«) und eine mit kommerzieller Software (»NonOSS-CD«) – etwa dem Adobe Flash Player, dem Webbrowser Opera und einigen kommerziellen Treibern.

Nach dem Download müssen Sie die ISO-Images auf DVD beziehungsweise CD brennen. Wie dies unter openSUSE geht, erfahren Sie in Kapitel 9, »Datenträger und Dateisysteme«; unter Windows können Sie Nero oder ein anderes Brennprogramm Ihrer Wahl verwenden.

- ▶ Discover it (Wiki) oder <http://de.opensuse.org> ist das openSUSE-Wiki mit umfangreichen Handbüchern und Zusatzinformationen. Sie können eine Sprache wählen und werden dann entsprechend auf de.opensuse.org für Deutsch, en.opensuse.org für Englisch und so weiter umgeleitet. Die englische Version ist allerdings oft aktueller und ausführlicher. In Abbildung 1.5 sehen Sie die deutsche Startseite. Außerdem erhalten Sie hier die noch nicht stabilen Developer-Versionen; in der Regel hält das openSUSE-Team sich sehr genau an die in der rechten Spalte der Webseite sichtbaren *Projektmeilensteine*.



Abbildung 1.5 Die Startseite des deutschsprachigen openSUSE-Wikis; die (meist aktuellere) englische Version finden Sie unter <http://en.opensuse.org>

- ▶ Create it (Build Service) oder <http://build.opensuse.org> bietet Zugang zum openSUSE Build Service. Dieser stellt eine sehr einfache und praktische Möglichkeit dar, Software-Installationspakete für openSUSE, aber auch für andere Linux-Distributionen zu erstellen. Wie Sie Pakete aus dem Build Service suchen, erfahren Sie in Kapitel 2, »openSUSE installieren«; seine Verwendung für eigene Softwarepakete geht dagegen über den Umfang dieses Buches hinaus.

1.3 Zusammenfassung

Linux steht zweifellos »auf den Schultern von Riesen«: Das Betriebssystem wurde so konstruiert, dass es möglichst kompatibel mit seinem großen Vorbild UNIX ist; durch jahrzehntelange Erfahrungen gehört diese Systemfamilie zur ausgereiftesten und stabilsten Software, die es gibt. Dies soll aber keineswegs die Leistungen von Linus Torvalds und seinen späteren Mitstreitern herabwürdigen, die den Linux-Kernel (entgegen den lächerlichen Behauptungen der SCO Group) von Grund auf neu geschrieben haben. Zusammen mit den freien, UNIX-kompatiblen Systemprogrammen des GNU-Projekts bildet er ein gleichermaßen traditionsbewusstes wie hochmodernes Betriebssystem, das sich zudem als universell bezeichnen lässt: Kaum ein anderes System unterstützt derart viele unterschiedliche Hardwareplattformen.

Die Erfolgsgeschichte von Linux ist auch den Distributoren zu verdanken – Freiwilligen und später auch Firmen, die Softwarepakete sammelten, mit mehr oder weniger benutzerfreundlichen Installationsprogrammen auf Datenträger kopierten und dann verbreiteten. Heute gibt es Hunderte von Linux-Distributionen; jede von ihnen hat ihre eigenen Stärken und Schwächen. Für den Einsatz von openSUSE spricht vieles; einige wichtige Gründe sind die exzellente Hardwareunterstützung, der immense Software-Lieferumfang und das durchdachte Installations- und Konfigurationsprogramm YaST. Dass Novell die Distribution vollkommen freigegeben hat, ist ein guter Grund mehr.

Have a lot of fun!
– Das offizielle Motto von openSUSE

2 openSUSE installieren

Bevor Sie die ersten Schritte mit Ihrem neuen Betriebssystem unternehmen können, müssen Sie es auf Ihrem Rechner installieren. In der Frühzeit von Linux war dies eine Aufgabe für Experten. Wie einfach und unproblematisch die Installation dagegen bei einer modernen Distribution wie openSUSE vonstattengeht, zeigt dieses Kapitel Schritt für Schritt. Eine der erfreulichsten Neuerungen von openSUSE 11 ist übrigens die Tatsache, dass die Installation um ein Vielfaches beschleunigt wurde; während sie früher oft zwei Stunden oder länger dauerte, ist sie heute oft in 30 bis 45 Minuten erledigt – und das auf demselben Rechner.

Zum Ausprobieren: die openSUSE-Live-CD

In den letzten Jahren wurden mehr und mehr sogenannte Live-Linux-Distributionen eingeführt. Hierbei handelt es sich um Linux-Versionen, die direkt vom Datenträger (CD oder DVD) booten und standardmäßig ausschließlich im Arbeitsspeicher ausgeführt werden. Die bekannteste derartige Distribution ist wahrscheinlich Knoppix (<http://www.knopper.net/knoppix>); auch Ubuntu wird standardmäßig als Live-System geliefert. Bei SUSE haben die Live-Datenträger allerdings eine noch längere Tradition; bereits 6.x-Versionen der Distribution enthielten neben den vier bis sechs Installations-CDs auch eine Live-CD¹.

Für openSUSE 11.2 steht ebenfalls eine Live-CD zur Verfügung. Sie können sich das ISO-Image über <http://www.opensuse.de> von einem der zahlreichen Mirrors herunterladen; die Struktur der Site wurde bereits im vorigen Kapitel beschrieben. Bei der Live-CD müssen Sie sich zwischen einem der Desktops KDE und GNOME entscheiden (siehe Kapitel 5, »Grafische Oberflächen«). Das Paket trägt je nach Auswahl einen der folgenden Namen:

- ▶ *openSUSE-11.2-KDE4-LiveCD-i686.iso* – KDE, 32 Bit
- ▶ *openSUSE-11.2-KDE4-LiveCD-x86_64.iso* – KDE, 64 Bit

¹ Dieses »Live File System« setzte allerdings voraus, dass ein Minimalsystem auf der Festplatte installiert wurde. Danach konnte die CD eingebunden werden und bot zahlreiche zusätzliche Programme ohne Installation.

- ▶ *openSUSE-11.2-GNOME-LiveCD-i686.iso* – GNOME, 32 Bit
- ▶ *openSUSE-11.2-GNOME-LiveCD-x86_64.iso* – GNOME, 64 Bit

Nach dem Download können Sie das Image auf eine CD brennen; beispielsweise mit Nero unter Windows oder mit K3b unter Linux (für Letzteres siehe Kapitel 9, »Datenträger und Dateisysteme«). Legen Sie die fertige CD anschließend in Ihr DVD-ROM-Laufwerk ein, führen Sie einen Neustart durch, und erleben Sie einen völlig unverbindlichen openSUSE-Einstieg. Falls Ihr Rechner nicht für das Booten von CD konfiguriert sein sollte, finden Sie weiter unten entsprechende Hinweise.

Auf dem Desktop der Live-Distribution finden Sie ein Icon, mit dem Sie openSUSE auch auf der Festplatte installieren können. Nachdem Sie dies getan haben, funktioniert der Installationsverlauf genau so, wie er in diesem Kapitel beschrieben wird.

2.1 Vorbehandlung eines bereits installierten Windows-Systems

Die überwältigende Mehrheit der neu verkauften PCs enthält seit fast 25 Jahren bereits ab Werk ein Betriebssystem aus dem Hause Microsoft.² Seit einiger Zeit ist es in der Regel Windows 7 (meistens Home Pro, seltener Home Basic). Selbst wenn Sie Ihre Leidenschaft für Linux entdecken (wozu dieses Buch natürlich beitragen möchte), gibt es viele gute Gründe, das vorinstallierte Windows beizubehalten, wenn es schon einmal da ist.

Beispielsweise gibt es leider noch immer zu viele Softwareprodukte, die nicht unter Linux laufen. Hauptsächlich handelt es sich um Spiele, aber auch die alltäglichen Arbeitsprogramme mancher Berufsgruppen sind betroffen, etwa Dreamweaver und Flash für Webdesigner und -entwickler – zumindest für Letzteres gibt es unter Linux noch keinen anständigen Ersatz.

Wenn auf Ihrem Rechner bereits Windows installiert ist, sollten Sie vor der Installation von openSUSE die folgenden Vorbereitungsschritte vornehmen, damit es nicht zu Problemen kommt:

2 Das ist das »Geschäftsgeheimnis« von Microsoft! Stellen Sie sich zum Vergleich vor, jeder verkaufte Neuwagen würde mit Kraftstoff von, sagen wir Esso, vollgetankt (und die Kosten dafür würden – mit einem geringfügigen Rabatt – natürlich Ihnen in Rechnung gestellt, ohne als separater Posten auf derselben zu erscheinen ...)

► *Sichern Sie Ihre Daten*

Bei der möglicherweise erforderlichen Verkleinerung einer Windows-Partition³ durch den openSUSE-Installer kann es in seltenen Fällen zu Datenverlusten kommen. Brennen Sie alle Ihre Arbeitsdateien von der entsprechenden Partition auf eine CD oder DVD, oder kopieren Sie sie notfalls auf eine andere Partition.

► *Machen Sie eine Festplatte oder Partition frei*

Am leichtesten lässt sich ein Linux-System auf Ihrem Windows-Rechner installieren, wenn Sie ihm eine eigene Festplatte oder zumindest eine von Anfang an leere Partition gönnen. Schauen Sie sich dazu unter Arbeitsplatz Ihre vorhandenen Festplatten(-partitionen) an. Ist eine dabei, die nur wenige Daten enthält, und haben Sie auf einer anderen Partition genügend Platz, um die Dateien zu verschieben? Enthält diese Partition zudem keine installierten Programme (unter Windows ist es nicht ohne Weiteres möglich, diese zu verschieben)?

Wenn Sie beide Fragen mit »Ja« beantworten können, gehen Sie wie folgt vor: Erstellen Sie auf der Zielpartition einen neuen Ordner. Markieren Sie auf der Quellpartition alle Hauptverzeichnisse und einzelnen Dateien, und ziehen Sie sie in den neuen Ordner in der anderen Partition. Vergewissern Sie sich, dass alles korrekt kopiert wurde; am besten erstellen Sie zusätzlich eine Sicherungskopie dieser Daten auf CD oder DVD (siehe voriger Schritt).

Zum Schluss können Sie die Partition löschen. Doppelklicken Sie unter START • SYSTEMSTEUERUNG auf das Symbol VERWALTUNG. Führen Sie im erscheinenden Fenster einen Doppelklick auf COMPUTERVERWALTUNG aus, und wählen Sie im Baummenü auf der linken Seite die DATENTRÄGERVERWALTUNG. Es erscheint der Dialog aus Abbildung 2.1 (natürlich mit anderer Partitionierung). Klicken Sie im unteren Bereich die gewünschte, am Laufwerksbuchstaben erkennbare Partition mit der rechten Maustaste an, und wählen Sie PARTITION LÖSCHEN beziehungsweise LOGISCHES LAUFWERK LÖSCHEN aus dem Kontextmenü (der Unterschied zwischen diesen beiden Arten virtueller Laufwerke wird ebenfalls weiter unten erläutert). Nach einer Sicherheitsabfrage werden Sie wahrscheinlich zum (automatischen) Neustart des Systems aufgefordert, weil eine Partition, auf die bereits zugegriffen wurde, im laufenden Betrieb nicht gelöscht werden kann. Sie brauchen anschließend keine neue Partition zu erstellen; dies erledigt die openSUSE-Installation für Sie.

3 Falls Ihnen der Begriff Partition nichts sagt – er wird weiter unten im Rahmen der Installation genau erklärt. Unter Windows erkennen Sie Partitionen am leichtesten daran, dass jede von ihnen einen eigenen Laufwerksbuchstaben (C:, D:, E: und so weiter) trägt.

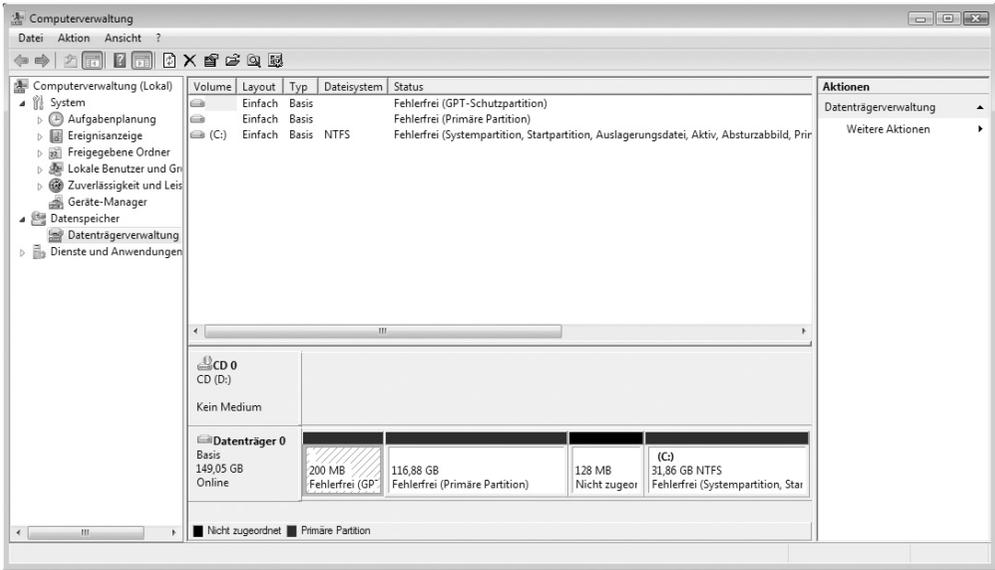


Abbildung 2.1 Verwenden Sie die Datenträgerverwaltung (hier die Windows Vista-Version), um eine Partition für Linux zu löschen.

► *Schaffen Sie sich eine neue Festplatte an*

Wenn Sie keine Festplatte oder Partition frei bekommen, könnten Sie eine neue Festplatte kaufen und einbauen. Beachten Sie dabei, dass diese bei einem Rechner, der älter als etwa sieben Jahre (Baujahr 2002 oder früher) ist, nicht größer als 137 GB (Herstellerrangabe) sein sollte (das Problem, und wie Sie es eventuell durch ein BIOS-Update lösen könnten, wird weiter unten im Kasten »Unterstützung großer Festplatten« beschrieben). Falls die Neuanschaffung einer Festplatte nicht infrage kommt, machen Sie mit dem nächsten Punkt weiter.

► *Verschieben Sie die Auslagerungsdatei*

Wenn weder das Löschen einer bestehenden Partition noch der Neukauf einer Festplatte zur Debatte steht, müssen Sie mithilfe der openSUSE-Installation eine vorhandene Partition verkleinern. Das kann problematisch werden, wenn sich die Windows-Auslagerungsdatei (virtuelle Speichererweiterung) auf der entsprechenden Partition befindet, da diese Datei sich durch das im Anschluss beschriebene Defragmentieren nicht verschieben lässt. Also sollten Sie sie manuell auf einer anderen Festplatte oder Partition anlegen oder – falls Sie nur eine einzige Partition haben – vorübergehend deaktivieren und nach dem Defragmentieren wieder anlegen.

Wählen Sie in beiden Fällen **START • SYSTEMSTEUERUNG**, und doppelklicken Sie auf **SYSTEM**. Klicken Sie auf der Registerkarte **ERWEITERT UNTER SYSTEMLEISTUNG**

auf die Schaltfläche EINSTELLUNGEN. In dem neuen Dialogfeld müssen Sie wiederum auf die Registerkarte ERWEITERT wechseln und unter VIRTUELLER ARBEITS-SPEICHER auf ÄNDERN klicken. Der Dialog aus Abbildung 2.2 wird angezeigt.

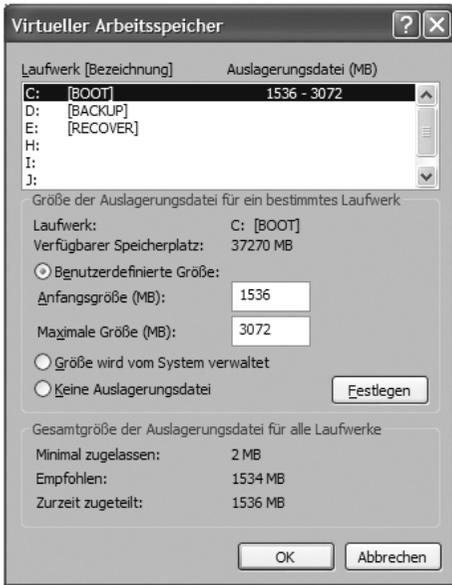


Abbildung 2.2 Verschieben Sie die Windows-Auslagerungsdatei in eine andere Partition.

Wählen Sie in der Liste oben im Dialog das Laufwerk aus, auf dem sich die Auslagerungsdatei zurzeit befindet. Merken Sie sich die bisherigen Größenangaben der Auslagerungsdatei. Wählen Sie die Option KEINE AUSLAGERUNGS-DATEI, und klicken Sie auf FESTLEGEN. Wenn Sie auf einer anderen Partition genügend Platz für eine Auslagerungsdatei haben, können Sie unter LAUF-WERK zu dieser Partition wechseln, die bisherigen Größen eintragen und erneut auf FESTLEGEN klicken. Andernfalls müssen Sie zuerst die Defragmentierung durchführen und diesen Dialog anschließend erneut aufrufen, um die Auslagerungsdatei wieder anzulegen.

► *Defragmentieren Sie die zu verkleinernde Partition*

Falls Sie sich für die Verkleinerung einer Partition entschieden haben, müssen Sie diese zuletzt defragmentieren. Festplatten fragmentieren nämlich im Laufe der Zeit; das bedeutet, dass Dateien immer seltener in einem Stück gespeichert werden, weil Löschvorgänge Lücken hinterlassen, die wieder gefüllt werden; zum Teil bleiben die Lücken auch frei. In der Folge ist der verfügbare Platz für die Linux-Partition viel kleiner als der eigentlich freie Speicher, da Partitionen auf jeden Fall einen kontinuierlichen Block benötigen.

Öffnen Sie den ARBEITSPLATZ, klicken Sie die gewünschte Festplatte beziehungsweise Partition mit der rechten Maustaste an, und wählen Sie EIGENSCHAFTEN. Auf der Registerkarte EXTRAS können Sie die Option JETZT DEFRAGMENTIEREN anklicken. Wählen Sie die passende Festplatte, klicken Sie auf DEFRAGMENTIEREN, und haben Sie Geduld – je nachdem, wie groß und wie voll die Platte ist und wie lange das letzte Defragmentieren her ist, kann es einige Stunden dauern.

Unterstützung großer Festplatten

Schon mehrmals in der PC-Geschichte wurden Festplattengrößen überschritten, die vom BIOS der jeweiligen Rechner nicht mehr akzeptiert werden konnten. In den meisten Fällen ermöglichen Softwaretreiber dennoch den Zugriff auf solche Platten. Wenn die Festplatte allerdings schon bei der Installation des Betriebssystems falsch erkannt wird, kann dies nicht nur diese Installation verhindern, sondern womöglich die Partitionstabelle zerschießen, sodass sich auch bereits vorhandene Betriebssysteme ohne vorsichtige Reparatur nicht mehr booten lassen.

Dieses Problem betrifft Sie in der Regel nur, wenn Sie eine Festplatte von über 128 Gigabyte Größe (Herstellerangabe: 137 GB = 137 Milliarden Byte) in einen PC einbauen, der älter als sechs bis sieben Jahre ist und bisher keine solche Platte besaß. Wenn die Größe dieser Festplatte zu Beginn der Installation mit etwa 130 GByte angezeigt wird, obwohl Sie wissen, dass sie erheblich größer ist, müssen Sie die Installation umgehend abbrechen; Sie dürfen unter keinen Umständen zulassen, dass der openSUSE-Installer versucht, eine Partitionstabelle auf diese Festplatte zu schreiben.

Die genannte Grenze entsteht, weil für den Zugriff auf größere Platten das sogenannte 48-Bit-LBA benötigt wird. LBA (Logical Block Addressing) ist die moderne Methode der Festplattenadressierung; das klassische 28-Bit-LBA kann mit Festplatten über der genannten Größe nicht umgehen. openSUSE selbst unterstützt 48-Bit-LBA, der Installer aber leider noch nicht.

Wenn Sie an diese Grenze stoßen, können Sie versuchen, ein *BIOS-Update* durchzuführen – viele PC- und Mainboard-Hersteller bieten solche Aktualisierungen zur Anpassung ihrer Geräte an neuere Hardware an.

Nähere Informationen über 48-Bit-LBA bietet die Website <http://www.48bitlba.com>.

2.2 Vorbereitung der Installation

Bevor die eigentliche Installation, das heißt das Kopieren der Softwarepakete auf Ihren Rechner, stattfinden kann, müssen Sie diverse Vorbereitungen treffen: Sie müssen überprüfen, von welchen Datenträgern Ihr PC booten kann, und dies gegebenenfalls ändern; nach dem Start vom Installationsdatenträger müssen Sie anschließend einige vorbereitende Fragen beantworten.

2.2.1 PC und Boot-Datenträger aufeinander abstimmen

Falls Ihr PC nicht von DVD beziehungsweise CD booten kann, müssen Sie sich zuerst um dieses Problem kümmern. Bei so gut wie allen PCs, die in den letzten fünf Jahren gebaut wurden, ist dies kein grundsätzliches Problem, sondern eine Frage der BIOS-Einstellung. Starten Sie dazu den Rechner neu, und betätigen Sie die Taste für das BIOS-Setup; dies ist bei den meisten Rechnern **F1**, **Entf** oder **F2**.

Je nach BIOS-Variante heißt die Einstellung für die Datenträgerreihenfolge beim Booten etwas anders. Bei einem halbwegs modernen Award-BIOS lautet der zuständige Hauptmenüpunkt **ADVANCED BIOS FEATURES**. Stellen Sie die Boot-Reihenfolge so ein, dass das gewünschte DVD-Laufwerk vor der Festplatte an die Reihe kommt. Wählen Sie anschließend die Option, die die Änderungen speichert und das Setup verlässt – beim Award-BIOS **SAVE & EXIT SETUP**.

2.2.2 Die Installation einleiten

Dieses Buch enthält zwei openSUSE-DVDs – eine mit der 32-Bit- und die andere mit der 64-Bit-Version von openSUSE. Legen Sie die zu Ihrem Rechner passende DVD beziehungsweise Ihr jeweiliges openSUSE-Installationsmedium in Ihr DVD-Laufwerk ein, und starten Sie Ihren Rechner neu. Nach den üblichen BIOS-Meldungen und einer kurzen Willkommensnachricht sehen Sie das Bootmenü. Drücken Sie zunächst **F2**, um die Sprache zu wechseln. Drücken Sie die Pfeiltasten, bis **DEUTSCH** ausgewählt ist, und anschließend **↵**, um Ihre Auswahl zu bestätigen. Anschließend sieht der Bildschirm so aus wie in Abbildung 2.3.

Drücken Sie als Nächstes **↓** und dann **↵**, um den gewünschten Eintrag zu wählen. Die einzelnen Punkte in diesem Menü bedeuten Folgendes:

- ▶ **VON FESTPLATTE BOOTEN** ist die Vorauswahl – wenn Sie die DVD im Laufwerk vergessen, wird auf diese Weise nicht automatisch davon gestartet.
- ▶ **INSTALLATION** führt in den meisten Fällen zu dem in diesem Kapitel gesteckten Ziel: Diese Option führt die Installation mit Standardvoreinstellungen durch.
- ▶ **INSTALLIERTES SYSTEM REPARIEREN** führt alle unten beschriebenen Installationschritte durch, aber nicht als Neuinstallation, sondern zur Reparatur eines bereits installierten openSUSE-Systems.
- ▶ **RETTUNGSSYSTEM** startet keine Installation, sondern bootet die in Kapitel 9, »Datenträger und Dateisysteme«, beschriebene *Rettungskonsole*, die Ihnen helfen kann, falls Sie versehentlich Partitionen zerschossen haben oder wenn nach einem Stromausfall Dateisystemfehler vorliegen.

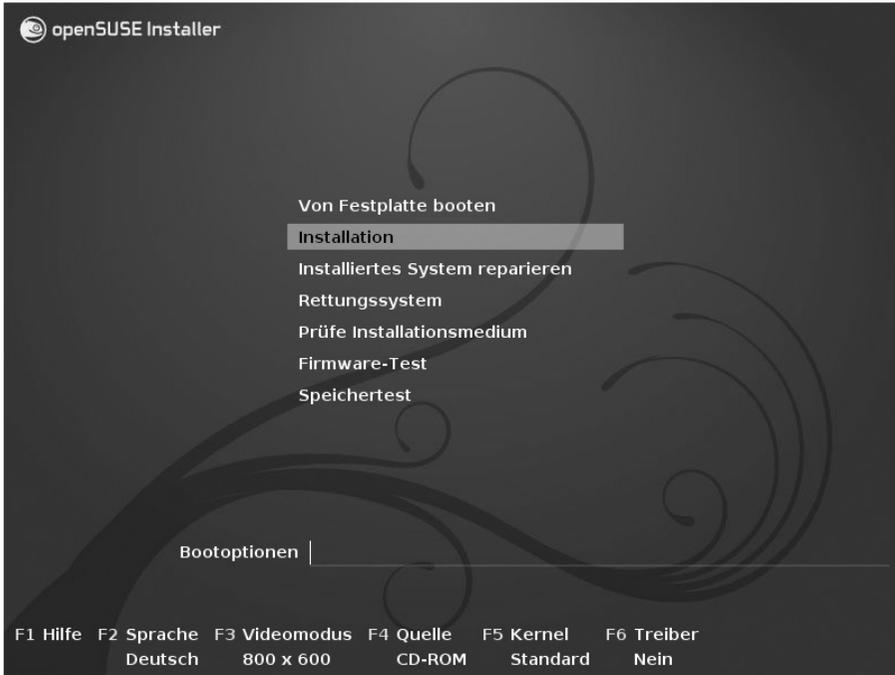


Abbildung 2.3 Der Auswahlbildschirm nach dem Booten der openSUSE-Installations-DVD

- **PRÜFE INSTALLATIONSMEDIUM** – hier haben Sie die Möglichkeit, die Integrität Ihrer Installationsmedien zu überprüfen. Dies ist immer dann zu empfehlen, wenn Sie CD- oder DVD-Images aus dem Internet heruntergeladen und selbst gebrannt haben, da sie durch Übertragungsfehler falsche Daten enthalten können. Zudem stammen sie in der Regel von einem Mirror und könnten daher theoretisch sogar absichtlich Schadsoftware enthalten. Allerdings brauchen Sie auch in diesem Fall nicht unbedingt die vergleichsweise langsame Prüfung während der Installation durchzuführen, sondern können die Integrität mithilfe von MD5-Prüfsummen berechnen, noch bevor Sie die Images auf Datenträger brennen. Für diesen Test wird aus jedem ISO-Image eine Prüfsumme berechnet, die Sie mit einem offiziellen Wert auf der openSUSE-Website vergleichen müssen. Die Vergleichswerte finden Sie auf den jeweiligen im vorigen Kapitel beschriebenen Download-Seiten.

Wenn Sie bereits Linux oder eine andere UNIX-Variante einsetzen, können Sie einfach ein Terminalfenster öffnen und im richtigen Verzeichnis `md5sum` Paketname eingeben, um die MD5-Prüfsumme des gewählten Pakets zu ermitteln. Hier ein Beispiel für das Image der openSUSE-Installations-DVD für x86_64-Plattformen:

```
$ md5sum openSUSE-11.2-DVD-x86_64.iso
6a09295e34dc030319d040f67f4742c6
*openSUSE-11.2-DVD-x86_64.iso
```

Vergleichen Sie die 32-stellige Hexadezimalzahl mit dem Wert von der openSUSE-Website – am einfachsten, indem Sie ihn in der Textdatei unter die entsprechende Zeile kopieren. Im obigen Beispiel stimmt der Wert übrigens.

Wenn Sie bisher nur ein Windows-System verwenden, steht Ihnen das Kommandozeilentool `md5sum` standardmäßig nicht zur Verfügung. Sie können aber leicht eine Windows-Version herunterladen; auf der Website zum Buch (<http://buecher.lingoworld.de/opensuse>) steht ein entsprechender Link. Kopieren Sie das Programm `md5sum.exe` in Ihr Windows-Systemverzeichnis (etwa C:\WINDOWS), öffnen Sie die Eingabeaufforderung (zum Beispiel START, dann `cmd` ins Suchfeld eingeben und  drücken) und geben Sie den oben gezeigten Befehl ein.

Wenn Sie die Datenträger aus einer gekauften openSUSE-Box oder die Buch-DVD verwenden, können Sie sich diesen Schritt dagegen sparen, weil diese bereits überprüft wurden. Um völlig sicherzugehen, können Sie die Prüfung allerdings auch in diesem Fall durchführen, zumal sie – gemessen an der Gesamtdauer der Installation – nicht allzu lange dauert.

Falls Sie sich für die Prüfung entschieden haben, wird der Kernel geladen; anschließend findet die eigentliche Prüfung im Textmodus statt. Haben Sie etwas Geduld; die Prüfung dauert mehrere Minuten. Wenn alles in Ordnung ist, erscheint die Meldung KEINE FEHLER GEFUNDEN. Andernfalls gibt es eine Fehlermeldung, und Sie müssen den entsprechenden Datenträger wahrscheinlich erneut herunterladen und/oder brennen. Nach einer erfolgreichen Prüfung wird automatisch der Rechner neu gestartet.

- ▶ **FIRMWARE-TEST** aktiviert ein spezielles Programm, um das BIOS des Rechners zu überprüfen. Manche Inkompatibilitäten, die bei der Installation ansonsten auftreten, lassen sich auf diese Weise lösen.
- ▶ **SPEICHERTEST** überprüft den Arbeitsspeicher des Rechners – einige unerklärliche Fehler bei der Installation oder im laufenden Betrieb kommen durch defekte Speichermodule zustande.

Unter **BOOTOPTIONEN** können Sie spezielle Kernelmodule angeben; dies kann erforderlich sein, wenn Sie exotischere Hardware verwenden. Einige denkbare Werte lernen Sie in Kapitel 6, »Wichtige Hardware konfigurieren«, kennen.

Am Fuß des Bildschirms finden Sie einige Spezialoptionen, die Sie mit den Tasten  bis  erreichen:

- ▶ **[F1]** (Hilfe) startet in einem kleinen Fenster die interaktive Installationshilfe. Sie enthält diverse (interne) Links, die Sie mit den Pfeiltasten und **[↩]** aktivieren können.
- ▶ Mit **[F2]** (Sprache) können Sie, wie oben bereits erwähnt, die Installations-sprache ändern.
- ▶ **[F3]** (Videomodus) erlaubt die Auswahl der Bildschirmauflösung für die grafi-sche Installation. Sollten Sie eine ältere, sehr neue oder seltene Grafikkarte haben, die während der Installation nicht erkannt wird, können Sie sich hier auch für den `TEXTMODUS` entscheiden.
- ▶ Die Taste **[F4]** (Quelle) ermöglicht einen Wechsel der Installationsquelle. Wenn Sie, wie hier beschrieben, von der DVD gebootet haben, ist diese vor-ausgewählt. Daneben stehen die `FESTPLATTE` sowie diverse Netzwerk- und Internetoptionen zur Auswahl.
- ▶ Mit **[F5]** (Kernel) können Sie auswählen, dass zur Installation ein spezieller Linux-Kernel gebootet werden soll. Vorausgewählt ist `STANDARD`. Die anderen Optionen sind:
 - ▶ `KEIN ACPI` – diese Option braucht Sie nur zu interessieren, wenn die nor-male Installation Probleme bereitet. `ACPI` (Advanced Configuration and Power Interface) ist die Power-Management-Funktion moderner PCs und Notebooks. Sollte die Standardinstallation mit einer entsprechenden Feh-lermeldung abbrechen, sollten Sie zuerst überprüfen, ob `ACPI` in Ihrem BIOS eingeschaltet ist – starten Sie dazu den Rechner neu, und drücken Sie die weiter oben erwähnte Taste für das BIOS-Setup. Wenn `ACPI` aktiv ist und die Probleme trotzdem weiterbestehen, können Sie die Installation mit dieser Auswahl durchführen. Einige besondere Systemfunktionen wie Standby oder die gezielte Abschaltung einzelner Hardwarekomponenten nach einer bestimmten inaktiven Zeitspanne stehen Ihnen dann allerdings nicht zur Verfügung.
 - ▶ `KEIN LOKALES APIC` schaltet den modernen Interrupt-Controller `APIC` (Advanced Programmable Interrupt Controller) aus, der Ein- und Ausgabe- operationen auf neueren PCs beschleunigt. Auch dies brauchen Sie nur in Erwägung zu ziehen, falls die normale Installation mit einer entsprechen- den Meldung fehlschlägt.
 - ▶ `SICHERE EINSTELLUNGEN` deaktiviert `ACPI`, `APIC` und einige andere moderne Rechner- und BIOS-Features. Dies hilft manchmal bei einigen besonders alten PCs, die die Installation ansonsten komplett verweigern.
- ▶ **[F6]** (Treiber) erlaubt schließlich das Laden von Drittanbieter-Treibern aus einer `DATEI` oder einer `URL`. Wählen Sie diese Option, falls Sie ein Gerät

haben, für das ein eigener, optimierter Treiber des Herstellers zur Verfügung steht. Im Prinzip brauchen Sie hier nur Treiber für installationsrelevante Hardware bereitzustellen, beispielsweise exotischere SCSI-Controller für Ihre Festplatten. Spezialisierte Treiber für 3-D-beschleunigte Grafikkarten und dergleichen können Sie auch nach der Installation einrichten (siehe besonders Kapitel 5, »Grafische Oberflächen«, und 6, »Wichtige Hardware konfigurieren«).

2.2.3 Lizenzvereinbarung

Nach der Fortschrittsanzeige LADEN DES LINUX-KERNELS und dem Start der grafischen Benutzeroberfläche wird das eigentliche Installationsprogramm *YaST* (»Yet Another Setup Tool«) gestartet, das Ihnen im nächsten Kapitel auch als Steuerzentrale des fertig installierten Betriebssystems wieder begegnen wird. Hier gelangen Sie zunächst auf den Bildschirm WILLKOMMEN. Dort können Sie bei Bedarf nochmals SPRACHE und TASTATURBELEGUNG auswählen; in der Regel entspricht diese aber bereits Ihrer Auswahl im Bootmenü.

Darunter befindet sich die LIZENZVEREINBARUNG. Wie bereits im vorigen Kapitel klargestellt, bedeutet freie Software nicht, dass es keinerlei Regeln für ihren Einsatz gäbe. openSUSE besteht aus dem Linux-Betriebssystem und seinen Tools sowie unzähligen Anwendungsprogrammen, die unter verschiedenen freien Softwarelizenzen (GNU GPL, BSD-Lizenz, Apache-Softwarelizenz und so weiter) stehen; auch einige kommerzielle Softwarepakete mit eigenen Lizenzbedingungen sind enthalten.

Es ist klar, dass kein normaler Mensch Zeit hat, sich Lizenzvereinbarungen Wort für Wort durchzulesen; zudem fehlt den meisten von uns ohnehin das juristische Fachwissen, um ihre Gültigkeit und alle ihre Folgen einschätzen zu können. Kurz überfliegen sollten Sie den Text dennoch, um herauszufinden, ob er nichts enthält, womit Sie partout nicht leben können – in diesem Fall sollten Sie nämlich von der Installation der Software absehen.

Die wichtigste Aussage der Lizenz für den Alltag lautet: »Sie können unbegrenzt Kopien der Software innerhalb Ihrer Organisation erstellen und verwenden. Von Versionen, die die Buchstaben ›OSS‹ im Produktnamen führen (mit Ausnahme der Versionen mit ›NON-OSS‹ im Produktnamen) dürfen Sie unbegrenzt Kopien erstellen und außerhalb Ihrer Organisation verteilen.« (Der Unterschied bezieht sich darauf, dass die kommerziellen »NON-OSS«-Pakete nicht frei sind und daher nur innerhalb derselben Firma oder Institution oder im selben Haushalt verteilt werden dürfen. Andererseits ist die Frage, worin der Unterschied zwischen meh-

renen separaten Downloads der Installationsmedien oder beliebig vielen Kopien desselben Mediums besteht.

In den allermeisten Fällen sollte es auch mit den restlichen Bedingungen der Lizenzvereinbarung keine Probleme geben, sodass Sie die Option **JA, ICH AKZEPTIERE DIE LIZENZVEREINBARUNG** aktivieren und **WEITER** drücken können.

2.2.4 Neuinstallation oder Aktualisierung?

Im nächsten Schritt können Sie sich zwischen **NEUINSTALLATION**, **AKTUALISIERUNG** oder **REPARATUR DES INSTALLIERTEN SYSTEMS** entscheiden. Wählen Sie hier die erste Option, um openSUSE neu zu installieren – entweder auf einem Rechner, auf dem es sich noch nicht befindet, oder als Zweitinstallation neben einer älteren Version. Wenn Sie **AKTUALISIERUNG** wählen, analysiert der Installer das vorhandene System; in den meisten Fällen ist das Upgrade relativ unproblematisch möglich, allerdings sollten Sie vorher die wichtigsten Daten sichern. Die **REPARATUR DES INSTALLIERTEN SYSTEMS** schließlich repariert einen defekten Bootloader, ergänzt versehentlich gelöschte Pakete oder dergleichen.

Der Ablauf der Installation ist ansonsten in allen drei Fällen identisch; Besonderheiten bei der Paketauswahl werden weiter unten erläutert. Schließen Sie auch diesen Schritt mit einem Klick auf **WEITER** ab.

Optional können Sie an dieser Stelle den Punkt **ADD-ON-PRODUKTE AUS SEPARATEN MEDIEN EINSCHLIESSEN** aktivieren; hier lässt sich beispielsweise die im vorigen Kapitel angesprochene CD mit der Unterstützung für zusätzliche Sprachen einbinden.

Die standardmäßig aktivierte Option **BENUTZE AUTOMATISCHE KONFIGURATION** führt die weiter unten beschriebene Grundkonfiguration nach der eigentlichen Installation weitgehend automatisch durch.

2.2.5 Zeitzone, Datum und Uhrzeit einstellen

Auf dem nächsten Bildschirm, **UHR UND ZEITZONE**, werden Sie aufgefordert, die Zeitzone Ihres Rechners zu wählen sowie Datum und Uhrzeit zu überprüfen, die beim Booten aus der Systemuhr ausgelesen werden. Wählen Sie unter **REGION** Ihren Kontinent – für die meisten Leser dieses Buches ist dies wahrscheinlich **EUROPA**. In der Liste **ZEITZONE** werden daraufhin die passenden Länder angeboten; wählen Sie hier Ihr Land oder – falls es sich nicht in der Liste befindet – ein Land, von dem Sie wissen, dass es in derselben Zeitzone liegt und dieselben Gepflogenheiten bezüglich Sommer- und Winterzeit hat wie Ihres.

Wichtig ist die Option `RECHNERUHR IST AUF UTC GESTELLT`: Manche User stellen die Uhr ihres Computers auf UTC ein, sodass das Betriebssystem die Differenz zur eingestellten Zeitzone berechnen muss. Sollte dies bei Ihnen der Fall sein, können Sie diese Option aktivieren; andernfalls muss sie ausgeschaltet bleiben.

Beachten Sie in diesem Zusammenhang, dass UTC (Coordinated Universal Time) und GMT (Greenwich Mean Time) eigentlich nicht ganz dasselbe sind: GMT ist die Uhrzeit am nullten Längengrad, der durch Greenwich in London verläuft; wie alle Länder der EU und viele andere stellt aber auch Großbritannien die Uhren im Sommer um eine Stunde vor. UTC ist dagegen die durch Atomuhren definierte Weltzeit, die nur im Winter ungefähr gleich GMT ist. In Deutschland, Österreich und der Schweiz gilt daher vom letzten Sonntag im März bis zum letzten Sonntag im Oktober UTC plus zwei Stunden, in der übrigen Zeit des Jahres dagegen UTC+1.

Unter `ZEIT UND DATUM` können Sie schließlich die Schaltfläche `ÄNDERN` betätigen, falls die Rechnerzeit nicht stimmen sollte. Zu einem späteren Zeitpunkt erfahren Sie, wie Sie Systemdatum und -uhrzeit mit einem Zeit-Server (NTP) im Internet verbinden können, um sie automatisch zu synchronisieren. Bestätigen Sie Ihre Einstellungen schließlich mit `WEITER`.

2.2.6 Desktop auswählen

Der letzte wichtige Vorbereitungsschritt besteht darin, sich eine Desktop-Umgebung für die grafische Benutzeroberfläche auszusuchen. Neben den beiden wichtigsten Desktops GNOME und KDE 4 (die ausführlich in Kapitel 5, »Grafische Oberflächen«, besprochen werden) finden Sie unter `ANDERE` noch die Auswahlmöglichkeiten XFCE, `MINIMALES GRAFISCHES SYSTEM` und `TEXTMODUS`. Gegenüber den »Full-Service-Desktops« GNOME und KDE steigern diese Optionen die Performance Ihres Systems und sind daher vor allem für Serverrechner geeignet.

In alten SUSE-Linux-Versionen wurde standardmäßig KDE installiert. Novell wollte mit Version 11.0 ursprünglich auf GNOME als Standard-Desktop umsteigen; nach zahlreichen User-Protesten haben die Entwickler sich aber letzten Endes entschlossen, die Auswahl schon bei der Installation den Benutzern zu überlassen.

Ihre Entscheidung an dieser Stelle bedeutet aber keineswegs, dass Sie sich exklusiv für eine Desktop-Umgebung entscheiden müssen. Wenn Sie die eigentliche Installation in Gang setzen, können Sie zusätzlich den jeweils anderen Desktop und/oder einige alternative Window-Manager auswählen. Hier bestimmen Sie also hauptsächlich, welcher Desktop beim Booten automatisch gestartet werden soll, und selbst diese Einstellung können Sie später bei Bedarf ändern.

Möglicherweise haben Sie bereits einen Favoriten unter den beiden Desktops, dann wählen Sie ihn einfach aus. Besonders Anwender mit Windows-Kenntnissen und wenig Linux-Erfahrung könnten mit KDE vermutlich besser zurechtkommen – dies ist absolut nicht abwertend gemeint; ich persönlich bevorzuge ebenfalls KDE.

Wenn Sie sich nach diesen Vorgaben oder nach einem Blick in Kapitel 5, »Grafische Oberflächen«, für einen Desktop entschieden haben, aktivieren Sie die entsprechende Option, und drücken Sie WEITER.

2.2.7 Die Partitionierung

Der nächste Punkt in den Installationsvoreinstellungen heißt FESTPLATTE und betrifft die *Partitionierung*. Dieser Vorgang unterteilt einen Datenträger, meist eine Festplatte, in mehrere unabhängige Dateisysteme, die *Partitionen* genannt werden. Bei einem reinen Windows-Rechner ist die Partitionierung in der Regel optional; es ist möglich, wenn auch nicht empfehlenswert, die gesamte Festplatte als eine große Partition zu betreiben. Für ein vernünftig konfiguriertes Linux-System ist diese Vorgehensweise nicht möglich, weil der Plattenplatz für die Auslagerung von Arbeitsspeichereinhalten (sehr wichtig für das korrekte Funktionieren eines modernen Betriebssystems) hier nicht wie unter Windows in einer Auslagerungsdatei, sondern in einer eigenständigen Auslagerungspartition angelegt wird.

Darüber hinaus ist es empfehlenswert, System und Anwendungsprogramme aus Sicherheitsgründen auf einer anderen Partition unterzubringen als die Benutzerdaten, die auf allen UNIX-artigen Systemen in sogenannten Home-Verzeichnissen gespeichert werden: Auf diese Weise können Anwenderfehler nicht ohne Weiteres die System- und Anwendungsinstallation in Mitleidenschaft ziehen.

Zunächst wird ein automatisch generierter *Partitionierungsvorschlag* angezeigt. Bei einem Rechner ohne Betriebssystem wird die Unterteilung des gesamten verfügbaren Platzes in drei Partitionen vorgeschlagen: die doppelte Größe des physischen Arbeitsspeichers als Swap-Partition (Auslagerungspartition), etwa 40 Prozent des verfügbaren Platzes für die sogenannte *Root-Partition* / (System und Programme) und circa 60 Prozent für die User-Verzeichnisse unter */home*.

Bei einem Rechner mit vorhandenem Betriebssystem wird eine freie Festplatte oder Partition gewählt; falls keine vorhanden ist, schlägt das Installationsprogramm die automatische Verkleinerung einer vorhandenen Partition vor. Der entsprechend frei gewordene Platz wird anschließend genauso aufgeteilt wie bei einem freien Rechner.

Prinzipiell können Sie sich zwischen den Partitionierungsvarianten PARTITIONSBASIEREND und LVM BASED entscheiden. Die standardmäßig ausgewählte Option PARTITIONSBASIEREND partitioniert jeden physischen Datenträger getrennt. Der modernere *Logic Volume Manager* (LVM) erstellt dagegen auf Wunsch aus mehreren Datenträgern virtuelle Geräte namens */dev/system* (bei mehreren nummeriert), die unabhängig von den Grenzen der tatsächlichen physischen Festplatten partitioniert werden können. Partitionen können also kleiner oder auch größer sein als die eigentlichen Festplatten.

Als Dateisystem stellt openSUSE seit Version 11.2 automatisch EXT4 ein (zwischen 10.2 und 11.1 war der Vorgänger Ext3 vorausgewählt), früher wurde dagegen REISERFS voreingestellt. Wenn Sie sich entscheiden, den Vorschlag nicht unverändert anzunehmen, können Sie auch diese Einstellung modifizieren. Weiter unten erhalten Sie eine Kurzübersicht über die verfügbaren Dateisysteme; in Kapitel 9, »Datenträger und Dateisysteme«, werden sie ausführlicher diskutiert.

Sie haben drei Möglichkeiten, mit der Partitionierungsvorgabe umzugehen:

- ▶ Klicken Sie einfach auf WEITER, um die Partitionierung wie empfohlen durchzuführen.
- ▶ Wenn Sie sich für die Auswahl PARTITIONSAUFBAU ERSTELLEN entscheiden, können Sie alle Einstellungen ohne Vorgabe selbst vornehmen. Dazu ist ein gewisses Hintergrundwissen über Partitionstabellen und Dateisysteme erforderlich. In diesem Buch kommen diese Themen erst in Kapitel 9 an die Reihe, sodass Sie von dieser Option die Finger lassen sollten, wenn Sie nicht ganz genau wissen, was Sie tun. Wenn Sie einen ganz leeren Rechner zur Verfügung haben, können Sie natürlich nach Herzenslust experimentieren; viel Zeit sollten Sie in diesem Fall auch haben, denn möglicherweise müssen Sie die Installation bei zu spät erkannten Problemen wiederholen. Auf einem Computer mit vorhandenem Betriebssystem sollten Sie diese Option dagegen *auf keinen Fall* wählen, solange Sie kein Experte sind, denn es kann bei den geringsten Fehlern zu Datenverlusten kommen.
- ▶ Die Option PARTITIONSAUFBAU BEARBEITEN ermöglicht Ihnen einzelne Änderungen an der vorgeschlagenen Partitionierung. Anhand dieser Auswahlmöglichkeit wird hier der eigentliche Partitionsdialog beschrieben.

Wenn Sie sich für eine der beiden Änderungsoptionen entschieden haben, gelangen Sie auf den in Abbildung 2.4 gezeigten Bildschirm FESTPLATTE VORBEREITEN: EXPERTENMODUS (beim völlig benutzerdefinierten Setup müssen Sie sich zuvor noch zwischen den verfügbaren Festplatten entscheiden und auf WEITER klicken). Je nach vorheriger Auswahl sind hier bereits Partitionierungsvorschläge eingetragen oder nicht.

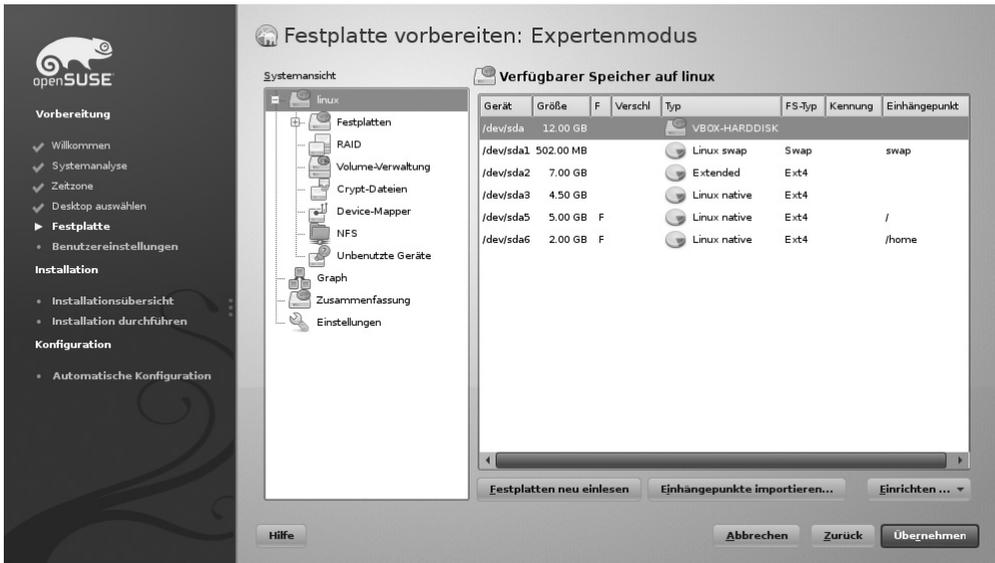


Abbildung 2.4 Der detaillierte Partitionsdialog »Festplatte vorbereiten (Expertenmodus)«

Der größte Teil des Bildschirms besteht aus einer Tabelle, in deren Zeilen die verfügbaren Festplatten beziehungsweise die bereits vorhandenen Partitionen eingetragen sind. Sie enthält von links nach rechts folgende Spalten (einige von ihnen werden nur angezeigt, wenn Sie **DETAILS ANZEIGEN** aktivieren):

► **GERÄT**

Der Name (genauer gesagt Pfad) der jeweiligen Festplatte oder Partition. Bereits in der Einleitung haben Sie erfahren, dass in UNIX-Systemen »alles eine Datei« ist, so auch sämtliche Hardwaregeräte. Diese liegen im speziellen Geräteverzeichnis `/dev`. EIDE- oder S-ATA-Festplatten heißen *hda*, *hdb* und so weiter; die entsprechenden Partitionen sind durchnummeriert (*hda1*, *hda2* etc.). SCSI-Festplatten wie in der Abbildung⁴ heißen entsprechend *sda*, *sdb* und so weiter. Jede PC-Festplatte kann bis zu vier Partitionen enthalten, die von 1 bis 4 durchnummeriert werden. Eine dieser Partitionen kann eine **ERWEITERTE PARTITION** sein, die wiederum beliebig viele **LOGISCHE PARTITIONEN** enthalten darf; die normalen Partitionen, für die das nicht gilt, werden als **PRIMÄRE PARTITIONEN** bezeichnet. Die Nummerierung der logischen Partitionen beginnt stets bei 5 (zum Beispiel *hda5* oder *sdb5*), egal, wie viele primäre Partitionen neben der erweiterten vorhanden sind.

⁴ In Wirklichkeit handelt es sich um eine virtuelle Festplatte unter VMware Workstation; nur so ließen sich problemlos Screenshots der Installation anfertigen.

- ▶ **GRÖSSE**
Hier können Sie ablesen, wie groß die jeweilige Festplatte, Partition oder der unpartitionierte Bereich ist; je nach Größenbereich wird der Wert in Megabyte oder Gigabyte angegeben.
- ▶ **F**
In dieser Spalte zeigt der Wert F an, dass die entsprechende Partition neu angelegt und daher *formatiert* (gelöscht und mit einem neuen Dateisystem versehen) wird. Ein C gibt an, dass die Partition zusätzlich *verschlüsselt* (encrypted) wird.
- ▶ **TYP**
Hier wird der jeweilige bestehende oder vorgesehene Partitionstyp eingetragen. Der Partitionsvorschlag trägt in der Regel nur zwei mögliche Typen ein: LINUX SWAP für die Auslagerungspartition und LINUX NATIVE (EXT4) für Linux-Standardpartitionen mit dem konkreten Dateisystem Ext4.
- ▶ **EINHG.**
Bereits im vorigen Kapitel wurde erwähnt, dass es unter Linux keine Laufwerksbuchstaben oder dergleichen gibt, sondern dass die Laufwerke und Partitionen an frei wählbaren Stellen in einen einzigen Verzeichnisbaum eingehängt (gemountet) werden. Der *Mountpoint* ist also das Verzeichnis, unter dem die jeweilige Partition im Verzeichnisbaum zur Verfügung steht. Wie bereits beschrieben, wird jedes Volume zusätzlich durch eine Gerätedatei unter */dev* repräsentiert, die auch ohne Mounten bereitsteht. Über diesen Verzeichniseintrag können allerdings keine normalen Dateioperationen durchgeführt werden, sondern man könnte lediglich auf Blockebene auf die tatsächlich gespeicherten Bytes zugreifen, was im Alltag weder erforderlich noch wünschenswert ist.
- ▶ **EINHÄNGEN MITTELS**
Der Eintrag in dieser Spalte gibt an, wodurch die Partition beim Mounten identifiziert wird: I steht für die Geräte-ID, K (Kernel-Name) ist die */dev*-Gerätedatei; L (Label) ist eine optionale, selbstgewählte Bezeichnung; U (**U**niversally **U**nique **I**dentifier oder kurz UUID) steht für eine weltweit einmalige, aber ebenfalls optionale Nummer; P (Path) schließlich ist der Gerätepfad, der weiter unten als letzter Eintrag der Tabelle beschrieben wird. In aller Regel gibt es keinen Grund, an der Voreinstellung I etwas zu ändern.
- ▶ **ANFANG**
Diese Spalte gibt den Startzylinder der Partition an. Das Partitionierungstool verwendet die sogenannte LBA-Methode zur Adressierung der Festplattenbereiche: Sie werden unabhängig von der konkreten Einzelplatte und der Spur einfach durchnummeriert. Deshalb genügt jeweils eine einzelne Nummer zur

Kennzeichnung der Start- und Endbereiche. Einzelheiten zur Festplatten-adressierung erfahren Sie in Kapitel 9, »Datenträger und Dateisysteme«.

- ▶ **ENDE**
Hier wird der Endzylinder der Partition angegeben.
- ▶ **BELEGT VON**
Eventuelle zusätzliche Informationen – meist nur, wenn die Partition durch ein anderes Betriebssystem oder Programm angelegt wurde.
- ▶ **LABEL**
Eine frei gewählte Partitionsbezeichnung, die Sie in den weiter unten beschriebenen Dialogen eingeben können.
- ▶ **GERÄTE-ID**
Die bereits erwähnte eindeutige ID einer Festplatte oder Partition.
- ▶ **GERÄTE-PFAD**
Eine alternative Bezeichnung für die Festplatte oder Partition, die die konkrete physische Anschlusslogik wiedergibt.

Wenn Sie die hier beschriebene Modifikation des Partitionierungsvorschlags gewählt haben, sind bereits alle Informationen über die künftigen Partitionen eingetragen. Klicken Sie eine von ihnen an, und drücken Sie auf **BEARBEITEN**, um die entsprechenden Einstellungen zu ändern. Der Dialog aus Abbildung 2.5 (hier am Beispiel der Partition mit dem Mountpoint `/`) wird angezeigt.



Abbildung 2.5 Der Dialog zum Ändern der Einstellungen einer Partition

Unter `FORMATIERUNGSOPTIONEN` können Sie zunächst einmal einstellen, ob und wie die Partition formatiert werden soll. Die Option `NICHT FORMATIEREN` ist einerseits für bestehende Partitionen geeignet, die nicht gelöscht oder verändert werden sollen, andererseits aber auch für leere Partitionen, falls Sie diese später selbst formatieren möchten. Wenn Sie sich für diese Einstellung entscheiden, können Sie unter `DATEISYSTEM-ID` einen der folgenden Werte auswählen:

- ▶ `0x83 LINUX` wird für normale Linux-Datenpartitionen verwendet.
- ▶ `0x8E LINUX LVM` ist der korrekte Wert für den Logical Volume Manager (siehe unten), der virtuelle Partitionen über mehrere physische Laufwerke hinweg anlegen kann.
- ▶ `0xFD RAID` kann ausgewählt werden, wenn die Festplatte oder Partition Teil eines RAID-Arrays werden soll, einem logischen Zusammenschluss von Festplatten zur Steigerung der Performance und/oder Datensicherheit. Näheres über RAID erfahren Sie in Kapitel 9, »Datenträger und Dateisysteme«.
- ▶ `0x82 SWAP` ist die ID für die bereits erwähnten Swap-Partitionen.
- ▶ `0x0C WIN95 FAT32` wird für das gleichnamige Windows-Dateisystem verwendet und ist für manche Windows-Partitionen, die in der Liste auftauchen, bereits eingetragen.
- ▶ `0xA0 HIBERNATION` ist ein spezieller Partitionstyp, der für die »Suspend to Disk«-Funktion moderner Power-Management-Systeme verwendet werden kann – der komplette aktuelle Zustand des Rechners wird auf der Festplatte gespeichert und nach dem Wiedereinschalten erneut geladen.

Wenn Sie sich für die Hauptoption `FORMATIEREN` entscheiden, wird die Dateisystem-ID je nach Auswahl einer Formatierungsart automatisch festgelegt. Im Pull-down-Menü `DATEISYSTEM` stehen folgende Optionen zur Auswahl (genauere Details über die Dateisysteme stehen in Kapitel 9):

- ▶ `BTRFS` (B-Tree File System oder Butter File System) ist eine Neuentwicklung des Datenbankspezialisten Oracle. Es unterstützt wie Ext4 (siehe unten) sehr große Partitionen, aber im Gegensatz zur Linux-Ext-Dateisystemfamilie wurden eine Reihe neuer Ideen verwirklicht. Insbesondere wird Copy on Write verwendet (Anlegen echter physischer Kopien erst beim Auftreten von Unterschieden), um die Performance zu steigern. Wenn Sie dieses Dateisystem auswählen, wird eine ernsthafte Warnung ausgegeben, dass es noch experimentell sei. Für ein Produktivsystem sollten Sie es daher nur mit großer Vorsicht einsetzen.
- ▶ `EXT2` (Extended File System Version 2) ist das klassische Linux-Dateisystem. Es ist schnell und stabil, verfügt aber nicht über die Journaling-Funktionen

modernerer Dateisysteme, sodass die Wiederherstellung nach einem Absturz, etwa durch Stromausfall, länger dauert und eher schiefgeht. Heute wird Ext2 nur noch aus Kompatibilitätsgründen unterstützt. Solange Sie auf Ihrem Rechner nicht parallel ein besonders altes Linux-System betreiben und Daten mit diesem austauschen müssen, ist von einer Neuinstallation mit Ext2 abzuraten. Zudem ist Ext3 ohnehin abwärtskompatibel.

- ▶ EXT3 und EXT4 (Extended File System Version 3 beziehungsweise 4), die Nachfolger von Ext2, wurden insbesondere um die Journaling-Funktion erweitert: Das Dateisystem notiert alle Änderungen in einem sogenannten Journal, sodass das Dateisystem nach einem Absturz relativ problemlos wieder auf den letzten funktionierenden Stand gebracht werden kann. Ext4 unterstützt zusätzlich sehr große Festplatten (erheblich größer als diejenigen, die heutzutage für gewöhnliche Rechner verkauft werden).
- ▶ JFS ist ein sehr neues Journaling-Dateisystem. Es sollte nicht für den Produktiveinsatz verwendet werden, da noch kaum Erfahrungen damit vorliegen; der Installer selbst gibt ebenfalls eine entsprechende Meldung aus.
- ▶ REISER (das Reiser File System) besitzt ebenfalls eine Journaling-Funktion. Im Gegensatz zu Ext3 schrieb es ursprünglich keine Nutzdaten, sondern nur Metadaten (Verzeichnisstrukturen und Ähnliches) ins Journal; inzwischen lässt sich das Journaling von Nutzdaten optional aktivieren. Ein wichtiger Vorteil von ReiserFS besteht darin, dass es den verfügbaren Platz auf Festplatten besser ausnutzt, indem sich mehrere kleine Dateien oder übrig bleibende Teilstücke von Dateien eine Zuordnungseinheit teilen können; bei den meisten anderen Dateisystemen belegen Dateien stets ganze Zuordnungseinheiten.
- ▶ FAT (File Allocation Table) ist das klassische Windows-Dateisystem, genauer gesagt eine Gruppe von Dateisystemen; auf der Windows-Plattform wurde es inzwischen fast vollständig durch die Neuentwicklung NTFS abgelöst. Auf einem reinen Linux-System überwiegen die Nachteile der technisch überholten FAT-Dateisysteme; hier sollten Sie FAT nicht einmal in Betracht ziehen. Interessant ist eine FAT-Partition dagegen auf einem Rechner, der Linux und Windows nebeneinander enthält; hier kann sie für den Datenaustausch zwischen den abwechselnd zu bootenden Systemen eingesetzt werden, zumal die NTFS-Dateisysteme der modernen Windows-Versionen unter Linux bis vor Kurzem nur gelesen, aber nicht geschrieben werden konnten.
- ▶ XFS ist ebenfalls ein Journaling-Dateisystem, das ursprünglich von Silicon Graphics (SGI) für deren kommerzielles UNIX-System IRIX entwickelt wurde, inzwischen aber Open Source und offizieller Bestandteil des Linux-Kernels ist. Die Abkürzung steht eigentlich für »Extended File System«; der Name wird

allerdings nie ausgeschrieben, um das Dateisystem nicht mit Ext2 und Ext3 zu verwechseln. XFS war eines der ersten 64-Bit-Dateisysteme und eignet sich daher für sehr große Festplatten oder Partitionen.

- SWAP ist kein Datendateisystem, sondern – wie bereits erwähnt – das spezielle Dateisystem für Auslagerungsdateien.

Wenngleich jedes der vier Journaling-Dateisysteme eine andere Entwicklungsgeschichte hinter sich hat, gleichen sich die Features von Ext3/4, JFS, ReiserFS, und XFS in der Praxis immer mehr aneinander an; die verbliebenen Detailunterschiede werden in Kapitel 9, »Datenträger und Dateisysteme«, diskutiert. Im alltäglichen Betrieb Ihres openSUSE-Systems spielt es keine allzu große Rolle, welches von den dreien Sie verwenden. Da Ext4 inzwischen von den openSUSE-Entwicklern bevorzugt wird, ist allerdings zu erwarten, dass die Unterstützung dieses Dateisystems durch mit der Distribution gelieferte Tools etwas besser ist als bei den anderen.

Die Schaltfläche **OPTIONEN** bietet je nach gewähltem Dateisystem verschiedene zusätzliche Einstellungen. Wenn Sie möchten, können Sie einen Blick hineinwerfen; in der Regel sind die Standardwerte aber erst einmal in Ordnung.

Wenn Sie **DATEISYSTEM VERSCHLÜSSELN** ankreuzen, müssen Sie beim Bestätigen der Einstellungen mit **OK** ein Passwort eingeben. Ohne Kenntnis dieses Passwortes kann die Partition nicht mehr gemountet werden – auch nicht von Ihnen! Wie Sie sichere Passwörter finden, die Sie sich trotzdem leicht merken können, wird weiter unten in diesem Kapitel erläutert.

In der Rubrik **GRÖSSE** wird festgelegt, welche Zylinder der Festplatte die Partition belegen soll. Der **STARTZYLINDER** bestimmt den Anfang der Partition; unter **ENDE** ist zunächst der Endzylinder eingetragen. Alternativ können Sie das Ende relativ angeben, indem Sie ein Pluszeichen (+) vor den Wert und eine der Einheiten **M** (Megabyte) oder **GB** (Gigabyte) dahintersetzen. Geben Sie etwa **+1024M** für eine 1.024 Megabyte große Swap-Partition oder **+20GB** für eine 20-Gigabyte-Datenpartition ein. Beachten Sie allerdings, dass sich die Größe einer Partition intuitiver über den Dialog **GRÖSSE ÄNDERN** modifizieren lässt.

Der Dialog **FSTAB-OPTIONEN** (siehe Abbildung 2.6) bestimmt die Einträge in der Datei */etc/fstab*, die für das automatische Mounten Ihrer Partitionen beim Systemstart verantwortlich ist. Die Einstellung **EINHÄNGEN IN /ETC/FSTAB MITTELS** wurde bereits weiter oben in der Beschreibung der Partitionstabelle erläutert; die Voreinstellung **GERÄTE-ID** (die Sie nur in Ausnahmefällen ändern müssen) entspricht dem Wert **I** aus der Tabelle. Unter **VOLUME-KENNUNG** können Sie die bereits beschriebene, selbstgewählte Bezeichnung für die Partition eingeben, was

in der Regel nur erforderlich ist, wenn Sie das Dateisystem anhand dieser Information mounten möchten.



Abbildung 2.6 Der Dialog »Fstab-Optionen« bestimmt die Einstellungen für das automatische Mounten.

Als Nächstes stehen fünf spezielle Optionen zur Verfügung:

- ▶ **SCHREIBGESCHÜTZT EINHÄNGEN** (in der Optionsliste in */etc/fstab* `ro` für »read only«; das Gegenteil ist `rw` für »read/write«). Der Datenträger wird so gemountet, dass er nur gelesen, aber nicht beschrieben werden kann. Dieser Wert ist natürlich Voreinstellung bei CD- und DVD-ROM-Laufwerken sowie traditionell bei vorhandenen NTFS-Partitionen; für normale Datenpartitionen auf Festplatten ist er in aller Regel nicht sinnvoll.
- ▶ **KEINE ZUGRIFFSZEIT** (*/etc/fstab*-Option `noatime`; Gegenteil `atime`). Beim Zugriff auf Dateien und Verzeichnisse in dem Dateisystem werden Datum und Uhrzeit des letzten Zugriffs nicht aktualisiert. Da diese Option die Suche nach Problemen oder gar Angriffen erschwert, sollte sie nur in Ausnahmefällen eingeschaltet werden.

- ▶ DURCH BENUTZER EINHÄNGBAR (*user*). Normalerweise darf nur der Superuser *root* Partitionen und Datenträger mounten (der entsprechende Standardwert heißt *nouser*). Wenn Sie diese Option aktivieren, erhalten auch normale User dieses Recht für das betreffende Volume.
- ▶ NICHT BEIM SYSTEMSTART EINHÄNGEN (*noauto*; der Normalwert ist *auto*). Wenn Sie diese Option ankreuzen, wird das Volume beim Booten nicht automatisch gemountet. Sinnvoll ist das vor allem für Laufwerke mit Wechseldatenträgern wie etwa CD-ROM-Laufwerken.
- ▶ KONTINGENTUNTERSTÜTZUNG (QUOTA) AKTIVIEREN (*/etc/fstab*-Option: *quota*) ermöglicht es dem Administrator, festzulegen, wie viel Speicherplatz jedem Benutzer in der Partition zur Verfügung steht. Dies ist insbesondere für ausgelagerte Home-Verzeichnisse auf Mehrbenutzer-Servern empfehlenswert.

Für die Standardwerte *rw,atime,nouser,auto* (die Werteliste in */etc/fstab*-Einträgen wird durch Kommas getrennt, darf aber kein Leerzeichen enthalten) gibt es die spezielle Kompaktschreibweise *defaults*.

Die Option DATEN-JOURNAL-MODUS steht selbstverständlich nur für die Journaling-Dateisysteme zur Verfügung. Sie legt fest, ob und auf welche Weise Nutzdaten im Journal gespeichert werden. Die drei möglichen Werte sind:

- ▶ JOURNAL
Nutzdaten zuerst ins Journal und erst danach in das Dateisystem schreiben.
- ▶ ORDERED
Daten in das Dateisystem schreiben, danach nur Metadaten im Journal speichern.
- ▶ WRITEBACK
Nutzdaten sowohl im Journal als auch im Dateisystem ablegen, allerdings ohne Vorgabe einer Reihenfolge.

Die Voreinstellung ist *ORDERED*; dies ist besser für die Performance, aber nicht ideal für die Datensicherheit. *JOURNAL* ist die sicherste Option, kann aber zu Geschwindigkeitsnachteilen führen. *WRITEBACK* schließlich ist oft ein erträglicher Kompromiss.

Die ZUGRIFFSKONTROLLLISTEN (ACL) – Letzteres ist eine Abkürzung für »Access Control Lists« – bieten eine über die traditionellen benutzer- und gruppenbasierten Dateirechte hinausgehende Möglichkeit, den Zugriff auf Dateien und Verzeichnisse freizugeben oder einzuschränken. Es ist empfehlenswert, diese Option zu aktivieren.

ERWEITERTE BENUTZERATTRIBUTE sind spezielle Dateirechte, die unter anderem die Ausführung von Programmen mit den Rechten des Eigentümers statt des aufrufenden Benutzers gewähren (Setuid-Bit). Diese Option sollte in der Regel eingeschaltet werden; für bestimmte Teile des Verzeichnisbaums wie */var* ist sie sogar notwendig.

Unter WEITERE OPTIONEN können Sie zusätzliche Mount-Optionen eintragen. Diese müssen wiederum durch ein Komma voneinander getrennt werden und dürfen kein Leerzeichen enthalten. Einige mögliche Werte lernen Sie in Kapitel 9, »Datenträger und Dateisysteme« kennen. Die Mount-Optionen können jederzeit geändert werden und lassen sich durch einen Neustart des Systems oder durch erneutes Mounten der entsprechenden Partition aktivieren, sodass Sie keine Probleme bekommen, wenn Sie hier erst einmal nichts eintragen.

Wenn Sie Ihre Mount-Einstellungen mittels OK bestätigt haben, finden Sie unter der Schaltfläche FSTAB-OPTIONEN als letzten Eintrag den EINHÄNGEPUNKT (Mountpoint). Wie bereits erwähnt, geht es darum, an welcher Stelle die Partition in den Verzeichnisbaum eingehängt werden soll. Sobald Sie eines der standardmäßig bei der Installation erzeugten Hauptverzeichnisse (oder auch eines ihrer Unterverzeichnisse) als Mountpoint für eine Partition angeben, wird das entsprechende Verzeichnis auf diesem Volume angelegt.

Standardmäßig wird nur */home*, die Struktur der Benutzerverzeichnisse, als eigenständige Partition angelegt. Wenn Sie genug Plattenplatz haben, finden Sie hier einige weitere gute Kandidaten für eigene Partitionen:

- ▶ */var* ist das Verzeichnis für allerlei Dateien, auf die besonders viele Schreibzugriffe stattfinden – besonders Logdateien und sogenannte PID-Dateien, die die aktuelle Prozess-ID eines laufenden Programms enthalten, um unter anderem klarzustellen, dass das entsprechende Programm bereits läuft. Die Auslagerung von */var*, vorzugsweise auf eine separate physische Festplatte, kann die Systemperformance verbessern.
- ▶ */root* ist das eigenständige Home-Verzeichnis des gleichnamigen Superusers. Es lässt sich darüber streiten, ob es eine eigene Partition erhalten sollte oder nicht. Für die Auslagerung spricht, dass es sich nicht um ein System- oder Programmverzeichnis, sondern wie bei den */home/username*-Verzeichnissen der gewöhnlichen Benutzer eben um ein Verzeichnis mit Benutzerdaten handelt. Andererseits sollten Sie Ihre alltägliche Arbeit mit Anwendungsprogrammen ohnehin nicht als *root* durchführen, sodass das Verzeichnis im Normalfall keine nennenswerte Menge an Benutzerdateien enthält. Zudem kann es vorkommen, dass Ihnen für Wartungsarbeiten keine Partition außer der Stammpartition (auf der sich */* befindet) zur Verfügung steht. Falls */root* in einem sol-

chen Fall nicht auf dieser Partition liegt, können die benutzerorientierten Konfigurationseinstellungen des Superusers nicht eingelesen werden.

- ▶ */boot* enthält den Linux-Kernel, ist also gewissermaßen das wichtigste Verzeichnis Ihrer Hierarchie. Manche älteren Rechner können nur von einer Partition booten, die innerhalb der ersten 1.024 Zylinder liegt. Wenn Sie in diesem Bereich nur wenig freien Platz haben, könnten Sie gemäß der Anleitung am Anfang dieses Kapitels eine dort liegende Windows-Partition zum Verkleinern vorbereiten, sie während der Installation geringfügig verkleinern und auf dem frei gewordenen Plattenplatz eine kleine Partition mit dem Mountpoint */boot* anlegen.

Unter EINHÄNGEPUNKT stehen noch einige andere Verzeichnisse zur Auswahl, beispielsweise die Software-Installationsverzeichnisse */usr* und */opt* oder das Verzeichnis */srv* für Server-Freigaben wie etwa Websites. Selbstverständlich steht es Ihnen frei, statt der vorgefertigten Mountpoints einen beliebigen eigenen Verzeichnisnamen anzugeben. Dies verletzt allerdings die unter anderem von der Linux Standard Base (LSB) festgelegten Verzeichniskonventionen, sodass Software, die Sie in einem Hauptverzeichnis mit einem selbsterfundenen Namen installieren, möglicherweise nicht erwartungsgemäß funktioniert. Es ist auch kein großer Sicherheitsgewinn, die Standardverzeichnisse umzubenennen – ein solches Vorgehen wird als »Security by Obscurity« bezeichnet und taugt nicht viel.

Wenn Sie für eines dieser Verzeichnisse eine eigene Partition anlegen möchten, müssen Sie zunächst eine der vorgesehenen verkleinern. Dies geschieht mithilfe der weiter unten beschriebenen Option GRÖSSE ÄNDERN.

Nachdem Sie alle Einstellungen für eine Partition vorgenommen haben, können Sie auf OK klicken; sollten Ihnen die Optionen dagegen falsch vorkommen, klicken Sie VERWERFEN an. In beiden Fällen gelangen Sie zurück in den Hauptdialog, der neben der soeben beschriebenen Schaltfläche BEARBEITEN noch diverse andere Möglichkeiten bietet:

- ▶ Mithilfe der Schaltfläche ANLEGEN können Sie eine neue Partition erstellen. Zunächst werden Sie aufgefordert, den PARTITIONSTYP zu wählen. Wenn Sie sicher sind, dass Sie auf dem entsprechenden Datenträger nicht mehr als vier Partitionen benötigen, können Sie eine PRIMÄRE PARTITION erstellen; andernfalls müssen Sie sich für eine ERWEITERTE PARTITION entscheiden. Wenn Sie sich für eine primäre Partition entscheiden, gelangen Sie sofort in den bereits für die Funktion BEARBEITEN ausführlich beschriebenen Dialog. Bei einer erweiterten Partition müssen Sie dagegen zunächst Start- und Endzylinder angeben. Standardmäßig wird der gesamte verfügbare Platz für die erweiterte

Partition verwendet, was in der Regel kein Problem ist, da Sie darin eine Reihe logischer Partitionen anlegen können. In den meisten Fällen können Sie hier also auf OK klicken. Wenn Sie danach erneut auf ANLEGEN klicken, wird ohne weitere Nachfrage eine logische Partition erstellt.

- ▶ Die Schaltfläche LÖSCHEN ermöglicht das Entfernen der ausgewählten existierenden oder vorgesehenen Partition. Seien Sie vorsichtig – nach einer einmaligen Sicherheitsabfrage ist die Partition weg. Überschrieben ist sie allerdings noch nicht; solange der frei gewordene Platz noch nicht durch neue Daten überschrieben wurde, lässt sie sich wahrscheinlich mithilfe geeigneter Tools wiederherstellen, deren Betrachtung jedoch über den Umfang dieses Buches hinausgeht.
- ▶ Mithilfe der Funktion GRÖSSE ÄNDERN können Sie eine bestehende oder geplante Partition vergrößern beziehungsweise verkleinern. Der Dialog wird in Abbildung 2.7 gezeigt. Sie haben die Auswahl zwischen folgenden Optionen:
 - ▶ MAXIMALE GRÖSSE: Es wird die größtmögliche Größe ausgewählt, die an der aktuellen Position auf dem Datenträger zur Verfügung steht. Der konkrete Wert wird in Klammern angezeigt, in der Regel in Gigabyte.
 - ▶ MINIMALE GRÖSSE: Dies ist die Mindestgröße (im Prinzip eine Zuordnungseinheit plus genügend Platz für Verwaltungsinformationen). Diese kann je nach Dateisystem schwanken und wird ebenfalls dargestellt.
 - ▶ BENUTZERDEFINIERTER GRÖSSE: Sie können den gewünschten Wert selbst in das Eingabefeld unter der Option eintragen; dabei sind Einheitenkürzel wie GB oder MB erlaubt.

Unter den Auswahlmöglichkeiten wird zur Information die DERZEITIGE GRÖSSE angezeigt.

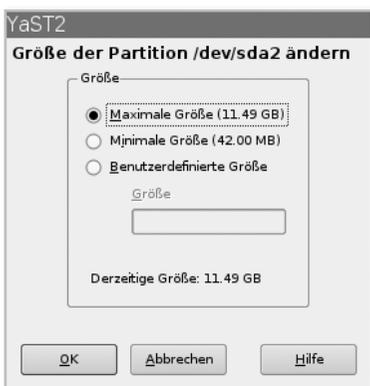


Abbildung 2.7 Der Dialog zum Verändern von Partitionsgrößen

Die Schaltflächen und Menüs in der unteren Reihe ermöglichen speziellere Einstellungen: LVM und RAID wurden bereits bei der Beschreibung der Partitionstypen erwähnt. Eine KRYPTODATEI entspricht dem bereits erwähnten verschlüsselten Dateisystem – mit einer Ausnahme: Es handelt sich nicht um eine echte Partition, sondern um ein sogenanntes *Loop-Device*, das heißt eine normale Datei, die ein Dateisystem enthält und in den Verzeichnisbaum eingehängt werden kann. Der Vorteil: Im Gegensatz zu normalen Partitionen können Loop-Devices auch von gewöhnlichen Benutzern erstellt und gemountet werden. Eine weitere Anwendung dieses Konzepts ist das Mounten von CD- oder DVD-Images vor dem eigentlichen Brennen, um sie auf Fehler zu prüfen (Näheres über Loop-Devices erfahren Sie in Kapitel 9, »Datenträger und Dateisysteme«).

Mithilfe der Schaltfläche NFS können Sie ein Netzwerkvolume zu Ihrer `/etc/fstab` hinzufügen, das heißt eine Partition, die sich auf einem anderen Rechner im lokalen Netzwerk befindet. In manchen Firmennetzwerken befinden sich beispielsweise die Home-Verzeichnisse auf zentralen Servern, damit die Mitarbeiter sich von jedem Rechner aus persönlich anmelden und mit ihren eigenen Daten arbeiten können. Näheres zu diesem Thema finden Sie in Kapitel 14, »Datei- und Verzeichnisdienstserver«.

Das Menü EXPERTEN enthält schließlich drei spezielle Anweisungen:

- ▶ PARTITIONSTABELLE NEU EINLESEN verwirft Ihre bisherigen Änderungen und liest die bestehende Partitionierung der Festplatten neu ein. Wenn bei einer Neuinstallation noch gar keine Partitionstabelle existiert, werden nur noch die leeren, unbehandelten Festplatten in der Liste angezeigt. In diesem Fall müssen Sie die Partitionierung manuell durchführen oder auf ZURÜCK klicken, um erneut einen Partitionierungsvorschlag zu erhalten.
- ▶ EINHÄNGEPUNKTE AUS VORHANDENER DATEI /ETC/FSTAB IMPORTIEREN kann nur auf einem Rechner verwendet werden, auf dem bereits Linux installiert ist, beispielsweise bei einer Aktualisierung auf die neueste Version oder wenn Sie den Partitionierer später im bereits installierten System aufrufen. Die Funktion liest die bisherigen Mountpoints ein und zeigt sie in der Tabellenansicht an.
- ▶ PARTITIONSTABELLE UND FESTPLATTENKENNUNG LÖSCHEN verwirft die gesamte vorhandene Liste, und Sie können eine komplett neue Partitionierung erstellen. Dies ist sinnvoll, wenn Sie openSUSE auf einem völlig leeren Rechner installieren oder alle vorhandenen Informationen löschen möchten.
- ▶ STARTE ISCSI-KONFIGURATION ermöglicht die Konfiguration von Volumes, die sich in speziellen Netzwerk-Storage-Systemen befinden.

Wie Sie die Partitionierung letztendlich durchführen, bleibt ganz Ihren Vorstellungen überlassen. Allerdings sollten Sie folgende Mindestvorgaben für die beiden Datenpartitionen `/` und `/home` sowie für die Swap-Partition beachten, um sinnvoll mit dem System arbeiten zu können (angesichts der heutigen Festplattengrößen und -preise sollte das alles kein Problem mehr sein):

- ▶ Die Standardinstallation ohne optionale Software belegt mit dem Desktop KDE etwa 1,9 Gigabyte, mit GNOME sogar 2,3 Gigabyte. Ihre `root`-Partition (`/`) sollte also mindestens 2,5 Gigabyte groß sein. Wenn Sie einen Großteil der mit der Distribution gelieferten Anwendungsprogramme verwenden oder auch nur ausprobieren möchten, kommen Sie nicht mit weniger als 6 Gigabyte aus. Falls Sie auf unterschiedlichen Festplatten jeweils weniger Platz haben, sollten Sie für `/` selbst eine Partition von etwa 3 GB und eine weitere für `/usr` von 3 GB oder mehr anlegen.

Es schadet auch nichts, die `/`-Partition größer anzulegen; wichtig ist aber stets, dass genug Platz (ab etwa 3 Gigabyte) für `/home` übrig bleibt.

- ▶ Die Größe der Swap-Partition sollte einem ganzzahligen Vielfachen Ihres physikalischen Arbeitsspeichers möglichst nahe kommen. Als absolute Mindestgröße empfehlen sich 512 Megabyte; falls Ihr RAM größer ist, sollten Sie das Doppelte desselben wählen.
- ▶ Die `/home`-Partition kann im Grunde kaum zu groß sein – hier sammeln sich im Laufe der Zeit Ihre Arbeitsdateien an. Teilen Sie den restlichen verfügbaren Platz einfach dieser Partition zu.

Bestätigen Sie Ihre Partitionierungseinstellungen wie gehabt über die Schaltfläche WEITER.

2.2.8 Einrichten des ersten Standardbenutzers

Als Nächstes erscheint der Bildschirm NEUEN BENUTZER ERSTELLEN (siehe Abbildung 2.8); hier wird der erste normale Systembenutzer angelegt. Geben Sie hier folgende Daten ein:

- ▶ VOLLSTÄNDIGER NAME DES BENUTZERS
Hier können Sie den Realnamen des gewünschten Benutzers eingeben. Dieser wird nicht für die eigentliche Anmeldung verwendet, sondern dient Ihrem eigenen Komfort. Er wird beispielsweise bei der grafischen Anmeldung in der Benutzerliste angezeigt. Der erste reguläre Benutzer sind Sie selbst, also geben Sie hier Ihren eigenen Vor- und Nachnamen ein. Als Beispiel soll hier »Sascha Kersken« gelten.

► **BENUTZERNAME**

Dies wird der tatsächliche Anmeldename des Benutzers. Manche Administratoren verwenden ein standardisiertes Schema für die Kombination aus Vor- und Nachnamen. Automatisch wird der Vorname vorgeschlagen; in einem sehr kleinen Netzwerk ist dies sinnvoll. Gemäß dem obigen Beispiel wäre der automatisch erstellte Benutzername *sascha*.

► **PASSWORT**

Hier wird das Passwort des neuen Benutzers eingegeben. Folgen Sie den im nächsten Unterabschnitt diskutierten Ratschlägen für ein gutes Passwort. Unter **PASSWORT BESTÄTIGEN** müssen Sie die Eingabe wiederholen.

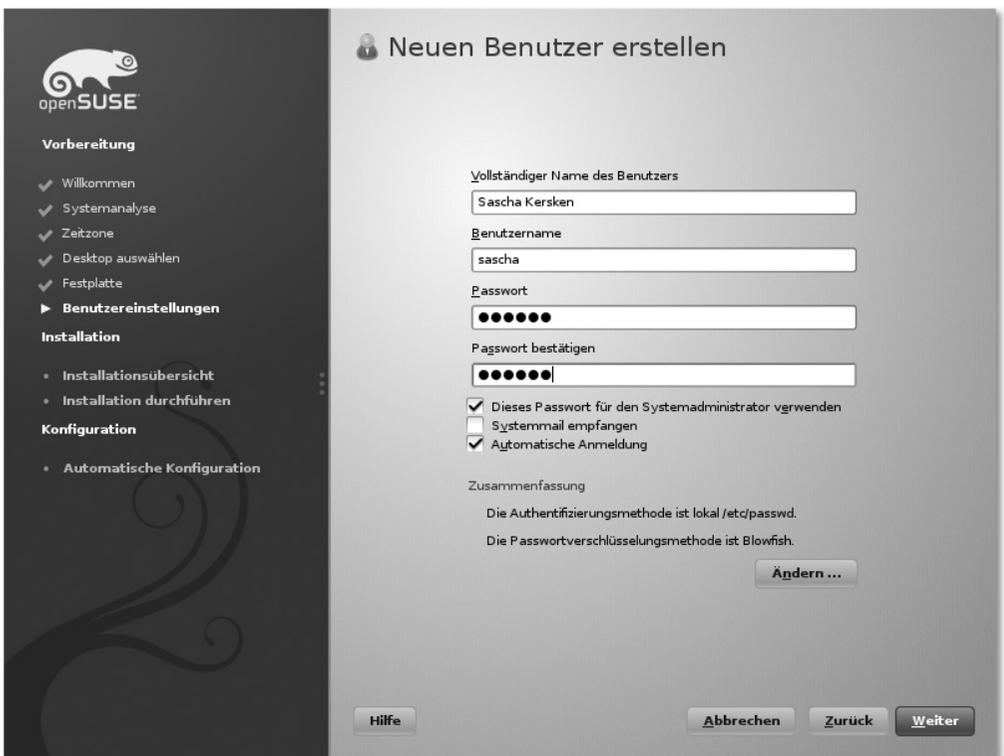


Abbildung 2.8 Erstellung des ersten normalen Benutzerkontos

► **DIESES PASSWORT FÜR DEN SYSTEMADMINISTRATOR BENUTZEN**

Diese Option weist dem Systemverwalter `root` dasselbe Passwort zu wie diesem ersten Standardbenutzer. Zu Hause können Sie dies aus Bequemlichkeitsgründen möglicherweise in Erwägung ziehen; in einem Firmennetzwerk ist strikt davon abzuraten.

► SYSTEMMAIL EMPFANGEN

Manche wichtigen Systemmeldungen werden per (lokaler) Mail verschickt. Wenn Sie der Systemverwalter sind, kann es nützlich sein, diese Mails auf Ihrem Standardbenutzerkonto zu empfangen und nicht nur als `root`.

► AUTOMATISCHE ANMELDUNG

Diese Option sorgt dafür, dass dieser Benutzer nach dem Booten automatisch (das heißt ohne Passworteingabe) angemeldet wird. Auch dies sollten Sie höchstens bei einem Heimrechner machen; in einer Firma, in der mehrere Personen physisch auf den Rechner zugreifen können, ist diese Einstellung grob fahrlässig.

Mithilfe der Schaltfläche **ÄNDERN** gelangen Sie auf den Bildschirm **EINSTELLUNGEN FÜR EXPERTEN**; hier wird zunächst die **AUTHENTIFIZIERUNGSMETHODE** eingestellt. In einem Firmennetzwerk, zumal einem größeren, wäre es lästig und vor allem ein Sicherheitsproblem, die Anmeldedaten auf jedem einzelnen Rechner vorzuhalten. Mit einer geeigneten Methode lassen sich diese Daten im Netzwerk zentralisieren, das heißt von einem speziellen Anmeldeserver überprüfen. Wenn dann noch der `/home`-Verzeichnisbaum auf einen Serverrechner ausgelagert wird, könnten die Benutzer bei jeder Anmeldung dieselbe Arbeitsumgebung vorfinden, egal an welchem Rechner sie sich anmelden. Die vier möglichen Optionen sind:

► LOKAL (/ETC/PASSWD)

Die Benutzerdaten werden auf dem lokalen Rechner gespeichert; es wird kein Authentifizierungs-Server verwendet. Dieses Verfahren wird im nächsten Kapitel genauer behandelt.

► LDAP

Das *Lightweight Directory Access Protocol* ist eine weitverbreitete Methode für den Zugriff auf sogenannte Verzeichnisdienste. Der Begriff Verzeichnis beschreibt hier keine Verzeichnisstruktur auf einem Datenträger, sondern eine Datenbank mit detaillierten Informationen über Benutzer, Rechner und weitere Ressourcen. Bekannte LDAP-basierte Verzeichnisdienste sind OpenLDAP sowie Microsoft Active Directory. OpenLDAP wird in Kapitel 14, »Datei- und Verzeichnisdienstserver«, beschrieben.

► NIS

Der *Network Information Service* (früher auch »Yellow Pages« oder kurz »yp« genannt) wurde von Sun Microsystems entwickelt und ist in der UNIX-Welt weitverbreitet. Er ist einfacher einzurichten als ein LDAP-kompatibles Verzeichnis, aber auch nicht ganz so leistungsfähig. Auch NIS wird in Kapitel 14 besprochen.

► WINDOWS-DOMÄNE

Wenn Sie ein heterogenes Netzwerk mit Windows- und Linux-Rechnern betreiben, können Sie sich sogar an einer Windows-Domäne anmelden. Diese wird von einem Windows-Server-Betriebssystem wie Windows Server 2008 oder der Vorgängerversion Windows Server 2003 bereitgestellt. Interessanterweise kann auch ein Linux-Rechner eine Windows-Domäne kontrollieren; dies ist eine der Fähigkeiten des freien, Windows-Netzwerk-kompatiblen Dateiservers Samba.

Momentan sollten Sie einstweilen LOKAL wählen, um ein Benutzerkonto zu erstellen, dessen Anmeldedaten auf dem lokalen Rechner selbst gespeichert und überprüft werden.

2.2.9 Das root-Passwort festlegen

Wenn Sie sich bei den Benutzereinstellungen dagegen entschieden haben, das Benutzerpasswort auch für `root` zu verwenden, müssen Sie im nächsten Schritt das `root`-Passwort festlegen. Wie bereits im vorigen Kapitel erwähnt, ist Linux aufgrund seiner UNIX-Tradition ein Multiuser-System. Normale Benutzer besitzen nur eingeschränkte Rechte, was die Auswirkungen sowohl vorsätzlicher als auch versehentlicher Fehler abmildert. Der spezielle Benutzer, der *alles* darf und damit für die Systemadministration verantwortlich ist, heißt `root`. Da Sie das System selbst installieren, sind Sie das auf Ihrem Rechner. Beachten Sie allerdings, dass Sie normale Anwendertätigkeiten aus Sicherheitsgründen niemals als `root` durchführen sollten.

Auf diesem Bildschirm wird das Passwort für das `root`-Benutzerkonto eingestellt. Es muss zwei Anforderungen gleichermaßen erfüllen:

- Es darf nicht einfach zu erraten sein; aus diesem Grund dürfen Sie niemals ein existierendes Wort irgendeiner Sprache oder einen Eigennamen verwenden.
- Sie selbst müssen es sich merken können, weil Sie ansonsten enorme Schwierigkeiten haben, auf Ihren Rechner zuzugreifen (was Sie tun können, falls Sie doch einmal Ihr `root`-Passwort vergessen sollten, steht in Kapitel 9, »Datenträger und Dateisysteme«).

Die einfachste Methode, ein sicheres und doch leicht zu merkendes Passwort zu konstruieren, besteht darin, sich einen nicht zu offensichtlichen Satz auszudenken und dessen Anfangsbuchstaben zu verwenden. Dabei sollten Sie Groß- und Kleinschreibung mischen und manche visuell passenden Zeichen durch Ziffern ersetzen.

Als Beispiel soll der Satz »There's more than one way to do it«⁵ dienen. Die Anfangsbuchstaben lauten: TMTOWTDI. Wenn Sie beschließen, den ersten Buchstaben und das W als den Anfangsbuchstaben des Substantivs großzuschreiben, das O durch eine Null und das I durch eine Eins ersetzen, erhalten Sie **Tmt0Wtd1**. Eine solche Zeichenfolge steht garantiert nicht in einer Wörterbuchdatei. Aber natürlich dürfen Sie *diesen* Satz in der Praxis nicht verwenden – er steht in diesem Buch, und da dieses Buch und seine Voraufgaben einen gewissen Verbreitungsgrad besitzen, könnten Cracker sich einen »Spaß« daraus machen, dieses Passwort auf Rechnern im Netz auszuprobieren.

2.3 Softwareauswahl und Installation

Mit den Benutzer- und Passworteinstellungen sind die Vorarbeiten abgeschlossen – Sie gelangen auf den Bildschirm `INSTALLATIONSEINSTELLUNGEN`, wo Sie die Einstellungen für die eigentliche Installation vornehmen. Die wichtigste ist die Auswahl der zu installierenden Software.

2.3.1 Software-Auswahl

Linux-Distributionen bestehen bekanntlich nicht nur aus dem Betriebssystem, sondern enthalten zahllose Anwendungsprogramme aus den unterschiedlichsten Bereichen. An dieser Stelle des Installationsprozesses treffen Sie die Erstauswahl; Sie können später jederzeit Software hinzufügen, entfernen oder aktualisieren (siehe nächstes Kapitel).

Voreingestellt ist je nach Ihrer Desktop-Auswahl ein `SYSTEMTYP` wie KDE 4.3 oder GNOME. Klicken Sie den Link `SOFTWARE-AUSWAHL` an, um weitere Programme hinzuzufügen. Dadurch gelangen Sie auf den Bildschirm aus Abbildung 2.9.

Die linke Spalte des Hauptdialogs zeigt die sogenannten *Schemata* an, das heißt die thematischen Kategorien, in die die mitgelieferte Software unterteilt wird. Wenn eines der Kontrollkästchen angekreuzt ist, bedeutet dies, dass alle Pakete des entsprechenden Schemas ausgewählt wurden. Wenn Sie eine Selektion vollständig installieren möchten, klicken Sie das entsprechende Kontrollkästchen an. Wenn Sie es bei einer bereits ausgewählten Selektion deaktivieren, dann wird kein Paket dieser Selektion installiert.

⁵ Dies ist eines der offiziellen Mottos der Programmiersprache Perl (siehe Kapitel 17, »System-Automatisierung«).

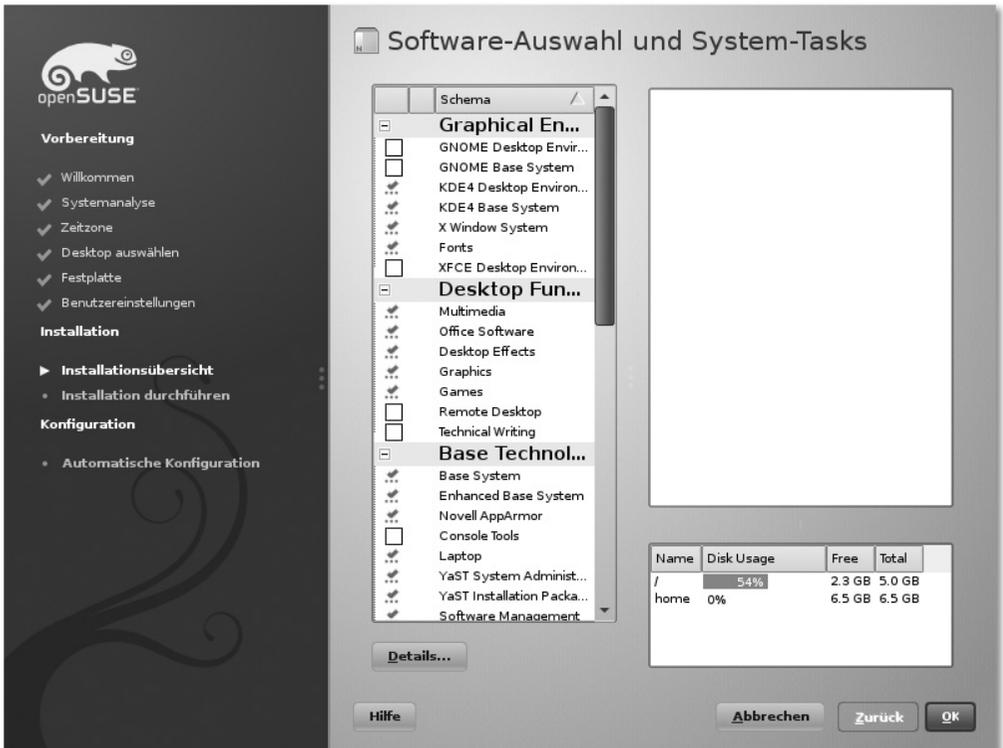


Abbildung 2.9 Die Auswahl der zu installierenden Softwarepakete

Bei einem Häkchen mit drei Punkten sind einige Pakete des Schemas ausgewählt. Klicken Sie auf die Schaltfläche DETAILS, um einzelne Pakete an- oder abzuwählen.

In Tabelle 2.1 erhalten Sie zunächst eine Übersicht über die verfügbaren Schemata für openSUSE 11.2, die jeweils enthaltene Software und gegebenenfalls die Kapitel dieses Buches, in denen die Inhalte der einzelnen Schemata behandelt werden (in Klammern, falls es nur am Rande darum geht). Die Schema-Namen sind auch bei einer deutschsprachigen Installation durchweg Englisch; dies war in früheren openSUSE-Versionen anders. Möglicherweise handelt es sich um einen Bug, der mit einem kommenden Softwareupdate oder aber in der nächsten Version behoben wird.

Einige Pakete sind übrigens in mehr als einem Schema zu finden. Beachten Sie außerdem, dass manche Pakete in etwas unlogisch erscheinenden Schemata untergebracht wurden. Aber keine Panik – konkrete Einzelfälle, die die in diesem Buch behandelten Programme betreffen, werden in den jeweiligen Kapiteln besprochen.

Schema	Kapitel	Beschreibung
Kategorie »Graphical Environments«		
GNOME Desktop Environment	3, 5	Der GNOME-Desktop mit zahlreichen Zusatzprogrammen
GNOME Base System	3, 5	Die grundlegenden Pakete für den GNOME-Desktop
KDE4 Desktop Environment	3, 5	Der KDE4-Desktop mit zahlreichen Zusatzprogrammen
KDE4 Base System	3, 5	Die grundlegenden Pakete für den KDE4-Desktop
Fonts	–	Diverse Schriftarten
XFCE Desktop Environment	(5)	Ein alternativer Desktop/Window-Manager
Kategorie »Desktop Functions«		
Multimedia	5, 8	Zahlreiche Player sowie Audio- und Videoprogramme
Office Software	7	OpenOffice.org und andere Office-Programme
Desktop Effects	(5)	3-D-Desktop-Effekte
Graphics	8	Grafik- und Bildbearbeitungsprogramme
Games	–	Diverse Spiele
Remote Desktop	(14)	Zugriff auf die Desktops entfernter Rechner bzw. Fernzugriff auf den eigenen Desktop
Technical Writing	(11)	LaTeX, DocBook und andere Software für Dokumentationen
Kategorie »Base Technologies«		
Base System	2, 3, 4	Das openSUSE-Basissystem (viele Pakete werden automatisch ausgewählt)
Enhanced Base System	(2, 3, 4)	Zusätzliche Pakete für das Basissystem
Novell AppArmor	–	Anwendungs-Sicherheitssystem
Console Tools	(4, 9)	Dienstprogramme für die Konsole

Tabelle 2.1 Die verfügbaren Software-Schemata unter openSUSE 11.2

Schema	Kapitel	Beschreibung
Laptop	–	Spezielle Tools für den Betrieb auf Laptops
YaST System Administration	2, 3	YaST-Pakete für die Systemadministration
YaST Installation Packages	2	YaST-Pakete für die Systeminstallation (automatisch ausgewählt)
Software Management	3	Tools für automatische Systemupdates
TabletPC	–	Spezielle Hilfsprogramme für Tablet- und Touchscreen-PCs
Kategorie »Server Functions«		
File Server	14	Dateiserver (NFS, Samba usw.)
Miscellaneous Server	–	Sonstige Server
Network Administration	13	Tools zur Netzwerkadministration
Print Server	(6)	Druckserver (CUPS und weitere Software)
Mail and News Server	(16)	Server für E-Mail und Newsgroups
Web and LAMP Server	15	Der Webserver Apache, PHP, MySQL und weitere Webanwendungs-Server-Software
Internet Gateway	14	Proxyserver, Firewall, VPN-Gateway usw.
DHCP and DNS Server	13, 14	Automatische TCP/IP-Konfiguration (DHCP) und Nameserver (DNS)
Directory Server (LDAP)	14	Der Verzeichnisdienstserver OpenLDAP und Zusatzprogramme
Xen Virtual Machine Host Server	–	Eine Kernel-nahe Lösung zur Betriebssystemvirtualisierung
Kategorie »Proprietary Software«		
Misc. Proprietary Packages	–	Proprietäre Software (Nicht-Open-Source). Die eigentlichen Pakete befinden sich auf der Non-OSS-Zusatz-CD (siehe Kapitel 1) oder in Online-Repositories

Tabelle 2.1 Die verfügbaren Software-Schemata unter openSUSE 11.2 (Forts.)

UNGÜLTIG

In der Hauptansicht (nicht in der Detailansicht) erscheint die Belegungsstatistik für die verschiedenen Festplattenpartitionen, wie sie nach der aktuell ausgewählten Installation aussehen würde. Achten Sie darauf, dass die Partitionen nicht zu voll werden; gegebenenfalls müssen Sie zurückblättern und die Partitionierungseinstellungen noch einmal ändern.

Wenn Sie die Schaltfläche DETAILS drücken, wird der Bildschirm aus Abbildung 2.10 angezeigt. Hier können Sie einzelne Pakete zur Installation auswählen oder gezielt abwählen.

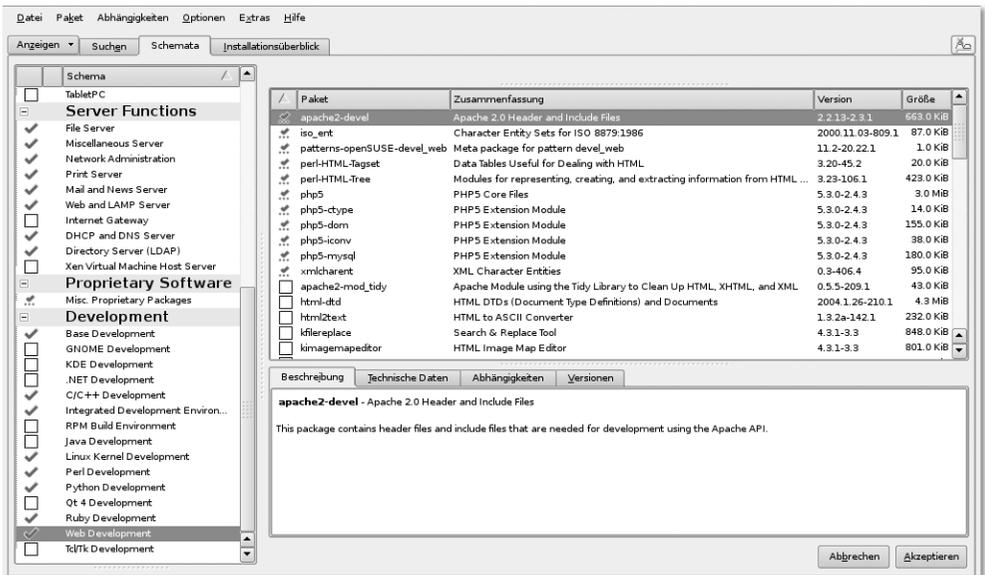


Abbildung 2.10 Die Detailansicht der Softwareinstallation

Neben der gezeigten Schema-Ansicht können Sie unter FILTER folgende Optionen einstellen:

► **PAKETGRUPPEN**

Eine etwas detailliertere Kategorisierung der Softwarepakete. Der linke Fensterbereich zeigt ein aufklappbares Baummenü mit zahlreichen Softwaregruppen und -untergruppen, während rechts wie gehabt die einzelnen Pakete der jeweiligen Gruppe angezeigt werden.

► **SPRACHEN**

Spezielle Pakete für die Unterstützung zahlreicher Sprachen (gemeint sind tatsächlich menschliche Sprachen, nicht etwa Programmiersprachen). Angeboten werden unter anderem Wörterbücher, Tastaturschemata und Übersetzungen der Desktop-Umgebungen.

► **INSTALLATIONSQUELLEN**

Hier können Sie optional auch die Quellcodepakete zu sämtlichen Open-Source-Programmen der Distribution wählen. Dies dient gegebenenfalls der Anpassung von Software an spezielle Erfordernisse. Zusätzlich ist das Studium von Quellcode eine der wichtigsten Informations- und Weiterbildungsquellen für Entwickler.

► **SUCHE**

Wenn Sie ein bestimmtes Programm installieren möchten, aber nicht wissen, in welcher Selektion oder Paketgruppe es sich befindet, empfiehlt sich die Suchfunktion. Geben Sie dazu unter **SUCHE** einen beliebigen Suchbegriff ein. Das Feld **SUCHEN IN** ermöglicht Ihnen die Angabe, welche Informationen durchsucht werden sollen. Die standardmäßig angekreuzten Felder **NAME** und **ZUSAMMENFASSUNG** reichen in den meisten Fällen aus. Falls Sie nur die ungefähre Funktion, aber nicht den Namen eines Pakets kennen, kann eventuell die Suche in der **BESCHREIBUNG** weiterhelfen. Da die Beschreibungen umfangreicher sind als die anderen Informationen – was die Suchdauer erheblich verlängert –, sollten Sie diese Option allerdings nach Gebrauch wieder deaktivieren. Die beiden letzten Felder beziehen sich auf Paketabhängigkeiten: Die erste der beiden Optionen sucht in der Liste der Pakete, für die das jeweilige Paket erforderlich ist; die zweite findet umgekehrt diejenigen Pakete, die ein bestimmtes Paket zum Funktionieren benötigt.

► **ZUSAMMENFASSUNG**

Hier erhalten Sie die Liste aller Pakete, die während der Installation modifiziert werden sollen oder einen bestimmten Installationsstatus besitzen. Links können Sie sich beliebig viele Statusoptionen aussuchen; in der Liste rechts werden daraufhin die entsprechenden Pakete angezeigt. Standardmäßig sind fast alle Optionen aktiviert. Eine Ausnahme bilden die beiden letzten Statusinformationen, da sie auf diejenigen Pakete zutreffen, die weder eine Änderung erfahren noch einen besonderen Status haben: **BEHALTEN** gilt für Pakete, die bereits installiert sind (dies kommt natürlich nicht bei der Erstinstallation, sondern nur bei Updates oder Nachinstallationen einzelner Pakete vor); **NICHT INSTALLIEREN** beschreibt dagegen alle Pakete, die nicht installiert werden sollen – im Gegensatz zur Option **TABU** können diese allerdings im Zuge der automatischen Auflösung von Paketabhängigkeiten installiert werden.

Rechts wird die Liste aller Einzelpakete angezeigt, die dem links eingestellten Filter entsprechen. Die Liste besitzt folgende Felder:

- Das Kontrollkästchen am Anfang jeder Zeile zeigt den Installationsstatus an.
- **PAKET** enthält den Namen des Pakets.

- ▶ ZUSAMMENFASSUNG ist eine Kurzbeschreibung der Aufgabe des Pakets.
- ▶ VERSION gibt die Versionsnummer an, in der das Paket vorliegt.
- ▶ GRÖSSE zeigt die Größe des Pakets in Kilobyte oder Megabyte an.

Wenn Sie eine Zeile anklicken, erhalten Sie rechts unten auf vier Registerkarten Detailinformationen zum jeweiligen Paket:

- ▶ BESCHREIBUNG liefert mehr oder weniger ausführliche Informationen über die im Paket enthaltene Software.
- ▶ TECHNISCHE DATEN enthält eine Tabelle mit zusätzlichen Informationen wie Version, Größe, Lizenz und Autoren.
- ▶ ABHÄNGIGKEITEN beschreibt, welche Elemente das Paket zur Verfügung stellt (PROVIDES) und welche es benötigt (REQUIRES). Die letzte Rubrik – OBSOLETES – zeigt an, welche älteren Pakete durch die vorliegende Version überflüssig werden.
- ▶ VERSIONEN stellt noch einmal ausführlich die Versionsnummern der diversen Paketbestandteile dar.

Wenn Sie ein Paket installieren möchten, müssen Sie das Kontrollkästchen ganz links in seiner Zeile anklicken. Um ein bereits zur Installation ausgewähltes Paket abzuwählen, klicken Sie das Kästchen erneut an. Weitere Statusvarianten finden Sie unter PAKET in der Menüleiste am oberen Bildschirmrand:

- ▶ INSTALLIEREN wählt das Paket zur Installation aus.
- ▶ NICHT INSTALLIEREN merkt vor, dass das Paket nicht installiert werden soll.
- ▶ BEHALTEN belässt das Paket bei einem Update oder einer Nachinstallation unverändert.
- ▶ LÖSCHEN entfernt ein bereits installiertes Paket.
- ▶ AKTUALISIEREN ersetzt die vorhandene Version des Pakets durch eine neuere.
- ▶ TABU – NIEMALS INSTALLIEREN sorgt dafür, dass das Paket auf keinen Fall installiert wird – selbst dann nicht, wenn dies Paketabhängigkeiten verletzen sollte. Für diese Option müssen Sie ganz genau wissen, was Sie tun – wählen Sie sie beispielsweise aus, wenn Sie die betreffende Software manuell installieren.
- ▶ GESCHÜTZT • NICHT VERÄNDERN behält den aktuellen Status des Pakets um jeden Preis bei.
- ▶ QUELLEN INSTALLIEREN fügt den Quellcode des jeweiligen Pakets hinzu, falls vorhanden.
- ▶ QUELLE NICHT INSTALLIEREN sorgt dafür, dass das Quellcodepaket nicht installiert wird – dies ist Standard.

Bei einer Neuinstallation sind nur die Optionen **INSTALLIEREN**, **NICHT INSTALLIEREN** oder **TABU** interessant; zusätzlich können Sie bestimmen, was gegebenenfalls mit den jeweiligen Quellpaketen geschehen soll. Alle anderen Optionen sind nur bei einem Upgrade oder bei der späteren Deinstallation beziehungsweise Nachinstallation von Software interessant.

Welche Pakete Sie installieren, hängt natürlich fast ausschließlich davon ab, was Sie mit dem System tun möchten. Lesen Sie sich die Zusammenfassungen und gegebenenfalls auch die genaueren Beschreibungen durch. Erste Anhaltspunkte bietet die Liste der verfügbaren Schemata weiter oben in diesem Kapitel; zusätzlich enthalten viele Kapitel dieses Buches detailliertere Paketlisten. Mitunter ist es am einfachsten, ein komplettes Schema zur Installation auszuwählen.

Nachdem Sie Ihre Paketauswahl getroffen haben, können Sie auf **PRÜFEN** klicken, um zu testen, ob alle Paketabhängigkeiten erfüllt sind. Der Überprüfungsablauf wählt dabei vollautomatisch die fehlenden Pakete zur Installation oder Aktualisierung aus. Häufig entstehen Paketkonfliktmeldungen also erst durch ein ausdrückliches Tabu. Wenn Sie eine solche Meldung erhalten, haben Sie in der Regel drei Möglichkeiten:

- ▶ Das fragliche Paket nachinstallieren beziehungsweise entsperren – meist die richtige Wahl.
- ▶ Das davon abhängige Paket nicht installieren – dies kann allerdings zu weiteren Konflikten führen.
- ▶ Den Konflikt ignorieren – bitte nur, wenn Sie wissen, warum er besteht, beispielsweise, weil Sie eine bestimmte Software manuell installiert haben.

Sofern Sie sich nicht für das Ignorieren entschieden haben, sollten Sie **OK • NEU PRÜFEN** anklicken, um sicherzugehen, dass keine anderen Konflikte mehr bestehen.

Sobald alle Abhängigkeiten aufgelöst sind, erhalten Sie die Meldung **ALLE PAKETABHÄNGIGKEITEN SIND OK**. Alternativ können Sie die Option **AUTOMATISCHE ÜBERPRÜFUNG** aktivieren, um die Prüfung in den Installationsablauf zu integrieren.

Wenn Ihre Paketauswahl gemäß den obigen Beschreibungen abgeschlossen ist, klicken Sie rechts unten im Fenster auf **ÜBERNEHMEN**.

2.3.2 Länderspezifische Einstellungen

In der Registerkarte **ÜBERBLICK** können Sie Unterstützung für zusätzliche Sprachen hinzufügen. Klicken Sie dazu den Link **LÄNDERSPEZIFISCHE EINSTELLUNG** an.

Im oberen Feld wird die primäre Sprache des Systems eingestellt. Wenn Sie sich vor der Installation für Deutsch entschieden haben, wird dies hier bereits eingetragen. Die Option `TASTATURBELEGUNG AN DEUTSCH ANPASSEN` ist nützlich, damit die Beschriftung einer deutschen Standardtastatur den erzeugten Zeichen entspricht. Die `ZEITZONE` haben Sie dagegen bereits eingestellt, wenn Sie der Anleitung in diesem Kapitel bis hierhin gefolgt sind.

Die Schaltfläche `DETAILS` öffnet einen Dialog mit zusätzlichen Einstellungen:

- ▶ Die `LOCALE-EINSTELLUNGEN FÜR DEN BENUTZER ROOT` legen fest, ob für den Superuser dieselben Spracheinstellungen gelten sollen wie für gewöhnliche Benutzer. Die Standardeinstellung ist `NUR CTYPE`, was unter anderem die Datums- und Uhrzeitformate für `root` auf die primäre Sprache einstellt, aber zahlreiche andere Einstellungen international (das heißt englisch) belässt. Die anderen verfügbaren Optionen sind `JA` (alles auf Deutsch einstellen) oder `NEIN` (alles auf Englisch). Die Voreinstellung für `ctype` bereitet in der Regel die wenigsten Probleme; falls Sie nur sehr wenig Englisch können, sollten Sie dagegen `JA` einstellen.
- ▶ `UTF-8 ALS KODIERUNG VERWENDEN` stellt den zu ASCII abwärtskompatiblen Unicode-Zeichensatz UTF-8 ein. Da dies auf Linux-Systemen inzwischen Standard ist, sollten Sie diese Einstellung aktiviert lassen. Wenn UTF-8 später in einzelnen Programmen Probleme bereitet, können Sie den Zeichensatz dort gezielt umstellen.
- ▶ `DETAILLIERTE LOCALE-EINSTELLUNG` erlaubt den Zugriff auf die geringfügigen Unterschiede zwischen dem Deutsch der verschiedenen deutschsprachigen oder teilweise deutschsprachigen Länder: Österreich (`DE_AT`), Belgien (`DE_BE`), Schweiz (`DE_CH`), Deutschland (`DE_DE`) und Luxemburg (`DE_LU`).

Unter `SEKUNDÄRE SPRACHEN` können Sie die Unterstützung für diverse zusätzliche Sprachen aktivieren. Dies betrifft unter anderem Tastaturschemata und Zeichensätze, Datums- und Uhrzeitformate, Wörterbücher und Silbentrennalgorithmus sowie in Einzelfällen Übersetzungen von Anwendungsprogrammen.

Wenn Sie den Unterpunkt `TASTATURBELEGUNG` anklicken, können Sie gegebenenfalls Detaileinstellungen zur Tastaturbelegung vornehmen. Unter Linux erzeugen die diversen Akzent-Tasten normalerweise direkt das jeweilige Einzelzeichen. Wenn Sie häufig Texte in europäischen Fremdsprachen wie etwa Französisch eingeben, könnten Sie den Wechsel auf die Belegung `DEUTSCH (MIT »TOTEN« TASTEN)` erwägen; in diesem Modus funktionieren die Akzenttasten wieder wie unter Windows: Der Tastendruck selbst erzeugt noch kein Zeichen; je nach der Art des nächsten Zeichens wird der Akzent dann aufgesetzt oder separat davor gesetzt. Die Schaltfläche `EINSTELLUNGEN FÜR EXPERTEN` öffnet einen zusätzlichen Dialog,

in dem Sie erweiterte Aspekte der Tastatur wie Wiederholrate oder Status der Num-Lock-Taste einstellen können.

2.3.3 Weitere Einstellungen

In der Installations-Übersicht finden Sie noch einige weitere Optionen, die Sie in vielen Standardfällen nicht zu ändern brauchen – einige von ihnen, beispielsweise Partitionierung, Benutzerkonto und Zeitzone, haben Sie zudem bereits vorher eingestellt. Ein Blick auf die voreingestellten Werte in den zusätzlichen Optionen ist jedoch in jedem Fall empfehlenswert, damit Sie keine unliebsamen Überraschungen erleben, die Sie später zur Neuinstallation zwingen. Die Registerkarte enthält folgende Punkte:

- ▶ **SYSTEMSTART**
Hier können Sie Details für den Bootloader einstellen. Näheres dazu erfahren Sie in Kapitel 9, »Datenträger und Dateisysteme«.
- ▶ **SYSTEMSTART**
Hier wird der Bootmanager konfiguriert. Sie sollten die Einstellungen unter diesem Punkt besonders dann genauer überprüfen, wenn Ihr Rechner mehrere Betriebssysteme enthält. Details werden in Kapitel 9 erläutert.
- ▶ **STANDARD-RUNLEVEL**
Die Runlevel sind verschiedene Betriebsmodi eines Linux-Rechners. Dieses Konzept ermöglicht das automatische Starten und Beenden bestimmter Programme und Dienste für jeden dieser Modi; es wird im nächsten Kapitel behandelt. Das **RUNLEVEL 5: VOLLER MEHRBENUTZEBETRIEB MIT NETZWERK UND DISPLAY-MANAGER** ist in den allermeisten Fällen die korrekte Standardeinstellung.
- ▶ **INSTALLATION VON ABBILDERN**
Diese Option beschleunigt die Installation noch einmal erheblich, da sie größere Softwaresammlungen nicht aus Einzelpaketen, sondern aus Images installiert. Deshalb sollten Sie sie unbedingt aktivieren.

2.4 Erstkonfiguration nach der Installation

Nachdem Sie alle oben beschriebenen Einstellungen zur Installation vorgenommen haben, können Sie auf die Schaltfläche **INSTALLIEREN** am rechten unteren Bildschirmrand klicken. Ein Dialogfeld fordert Sie auf, die Installation zu bestätigen. Klicken Sie nochmals auf **INSTALLIEREN**, wenn Sie sicher sind, dass Sie alles korrekt eingestellt haben, oder auf **ZURÜCK**, um Ihre Einstellungen nochmals zu überprüfen und gegebenenfalls zu überarbeiten.

Die eigentliche Installation kann eine Weile dauern; seit openSUSE 11.0 sollte sie aufgrund der diversen Beschleunigungen aber nicht mehr länger als eine halbe bis Dreiviertelstunde benötigen.

Nachdem die eigentliche Installation abgeschlossen ist, werden noch einige Konfigurationsschritte durchgeführt. Diese Abschlussarbeiten werden hier eher kurz beschrieben, weil Sie sämtliche Einstellungen später noch einmal ändern können. Dies wird in späteren Kapiteln dieses Buches im Zusammenhang mit dem nötigen Hintergrundwissen genauer erläutert.

Die nachfolgenden Unterabschnitte behandeln nacheinander die einzelnen Bildschirme, die während der Abschlussarbeiten zur Installation angezeigt werden. Falls Sie sich bei den Voreinstellungen zur Installation für die automatische Konfiguration entschieden haben, entfallen sie zum größten Teil.

2.4.1 Host- und Domainname

Wenn Ihr Rechner Teil eines lokalen Netzwerks werden soll, benötigt er darin einen eindeutigen Namen, der als *Hostname* bezeichnet wird (*Host*, englisch für Wirt oder Gastgeber, ist eine allgemeine Bezeichnung für Rechner in einem Netzwerk). Ein Einzelplatzrechner mit Wählverbindung ins Internet (dazu gehören auch DSL-Anschlüsse) braucht eigentlich keinen Hostnamen, da ihm bei jeder Einwahl dynamisch eine Adresse zugewiesen wird. Viele Anwendungen bestehen allerdings auch bei Einzelplatzrechnern auf einem Hostnamen, sodass Sie hier auf jeden Fall etwas eintragen sollten. In diesem Buch wird durchgehend der Beispielhostname `tux` verwendet, solange nicht die Vernetzung mehrerer Rechner beschrieben wird. Es steht Ihnen allerdings frei, hier einen beliebigen Namen aus Buchstaben, Ziffern und Bindestrichen (-) einzugeben; zwischen Groß- und Kleinschreibung wird nicht unterschieden.

Der *Domainname* ist das gemeinsame Namenssuffix aller Rechner innerhalb eines Netzwerks, das deren Zusammengehörigkeit etwa auch im Internet kennzeichnet. In Heimnetzen oder kleinen Firmennetzen ist es im Allgemeinen nicht üblich, den internen Arbeitsplatzrechnern einen im Internet gültigen Domainnamen zuzuweisen; dies können Sie auch gar nicht ohne Weiteres selbst erledigen, weil dieser Name registriert werden muss. Wenn Sie also keinen offiziellen Domainnamen für den lokalen Gebrauch besitzen, sollten Sie hier einen privaten Namen eintragen, gekennzeichnet durch die inoffizielle Top-Level-Domain `.local` (im Gegensatz zu offiziellen TLDs wie `.de` oder `.com`, die im Internet zum Einsatz kommen). Im Rahmen dieses Buches kommt `test.local` als Standardbeispiel zum Einsatz.

Die Option `HOSTNAMEN ÜBER DHCP ÄNDERN` sorgt dafür, dass Ihnen Host- und Domainname gegebenenfalls automatisch über einen DHCP-Server zugeteilt werden. Diese Serversoftware ermöglicht die vollautomatische Konfiguration der Netzwerkparameter auf Rechnern im lokalen Netzwerk. Wie Sie den DHCP-Server auf Ihrem openSUSE-Rechner einrichten können, wird in Kapitel 14, »Datei- und Verzeichnisdienstserver«, besprochen.

Beachten Sie, dass die Vergabe von Host- und Domainnamen allein nicht dafür sorgt, dass der Rechner unter diesem Namen im Netzwerk erreichbar ist. Dazu müssen Sie mindestens die Namensauflösungsdatei `/etc/hosts` anpassen (es genügt, hier die Option `HOSTNAMEN IN /ETC/HOSTS` schreiben zu aktivieren) oder noch besser einen Domain Name Server (DNS-Server) wie BIND installieren. Beides wird ausführlich in Kapitel 13, »Netzwerkgrundlagen«, beschrieben.

2.4.2 Netzwerkkonfiguration

Der in Abbildung 2.11 gezeigte Bildschirm ist der umfangreichste Dialog der Erstkonfiguration. An dieser Stelle werden die Einstellmöglichkeiten relativ kurz erläutert; ausführlichere Informationen zur Einrichtung von Netzwerkhardware, -protokollen und -diensten erhalten Sie in Kapitel 13 und den nachfolgenden Kapiteln. Falls Sie also komplizierte Netzwerkhardware oder ausgefallene Konfigurationswünsche haben, aktivieren Sie jetzt einfach die Option `KONFIGURATION ÜBERSPRINGEN`, und nehmen Sie die Netzwerkeinstellungen später in Ruhe vor.

Im Einzelnen enthält der Dialog folgende Rubriken, die Sie wie bei den Installationseinstellungen entweder über die Hyperlink-Überschriften oder über das Menü `ÄNDERN` modifizieren können:

- ▶ **NETZWERKEINSTELLUNGEN** mit den Unterpunkten `NETZWERKMODUS` und `IPV6`. Die Option `NETZWERKMODUS` bestimmt, ob das neuartige Modul `NETWORKMANAGER` (siehe Kapitel 13) installiert werden soll oder nicht. Die Installation ist besonders empfehlenswert, wenn Ihr Rechner oft das Netzwerk wechselt (etwa ein Notebook, das am Arbeitsplatz und zu Hause eingesetzt wird). Standardmäßig ist der `NETWORKMANAGER` deaktiviert; jeder Klick wechselt seinen Status. Ebenso können Sie die Unterstützung für `IPV6` (Neufassung des IP-Protokolls mit 128 Bit langen Adressen; siehe Kapitel 6, »Wichtige Hardware konfigurieren«) einschalten.
- ▶ **FIREWALL**
openSUSE beziehungsweise SUSE Linux wird seit einigen Versionen mit einer leistungsfähigen lokalen Firewall ausgeliefert, die auf dem Kernelmodul `netfilter/iptables` basiert. Standardmäßig ist die `FIREWALL` aktiviert, was Sie in aller Regel beibehalten sollten. Wenn Sie den Link anklicken, können Sie wäh-

len, ob die Firewall BEIM SYSTEMSTART (die zu empfehlende Voreinstellung) oder MANUELL aktiviert werden soll. Klicken Sie danach auf ÜBERNEHMEN, um Ihre Wahl zu bestätigen. Darunter finden Sie ein bis zwei weitere Links: Wenn Sie das letzte Wort von FIREWALL IST EINGESCHALTET/AUSGESCHALTET anklicken, wird der jeweils andere Zustand herbeigeführt.

Wenn die Firewall aktiviert ist, können Sie zusätzlich bestimmen, ob der SSH-PORT GEBLOCKT oder GEÖFFNET werden soll. Mit SSH, der Secure Shell, können Sie von einem anderen Rechner aus über eine verschlüsselte Netzwerkverbindung auf dem Computer arbeiten, als säßen Sie direkt davor. Wenn Sie die Firewall für weitere Netzwerkdienste öffnen möchten, müssen Sie dies nach der Installation tun. All das wird ausführlich in Kapitel 13, »Netzwerkgrundlagen«, und im Folgenden besprochen.

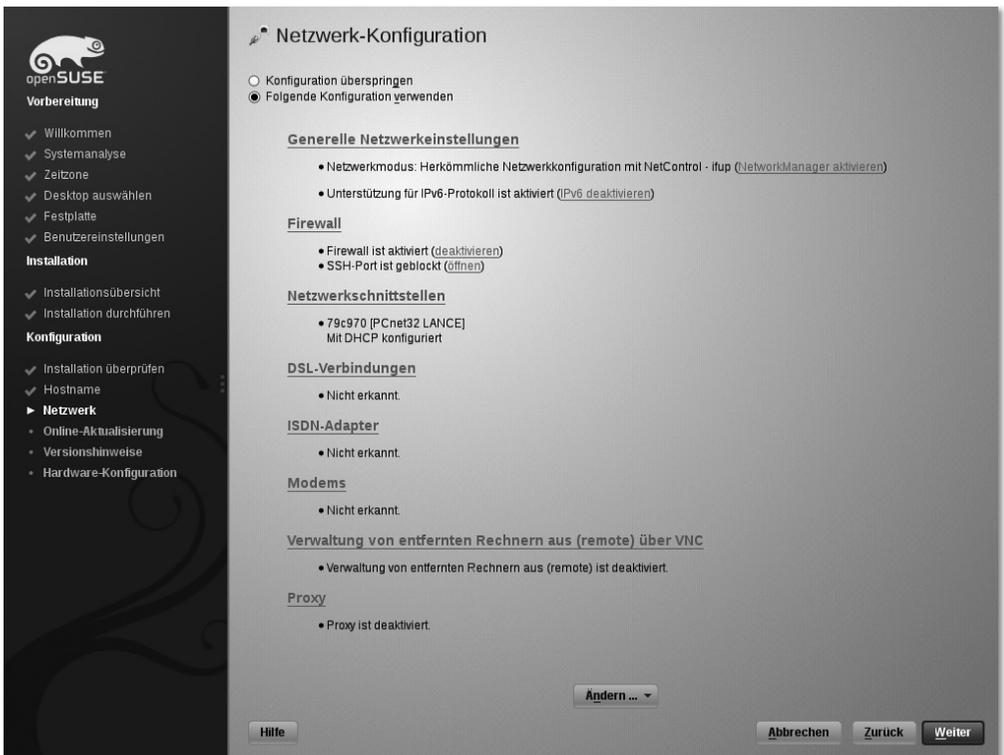


Abbildung 2.11 Der Dialog zur Netzwerkkonfiguration nach der Installation

► NETZWERKSCHNITTSTELLEN

Hier werden die verschiedenen in Ihrem Rechner verfügbaren Netzwerkkarten (oder Onboard-Netzwerkanschlüsse) konfiguriert. Neben Ethernet kön-

nen auch Wireless LAN, USB-Netzwerkschnittstellen und weitere Varianten konfiguriert werden. Klicken Sie den Link an, und Sie erhalten eine Liste der bereits erkannten Netzwerkkarten. Gängige Hardware wird in der Regel automatisch erkannt.

Wenn eine Netzwerkschnittstelle in der Liste fehlt, können Sie auf HINZUFÜGEN klicken, um sie manuell zu ergänzen. Unter Gerätetyp (Voreinstellung ETHERNET) wird die grundsätzliche Art der Schnittstelle gewählt. KONFIGURATIONSNAMEN ist eine unterscheidende Nummer, falls mehrere Schnittstellen desselben Typs vorhanden sind. Wenn Sie das für Ihre Karte zuständige Kernelmodul kennen (zum Beispiel aus dem Handbuch der Netzwerkkarte), können Sie MODULNAME und OPTIONEN manuell eintragen. In der Regel ist dies allerdings nicht der Fall, sodass Sie die Schaltfläche AUSWAHL AUS LISTE anklicken sollten.

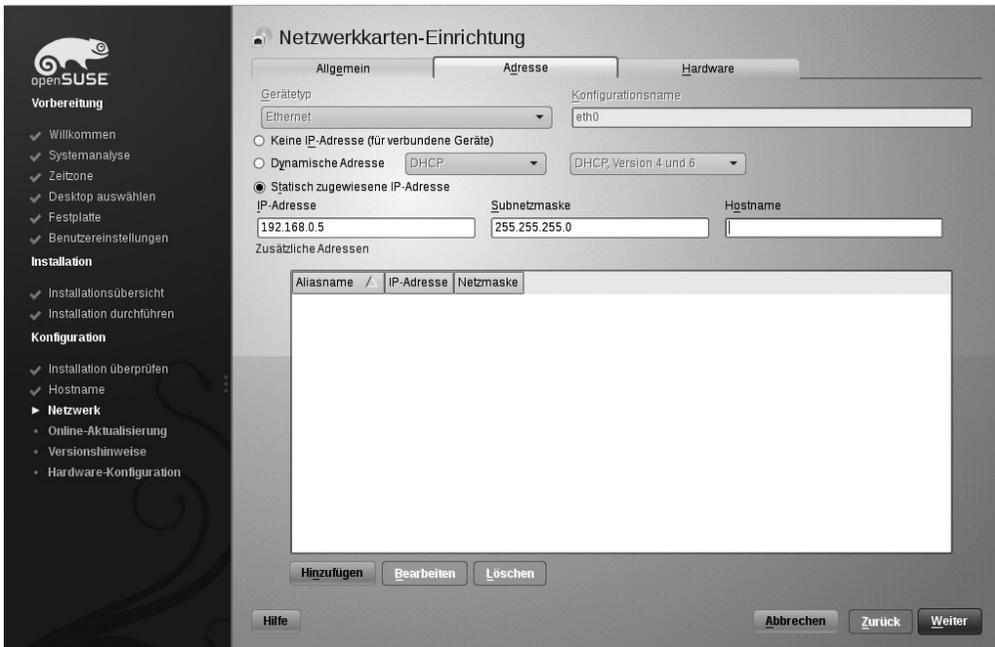


Abbildung 2.12 Einstellen der IP-Adresse einer Netzwerkkarte

Um die Einstellungen für eine vorhandene Karte zu ändern, wählen Sie sie aus der Liste aus, und klicken Sie auf BEARBEITEN. Der Bildschirm aus Abbildung 2.12 wird angezeigt. Auf der Registerkarte ADRESSE wird die IP-ADRESSE der Netzwerkkarte eingestellt. Dies ist eine 32 Bit lange, per Software konfigurierbare Nummer, die eine Netzwerkschnittstelle im Netzwerk beziehungsweise

Internet eindeutig kennzeichnet. Näheres über IP-Adressen erfahren Sie in Kapitel 13; dort werden übrigens auch die Einstellungen der Registerkarte ALLGEMEIN besprochen.

Standardmäßig ist die Option DYNAMISCHE ADRESSE, DHCP voreingestellt. Wenn sich in Ihrem Netzwerk ein DHCP-Server befindet, brauchen Sie nichts weiter einzustellen, denn dann bezieht die Schnittstelle alle erforderlichen Einstellungen automatisch. Da ein DHCP-Server nicht von selbst vorhanden ist, wissen Sie normalerweise Bescheid, dass er da ist. Allerdings gibt es auch zahlreiche moderne DSL- und/oder WLAN-Router, die ab Werk mit DHCP-Funktionen ausgestattet sind. Dies können Sie dem Handbuch eines solchen Geräts entnehmen.

Statt DHCP können Sie in dem Pull-down-Menü auch die Option ZEROCONF auswählen, die dafür sorgt, dass der Rechner der Netzwerkkarte selbst eine IP-Adresse zuweist, die aufgrund bestimmter Konventionen nicht mit anderen Adressen kollidiert. Die Variante DHCP + ZEROCONF schließlich verwendet immer dann Zeroconf, wenn kein DHCP-Server erreichbar ist.

Wenn kein DHCP-Server vorhanden ist und Sie kein Zeroconf verwenden möchten, müssen Sie die Einstellungen manuell vornehmen. Wählen Sie KONFIGURATION DER STATISCHEN ADRESSE, und geben Sie passende Werte ein. Wenn bereits ein lokales Netzwerk besteht, dann müssen die IP-Adresse und die anderen Werte des neuen Rechners dazu passen. Wenn Sie das Netzwerk dagegen ganz neu anlegen, gelten kurz gesagt folgende Regeln:

- ▶ Geben Sie unter IP-ADRESSE etwas aus dem Nummernkreis 192.168.x.y ein – das »x« muss eine ganze Zahl zwischen 0 und 255 sein und bei allen Rechnern gleich sein; typischerweise wird die 0 gewählt. Das »y« muss dagegen für jeden Rechner (genauer gesagt: für jede Schnittstelle) einen anderen Wert haben, wobei die zulässigen Werte hier 1 bis 254 sind. 192.168.x.y ist einer der privaten IP-Adressbereiche, die Sie ohne Anmeldung für Ihr lokales Netz benutzen dürfen, weil sie nicht mit dem Internet in Konflikt geraten können. Wenn Sie beispielsweise drei Rechner haben, könnten Sie ihnen die Adressen 192.168.0.2, 192.168.0.3 und 192.168.0.4 zuweisen.
- ▶ Die SUBNETZMASKE muss 255.255.255.0 lauten (drei Byte kennzeichnen das Netzwerk, ein Byte den konkreten Rechner). Dieser Wert ist bereits voreingestellt.
- ▶ Unter HOSTNAME brauchen Sie für eine lokale Netzwerkschnittstelle nur dann etwas einzutragen, wenn der Rechner Teil eines Firmennetzwerks (meist mit gemeinsamem Internetzugang) wird. Details erfahren Sie in diesem Fall von Ihrem Administrator oder in Kapitel 13.

Die Registerkarte **HARDWARE** ermöglicht die Einstellung von Hardwareadresse und Treiber.

- ▶ **DSL-GERÄTE** – an dieser Stelle können Sie den DSL-Anschluss zu Ihrem Internetprovider konfigurieren. Auf der Registerkarte **DSL-GERÄTE** werden die gegebenenfalls erkannten DSL-Anschlüsse aufgelistet; falls Ihre DSL-Schnittstelle nicht erkannt wurde, müssen Sie auf **HINZUFÜGEN** klicken und sie manuell konfigurieren. Wählen Sie den **PPP-MODUS**, je nachdem, wie Ihr Rechner mit dem DSL-Modem oder -Router verbunden ist. Unter **GERÄTE-AKTIVIERUNG** können Sie festlegen, ob die Verbindung automatisch oder manuell hergestellt werden soll. Die weiteren Einstellungsmöglichkeiten entnehmen Sie bitte der detaillierten Anleitung in Kapitel 13. Falls ein DSL-Gerät erkannt wurde, können Sie auf **BEARBEITEN** klicken und seine Einstellungen überprüfen.

Ein Klick auf **WEITER** – oder auf **HINZUFÜGEN** auf der Registerkarte **PROVIDER** – erlaubt Ihnen die Auswahl oder Eingabe Ihres Internetproviders. Wählen Sie diesen aus der Liste aus, oder klicken Sie auf **NEU**, um Ihren eigenen ISP anzugeben. In beiden Fällen müssen Sie anschließend Ihren Benutzernamen und Ihr Passwort eingeben; diese sollten Sie vom Provider erhalten haben.

- ▶ Wenn Sie eine ISDN-Karte besitzen, können Sie diese unter **ISDN-ADAPTER** konfigurieren. Dies funktioniert sinngemäß genauso wie bei den DSL-Schnittstellen; die Details erfahren Sie auch hier in Kapitel 13.
- ▶ Unter **MODEMS** können Sie analoge Wählverbindungen ins Internet mithilfe eines klassischen Modems konfigurieren. Auch dies wird im Netzwerk-Kapitel 13 genauer beschrieben.
- ▶ **VERWALTUNG VON ENTFERNTEN RECHNERN AUS (REMOTE) ÜBER VNC** erlaubt das vollwertige Arbeiten mit der grafischen Oberfläche des Rechners über das Netzwerk, entweder mithilfe eines speziellen VNC-Clients oder mit einem Java-fähigen Webbrowser. Wenn Sie den Link anklicken, können Sie den VNC-Zugriff ein- oder ausschalten. In diesem Zusammenhang können Sie auch gleich den entsprechenden Firewall-Port öffnen.
- ▶ **PROXY** – hier können Sie Proxy-Server für diverse Internet-Anwendungsprotokolle konfigurieren. Dies ist notwendig, wenn kein vollwertiger Internetzugang über einen Router zur Verfügung steht. Auch in anderen Fällen kann es nützlich sein, Proxies zu aktivieren, weil diese häufig genutzte Webinhalte in einem Cache aufbewahren, was Zugriffe erheblich beschleunigen kann. Die Details stehen in Kapitel 14, »Datei- und Verzeichnisdienstserver«.

Wenn Sie Ihre Netzwerkeinstellungen vorgenommen haben, klicken Sie auf **WEITER**. Anschließend besteht die Möglichkeit, die soeben eingestellte Internetverbindung zu testen.

Wenn die Internetverbindung erfolgreich hergestellt werden konnte, können Sie anschließend Ihr erstes Online-Update durchführen. openSUSE sucht auf wählbaren Mirror-Servern nach Sicherheits- und/oder Feature-Updates aller installierten Softwarepakete, lädt sie automatisch herunter und installiert sie. Dieses Online-Update können Sie später regelmäßig durchführen, entweder vollautomatisch oder auf Anfrage (siehe nächstes Kapitel).

2.4.3 Abschluss der Installation

Abschließend werden noch einige Arbeitsschritte durchgeführt. Nach dem Erstellen des ersten Benutzers sehen Sie die Meldung `UPDATE DER KONFIGURATION`. Hier werden Ihre soeben vorgenommenen Einstellungen in die Konfigurationsdateien geschrieben. Ein Eingreifen Ihrerseits ist an dieser Stelle nicht möglich.

Der nächste Bildschirm ist die `REGISTRIERUNG`. Falls Sie im vorherigen Schritt eine Internetverbindung eingerichtet haben, können Sie sich an dieser Stelle bei Novell registrieren, um Online-Updates und Support zu erhalten. Danach folgt der Bildschirm `ONLINE-UPDATE`, in dem Sie nach neueren Versionen der installierten Pakete suchen können – natürlich auch wieder nur dann, wenn bereits ein Internetzugang besteht. Die nachfolgenden `BENUTZER` haben Sie bereits im Rahmen der Voreinstellungen konfiguriert.

Nachdem die Konfiguration geschrieben wurde, werden die `VERSIONSHINWEISE` (Release Notes) angezeigt; hier erhalten Sie zusätzliche Tipps und Hinweise zur aktuellen Version. Lesen Sie sie durch, soweit Sie möchten, und klicken Sie erneut auf `WEITER`.

Der letzte Schritt ist die `HARDWARE-KONFIGURATION`. Hier können Sie einige zusätzliche Hardwarekomponenten einrichten. Da dies in den nachfolgenden Kapiteln jeweils detailliert erläutert wird, können Sie sich die einzelnen Punkte dort kurz anschauen, aber erst einmal unverändert lassen. Dass ein bestimmtes Gerät hier noch nicht erkannt wird, heißt noch lange nicht, dass es nicht unterstützt würde.

Wenn Sie ein letztes Mal auf `WEITER` klicken, wird die neu erkannte Hardware eingerichtet; anschließend erscheint der Bildschirm `INSTALLATION ABGESCHLOSSEN`. Die Option `DIESES SYSTEM FÜR AUTOYAST KLONEN` ist interessant, wenn Sie das System anschließend auf mehreren baugleichen Rechnern installieren möchten.

Klicken Sie nun auf `BEENDEN`. Ihre Installation ist abgeschlossen. Im nächsten Kapitel erfahren Sie nach einem kurzen Rundgang durch Ihre neue Systemumgebung, wie Sie weitere Konfigurationsschritte durchführen können.

2.5 Zusammenfassung

In der Frühzeit von Linux war die Installation eine beinahe wissenschaftliche Angelegenheit für Experten. Die Schritt-für-Schritt-Installation moderner Distributionen ist dagegen für jeden zu bewältigen, der über etwas allgemeine Computererfahrung verfügt. Dennoch kann es einige Tücken geben, die aber in diesem Kapitel ausführlich erläutert wurden.

Zunächst müssen Sie klären, wohin Sie openSUSE installieren möchten. Idealerweise ist eine ganze Partition oder Festplatte frei, oder Sie haben genug Platz, die Inhalte einer bestehenden Partition zu verschieben. Wenn nicht, dann muss das Installationsprogramm eine existierende Partition verkleinern, um Platz für Ihre Linux-Installation zu schaffen. Dazu müssen Sie diese Partition zunächst defragmentieren und eventuell die Windows-Auslagerungsdatei verschieben.

Für die eigentliche Installation steht mit YaST ein komfortables und durchdachtes Werkzeug zur Verfügung. Der Ablauf der Installation lässt sich in die drei Hauptschritte Voreinstellungen, Installationseinstellungen und Erstkonfiguration unterteilen. Die eigentliche Installation, das Kopieren der Softwarepakete auf die Festplatte, findet nach den Installationseinstellungen statt.

Zu den Voreinstellungen gehören unter anderem die Anerkennung der Lizenzbedingungen, die Auswahl der voreingestellten Desktop-Umgebung (KDE oder GNOME), die Einrichtung des ersten Benutzerkontos und die Partitionierung der Festplatten.

Die Installationseinstellungen umfassen standardmäßig die Auswahl der Softwarepakete und die Konfiguration des Bootloaders. Auch die bereits im Rahmen der Voreinstellungen vorgenommene Konfiguration können Sie hier nochmals ändern. Nachdem Sie alles eingestellt haben, heißt es Geduld haben – das Installieren der Softwarepakete kann etwa 30–45 Minuten dauern.

Zum Schluss können Sie erste Konfigurationseinstellungen vornehmen, beispielsweise Netzwerk und diverse Hardware. Mit weiteren Konfigurationsoptionen, die Sie nach der Installation durchführen können, geht es im nächsten Kapitel weiter.

*Think I'll go on a walkabout
and find out what it's all about.
– Red Hot Chili Peppers*

3 Systemrundgang und Grundkonfiguration

Nachdem Sie das System mithilfe der ausführlichen Anleitung im vorigen Kapitel hoffentlich erfolgreich installiert haben, wird es Zeit, sich ein wenig darin umzuschauen. Der erste Abschnitt dieses Kapitels bietet daher eine erste Orientierungshilfe auf Ihrem KDE- oder GNOME-Desktop. Im zweiten erfahren Sie das Wichtigste über die verschiedenen Hilfe-Systeme und elektronischen Handbücher. Im dritten Abschnitt schließlich wird dann grundlegend erläutert, wie Sie das System an Ihre Bedürfnisse anpassen können: Sie erfahren, wie Sie das bereits für die Installation eingesetzte Tool YaST für Konfigurationsaufgaben nutzen und wie zusätzliche Software installiert wird.

3.1 Das System kennenlernen

In diesem kurzen Abschnitt erhalten Sie erste Orientierungshilfen, um sich in Ihrem neuen System zurechtzufinden. Nach der Anmeldung werden die ersten Schritte mit den Desktops KDE und GNOME beschrieben.

3.1.1 Booten und Anmeldung

Nachdem die Installation erfolgt ist, können Sie das fertig eingerichtete System sofort benutzen. Wenn Sie es zu einem späteren Zeitpunkt in Betrieb nehmen möchten, schalten Sie wie gewohnt den Rechner ein. Sofern Sie den Bootmanager nicht manuell umkonfiguriert haben (was in diesem Buch ohnehin erst später zur Sprache kommt), wird nach den BIOS-Meldungen Ihres Rechners ein Bootmenü angezeigt, das etwa so aussieht wie in Abbildung 3.1.

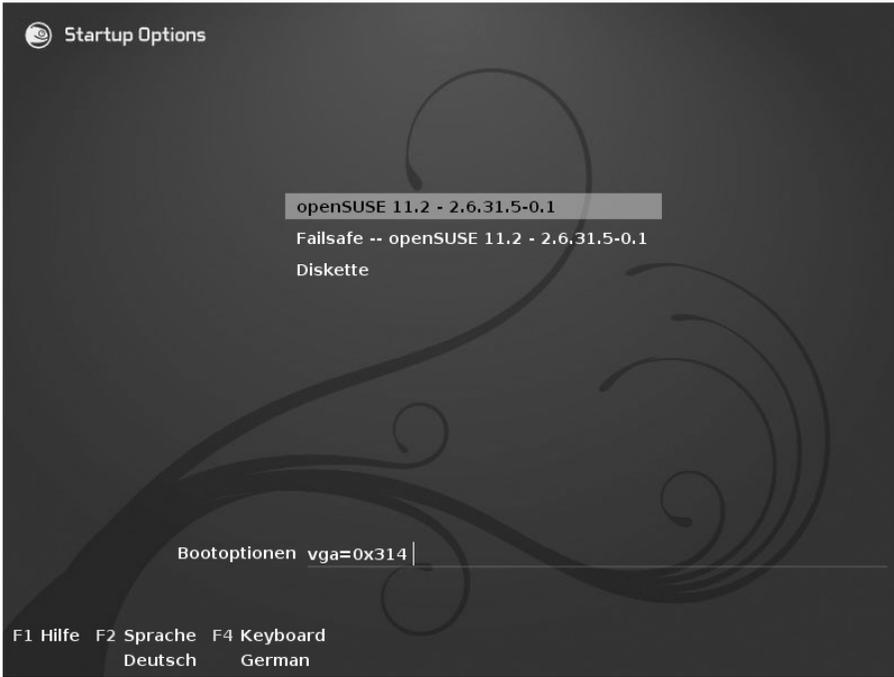


Abbildung 3.1 Das Bootmenü von openSUSE nach der Installation

Wenn auf Ihrem Rechner bereits vor der openSUSE-Installation ein Betriebssystem existierte, gibt es dafür einen zusätzlichen Menüeintrag, beispielsweise WINDOWS. Auf die restlichen Einträge wird in Kapitel 9, »Datenträger und Dateisysteme«, im Rahmen der Konfiguration des Bootloaders näher eingegangen. Im Moment können Sie einfach den Eintrag OPEN SUSE 11.2 (dahinter steht noch die aktuelle Kernelversion) wählen und **↵** drücken; bei einer Standardkonfiguration geschieht dies nach zehn Sekunden automatisch.

Näheres über die wichtigsten BOOTOPTIONEN erfahren Sie, wenn Sie die Taste **F1** für HILFE betätigen. Diese Optionen ermöglichen es Ihnen, auf spezifische Probleme beim Booten zu reagieren – beispielsweise können Sie das Power-Management ganz oder teilweise deaktivieren.

Das eigentliche Booten dauert eine Weile; wenn Sie möchten, können Sie sich die Zeit vertreiben, indem Sie **ESC** drücken und die Meldungen über die startenden Systemdienste und Programme lesen. Dies gibt eventuell auch Hinweise auf Schwierigkeiten. Wenn Sie Ihr System bei der Installation nicht für die automatische Anmeldung des ersten Benutzers konfiguriert haben, erscheint nach einiger

Zeit ein Bildschirm wie in Abbildung 3.2. Hier handelt es sich um den KDE-Anmeldemanager KDM; alternativ stehen GDM (GNU), XDM (Klassiker des X Window Systems) und einige weitere zur Verfügung – siehe Kapitel 5.

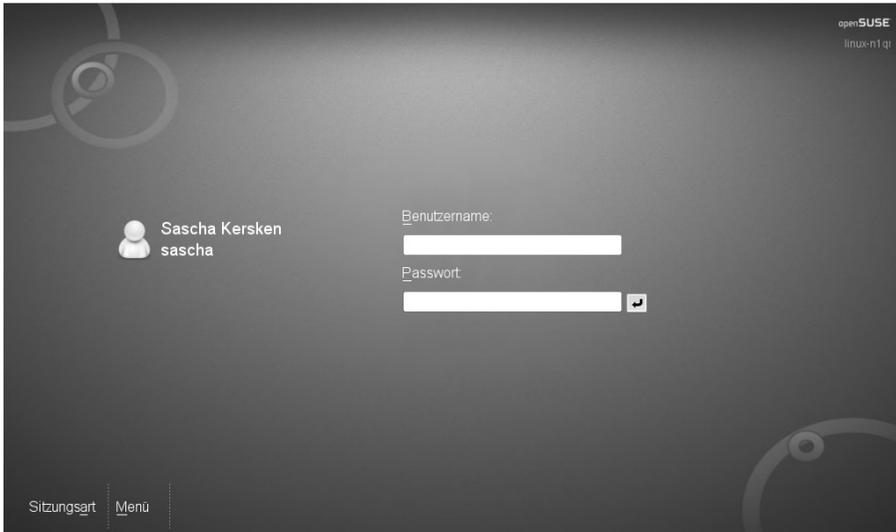


Abbildung 3.2 Der grafische Anmeldebildschirm von openSUSE (hier KDM)

Unter **BENUTZERNAME** können Sie einen existierenden Benutzer eingeben oder per Mausclick aus der Liste links auswählen; oft ist der erste Benutzer hier bereits eingetragen. Geben Sie als Nächstes das **PASSWORT** des entsprechenden Users ein (es werden nur ******* angezeigt). Klicken Sie danach auf den Pfeil daneben oder drücken Sie , um sich mit diesen Benutzerdaten anzumelden.

Theoretisch können Sie auch den Benutzernamen `root` eingeben, wenngleich er nicht in der Liste erscheint. In der Praxis ist dies aus Sicherheitsgründen nicht empfehlenswert; wenn Sie es dennoch tun, zeigt KDE zur Warnung einen knallroten Desktop-Hintergrund mit Bomben-Symbolen. In aller Regel ist es ratsamer, sich als normaler Benutzer anzumelden und dann gegebenenfalls einzelne Programme mit `root`-Rechten zu starten.

Wenn auf Ihrem System mehrere Desktops oder Window-Manager installiert sind, können Sie sich unter **SITZUNGSART** einen von ihnen aussuchen – neben KDE und/oder GNOME steht meist noch eine Reihe von Alternativen zur Verfügung, die in Kapitel 5, »Grafische Oberflächen«, kurz angesprochen werden. Standardmäßig wird der Desktop der vorigen Sitzung gestartet.

Unter MENÜ finden Sie einige zusätzliche Optionen:

- ▶ Mit **BENUTZER WECHSELN** können Sie zu einer eventuell parallel laufenden grafischen Oberfläche wechseln.
- ▶ **ANMELDUNG AUF FREMDRECHNER** startet eine grafische Sitzung auf einem anderen Rechner im Netzwerk, das heißt, die Programme werden auf dem anderen Computer ausgeführt, aber die Anzeige erfolgt bei Ihnen. Geben Sie den Netzwerknamen des betreffenden Rechners ein, und klicken Sie auf **HINZUFÜGEN**. Näheres dazu erfahren Sie in Kapitel 14, »Datei- und Verzeichnisdienstserver«. Wählen Sie **MENÜ • LOKALE ANMELDUNG**, wenn Sie sich doch lieber auf Ihrem eigenen Rechner einloggen möchten.
- ▶ Mit **TEXTKONSOLEN-MODUS** wechseln Sie zum ersten der sogenannten **VIRTUELLEN TERMINALS** (siehe nächstes Kapitel), in denen Sie die Shell zur manuellen Befehlseingabe nutzen können. Drücken Sie **[Alt]+[F7]**, um zum grafischen Anmeldebildschirm zurückzukehren.
- ▶ Wenn Sie **HERUNTERFAHREN** wählen, können Sie den Computer ausschalten oder neu starten.

Nachdem die grafische Oberfläche gestartet ist, geht es je nach Desktop im nächsten (KDE) oder übernächsten Abschnitt (GNOME) weiter.

Sollte Ihr Rechner eine der wenigen Grafikkarten besitzen, mit denen openSUSE gar nichts anfangen kann, dann befinden Sie sich nach dem Booten auf der Textkonsole. Hier wird ebenfalls ein Login verlangt – geben Sie Ihren Benutzernamen ein, gefolgt von **[↵]**. Anschließend werden Sie nach Ihrem Passwort gefragt; die Eingabe erzeugt überhaupt kein visuelles Feedback. Wie es danach an dieser Stelle weitergeht, erfahren Sie im nächsten Kapitel. Wenn Sie die grafische Oberfläche manuell konfigurieren möchten, können Sie in Kapitel 5 weiterlesen.

3.1.2 Erste Schritte mit KDE

Wie bereits im Rahmen der Installation im vorigen Kapitel erwähnt, wird openSUSE 11.2 nur noch mit der KDE-Version 4 geliefert. Diese gilt seit über einem Jahr als stabil, und die aktuelle Unterversion 4.3 hat die meisten Mängel und Probleme überwunden. Deshalb wird in dieser Auflage zum ersten Mal ausführlich KDE 4 behandelt.

In Abbildung 3.3 sehen Sie die openSUSE-Version des KDE-4-Desktops unmittelbar nach dem Start. Die wichtigsten Elemente sind die Desktop-Symbole, auf die Sie per Einzelklick zugreifen können, sowie die Leiste am unteren Bildschirmrand, die bei KDE als *Panel* bezeichnet wird. Die Schaltfläche ganz links öffnet ein umfangreiches Menü, das nicht nur bezüglich seiner Position dem bekannten

Windows-Startmenü nachempfunden wurde. Es wird K-Menü genannt, obwohl bei openSUSE die Form des hauseigenen Chamäleons *Gecko* statt des blauen K-Logos von KDE erscheint. Hier erhalten Sie schnellen Zugriff auf thematisch sortierte Untermenüs mit Programmen und Systemeinstellungen sowie Funktionen wie Ausschalten oder Neustart des Rechners. Rechts daneben finden Sie in der Standardkonfiguration folgende Symbole, die Sie durch einen einzelnen Klick aktivieren können:

► PLASMA-DASHBOARD ANZEIGEN

Blendet alle aktuellen Fenster aus und zeigt stattdessen das Dashboard der KDE-4-Oberfläche Plasma an. Neben den (per Konfiguration ein- oder ausblendbaren) Desktopsymbolen können Sie hier eine Reihe von Miniprogrammen hinzufügen, die ohne Start an der gewünschten Stelle angezeigt werden. Rechts oben auf dem Bildschirm befindet sich ein Menü zur Verwaltung der Miniprogramme und anderer Aspekte des Dashboards.

► FIREFOX WEBBROWSER

KDE bringt zwar einen eigenen Webbrowser namens *Konqueror* mit, aber openSUSE konfiguriert sowohl unter KDE als auch unter GNOME den *Mozilla Firefox* als Standardbrowser, was den Wünschen der meisten Anwender entsprechen dürfte. Näheres über Firefox erfahren Sie in Kapitel 8, »Desktop-Software«.

► DOLPHIN DATEIMANAGER

Öffnet Ihr Home-Verzeichnis im KDE-4-Dateimanager *Dolphin*. Von hier aus können Sie die Festplatte und andere Datenträger durchstöbern sowie Datei- und Verzeichnisoperationen durchführen, soweit Ihre Benutzerrechte dies zulassen.

Sie selbst können beliebige Programmverknüpfungen zum Panel hinzufügen, indem Sie ihre Symbole aus einem Verzeichnis oder vom Desktop hineinziehen.

Die mit 1 bis 4 beschrifteten rechteckigen Felder ermöglichen die Verwaltung mehrerer separater Desktop-Oberflächen oder Arbeitsflächen. Sie können zwischen ihnen hin- und herwechseln und sich so aussuchen, auf welcher Sie Ihre Programme und sonstigen Fenster öffnen – dies ermöglicht oft einen schnelleren Wechsel zwischen verschiedenen Arbeitsbereichen als das Verschieben oder vorübergehende Minimieren diverser Fenster. Desktopsymbole liegen dagegen auf allen Desktops an derselben Stelle.

Rechts neben den Panel-Symbolen finden Sie für jedes geöffnete Programm oder Fenster eine Schaltfläche; durch Anklicken dieser Schaltflächen können Sie zwischen den einzelnen Fenstern umschalten oder vorübergehend verkleinerte Fenster wiederherstellen. Genau wie unter Windows lässt sich aber auch in KDE

schneller zwischen den einzelnen geöffneten Programmen beziehungsweise Fenstern hin- und herwechseln, indem Sie die **[Alt]**-Taste gedrückt halten und mehrfach hintereinander **[F3]** drücken. Dies betrifft interessanterweise nur den jeweils aktiven der vier Desktops.

Der Bereich ganz rechts ist für Datum und Uhrzeit sowie für einige Schnellzugriffssymbole reserviert, etwa Lautstärkeregelung, openSUSE-Online-Updates oder Netzwerkverbindungen.



Abbildung 3.3 Der KDE-4-Desktop in openSUSE 11.2

Der erste Arbeitsschritt beim Lernen einer neuen Programmiersprache besteht traditionell darin, ein Programm zu schreiben, das die Worte »Hallo Welt« ausgibt. Etwas Ähnliches lässt sich auch als erste Übung für den Einsatz einer Desktop-Umgebung durchführen. Im Einzelnen sind folgende Aufgaben geplant:

1. Mit einem Texteditor wird eine Skriptdatei erstellt, die »Hallo Welt« ausgibt.
2. Im Dateimanager wird das Skript zur ausführbaren Datei gemacht.
3. In einem Terminalfenster wird das Skript gestartet.

Dieser kleine Testparcours wird hier für KDE und im nächsten Unterabschnitt noch einmal für GNOME beschrieben. Gehen Sie unter KDE wie folgt vor:

1. Der Editor der Wahl heißt *Kate*; es handelt sich um einen besonders leistungsfähigen Texteditor, der zum Lieferumfang von KDE gehört. Starten Sie ihn über **K-MENÜ • DIENSTPROGRAMME • EDITOR • ERWEITERTER TEXTEDITOR KATE**.

Eventuell müssen Sie in einem Vorabdialog die Schaltfläche **SITZUNG ÖFFNEN** betätigen; danach wird das eigentliche Kate-Hauptfenster angezeigt (siehe Abbildung 3.4).

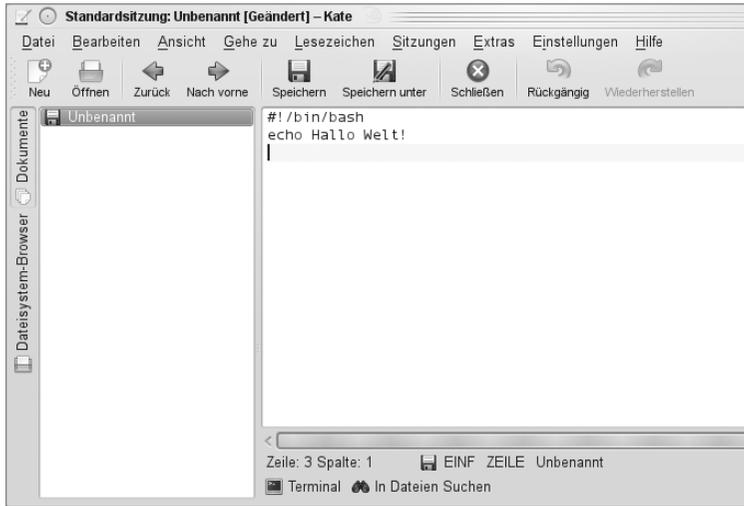


Abbildung 3.4 Das kleine Shell-Skript im KDE-Editor Kate

Geben Sie die folgenden beiden Codezeilen ein:

```
#!/bin/bash
echo Hallo Welt!
```

Die erste Zeile wird *Shebang* genannt – eine Abkürzung für Sharp (#) und Bang (!). Sie klärt, welcher Interpreter das Skript ausführen soll – in diesem Fall die Linux-Standard-Shell Bash. Die zweite Zeile verwendet das Kommando `echo`, das den gesamten Text bis zum Zeilenende auf der Konsole ausgibt, in der das Skript ausgeführt wird.

Wählen Sie **DATEI • SPEICHERN UNTER**. In der oberen Zeile des Dialogs **DATEI SPEICHERN** wird das Verzeichnis angezeigt, in dem Sie sich gerade befinden – bei einem frisch installierten openSUSE-System wahrscheinlich *Dokumente* unter Ihrem Home-Verzeichnis (zum Beispiel */home/sascha/Dokumente*). Bewegen Sie sich mithilfe der Pfeil-hoch-Schaltfläche ganz links oben ein Verzeichnis höher, also in das Home-Verzeichnis selbst. Geben Sie unter **ADRESSE** den Dateinamen *hallo.sh* ein, und klicken Sie auf **SPEICHERN** oder drücken Sie .

Wenn Sie korrekt gespeichert haben, erscheint der Quelltext nun mit sogenanntem Syntax-Highlighting – aufgrund der Dateiendung *.sh* werden die

typischen Elemente von Shell-Skripten durch spezielle Farben oder Formatierungen hervorgehoben.

Wenn Sie möchten, können Sie Kate nun beenden (wählen Sie DATEI • BEEN- DEN, oder klicken Sie auf die Kreuzchen-Schaltfläche ganz rechts oben in der Titelleiste des Fensters). In der Praxis ist es allerdings ratsamer, den Editor beim Programmieren offen zu halten und das Programm erst gründlich zu testen, weil es vorkommen kann, dass noch Fehler zu beheben sind.

2. Als Nächstes soll das Programm mithilfe des Dateimanagers – bei KDE heißt er wie erwähnt Dolphin – ausführbar gesetzt werden. UNIX-Systeme wie Linux unterscheiden für jede Datei und jedes Verzeichnis die drei Rechte Lesen, Schreiben (dazu gehören auch Verschieben, Umbenennen oder Löschen) und Ausführen; jedes dieser Rechte wird für den Besitzer der Datei, die Gruppe der Datei und alle anderen User gespeichert. Im vorliegenden Fall geht es darum, das Ausführen-Bit für alle drei Benutzerarten zu setzen.



Abbildung 3.5 Das Home-Verzeichnis mit dem Shell-Skript im KDE-Dateimanager Dolphin

Öffnen Sie zu diesem Zweck Ihr Home-Verzeichnis im Dolphin; dies geht am schnellsten über die Haus-Schaltfläche im KDE-Panel. Sie erhalten eine grafische Inhaltsübersicht des Verzeichnisses wie in Abbildung 3.5. Klicken Sie *hallo.sh* mit der rechten Maustaste an, und wählen Sie EIGENSCHAFTEN aus dem Kontextmenü. Nach einem Wechsel auf die zweite Registerkarte BERECHTIGUNGEN wird das Dialogfeld aus Abbildung 3.6 angezeigt.



Abbildung 3.6 Die Berechtigungen der Datei »hallo.sh« im Dolphin

Kreuzen Sie das Kontrollkästchen **AUSFÜHRBAR** an. Sie können auch die Schaltfläche **ERWEITERTE BERECHTIGUNGEN** anklicken, um sich die im nächsten Kapitel besprochene tatsächliche Struktur der Dateirechte anzusehen. Klicken Sie abschließend auf **OK**.

3. Nun soll das Skript ausgeführt werden. Da seine Ausgabe nur auf der Konsole stattfindet, ist es zwecklos, es innerhalb der grafischen Oberfläche auszuführen; Sie brauchen ein Terminalfenster dafür. Öffnen Sie dazu das KDE-Terminalprogramm **KONSOLE**; Sie finden es unter **K-MENÜ • SYSTEM • TERMINALS • KONSOLE**. Wenn Sie öfter mit der Konsole arbeiten, lohnt es sich, ein Panel-Symbol dafür anzulegen, indem Sie das Programm-Icon aus dem **K-Menü** ins Panel ziehen. Theoretisch können Sie auch eines der anderen Programme aus dem Menü **TERMINALS** verwenden; sie sind allerdings nicht so komfortabel wie die KDE-Konsole.

Die Eingabeaufforderung in der Konsole müsste schematisch folgendermaßen lauten:

```
Username@Rechner :~>
```

Die Tilde (~) steht für das Home-Verzeichnis des aktuellen Benutzers; wenn hinter dem Doppelpunkt etwas anderes angezeigt werden sollte, müssen Sie zunächst in dieses Verzeichnis wechseln, da Sie die Datei dort gespeichert

haben. Geben Sie dazu einfach Folgendes ein (das Fettgedruckte ist Ihre Eingabe):

```
Username@Rechner:/falsches/Verzeichnis> cd ~ 
Username@Rechner:~>
```

Nun können Sie das Skript ausführen, da Sie sich im richtigen Verzeichnis befinden. Geben Sie diese Anweisung ein:

```
Username@Rechner:~> ./hallo.sh 
Hallo Welt!
Username@Rechner:~>
```

Wie Sie sehen, wird das Skript sofort ausgeführt – die Ausgabe »Hallo Welt!« findet statt. Darunter erscheint sofort die nächste Eingabeaufforderung, denn die Ausführung ist beendet. Die Zeichen `./` vor dem Namen des Skripts stehen für das aktuelle Verzeichnis. Sie sind standardmäßig notwendig, weil das jeweilige Arbeitsverzeichnis unter Linux nicht automatisch zum Suchpfad für ausführbare Dateien gehört (wenn Sie Windows kennen, sind Sie dies möglicherweise gewöhnt). In Abbildung 3.7 wird der gesamte Vorgang der Skriptausführung gezeigt.

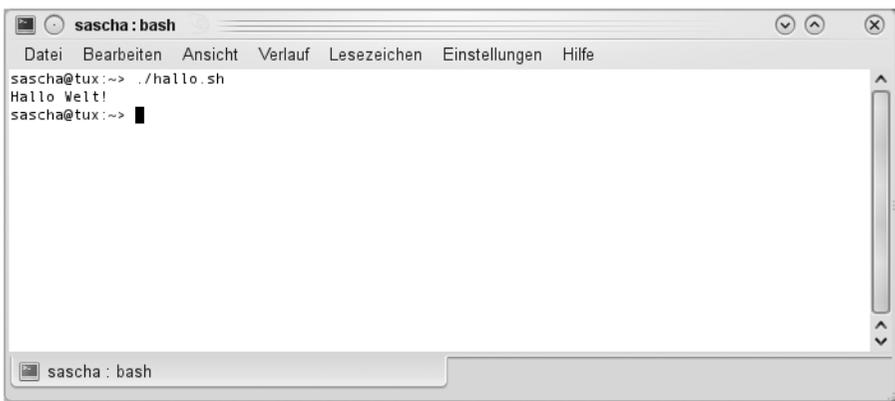


Abbildung 3.7 Ausführung des Shell-Skripts in der KDE-Konsole

Wenn Sie möchten, können Sie das Shell-Skript nun noch löschen. Klicken Sie es im Dolphin mit der rechten Maustaste an, und wählen Sie **IN DEN MÜLLEIMER WERFEN** aus dem Kontextmenü, oder drücken Sie einfach `[Entf]`. Der Mülleimer wird in KDE 4 standardmäßig nicht auf dem Desktop angezeigt, sondern in der Spalte **ORTE** links im Dolphin-Fenster. Um die darin enthaltenen Objekte endgültig zu entfernen, können Sie das Symbol ebenfalls mit rechts anklicken und **MÜLLEIMER LEEREN** wählen.

Alternativ lässt sich die Datei auch per Shell-Befehl löschen. Geben Sie in der Konsole Folgendes ein:

```
Username@Rechner:~> rm hallo.sh ↵
```

Beachten Sie, dass es hier keinen Mülleimer gibt. Die Datei wird sofort gelöscht und lässt sich nur noch mithilfe spezieller Maßnahmen wiederherstellen – und das in der Regel nur, wenn es sofort geschieht.

Damit ist die KDE-Einführung zunächst abgeschlossen. Mehr über diesen benutzerfreundlichen Desktop erfahren Sie in Kapitel 5, »Grafische Oberflächen«.

Wenn auf Ihrem System auch GNOME installiert ist, können Sie nun auch diese Desktop-Umgebung ausprobieren. Schließen Sie dazu alle KDE-Programme (optional; wenn Sie etwas nicht gespeichert hatten, wird automatisch nachgefragt). Wählen Sie **K-MENÜ • ABMELDEN**, und klicken Sie die Schaltfläche **AKTUELLE SITZUNG BEENDEN** an. Auf dem Anmeldebildschirm wählen Sie dann unter **SITZUNGSART** den Eintrag **GNOME** aus, melden sich neu an und fahren im nächsten Unterabschnitt fort.

3.1.3 Erste Schritte mit GNOME

Nachdem Sie sich angemeldet und Ihre GNOME-Sitzung begonnen haben, präsentiert sich der GNOME-Desktop etwa wie in Abbildung 3.8. Die verschiedenen Desktop-Symbole, *Starter* genannt, werden per Doppelklick aktiviert. Am unteren Bildschirmrand befindet sich eine Multifunktionsleiste, die wie bei KDE als *Panel* bezeichnet wird. Am linken Rand befindet sich der Button **RECHNER**. Wenn Sie ihn anklicken, öffnet sich eine Menüstruktur mit drei Hauptschaltflächen:

- ▶ **ANWENDUNGEN**
Enthält die beiden Bereiche **BEVORZUGTE ANWENDUNGEN** und **ZULETZT BENUTZTE ANWENDUNGEN**. Mithilfe der Schaltfläche **WEITERE ANWENDUNGEN** können Sie den Anwendungs-Browser starten, der eine thematisch sortierte Liste aller installierten Anwendungen enthält. Sowohl im Rechner-Menü selbst als auch im Anwendungs-Browser werden Programme per Einzelklick gestartet.
- ▶ **DOKUMENTE**
Bietet schnellen Zugriff auf die zuletzt geöffneten und bearbeiteten Dateien.
- ▶ **ORTE**
Öffnet verschiedene wichtige Verzeichnisse im GNOME-Dateimanager **Nautilus** – **PERSÖNLICHER ORDNER** geht beispielsweise vom eigenen Home-Verzeichnis aus, genau wie der gleichnamige Starter auf dem Desktop. Zusätzlich gibt es einige Einträge für Netzwerkressourcen sowie für die Suche nach Dateien.

Am rechten Rand des Menüs befindet sich der Bereich SYSTEM. Er bietet Zugriff auf diverse wichtige Systemfunktionen wie das KONTROLLZENTRUM (GNOME-Einstellungen), YAST (Systemkonfiguration), HILFE UND ABMELDEN beziehungsweise AUSSCHALTEN (Neustart oder Herunterfahren).

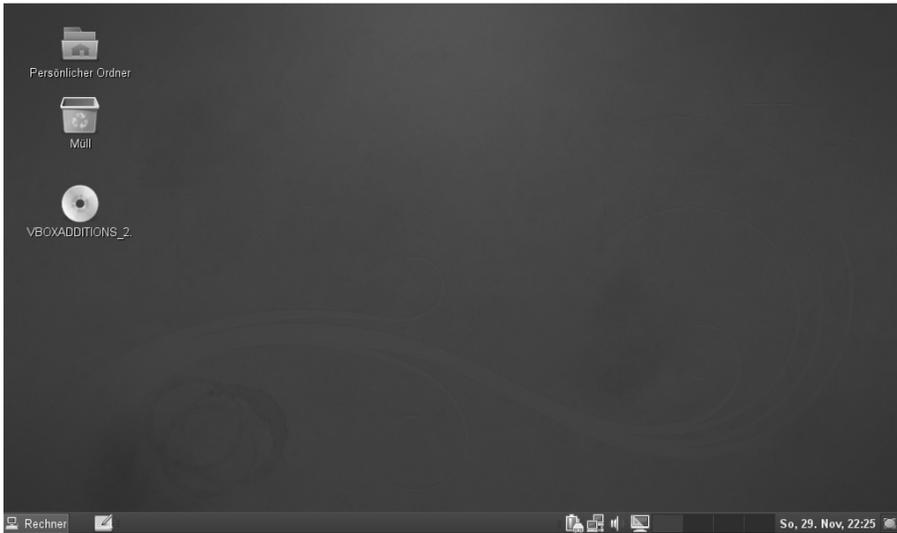


Abbildung 3.8 Der GNOME-Desktop in openSUSE 11.2

Rechts unten finden Sie Datum und Uhrzeit sowie kleine Icons für den schnellen Zugriff auf die Lautstärkeregelung sowie auf den Desktop (Verkleinern aller Fenster und Anzeigen der Desktop-Symbole).

Auch GNOME legt für jedes geöffnete Programm oder Fenster eine Schaltfläche im Panel an. Sie können jede von ihnen mit der Maus anklicken, um das jeweilige Fenster in den Vordergrund zu stellen, oder auch mithilfe der von Windows und KDE bekannten Tastenkombination **Alt** + **Tab** zwischen allen aktiven Anwendungen wechseln. Die Schaltfläche ganz rechts neben der Uhrzeit dient dazu, alle Fenster gleichzeitig auf das Panel zu reduzieren, um den Desktop anzuzeigen.

Nun aber zu der kleinen »Hallo Welt«-Übung, GNOME-Style:

1. Der GNOME-Standard-Texteditor heißt *gedit*. Starten Sie ihn über den Anwendungs-Browser (Panel-Menü ANWENDUNGEN • WEITERE ANWENDUNGEN); die korrekte Auswahl ist DIENSTPROGRAMME • GEDIT. Die Ausstattung dieses Editors ist recht spartanisch; wenn bei Ihnen auch KDE installiert ist, steht Ihnen im selben Menü wahrscheinlich der weiter oben beschriebene KDE-Editor

Kate zur Verfügung – die meisten für KDE geschriebenen und mit diesem gelieferten Anwendungen laufen auch unter GNOME und umgekehrt.

Geben Sie in den gewählten Editor folgenden Quelltext ein:

```
#!/bin/bash
echo Hallo Welt!
```

Eine Erläuterung dieser beiden Skriptzeilen finden Sie weiter oben im Unterabschnitt »Erste Schritte mit KDE«. Abbildung 3.9 zeigt den GNOME-Editor gedit mit dem Skript.

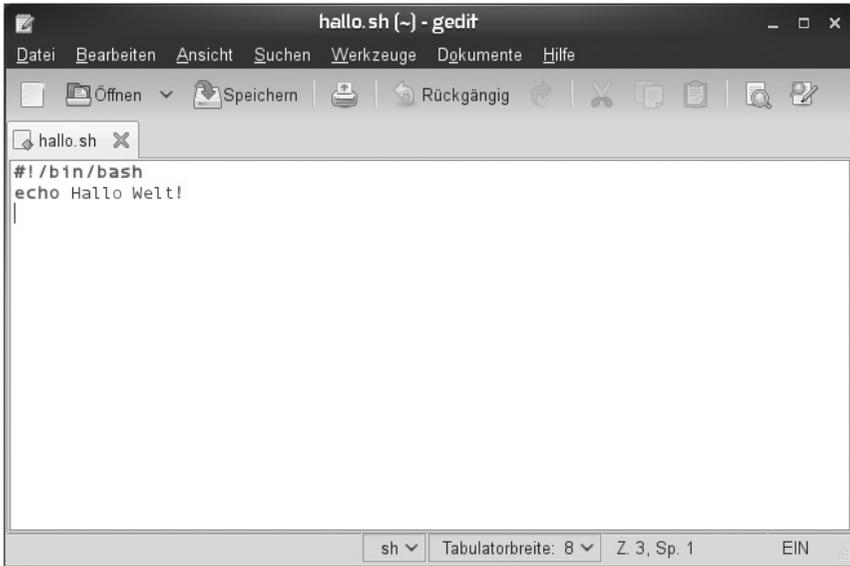


Abbildung 3.9 Das Shell-Skript »hallo.sh« im GNOME-Editor gedit

Wählen Sie **DATEI • SPEICHERN UNTER**; stellen Sie sicher, dass Sie sich in Ihrem Home-Verzeichnis befinden, und geben Sie den Dateinamen *hallo.sh* ein. gedit zeigt das Skript daraufhin mit farbiger Syntaxhervorhebung an (Kate, wie bereits erwähnt, ebenfalls).

- Öffnen Sie Ihr Home-Verzeichnis im GNOME-Dateimanager Nautilus, indem Sie auf das Desktop-Symbol **PERSÖNLICHER ORDNER** doppelklicken oder den gleichnamigen Eintrag aus dem Panel-Menü **ORTE** wählen. Es öffnet sich eine Verzeichnisanzeige wie in Abbildung 3.10.

Nun geht es darum, das Ausführen-Recht des Skripts zu setzen. Klicken Sie dafür das Dateisymbol **HALLO.SH** mit der rechten Maustaste an, und wählen Sie **EIGENSCHAFTEN**. Die Berechtigungen werden auf der Registerkarte **ZUGRIFFS-**

RECHTE eingestellt, die in Abbildung 3.11 zu sehen ist. Kreuzen Sie das Kontrollkästchen DATEI ALS PROGRAMM AUSFÜHREN an und klicken Sie dann auf SCHLIESSEN.



Abbildung 3.10 Das Home-Verzeichnis mit dem Shell-Skript im GNOME-Dateimanager Nautilus

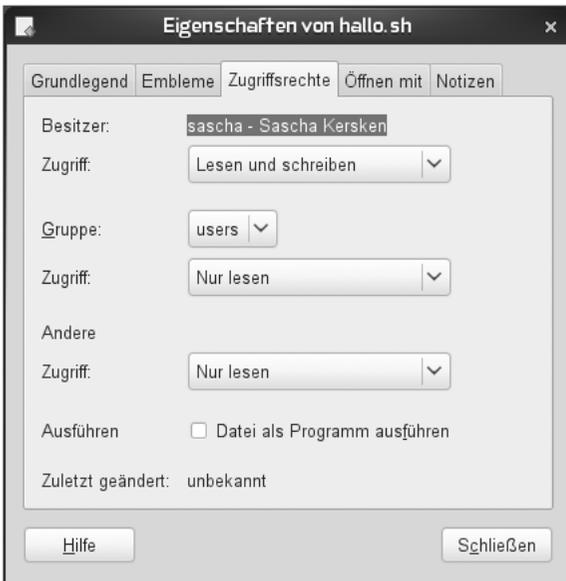


Abbildung 3.11 Das Shell-Skript in GNOME Nautilus ausführbar machen

3. Zum Ausführen des Skripts kommt unter GNOME bevorzugt das zugehörige Terminalprogramm *GNOME-Terminal* zum Einsatz. Starten Sie es über den Anwendungsbrowser mithilfe der Option SYSTEM • GNOME TERMINAL oder schneller, indem Sie mit der rechten Maustaste irgendwo auf den leeren Desktop klicken und IN TERMINAL ÖFFNEN aus dem Kontextmenü wählen. Im letzteren Fall ist das Arbeitsverzeichnis jedoch der Desktop und nicht das Home-Verzeichnis.

Vergewissern Sie sich, dass Sie sich in Ihrem Home-Verzeichnis befinden – die Eingabeaufforderung muss dazu folgendem Schema entsprechen:

```
Username@Recher:~>
```

Falls hinter dem Doppelpunkt ein anderes Verzeichnis als `~` angezeigt wird, wechseln Sie durch die Eingabe von `cd ~` dorthin. Geben Sie danach Folgendes (fett gedruckt) ein, um das Skript auszuführen:

```
Username@Rechner:~> ./hallo.sh 
Hallo Welt!
Username@Rechner:~>
```

Abbildung 3.12 zeigt das Terminal nach erfolgter Ausführung des Skripts. Sofort wird der Prompt für die nächste Befehlseingabe angezeigt. Hier können Sie beispielsweise Folgendes eingeben, um das Skript zu löschen:

```
Username@Rechner:~> rm hallo.sh 
```



Abbildung 3.12 Ausführen des Shell-Skripts im GNOME Terminal

Wenn Sie es dagegen mithilfe der grafischen Oberfläche löschen möchten, klicken Sie es im Nautilus-Verzeichnisfenster mit der rechten Maustaste an, und

wählen Sie **IN DEN MÜLL VERSCHIEBEN**. Alternativ können Sie einfach **Entf** drücken. Um die Datei endgültig loszuwerden, klicken Sie das Desktop-Symbol **MÜLL** mit der rechten Maustaste an, und wählen Sie **MÜLL LEEREN**. Standardmäßig erscheint eine Sicherheitsabfrage; wie Sie diese gegebenenfalls deaktivieren können, steht in Kapitel 5.

3.2 Hilfe und Online-Handbücher

Ein angenehmer Aspekt von Linux im Allgemeinen und Ihrem openSUSE-System im Besonderen ist, dass Sie nicht allein gelassen werden, wenn Sie Fragen oder Probleme haben. Schließlich haben Sie ja jetzt dieses Buch. Nein, Scherz beiseite; alle Systeme der UNIX-Tradition sind von jeher mit einem Online-Handbuch ausgestattet, den sogenannten *Manpages* – natürlich nicht, weil UNIX »nur was für Männer« wäre, sondern als Abkürzung für »Manual Pages«. Eine andere Option ist das moderne GNU info. In diesem Abschnitt wird beschrieben, wie Sie mit beiden umgehen.

Zusätzlich steht sowohl unter KDE als auch unter GNOME das Desktop-Icon **ONLINE HILFE** zur Verfügung. Es öffnet einen Webbrowser und darin die URL <http://help.opensuse.org>. Dazu benötigen Sie natürlich einen Internetzugang (siehe Kapitel 13, »Netzwerkgrundlagen«).

3.2.1 Manpages

Jede Manpage liefert Hilfe zu einem einzelnen Befehl. In der Regel geht es um System-Dienstprogramme und andere Konsolenanwendungen – Anwendungen für grafische Benutzeroberflächen besitzen meist ihre eigenen Hilfeformate. Folgerichtig ist `man`, das Programm zur Anzeige der Manpages, ein reines Konsolenprogramm. Öffnen Sie also das zu Ihrer grafischen Oberfläche passende Terminalfenster KDE-Konsole beziehungsweise GNOME Terminal.

Ein weiter oben bereits angesprochenes Konsolenprogramm, das Sie als Beispiel zum Testen verwenden können, ist das Löschkommando `rm`. Geben Sie also einfach folgenden Befehl ein, wie immer gefolgt von **↵**:

```
Username@Rechner:~> man rm
```

Da es mehrere Einträge mit dem Titel `man` gibt, zeigt das Programm in neueren Versionen zunächst die infrage kommenden Manual-Sektionen an; die Sektionen werden weiter unten vorgestellt. Sie können entweder die gewünschte Nummer eingeben (in diesem Fall 1) oder aber kurz warten – im letzteren Fall wird die

Sektion mit der kleinsten Nummer gewählt. Bei älteren man-Versionen geschieht dies automatisch.

Das Programm `man` lässt sich ein wenig Zeit, um die Manpage zu formatieren, und zeigt dann die gewünschte Hilfeseite an. Für die seitenweise Anzeige ist allerdings nicht `man` selbst, sondern ein sogenanntes Pager-Programm zuständig – unter Linux das im nächsten Kapitel explizit vorgestellte Kommando `less`. Daher können Sie sich unter anderem mithilfe der folgenden Tasten in einer Manpage bewegen:

- ▶ `E`, `↵` oder `↓` blättert um eine Zeile nach unten.
- ▶ `Y` oder `↑` wandert eine Zeile nach oben.
- ▶ `F`, `Leertaste` oder `Bild ↓` scrollt eine ganze Fensterhöhe nach unten.
- ▶ `B` oder `Bild ↑` springt eine Fensterhöhe nach oben.
- ▶ `Pos1` bewegt sich zum Anfang des Textes.
- ▶ `Ende` springt zur letzten Zeile.
- ▶ `/` (`↵`+`7`) ermöglicht die Eingabe eines Suchbegriffs, gefolgt von `↵`. Alle Vorkommen des Begriffs werden markiert, und die Zeile mit der ersten Fundstelle wird am oberen Fensterrand angezeigt.
- ▶ `?` (`↵`+`B`) leitet ebenfalls die Suche ein, allerdings ab der aktuellen Position rückwärts.
- ▶ `N` springt zum nächsten Vorkommen des aktuellen Suchbegriffs in derselben Richtung.
- ▶ `↵`+`N` sucht ebenfalls weiter, kehrt dabei aber die Richtung um.
- ▶ `H` zeigt eine umfangreiche Hilfeseite mit diesen und einigen weiteren Kommandos an.
- ▶ `Q` verlässt das Programm `less` und damit in diesem Fall auch die Manpage.

Die Manpages werden in verschiedene nummerierte Kategorien unterteilt. Moderne Linux-Systeme verwenden im Prinzip dieselbe Nummerierung wie das klassische UNIX, allerdings wurden einige neue Kategorien hinzugefügt. Im Einzelnen sind es folgende:

- ▶ **0:** C-Header-Dateien, die für das Einbinden von Bibliotheken in eigene Programme verwendet werden
- ▶ **1:** Systemprogramme, das heißt normale Konsolenbefehle – das Beispiel `rm` befindet sich in dieser Sektion.
- ▶ **2:** Systemaufrufe, also Dienste des Kernels

- ▶ **3:** Funktionen aus der C-Standardbibliothek und weiteren Systembibliotheken
- ▶ **4:** Beschreibung der Spezialdateien, vor allem der Gerätedateien unter */dev*
- ▶ **5:** Formate von Konfigurationsdateien und andere Systemkonventionen
- ▶ **6:** Spiele
- ▶ **7:** Makropakete (aus mehreren Einzelkomponenten bestehende Programme)
- ▶ **8:** Kommandos zur Systemadministration (meist für *root*)
- ▶ **9:** Kernelroutinen

Daneben gibt es noch die vier nicht-numerischen Sektionen *n* (*new*), *l* (*local*), *p* (*public*) und *o* (*old*), die allerdings veraltet sind und kaum noch verwendet werden. Zudem bringen manche optionalen Programme ihre eigenen Sektionsbezeichnungen mit.

In den meisten Fällen braucht Sie die Sektion beim Aufruf einer Manpage nicht zu interessieren. Das ist nur dann der Fall, wenn es in zwei oder mehr Sektionen gleichnamige Manpages gibt. Angenommen, Sie möchten über das Format der weiter unten beschriebenen Benutzerdatei */etc/passwd* Bescheid wissen. Die Seite heißt *passwd*, aber wenn Sie `man passwd` eingeben, wird standardmäßig die Beschreibung des gleichnamigen Systembefehls aus der Manual-Sektion 1 angezeigt – wenn Sie keine Sektionsnummer angeben, wird der Treffer mit der kleinsten ausgegeben. Die gewünschte Seite über */etc/passwd* befindet sich in Sektion 5; um sie aufzurufen, müssen Sie Folgendes eingeben:

```
Username@Rechner:~> man 5 passwd
```

Um herauszufinden, in welchen Sektionen sich Manpages mit dem gewünschten Titel befinden, können Sie das Kommando `whatis Befehlsname` verwenden. Als zusätzlichen Service zeigt es eine zwei bis drei Wörter lange Zusammenfassung des jeweiligen Kommandos an. Beispiel:

```
Username@Rechner:~> whatis passwd
passwd (1ssl)      - compute password hashes
passwd (1)         - change user password
passwd (5)         - password file
```

In Klammern wird jeweils die Sektion angezeigt. Statt `whatis` können Sie übrigens auch `man selbst` mit der Option `-f` aufrufen, also Folgendes schreiben:

```
Username@Rechner:~> man -f passwd
```

Wenn Sie den genauen Befehlsnamen nicht wissen und lieber nach einem Teil davon oder nach einem Begriff aus der Kurzbeschreibung suchen möchten, kön-

nen Sie den Befehl `apropos` oder das Synonym `man -k` verwenden. Hier eine stark gekürzte Beispielausgabe:

```
Username@Rechner:~> apropos passwd
htpasswd2 (1)      - Manage user files for basic authentication
ldappasswd (1)    - change the password of an LDAP entry
passwd (1)        - change user password
gpasswd (1)       - change group password
...
```

Da die Liste oft länger als eine Terminalfensterhöhe ist, können Sie die Ausgabe des Kommandos durch eine sogenannte Pipe (das Konzept wird im nächsten Kapitel genau erläutert) an den Pager `less` weiterleiten, um sie seitenweise durchzublättern. Hier die entsprechende Eingabe in der Schreibweise `man -k`:

```
Username@Rechner:~> man -k passwd | less
```

Durchblättern Sie die Ausgabe mithilfe der weiter oben für `man` selbst beschriebenen `less`-Tasten, und drücken Sie `Q`, wenn Sie fertig sind.

Wenn Sie noch mehr wissen möchten: Das Programm `man` hat natürlich ebenfalls eine eigene Manpage. Geben Sie `man man` ein, um alle Optionen nachzulesen. Eine detailliertere Liste erhalten Sie außerdem in Kapitel 19, »Referenz der Shell- und Systembefehle«.

3.2.2 GNU info

Aus Gründen der UNIX-Kompatibilität besitzen auch die Tools des GNU-Projekts bis heute Manpages. Zusätzlich wurde allerdings ein eigenes Hilfeseiten-Format namens *GNU info* eingeführt. Diese Seiten sind meist aktueller und ausführlicher als die Manpages; zudem verfügt das zugehörige Leseprogramm `info` über intelligentere Funktionen als `man`. In Abbildung 3.13 sehen Sie `info` beim Anzeigen der Hilfe zum Löschbefehl `rm`.

Ein erster Unterschied besteht darin, dass Sie `info` wahlweise mit Stichwort (`info` Befehl) oder auch ohne aufrufen können. Letzteres versetzt Sie in den sogenannten *Directory Node* (die einzelnen Hilfeseiten werden bei `info` als Nodes, also Knoten, bezeichnet). Diese Seite ist das Hauptmenü mit den wichtigsten Themen. Die Bedienung des Programms ist bis zu einem gewissen Grad kompatibel zu dem in Kapitel 10 vorgestellten Editor GNU Emacs. Folgende Tastenkombinationen dienen der Navigation auf dem Bildschirm:

- ▶ `Strg+F` (forward) bewegt den Cursor ein Zeichen nach rechts.
- ▶ `Strg+B` (backward) wandert um ein Zeichen nach links.

- ▶ `Strg`+`N` (next) springt in die nächste Zeile.
- ▶ `Strg`+`P` (previous) geht zurück zur vorherigen Zeile.
- ▶ `Strg`+`A` setzt den Cursor auf den Anfang der aktuellen Zeile.
- ▶ `Strg`+`E` geht zum Zeilenende.
- ▶ `Leertaste` blättert einen ganzen Bildschirm weiter.
- ▶ `Entf` oder `Backspace` springt einen Bildschirm zurück.

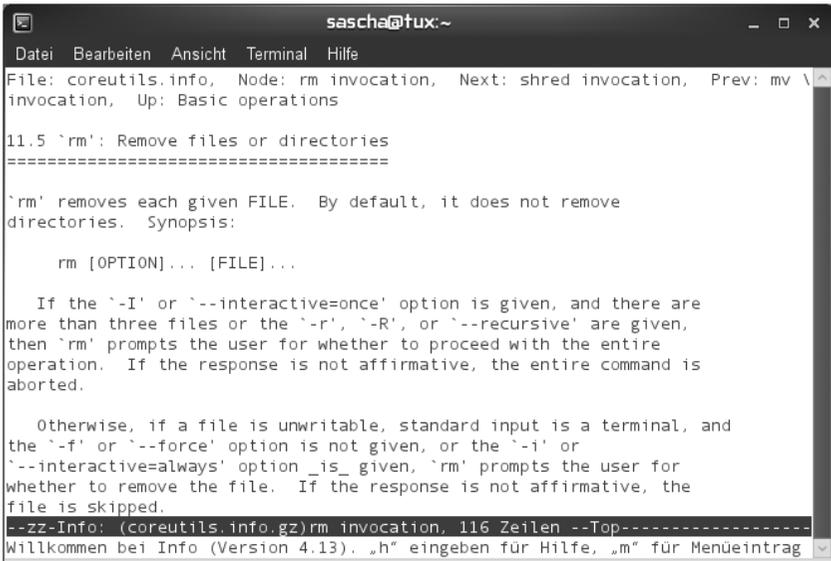


Abbildung 3.13 GNU info mit der Hilfeseite zum Löschbefehl `rm`

Neben diesen rein geometrischen Bewegungsoptionen gibt es einige Befehle zur thematischen Navigation, das heißt zum Blättern durch die Seiten und für die direkte Suche nach Themen:

- ▶ `M` (Menu) setzt den Cursor in die unterste Zeile des Terminals, den sogenannten Minibuffer. Hier können Sie den Namen eines Themas eingeben und mit `↵` abschließen, um Hilfe dazu aufzurufen. Versuchen Sie es beispielsweise mit `rm` für den erwähnten Löschbefehl. Wenn sich der Cursor beim Drücken der Taste `M` über einem Wort befindet, zu dem eine `info`-Seite existiert, wird diese vorgeschlagen. Genauso funktionieren die »Links« im Directory Node: Bewegen Sie den Cursor auf den Anfang eines dieser Texte, beispielsweise `Bash: (bash)` unter der Überschrift `Utilities`, und drücken Sie `M`. Im Minibuffer erscheint nun folgende Meldung:

Menu item (Bash):

Wenn Sie einfach `↵` drücken, wird der Vorschlag angenommen; falls Sie nicht damit einverstanden sind, können Sie einfach ein anderes Thema eingetippen.

- ▶ `N` (next) blättert eine Seite im aktuellen Oberthema weiter. Wenn dies möglich ist, erscheint am oberen Bildschirmrand der Text `Next: Themename`.
- ▶ `P` (previous) dient entsprechend dazu, eine Seite zurückzublättern. Diese Option steht zur Verfügung, wenn oben `Prev: Themename` steht.
- ▶ `U` (up) springt eine Strukturebene nach oben; die oberste Ebene ist stets der Directory Node. Wohin es ansonsten geht, können Sie dem Eintrag `Up: Themename` entnehmen.
- ▶ `L` (last) blättert zu der Seite zurück, die vor der aktuellen angezeigt wurde.
- ▶ `H` (help) öffnet ein grundlegendes Tutorial zum `info`-Programm. Drücken Sie danach `L`, um zur vorherigen Seite zurückzukehren.
- ▶ `?` zeigt eine tabellarische Hilfe zu `info` an, die die wichtigsten Tasten und erweiterten Befehle erläutert. Auch diese Seite können Sie mit `L` wieder verlassen.
- ▶ `Q` beendet `info`. Wenn Sie möchten, können Sie auch den offiziellen Befehl zum Beenden von Emacs verwenden, um sich daran zu gewöhnen: Drücken Sie nacheinander `Strg+X`, `Strg+C`.

3.3 Grundkonfiguration

Nachdem Sie nun einen ersten Überblick über die Funktionsweise des Systems erhalten haben, sollten Sie erfahren, wie sich die wichtigsten Einstellungen vornehmen lassen. Das Tool *YaST* (Yet Another Setup Tool), das bei openSUSE hauptsächlich dafür zuständig ist, haben Sie im vorigen Kapitel bereits für die Systeminstallation verwendet. Hier wird zunächst seine allgemeine Funktionsweise als Konfigurationswerkzeug beschrieben; anschließend werden exemplarisch einige konkrete Einstellungen wie die Benutzerverwaltung und der automatische Start von Programmen vorgestellt. Zahlreiche weitere Einzelkomponenten von YaST lernen Sie in späteren Kapiteln dieses Buches kennen.

3.3.1 YaST starten und einsetzen

Sie können YaST in zwei gleichberechtigten Modi ausführen: entweder fensterbasiert in der grafischen Oberfläche oder als Vollbild-Textanwendung in einem Terminal. Die Textvariante ist besonders nützlich, wenn Sie auf einem entfernten

Rechner arbeiten, etwa auf einem Webserver-Rechner, den Sie bei einem Hoster gemietet haben.

In beiden Fällen müssen Sie `root` sein, um YaST zu verwenden. Die grafische Version finden Sie in KDE unter **K-MENÜ • SYSTEM • YAST (KONTROLLZENTRUM)**. Bei GNOME liegt sie dagegen im Panel-Menü **SYSTEM • YAST**. In beiden Umgebungen werden Sie nach dem Aufruf aufgefordert, Ihr (bei der Installation im vorigen Kapitel festgelegtes) `root`-Passwort einzugeben; Abbildung 3.14 zeigt den entsprechenden Dialog in GNOME.



Abbildung 3.14 GNOME fordert Sie zur Eingabe des `root`-Passworts auf.

Wenn die Eingabe korrekt war, wird YaST geladen; die grafische Version sieht so aus, wie in Abbildung 3.15 gezeigt. Die linke Spalte enthält eine Reihe von Kategorien; wenn Sie eine von ihnen anklicken, werden rechts diverse Symbole für Einstellungen zum jeweiligen Thema angezeigt. Sobald Sie eines von ihnen anklicken, öffnet sich ein neues Fenster mit einem Dialog für die entsprechende Option.

Wenn Sie YaST im Textmodus betreiben möchten, benötigen Sie zunächst ein Terminalfenster wie die KDE-Konsole oder das GNOME Terminal. Danach müssen Sie `root` werden, um YaST starten zu dürfen. Dies geschieht durch die Eingabe von `su` und dem `root`-Passwort (Näheres dazu im nächsten Kapitel):

```
Username@Rechner:~> su
Passwort:
Rechner: /home/Username #
```

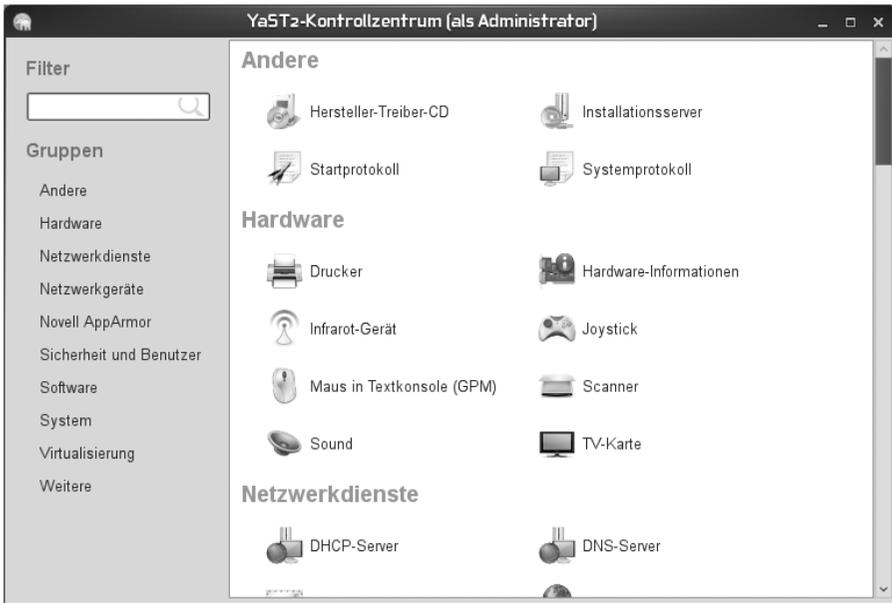


Abbildung 3.15 Die Startansicht von YaST in KDE

Dadurch ändert sich, wie Sie sehen, die Eingabeaufforderung – der Benutzername `root` wird nicht angezeigt, und da das Home-Verzeichnis des Normalusers nicht das Home-Verzeichnis von `root` ist, erscheint der konkrete Verzeichnisname statt der Tilde. Nun können Sie das Konfigurationsprogramm durch folgende Eingabe starten:

```
Rechner: /home/Username # yast
```

Die Textmodusvariante von YaST bietet genau dieselben Funktionen wie die grafische Version, wird aber per Tastatur gesteuert: Mit den Pfeiltasten bewegen Sie sich innerhalb der jeweiligen Menüstruktur. Für den Wechsel zwischen verschiedenen Bereichen oder Steuerelementen wird die `↵`-Taste (vorwärts) oder `⏪` (rückwärts) verwendet. Zusätzlich besitzt fast jede Beschriftung einen farblich hervorgehobenen Buchstaben. Wenn Sie die `Alt`-Taste festhalten und die entsprechende Taste drücken, wird der entsprechende Befehl ausgeführt. Verwenden Sie beispielsweise `Alt+V` für den Befehl `VERLASSEN`, um die jeweilige Konfigurationskomponente oder YaST selbst zu beenden. In Abbildung 3.16 sehen Sie die YaST-Startseite in der Textvariante.

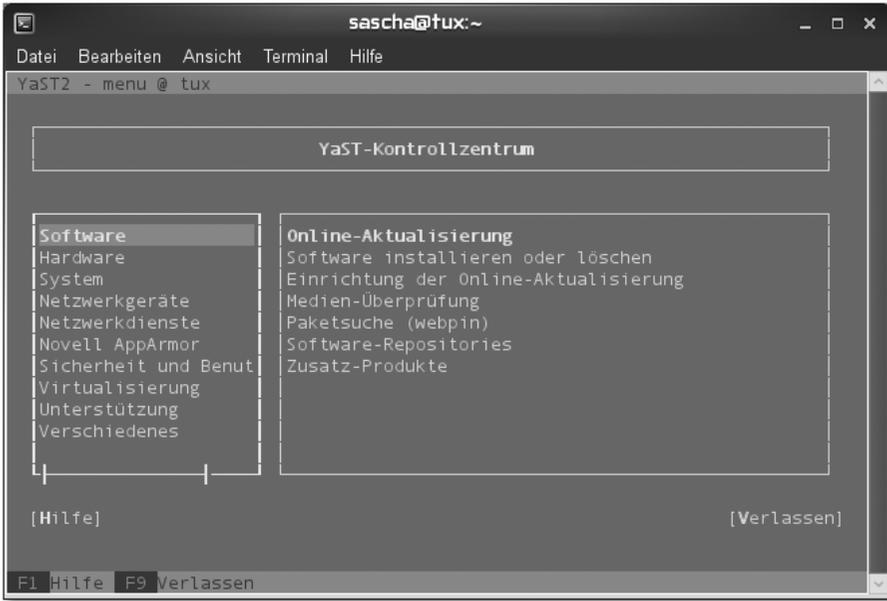


Abbildung 3.16 Die Startseite von YaST im Textmodus (GNOME Terminal)

In Tabelle 3.1 werden die einzelnen YaST-Hauptkategorien vorgestellt. Die meisten ihrer Unterpunkte kommen in späteren Kapiteln an Ort und Stelle zur Sprache; die entsprechende Spalte der Tabelle gibt darüber Aufschluss.

Kategorie	Bedeutung	Kapitel
SOFTWARE	Nachinstallation, Deinstallation oder Aktualisierung von Softwarepaketen	3
HARDWARE	Treiber und Einstellungen für diverse Geräte (außer Netzwerk) konfigurieren	6
SYSTEM	Wichtige Systemeinstellungen wie Partitionierung, Bootkonfiguration, Power-Management oder Datum und Uhrzeit	2, 3, 9 u. a.
NETZWERKGERÄTE	Treiber und Einstellungen für Netzwerkgeräte wie Netzwerkkarten, DSL, Modems oder WLAN-Konfiguration	13
NETZWERKDIENTSTE	Client- und Serverdienste für das Netzwerk konfigurieren, beispielsweise Dateiserver, Webserver oder Mailserver	13–16

Tabelle 3.1 Die verschiedenen Hauptkategorien von YaST

Kategorie	Bedeutung	Kapitel
NOVELL APPARMOR	Eine Sicherheitslösung, die sehr filigrane Einstellungen für die Rechte einzelner Anwendungsprogramme ermöglicht; wurde Anfang 2006 als Open Source freigegeben.	–
SICHERHEIT UND BENUTZER	Benutzerkonten, Sicherheitseinstellungen, Firewall	3, 13, 14
VIRTUALISIERUNG	Konfiguration der Virtualisierungslösung XEN	–
UNTERSTÜTZUNG	Versionshinweise und Support-Registrierung	–
VERSCHIEDENES	Einstellungen, die in keine andere Kategorie passen – zum Beispiel Anzeige diverser Logdateien und Einbinden von Datenträgern mit Drittanbieter-Treibern	–

Tabelle 3.1 Die verschiedenen Hauptkategorien von YaST (Forts.)

In den folgenden Unterabschnitten wird exemplarisch auf zwei der wichtigsten Konfigurationsthemen eingegangen: Benutzerverwaltung und Nachinstallation von Software.

3.3.2 Benutzer- und Gruppenverwaltung mit YaST

Bereits während der Installation haben Sie das Passwort für den Superuser `root` bestimmt und das erste reguläre Benutzerkonto angelegt. Wenn Sie Benutzerkonten bearbeiten oder neue hinzufügen möchten, können Sie dies über den Menüpunkt `BENUTZER- UND GRUPPENVERWALTUNG` in der YaST-Hauptkategorie `SICHERHEIT UND BENUTZER` erledigen. Wenn es schneller gehen soll: Im nächsten Kapitel lernen Sie die zugehörigen Shell-Befehle kennen, mit denen Sie die Benutzerverwaltung manuell durchführen können.

Benutzer- und Gruppenkonten

Die Verwaltung von Benutzern und Gruppen unter Linux sollte nicht ohne ein Mindestmaß an Theoriekenntnissen erfolgen. Deshalb wird hier zunächst ein wenig notwendiges Hintergrundwissen vermittelt. Wenn Sie bereits damit vertraut sind, können Sie im Abschnitt »Benutzer und Gruppen verwalten« weiterlesen.

UNIX-Benutzerkonten bestehen aus Einträgen in der Systemdatei `/etc/passwd`. Jeder solche Eintrag enthält verschiedene durch Doppelpunkt getrennte Informationen:

```
Username:Passwort:UID:GID:Info:Home:Shell
```

Im Einzelnen bedeuten sie Folgendes:

- ▶ Der `Username` ist der Login-Name, unter dem sich der entsprechende Benutzer anmeldet.
- ▶ Als Nächstes folgt das verschlüsselte `Password` oder ein `x`, wenn das Passwort in der Datei `/etc/shadow` steht (siehe unten).
- ▶ Die `UID` ist die numerische User-ID des Benutzers. `root` besitzt die UID 0, Systemkonten (für Systemdienste, die unter speziellen Benutzerkonten laufen) verwenden traditionellerweise UIDs unter 100 und normale Benutzer Werte ab 100.
- ▶ Die `GID` ist entsprechend die Nummer der Gruppe, der er »hauptamtlich« angehört (während jede Datei genau einer Gruppe gehört, kann ein Benutzer optional Mitglied in mehreren sein, was mitunter die Zuteilung von Rechten erleichtert). Auch hier sind Werte unter 100 traditionell dem System vorbehalten.
- ▶ Die `Info` (auch GECOS-Daten genannt) enthält eine Klartextinformation über den Benutzer, meistens seinen vollständigen Namen. Möglich sind aber etwa auch Telefonnummer oder E-Mail-Adresse.
- ▶ `Home` gibt das Home-Verzeichnis dieses Benutzers an (bei regulären Benutzern fast immer `/home/Username`, bei `root` einfach `/root`).
- ▶ `Shell` gibt schließlich an, welche Shell dem Benutzer nach dem (Textmodus-) Login präsentiert wird – die verschiedenen Shells werden im nächsten Kapitel erläutert.

Ein konkreter Eintrag könnte beispielsweise folgendermaßen aussehen:

```
user:x:102:100:Irgendjemand:/home/user:/bin/bash
```

Der Username und das Passwort sind die wichtigsten Informationen. Nur eine korrekt eingegebene Kombination aus beiden ermöglicht die Anmeldung eines Benutzers. Das Passwort steht an dieser Stelle nicht etwa im Klartext (es ließe sich leicht von jedem Benutzer oder sogar von einem externen Angreifer stehlen), sondern verschlüsselt. Das eingesetzte Verschlüsselungsverfahren ist so beschaffen, dass die Einträge nicht wieder entschlüsselt werden können. Das Passwort, das ein Benutzer eingibt, wird vielmehr auf dieselbe Art und Weise verschlüsselt wie die `/etc/passwd`-Einträge. Anschließend wird das Ergebnis mit dem gespeicherten, verschlüsselten Passwort verglichen – eine Übereinstimmung bedeutet, dass das Passwort wohl korrekt sein muss.

Da der Verschlüsselungsweg bekannt ist, könnte ein Angreifer, dem ein Diebstahl der */etc/passwd*-Datei gelingt, einfach nacheinander eine Liste von Wörtern verschlüsseln und mit den gespeicherten Passwörtern vergleichen. Es gibt sogar ein Programm namens `crack`, das diesen Job automatisch durchführt, und die passenden Wortlisten für die verschiedensten Sprachen erhalten Sie leicht aus dem Internet. Die Schlussfolgerung sollte klar sein: Verwenden Sie als Passwort niemals ein Wort, das in einem Wörterbuch vorkommen kann. Eine beliebige Kombination aus Großbuchstaben, Kleinbuchstaben und Ziffern ist dagegen ziemlich sicher. Wie Sie sich ein Passwort ausdenken, das diese Kriterien erfüllt und das Sie sich dennoch leicht merken können, wurde bereits im vorigen Kapitel erläutert.

Dass im obigen Beispieleintrag kein verschlüsseltes Passwort, sondern lediglich ein `x` steht, bedeutet Folgendes: Die eigentlichen Passwörter werden auf diesem System nicht in der von jedem lesbaren Datei */etc/passwd* verwahrt, sondern in */etc/shadow*. Diese Datei ist nur für `root` lesbar; ein entsprechender Eintrag sieht beispielsweise so aus (hier mit willkürlichem Umbruch mitten im verschlüsselten Passwort, damit er in die Zeile passt):

```
user:$2a$05$qxXD9s7DDMpEDjxVdwf/
zOzDeJZH9yDp2954oZvRGRLgHeyJmh4jK:13259:0:99999:7:-1::
```

Die einzelnen, durch Doppelpunkte getrennten Komponenten dieses Eintrags sind:

- ▶ *Benutzername* (wie in */etc/passwd*)
- ▶ *Verschlüsseltes Passwort*
- ▶ *Datum der letzten Passwortänderung*
Wird in Tagen seit *EPOCH* dargestellt, dem idealisierten UNIX-Erfindungsdatum (01.01.1970, 00:00 Uhr UTC).
- ▶ *Mindestalter des Passworts in Tagen*
Kann gesetzt werden, damit Benutzer es nach einer administrativ erzwungenen Änderung nicht sofort wieder auf den alten Wert zurücksetzen.
- ▶ *Höchstaltes des Passworts in Tagen*
Hier ist es aus Sicherheitsgründen empfehlenswert, Passwörter regelmäßig zu ändern; dieser Eintrag zwingt den betreffenden Benutzer dazu.
- ▶ *Warnungsdatum*
Gibt die Anzahl der Tage vor Ablauf des Passwortes an, ab der der Benutzer informiert wird.

▶ *Sperrdatum*

Diese Komponente bezeichnet die Anzahl der Tage nach Ablauf, ab der das Benutzerkonto gesperrt wird.

▶ *Reserviertes Feld*

Diese Komponente ist für künftige oder herstellereinspezifische Erweiterungen reserviert.

Was der jeweilige Benutzer darf oder nicht darf, steht nicht etwa in irgendeiner globalen Liste oder Konfigurationsdatei, sondern ist anhand der Besitzrechte an einzelnen Dateien und Verzeichnissen geregelt: Jede Datei gehört genau einem Benutzer und einer Gruppe, und wie Sie bereits gesehen haben, lassen sich die drei fundamentalen Rechte – Lesen, Schreiben und Ausführen – jeweils für den Besitzer, die Gruppe und alle anderen Benutzer erteilen. Da auf einem UNIX-System »alles eine Datei« ist, neben den regulären Dateien beispielsweise auch Hardwaregeräte oder Netzwerkeinstellungen, lassen sich die Berechtigungen der diversen Benutzer auf diese Weise recht genau einstellen.

Wenn Sie eine noch feiner eingestellte Kontrolle ausüben möchten, können Sie in einem modernen Linux-System zusätzlich sogenannte *Zugriffskontrolllisten* (Access Control Lists oder ACLs) verwenden. Diese sind eine Eigenschaft der Dateisysteme und werden daher in Kapitel 9, »Datenträger und Dateisysteme«, angesprochen.

Gruppenkonten werden in der Datei */etc/group* verwaltet. Ein typischer Eintrag sieht folgendermaßen aus:

```
users:x:100:
```

Von links nach rechts bedeuten die fünf Felder Folgendes:

▶ *Gruppenname*

▶ *Verschlüsseltes Gruppenpasswort*

x für */etc/shadow* oder leer, falls es keines gibt.

▶ *GID*

Bezeichnet die numerische Kennzeichnung der Gruppe.

▶ *Mitgliederliste*

Dieses Feld ist eine durch Kommas getrennte Liste von Benutzern, die Mitglieder dieser Gruppe sind. Sie ist nicht erforderlich, wenn es sich um die Hauptgruppe des jeweiligen Users handelt (diese steht, wie bereits erwähnt, in */etc/user*), sondern nur für zusätzliche Mitgliedschaften.

Benutzer und Gruppen verwalten

Aktivieren Sie den Eintrag **BENUTZER- UND GRUPPENMANAGEMENT** in der YaST-Gruppe **SICHERHEIT UND BENUTZER**, um den Dialog **VERWALTUNG VON BENUTZERN UND GRUPPEN** aufzurufen. Die Karteireiter am oberen Rand dienen der Auswahl, ob **BENUTZER**, **GRUPPEN**, **STANDARDEINSTELLUNGEN FÜR NEUE BENUTZER** oder **AUTHENTIFIZIERUNGSEINSTELLUNGEN** bearbeitet werden sollen. In Abbildung 3.17 ist die grafische Variante des Dialogs in der Einstellung **BENUTZER** zu sehen.

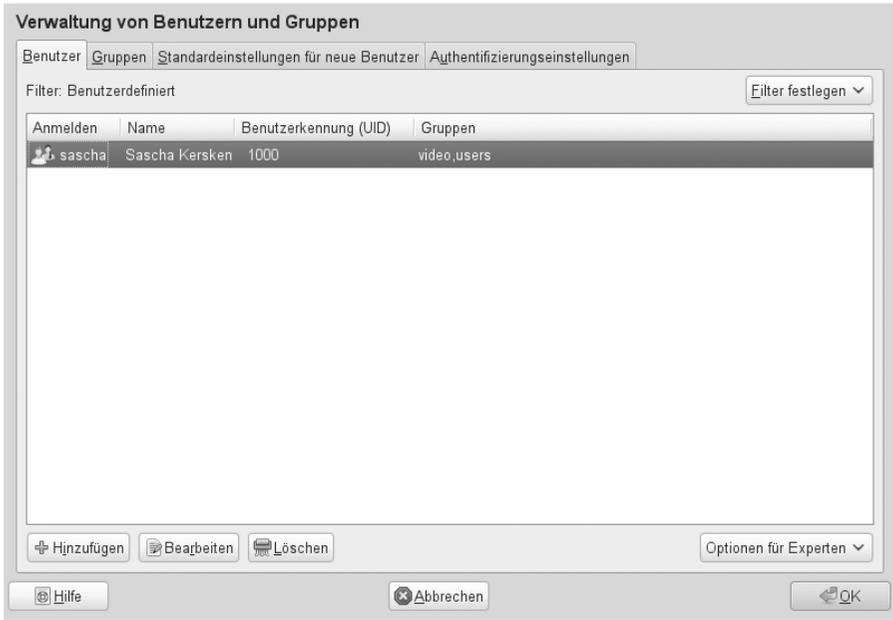


Abbildung 3.17 Die YaST-Benutzerverwaltung in KDE

Welche Benutzerkonten überhaupt in der Liste angezeigt werden, regelt das Pull-down-Menü **FILTER FESTLEGEN**. Prinzipiell stehen die drei Typen **LOKALE BENUTZER** (UID ab 100), **SYSTEMBENUTZER** (UID unter 100) und **NIS-BENUTZER** (Anmelde-daten aus dem Verzeichnisdienst NIS; siehe Kapitel 14) zur Verfügung; **BENUTZERDEFINIERTER FILTEREINSTELLUNG** ermöglicht Ihnen bei Bedarf auch, mehrere gleichzeitig auszuwählen.

Wenn Sie die Einstellungen eines vorhandenen Benutzerkontos ändern möchten, klicken Sie die entsprechende Zeile an, und betätigen Sie die Schaltfläche **BEARBEITEN**. Es öffnet sich ein weiterer Dialog mit drei oder vier Registerkarten:

- Im Bereich BENUTZER (in Abbildung 3.18 zu sehen) werden die grundlegenden Einstellungen für den jeweiligen Benutzer vorgenommen. VOLLSTÄNDIGER NAME DES BENUTZERS ist nur eine Verwaltungsinformation für Sie selbst; die eigentliche Anmeldeinformation ist der BENUTZERNAME. Wenn Sie das Passwort des Benutzers ändern möchten, müssen Sie es zweimal identisch eingeben. BENUTZERNAME DEAKTIVIEREN ermöglicht es Ihnen, das Konto vorübergehend zu sperren, sodass dieser Benutzer sich nicht anmelden kann.

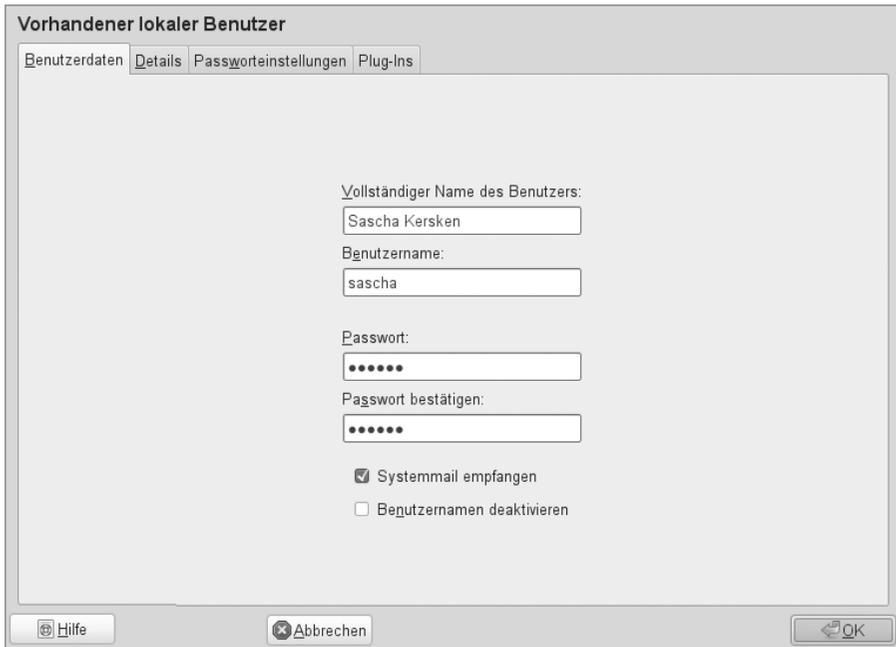


Abbildung 3.18 Die Grundeinstellungen für ein Benutzerkonto

- Die DETAILS regeln diverse genauere Einstellungen für das Benutzerkonto: Unter BENUTZERKENNUNG (UID) kann die numerische User-ID gelesen und (falls die neue frei ist) geändert werden. Bei neuen Benutzern wird automatisch die erste freie ID ab 100 vergeben. Das HOME-VERZEICHNIS können Sie entweder per Eingabe oder mithilfe der Schaltfläche DURCHSUCHEN ändern, wobei in der Regel kein Anlass dazu besteht. Lokale Benutzer haben meist das Home-Verzeichnis `/home/Benutzername`, Systemkonten dagegen mitunter `/var/lib/Dienstname`. Unter ZUSÄTZLICHE BENUTZERINFORMATIONEN können Sie beliebigen Text eingeben, der in das Info-Feld in `/etc/passwd` eingetragen wird. Die Anmelde-Shell legt fest, mit welcher Shell der Benutzer nach einem Konsolen-Login arbeitet. Bei normalen Usern wird meist `/bin/bash` verwendet. Systemkonten sind dagegen nicht für eine persönliche Anmeldung

gedacht, sodass sie keine Login-Shell brauchen. Um dies zu verdeutlichen, wird oft */bin/false* angegeben – ein Programm, dessen einzige Aufgabe darin besteht, mit einem Fehler abzubrechen. Die STANDARDGRUPPE ist die in */etc/passwd* eingetragene Gruppe, der der Benutzer primär angehört. Unter GRUPPEN können Sie zusätzliche Gruppen ankreuzen, wodurch der Benutzer in die Userlisten der entsprechenden Gruppen in */etc/group* aufgenommen wird.

- ▶ Die PASSWORTEINSTELLUNGEN entsprechen den weiter oben erläuterten Zusatzfeldern von */etc/shadow*; wenn Sie ein besonders sicheres System haben möchten, können Sie hier strikte Werte eingeben. Hier ein recht restriktives Beispiel:
 - ▶ TAGE VOR ABLAUF DES PASSWORTS WARNEN: 3
 - ▶ TAGE NACH ABLAUF DES PASSWORTS ANMELDEVORGANG MÖGLICH: -1 (das heißt deaktiviert)
 - ▶ MAXIMALE ANZAHL VON TAGEN FÜR DAS GLEICHE PASSWORT: 14 (noch weniger wäre eine Zumutung für Ihre Benutzer)
 - ▶ MINIMALE ANZAHL VON TAGEN FÜR DAS GLEICHE PASSWORT: 3

Klicken Sie auf ÜBERNEHMEN, um die Einstellungen aus allen drei Registerkarten zu bestätigen, oder auf VERWERFEN, um alles beim Alten zu lassen.

Wenn Sie einen neuen Benutzer erstellen möchten, klicken Sie im Hauptdialog auf HINZUFÜGEN. Die drei Registerkarten entsprechen denjenigen bei der Benutzeränderung, sodass Sie sie nach den obigen Informationen ausfüllen können.

Mithilfe des Buttons LÖSCHEN können Sie ein Benutzerkonto entfernen. Da dies recht gefährlich ist, erscheint zunächst eine Sicherheitsabfrage. Sie können keine aktuell angemeldeten Benutzer löschen. Ein besonderes Problem ist dies bei dem Benutzer, der sich ursprünglich für die grafische Oberfläche angemeldet hat – in diesem Fall hilft nur vollständiges Abmelden und erneutes Anmelden.

Das Pulldown-Menü OPTIONEN FÜR EXPERTEN enthält folgende besonderen Einstellungen:

- ▶ PASSWORTVERSCHLÜSSELUNG – hier können Sie sich einen von drei Algorithmen aussuchen, mit denen Passwörter verschlüsselt werden sollen:
 - ▶ DES (Data Encryption Standard) ist die Standardmethode. Dieser Algorithmus wurde in den 1970er-Jahren im Auftrag der US-Regierung entwickelt. Der größte Nachteil besteht darin, dass bei DES-verschlüsselten Passwörtern maximal acht Zeichen unterschieden werden. Zudem beträgt die Schlüssellänge nur 56 Bit, sodass man DES durch einfaches Durchprobieren aller Möglichkeiten (Brute-Force-Attacke) knacken kann – dies wurde bereits 1997 bewiesen. Danach wurden auch mathematische Schwächen offenbar, die ein schnelleres, gezieltes Knacken ermöglichen.

- ▶ MD5 (Message Digest Algorithm 5) wurde 1991 von *Ronald L. Rivest* entwickelt. Seine Hauptaufgabe war ursprünglich nicht die Verschlüsselung, sondern das *Hashing* – er erzeugt aus einem beliebig langen String eine 128 Bit lange, irreversible Zusammenfassung, die meist in 32 Hexadezimalziffern geschrieben wird. Es ist hinreichend sicher, dass selbst eine Abweichung von nur einem Zeichen im Ursprungstext einen anderen MD5-Hash erzeugt. MD5 wurde bereits in Kapitel 2, »openSUSE installieren«, zur Integritätsprüfung heruntergeladener openSUSE-Images vorgestellt. Der Hauptvorteil liegt darin, dass Sie mit MD5 Passwörter verwenden können, die länger als acht Zeichen sind.
- ▶ BLOWFISH ist ein Verschlüsselungsalgorithmus von *Bruce Schneier*. Er verwendet einen Schlüssel von 32 bis 448 Bit Länge und gilt als recht sicher. Auch mit Blowfish können Sie beliebig lange Passwörter verwenden.

Dass heute überhaupt andere Algorithmen als DES eingesetzt werden können, liegt an einer modernen Erweiterung namens PAM (Pluggable Authentication Modules). Diese Technik übergibt die eigentliche Passwortüberprüfung an einen separaten, austauschbaren Dienst. Darauf basiert auch die Anmeldung über externe Verzeichnisse (siehe unten).

- ▶ EINSTELLUNGEN FÜR DAS ANMELDEN – hier können Sie mithilfe der Option AUTOMATISCHE ANMELDUNG festlegen, dass ein bestimmter Benutzer nach dem Booten automatisch eingeloggt werden soll. Wie bereits im vorigen Kapitel angesprochen, sollten Sie dies nicht auf Firmenrechnern einstellen, sondern höchstens zu Hause. Die ANMELDUNG OHNE PASSWORT ist für öffentliche Umgebungen übrigens genauso gefährlich, weil sie allen normalen Benutzern den ungeschützten Zugang zum Desktop erlaubt.
- ▶ ÄNDERUNGEN NUN SCHREIBEN aktualisiert die bisherigen Einstellungen, ohne den Dialog zu verlassen.

Wenn Sie auf die Verwaltung von GRUPPEN umschalten, stehen Ihnen auch für diese die Optionen HINZUFÜGEN, BEARBEITEN und LÖSCHEN zur Verfügung. Der Dialog zum Erstellen oder Ändern einer Gruppe ist weniger umfangreich als derjenige für Benutzer; er besteht nur aus einer Seite, die in Abbildung 3.19 gezeigt wird.

Hier haben Sie im Einzelnen folgende Einstellmöglichkeiten:

- ▶ NAME DER GRUPPE
Dies ist die eindeutige Bezeichnung der Gruppe; das erste Feld in */etc/group*.
- ▶ GRUPPEN-ID (GID)
Diese Einstellung gibt die numerische Kennzeichnung der Gruppe an.

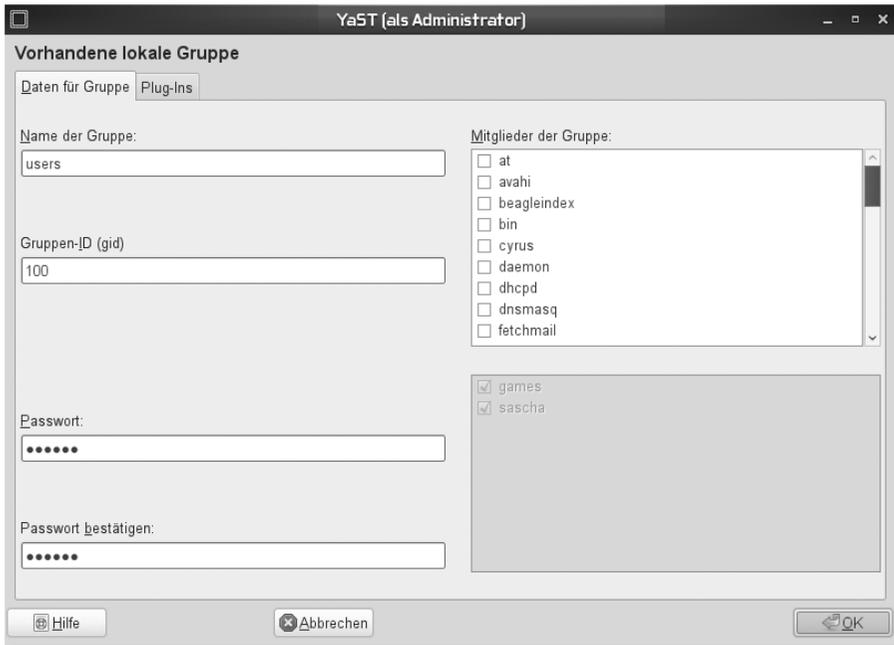


Abbildung 3.19 Die Einstellungen einer Gruppe ändern

- ▶ **PASSWORT UND PASSWORT BESTÄTIGEN**
Auch für Gruppen können Sie Passwörter festlegen; diese werden in der Praxis allerdings erheblich seltener benötigt als Benutzerpasswörter.
- ▶ **MITGLIEDER DER GRUPPE**
In dieser Liste aller verfügbaren Benutzerkonten können Sie diejenigen ankreuzen, die dieser Gruppe als Sekundärgruppe angehören sollen; als Hauptgruppe eines Benutzers kann sie hier nicht eingestellt werden, da diese eine Eigenschaft des Benutzerkontos ist.

Auf der Registerkarte **STANDARDEINSTELLUNGEN FÜR NEUE BENUTZER** können Sie detailliert einstellen, welche automatischen Voreinstellungen für neue Benutzerkonten vorgenommen werden sollen. Unter **STANDARDGRUPPE** wird festgelegt, welcher Gruppe ein neues Benutzerkonto angehören soll; die Vorgabe ist empfehlenswerterweise `users`. **SEKUNDÄRE GRUPPEN** regeln weitere vorgegebene Gruppenmitgliedschaften; vorgegeben sind `dialogout` für Wahlverbindungen ins Internet und `video` für Grafikeinstellungen. **STANDARD-LOGIN-SHELL** (Voreinstellung `/bin/bash`) und **PFADPRÄFIX FÜR HOME-VERZEICHNIS** (`/home`) dürften klar sein. Der **SKELETON FÜR HOME-VERZEICHNIS** gibt eine Vorlage an, deren Inhalt in neu erstellte Home-Verzeichnisse kopiert wird – bei den besonderen Bedürfnissen

moderner UNIX-Systeme mit grafischen Oberflächen eine gute Idee. STANDARD-ABLAUFDATUM und WIE VIELE TAGE NACH ABLAUF DES PASSWORTS KANN DER ANMELDEVORGANG DURCHGEFÜHRT WERDEN? schließlich sind die beiden wichtigsten Aspekte der oben beschriebenen Passwordeinstellungen.

Die letzte Registerkarte – AUTHENTIFIZIERUNGSEINSTELLUNGEN – ist nur zu sehen, wenn Sie einen der Verzeichnisdienste LDAP oder NIS konfiguriert haben; sie regelt die Details der Benutzeranmeldung über diese Dienste. Näheres dazu steht in Kapitel 14, »Datei- und Verzeichnisdienstserver«.

3.3.3 Software nachinstallieren, deinstallieren und aktualisieren

Wie die Softwarepaketverwaltung mit YaST funktioniert, haben Sie bereits im vorigen Kapitel erfahren. Auf dieselbe Weise können Sie damit auch nachträglich Pakete installieren, entfernen oder neue Versionen einspielen. Hier erhalten Sie einige zusätzliche Informationen. Danach lernen Sie eine Kommandozeilernalternative zur Installation kennen und erfahren, wie gängige Open-Source-Programme manuell aus dem Quellcode kompiliert werden.

Die Software-Sektion in YaST

Wenn Sie in YaST die Kategorie SOFTWARE auswählen, steht eine Reihe verschiedener Konfigurationsdialoge zur Verfügung. Einige von ihnen kennen Sie bereits von der Systeminstallation; eine Übersicht finden Sie in Tabelle 3.2.

Kategorie	Bedeutung
Online-Aktualisierung	Durchführung der Online-Aktualisierung
Einrichtung der Online-Aktualisierung	Einstellungen zur automatischen Durchführung der Online-Aktualisierung
Software-Repositories	Auswahl der verschiedenen regulären Installationsquellen (Datenträger und Online-Ressourcen)
Software installieren oder löschen	Pakete hinzufügen, entfernen oder aktualisieren (dieser Dialog wurde im vorigen Kapitel ausführlich beschrieben).
Medien-Überprüfung	Prüfung der Integrität von Installationsdatenträgern (siehe voriges Kapitel)
Zusatz-Produkte	Auswahl zusätzlicher, das heißt außerplanmäßiger Installationsquellen
Paketsuche (webpin)	Sucht in Online-Repositories nach Installationspaketen

Tabelle 3.2 Inhalt der YaST-Kategorie »Software«

Der wichtigste Punkt, den Sie noch nicht aus dem Installationskapitel kennen, sind Online-Updates.

Online-Updates

Online-Updates bieten eine praktische Möglichkeit, manuell oder automatisch die neuesten Sicherheitsupdates herunterzuladen und zu installieren. Wenn Sie eine einmalige Aktualisierung durchführen möchten, wählen Sie den Punkt ONLINE-UPDATES aus. Die aus dem vorigen Kapitel bekannte Paketverwaltungsoberfläche erscheint, und Sie können wie gewohnt die gewünschten Pakete auswählen. Klicken Sie danach auf PRÜFEN, um mögliche Abhängigkeitskonflikte zu bereinigen.

Auf Wunsch können Sie die Online-Updates auch automatisieren. Klicken Sie dazu den Punkt EINRICHTUNG DER ONLINE-AKTUALISIERUNG an. Der Dialog KONFIGURATION FÜR ONLINE-AKTUALISIERUNG wird angezeigt. Wählen Sie hier die Option AUTOMATISCHE ONLINE-AKTUALISIERUNG, um die Funktion grundsätzlich einzuschalten. Standardmäßig findet die Aktualisierung WÖCHENTLICH statt. Auf Wunsch können Sie auch zur Option TÄGLICH wechseln; in diesem Fall entfällt die Auswahl des Wochentags.

Die Schnellzugriffssymbole auf der rechten Seite des Panels in KDE oder GNOME bieten ebenfalls eine Möglichkeit, Online-Updates einzuleiten. Die Anwendung überprüft bei jedem Systemstart (genauer gesagt, sobald die Internetverbindung hergestellt wird), ob Updates verfügbar sind. Wenn dies der Fall ist, verändert sich das Symbol, um darauf hinzuweisen. Sobald Sie es anklicken, wird das Fenster SOFTWARE-AKTUALISIERUNGSFUNKTION geöffnet. Nach einer kurzen Prüfung erhalten Sie eine Liste aktualisierbarer Pakete. Kreuzen Sie die gewünschten an, und klicken Sie auf AKTUALISIEREN. Die Schaltfläche KONFIGURIEREN ermöglicht Ihnen unter anderem die Auswahl der gewünschten Mirror-Server.

RPM-Pakete manuell installieren

Die Softwarepakete, die über YaST verwaltet werden, besitzen traditionell das Dateiformat RPM (RedHat Package Manager). Wie der Name schon sagt, wurde dieses Format ursprünglich von dem Distributor RedHat eingeführt, aber bald darauf auch von SUSE und anderen Distributionen übernommen. Es handelt sich um ein Archivformat mit zusätzlichen Metainformationen – insbesondere werden die Namen der Installationsverzeichnisse sowie Paketabhängigkeiten gespeichert.

RPM-Pakete lassen sich nicht nur mit YaST installieren, sondern auch über das Kommandozeilentool `rpm`. Dies ist besonders nützlich, um externe RPMs zu

installieren, das heißt Pakete, die nicht mit Ihrer Distribution geliefert, aber dafür optimiert wurden. Die wichtigsten Optionen dieses Kommandos sind:

- ▶ `rpm -i Paketname`
`rpm --install Paketname`

Diese beiden synonymen Aufrufe installieren das angegebene RPM-Paket mit Standardoptionen.

- ▶ `rpm -U Paketname`
`rpm --upgrade Paketname`

Eine dieser Anweisungen aktualisiert das Paket, falls es bereits vorhanden ist, und installiert es andernfalls neu.

- ▶ `rpm -F Paketname`
`rpm --freshen Paketname`

Auch diese Variante aktualisiert das angegebene Paket, aber nur, falls bereits eine ältere Version existiert.

- ▶ `rpm -e Paketname`
`rpm --erase Paketname`

Ein Aufruf mit einer dieser Optionen deinstalliert das entsprechende Paket. Dabei müssen Sie die Endung `.rpm` weglassen.

- ▶ `rpm -q Paketname`
`rpm --query Paketname`

Gibt Auskunft darüber, ob das betreffende Paket bereits installiert ist, und wenn ja, in welcher Version – Sie können bei der Angabe des Paketnamens die Versionsnummer am Ende weglassen.

Für alle Varianten des Befehls außer der Auskunftsoption `-q` beziehungsweise `--query` müssen Sie `root` sein, weil Installationen oder Deinstallationen in der Regel auf Verzeichnisse zugreifen müssen, die für gewöhnliche Benutzer tabu sind. Starten Sie also – wie oben gezeigt – ein Terminalfenster, und geben Sie vor der Verwendung von `rpm` Folgendes ein:

```
Username@Rechner:~> su
Passwort:
Rechner: /home/Username #
```

Die Eingabeaufforderung wechselt wiederum nach dem weiter oben angegebenen Schema (im Folgenden wird sie einfach als `#` dargestellt). Als Beispiel soll das Paket `mysql-administrator-1.1.6-1.suse93.i586.rpm` installiert werden. Es handelt sich um ein grafisches Administrationstool für den in Kapitel 12 vorge-

stellten Datenbankserver MySQL, optimiert für das veraltete System SUSE Linux 9.3 (zumindest unter openSUSE 10.2 lief diese Version aber noch). Sie können es auf der Website des MySQL-Projekts herunterladen.

Sorgen Sie dafür, dass sich die RPM-Datei in Ihrem Home-Verzeichnis befindet. Nach dem Download liegt sie vermutlich auf dem Desktop oder im Verzeichnis *Documents*, die beide Unterverzeichnisse des Home-Verzeichnisses sind. Sie können sie mithilfe des weiter oben beschriebenen Dateimanagers Ihrer grafischen Oberfläche verschieben.

Geben Sie nun eine der beiden folgenden Anweisungen ein, um das Paket zu installieren (die zusätzliche Option `-h` beziehungsweise `--hash` gibt einen Fortschrittsbalken in Form von `#`-Zeichen aus):

```
# rpm -i -h mysql-administrator-5.0-r14-6.7.1.i586.rpm
```

oder

```
# rpm --install mysql-administrator-5.0-r14-6.7.1.i586.rpm
```

Wenn Paketabhängigkeiten nicht erfüllt sein sollten, erhalten Sie eine entsprechende Meldung, und die Installation schlägt fehl. In diesem Fall müssen Sie zuerst die fehlenden Pakete installieren. Sie können mithilfe der YaST-Suchfunktion überprüfen, ob diese Pakete zum Lieferumfang der Distribution gehören, ansonsten müssen Sie im Internet danach suchen. Eine bekannte Quelle für RPMs aller Art ist beispielsweise die Website <http://www.rpmfind.com>.

Falls eine frühere Version von MySQL Administrator installiert ist, können Sie diese mittels

```
# rpm -U -h mysql-administrator-5.0-r14-6.7.1.i586.rpm
```

beziehungsweise

```
# rpm --upgrade --hash \  
> mysql-administrator-5.0-r14-6.7.1.i586.rpm
```

aktualisieren. Der Backslash `\` in der Langform ist optional; er ermöglicht es Ihnen, innerhalb einer Anweisung  zu drücken, um diese auf mehrere Zeilen zu verteilen. Die Upgrade-Funktion installiert das Paket auch dann, falls es noch nicht vorhanden war. Um ausschließlich vorhandene Pakete zu aktualisieren, können Sie stattdessen die Option `-F` (ausführliche Schreibweise `--freshen`) verwenden.

Um dieses Paket wieder zu deinstallieren, müssen Sie folgende Anweisung eingeben (die zugehörige Langform wäre `--erase`):

```
# rpm -e mysql-administrator-5.0-r14-6.7.1.i586
```

Wie bereits erwähnt, muss die Endung `.rpm` dabei weggelassen werden.

Ein Test, ob und in welcher Version das Paket installiert ist, sieht schließlich so aus (hier in der ausführlichen Form; die Kurzschreibweise ist `-q`):

```
# rpm --query mysql-administrator
mysql-administrator-5.0-r14-6.7.1.i586
```

Ein weiteres, erst in openSUSE 10 eingeführtes Konsolentool zur Installation und zur Verwaltung von Installationspaketen ist `zypper`. Es lädt mehrere voneinander abhängige RPM-Pakete automatisch herunter und installiert sie. Dabei brauchen Sie noch nicht einmal eine Versionsnummer anzugeben; es wird automatisch die neueste zu Ihrem aktuellen System passende Version installiert. Die folgende Eingabe genügt beispielsweise, um `mysql-administrator` sowie die gesamte Familie der MySQL-GUI-Tools zu installieren:

```
# zypper install mysql-administrator
```

Sie erhalten die Meldung, dass die passenden Pakete gefunden wurden und heruntergeladen werden können:

```
8 neue Pakete zu installieren.
Gesamtgröße des Downloads: 10,5 MiB. Nach der Operation werden
zusätzlich 33,7 MiB belegt.
Fortfahren [j/n/?] (j): j
```

Bestätigen Sie die Installation durch Eingabe von `j` beziehungsweise Enter; der Rest geht automatisch vonstatten.

Gängige Open-Source-Programme manuell kompilieren

Viele freie Softwareprojekte bieten selbst keine vorkompilierten Binärpakete für Linux und andere UNIX-Systeme an. Wenn auch Ihre Distribution diese Programme nicht enthält, müssen Sie sie meist selbst kompilieren. Dazu muss – wie im vorigen Kapitel empfohlen – die GNU Compiler Collection (`gcc`) installiert sein, was openSUSE leider nicht standardmäßig erledigt.

Die meisten in den Programmiersprachen C oder C++ geschriebenen Open-Source-Programme werden heute mit den modernen GNU-Tools *Automake*, *Autoconf* und *Libtool* erstellt. Diese passen die Software mithilfe eines standardisierten Verfahrens für Ihre Systemplattform an. In der Regel müssen Sie eine Abfolge von fünf Befehlen eingeben, um solche Programme zu kompilieren und zu installieren:

1. Entpacken Sie die Archivdatei (am besten in Ihrem Home-Verzeichnis). Wenn sie die Endung `.tar.gz` besitzt, lautet die Anweisung:

```
$ tar xzvf Paketname.tar.gz
```

Das Dollarzeichen entspricht dem Prompt für einen normalen Benutzer. Bei Paketen mit der Erweiterung `.tar.bz2` muss der Parameter `j` statt `z` eingesetzt werden:

```
$ tar xjvf Paketname.tar.bz2
```

2. Das Entpacken erzeugt in aller Regel ein neues Unterverzeichnis namens *Paketname*, in das Sie mittels `cd` wechseln können:

```
$ cd Paketname
```

3. Nun muss das Autoconf-Skript `configure` ausgeführt werden, um angepasste *Makefiles* (Sammlungen von Kompilieranweisungen) für Ihr System zu erstellen. Meistens besitzt dieses Skript eine Option namens `--help`, die alle möglichen Parameter auflistet. Da die Liste häufig zu lang für das Terminalfenster ist, sollten Sie Folgendes eingeben, um den Inhalt seitenweise anzuzeigen:

```
$ ./configure --help |less
```

Das Blättern und die sonstige Bedienung funktionieren genau wie bei dem weiter oben beschriebenen Programm `man`, da es intern ebenfalls `less` verwendet. Wenn Sie die passenden Optionen gefunden haben, rufen Sie `configure` entsprechend auf:

```
$ ./configure -Option [--Option ...]
```

Diese Anweisung führt zu einer endlosen Folge von Ausgabezeilen, zunächst Systemüberprüfungen, die mit `Checking...` beginnen. Wenn das Skript mit einem Fehler abbricht, liegt dies meist daran, dass eine erforderliche Bibliothek fehlt. Versuchen Sie, sie mithilfe der YaST-Suchfunktion nachzuinstallieren. Wenn dies misslingt, ist die Bibliothek nicht in Ihrer Distribution vorhanden, und Sie müssen auf der Website des entsprechenden Open-Source-Projekts oder per Suchmaschine danach stöbern.

4. Wenn `configure` erfolgreich durchgelaufen ist, wird `make` aufgerufen – dies ist der eigentliche Kompilierbefehl, der die Quellcodes gemäß den Informationen aus den zuvor angepassten *Makefiles* übersetzt. Der Aufruf kommt stets ohne Parameter aus:

```
$ make
```

5. Sobald auch das Kompilieren abgeschlossen ist, findet die Installation statt – die fertig kompilierten Dateien werden in das vorgesehene Zielverzeichnis kopiert. Spätestens diesen Schritt müssen Sie als `root` durchführen, weil Schreibzugriff auf die entsprechenden Verzeichnisse erforderlich ist. Geben Sie dazu wie gehabt `su` und das `root`-Passwort ein. Danach erfolgt die Eingabe:

```
# make install
```

Hier als Beispiel die Befehlsabfolge zur manuellen Kompilierung und Installation des Nameservers BIND, der in Kapitel 13, »Netzwerkgrundlagen«, näher vorgestellt wird:

```
$ tar xzvf bind-9.6.1-P2.tar.gz
$ cd bind-9.6.1-P2
$ ./configure
$ make
$ su
Passwort:
# make install
```

3.3.4 Programme automatisch starten

Auf UNIX-Systemen werden laufende Programme *Prozesse* genannt; Details darüber erfahren Sie im nächsten Kapitel sowie an anderen passenden Stellen dieses Buches (die technischen Hintergründe gibt es dann in Kapitel 18, »Hinter den Kulissen«). Einige Prozesse werden nach dem Booten automatisch gestartet, und sie laufen im Hintergrund ohne Verbindung zu einem Terminal. Aus diesen Gründen werden sie als *Daemons* bezeichnet. Die entsprechenden Programme liegen häufig im Verzeichnis `/usr/sbin`, und ihre Namen enden meist mit dem Buchstaben `d` für *Daemon* – hier einige Beispiele:

- ▶ `/usr/sbin/sshd`
Der *Secure Shell Daemon* ermöglicht das Arbeiten auf einem entfernten Rechner über eine verschlüsselte Verbindung (siehe Kapitel 14, »Datei- und Verzeichnisdienstserver«).
- ▶ `/usr/sbin/cupsd`
Der *Common UNIX Printing System Daemon* stellt auf modernen UNIX-Systemen sämtliche Dienstleistungen rund um das Drucken zur Verfügung, sowohl lokal als auch im Netzwerk (siehe Kapitel 6, »Wichtige Hardware konfigurieren«).
- ▶ `/usr/sbin/httpd2`
Der *Apache HyperText Transfer Protocol Daemon* Version 2, ein mächtiger Webserver (siehe Kapitel 15, »LAMP«). Der Originalprogrammname lautet

übrigens `httpd`; die Variante `httpd2` führten einige Distributionen in der Übergangsphase zur neuen Version ein, weil sie beide Versionen enthielten.

Runlevels und init-Skripte

Linux hat das Programmsteuerungsprinzip des klassischen AT&T UNIX – das sogenannte *System V Init* – übernommen und steuert den Start und das Beenden von Daemons und anderen automatisch laufenden Programmen über sogenannte *Runlevels*. Ein Runlevel ist ein Systemzustand, in dem jeweils nur bestimmte Prozesse laufen. Beim Wechsel des Runlevels über den Befehl `init LEVELNR.` werden bestimmte Skripte aufgerufen, die manche Programme starten und andere beenden. Einige Runlevels haben eine spezielle Bedeutung:

- ▶ 0 heruntergefahrener Zustand
- ▶ S Single-User-Modus (für Wartungsarbeiten)
- ▶ 1 ebenfalls Single-User-Modus
- ▶ 2 Multi-User-Modus mit lokalem Netzwerk; nur Konsole
- ▶ 3 Multi-User-Modus mit vollem Netzwerkzugriff
- ▶ 5 Multi-User-Modus mit Netzwerk und grafischer Oberfläche (KDM, GDM oder XDM)
- ▶ 6 Systemneustart (Reboot)

Wenn Sie manuell in ein anderes Runlevel wechseln möchten, können Sie als `root` das Kommando `init LEVELNR.` eingeben – tippen Sie etwa Folgendes ein, um den Rechner herunterzufahren:

```
# init 0
```

Um herauszufinden, in welchem Runlevel der Rechner sich gerade befindet, geben Sie einfach `runlevel` ein (ebenfalls nur als `root`):

```
# runlevel
```

```
5
```

Für jedes Runlevel gibt es ein spezielles *Init-Verzeichnis*. Diese Verzeichnisse heißen `/etc/init.d/rcLEVELNR.d`, also etwa `/etc/init.d/rc1.d` für Runlevel 1 oder `/etc/init.d/rc5.d` für Runlevel 5. Die Shell-Skripte in diesen Verzeichnissen werden bei Erreichen des entsprechenden Levels automatisch ausgeführt, und zwar in alphabetischer Reihenfolge. Deshalb verwendet die übliche Konvention Namen, die mit *K* beginnen, für *Kill-Skripte* (die einen Prozess beenden) und solche mit *S* für *Start-Skripte*. Darüber hinaus bauen viele Daemons aufeinander auf. Deshalb wird hinter dem Anfangsbuchstaben eine zweistellige Zahl verwendet, die für eine bestimmte Reihenfolge sorgt.

In aller Regel sind die Einträge in diesen Verzeichnissen lediglich symbolische Links (Dateisystemverknüpfungen) mit Skripten, die sich eigentlich in einem anderen Verzeichnis befinden, meistens im übergeordneten Verzeichnis */etc/init.d*. Für den Start und das Beenden des jeweiligen Prozesses wird normalerweise dasselbe Skript verwendet: Ein *S*-Symlink ruft es automatisch mit der Kommandozeilenoption *start* auf, ein *K* mit *stop*. Geben Sie beispielsweise Folgendes ein, um zu erfahren, welche Daemons in Runlevel 5 gestartet beziehungsweise beendet werden (die Ausgabe wurde hier sehr stark gekürzt):

```
# ls /etc/init.d/rc5.d
K01SuSEfirewall12_setup
K04cupsrenice
K06cron
K06xinetd
K07apache2
[...]
S01acpid
S01dbus
[...]
S15apache2
S16cron
S16xinetd
S18cupsrenice
S21SuSEfirewall12_setup
```

Hier finden Sie die Anleitung für einen manuell kompilierten Apache-Webserver. Dieser beinhaltet ein Startskript namens *apachectl*, das standardmäßig unter */usr/local/apache2/bin* (eventuell auch in ein anderes Verzeichnis – siehe Kapitel 15) installiert wird.

Erzeugen Sie für die Runlevels mit voller Netzwerkfähigkeit (3 und 5) einen *S*-Symlink auf dieses Skript. Für die Runlevels 0 und 6 (Herunterfahren beziehungsweise Neustart) können Sie entsprechend einen *K*-Link anlegen. Da von Apache in der Regel keine anderen Dienste abhängen, können Sie ihn recht spät starten (wählen Sie einen Symlink-Namen wie *S95apache*) und ziemlich früh beenden (*K15apache* dürfte in Ordnung gehen).

Folgende Befehle sind für die Startskript-Links geeignet:

```
# ln -s /usr/local/apache2/bin/apachectl \
> /etc/init.d/rc3.d/S95apache
# ln -s /usr/local/apache2/bin/apachectl \
> /etc/init.d/rc5.d/S95apache
```

Als Nächstes werden die Stop-Links für die Runlevels 0 und 6 erzeugt:

```
# ln -s /usr/local/apache2/bin/apachectl \
> /etc/init.d/rc0.d/K15apache
# ln -s /usr/local/apache2/bin/apachectl \
> /etc/init.d/rc6.d/K15apache
```

Das Tool chkconfig

Unter openSUSE können Sie das Erstellen der besagten Start- und Stop-Links mithilfe des Kommandos `chkconfig` automatisieren. Dazu benötigen Sie ein allgemeines Init-Skript unter `/etc/init.d`. Um die Start-Links zu erstellen, wird `chkconfig` folgendermaßen aufgerufen:

```
# chkconfig -a Skriptname
```

Wenn Sie den automatischen Start des betreffenden Daemons wieder ausschalten möchten, wird diese Option verwendet:

```
# chkconfig -d Skriptname
```

Ein Statustest ist schließlich mithilfe des Parameters `-l` möglich:

```
# chkconfig -l Skriptname
```

Angenommen, Sie haben den Datenbankserver MySQL manuell kompiliert und installiert (siehe Kapitel 12). Wenn Sie dies mit Standardoptionen erledigt haben, finden Sie unter `/usr/local/mysql/support-scripts` das Skript `mysql.server`. Erstellen Sie zuerst einen Link nach `/etc/init.d`:

```
# ln -s /usr/local/mysql/support-scripts/mysql.server \
> /etc/init.d/mysql
```

Nun lässt sich der Symlink mithilfe von `chkconfig -a` aktivieren:

```
# chkconfig -a mysql
mysql 0:off 1:off 2:on 3:on 4:on 5:on 6:off
```

Den automatischen Start mit YaST konfigurieren

Über den *Runlevel-Editor* in YaST können Sie bequem einstellen, welche Programme beziehungsweise Daemons in welchem Runlevel ausgeführt werden sollen. Starten Sie dazu wiederum YaST, und wählen Sie SYSTEMDIENSTE (RUNLEVEL) in der Kategorie SYSTEM. In Abbildung 3.20 sehen Sie die Funktion in der grafischen Ansicht.



Abbildung 3.20 Der Runlevel-Editor in der grafischen Ansicht (GNOME)

Standardmäßig ist die Option `EINFACHER MODUS` aktiviert. Hier können Sie den jeweiligen Dienst in der Liste anklicken und seinen Status mithilfe der Schaltflächen `AKTIVIEREN` beziehungsweise `DEAKTIVIEREN` ändern. Wenn Sie in den Expertenmodus wechseln, können Sie sich dagegen für jedes Programm detailliert aussuchen, in welchen Runlevels es aktiv sein soll und in welchen nicht.

Wenn Sie den Start eines manuell installierten Programms mithilfe des Runlevel-Editors steuern möchten, müssen Sie zunächst einen Link auf sein Startskript in `/etc/init.d` ablegen, wie oben für MySQL beschrieben.

Die rc-Skripte

Aus Gründen der Bequemlichkeit enthält openSUSE noch einen weiteren Satz von Startskripten; sie befinden sich unter `/usr/sbin` und beginnen allesamt mit `rc`. Sie sind für die unmittelbare Anwendung durch den Benutzer `root` vorgesehen: Um einen Daemon oder Systemprozess zu starten, zu beenden oder neu zu starten, können Sie ein solches Skript als `root` mit einem der Argumente `start`, `stop` beziehungsweise `restart` aufrufen. `/usr/sbin` befindet sich dabei im Suchpfad von `root`, sodass die Eingabe aus einem beliebigen Arbeitsverzeichnis erfolgen kann.

Das folgende Beispiel beendet den Webserver Apache 2:

```
# rcapache2 stop
```

Die nachfolgende Eingabe startet den Datenbankserver MySQL, falls er noch nicht läuft:

```
# rcmysql start
```

Hier das letzte Beispiel – es startet Samba, den Dateiserver für Windows-Clients, neu. Falls er noch nicht lief, wird er entsprechend gestartet:

```
# rcsmb restart
```

Einige `rc`-Skripte besitzen noch mehr Optionen als die drei genannten; dies erfahren Sie, indem Sie `rc* help` eingeben. Beispiel:

```
# rcsmb help
```

```
Usage: /usr/sbin/rcsmb {start|stop|status|try-restart|restart|force-reload|reload|probe}
```

Beachten Sie, dass diese Skripte nur für Software zur Verfügung steht, die offizieller Teil Ihrer Distribution ist. Wenn Sie eines dieser Programme manuell kompiliert haben, können Sie aber in der Regel einen Symlink von dessen eigenem Startskript nach `/usr/sbin/rc*` erstellen. Hier als Beispiel die entsprechende Eingabe für einen mit Standardoptionen selbst kompilierten Apache-Webserver:

```
# ln -s /usr/local/apache2/bin/apachectl /usr/sbin/rcapache22
```

Der Name `rcapache22` (für Version 2.2) wurde aus Vorsichtsgründen gewählt, damit der selbst kompilierte Webserver nicht mit der eventuell vorhandenen Version aus der Distribution kollidiert.

3.4 Zusammenfassung

In einer modernen Linux-Distribution wie openSUSE ist es sehr wahrscheinlich, dass gleich nach der Installation eine komfortable grafische Benutzeroberfläche gestartet wird. In diesem Kapitel wurden die ersten Schritte im Umgang mit den beiden leistungsfähigen Desktops KDE und GNOME erläutert; Näheres über beide sowie über das zugrundeliegende X Window System erfahren Sie in Kapitel 5, »Grafische Oberflächen«.

Ein weiteres wichtiges Thema für Einsteiger, aber auch als Referenz für erfahrene Benutzer, sind die diversen Hilfe-Optionen und Online-Handbücher. Schon die frühesten UNIX-Systeme waren mit den sogenannten Manpages als elektroni-

scher Dokumentation ausgestattet. Dieses Format existiert bis heute; mithilfe von `man Befehlsname` erhalten Sie Hilfe zu jedem beliebigen Konsolenbefehl und vielen anderen Themen. Da GNU schon seiner Bezeichnung nach nicht UNIX ist, hat das GNU-Projekt sein eigenes Dokumentationsformat `GNU info` eingeführt. Auf Linux-Systemen sind die `info`-Seiten daher meist ausführlicher und aktueller als die entsprechenden `Manpages`, zudem lässt sich das Programm `info` noch komfortabler bedienen als `man`. Abgerundet wird das Dokumentationsangebot durch die `openSUSE-Hilfe`, die Hilfeseiten in einer komfortablen, Browser-ähnlichen Oberfläche darstellt.

Zu guter Letzt haben Sie in diesem Kapitel diverse wichtige Konfigurationseinstellungen kennengelernt. `openSUSE` verfügt seit Jahren über das komfortable Konfigurationsprogramm `YaST`, mit dem Sie die meisten Aspekte des Systems interaktiv einstellen können, sowohl grafisch als auch im Textmodus. Im Einzelnen wurden hier die Benutzerverwaltung, die nachträgliche Installation von Software sowie der automatische Start von Programmen besprochen. Weitere Konfigurationsoptionen lernen Sie in späteren Kapiteln dieses Buches kennen.

*I tried to live in the real world instead of a shell
but I was bored before I even began.
– Morrissey (The Smiths)*

4 Mit der Shell arbeiten

Auch wenn openSUSE und andere neuere Linux-Distributionen inzwischen schon bei der Installation eine grafische Benutzeroberfläche einrichten (die ausführlich in Kapitel 5, »Grafische Oberflächen«, beschrieben wird), sollten Sie den Umgang mit der Konsole erlernen. Die mächtigsten Funktionen des Systems werden nach wie vor über die Kommandozeile aufgerufen. Erst allmählich stehen grafische Steuerprogramme dafür zur Verfügung, und selbst diese beherrschen nur selten alle Optionen der jeweiligen Aufgabe.

Aber auch alltägliche Verwaltungsaufgaben kann die Shell oft erheblich schneller ausführen als eine grafische Oberfläche. Stellen Sie sich beispielsweise vor, Sie möchten in einem Verzeichnis alle JPEG-Bilder löschen, die zu einem speziellen Website-Projekt gehören und deren Namen daher alle mit *music_* beginnen. In gängigen grafischen Benutzeroberflächen gibt es natürlich Sortierfunktionen, mit denen Sie die entsprechenden Dateien gruppieren können. Anschließend können Sie sie mit gedrückter Maustaste umfahren oder Ähnliches, um sie zu markieren, und schließlich mit der `[Entf]`-Taste oder durch Ziehen auf ein Müll-eimersymbol löschen. (Den Mülleimer müssen Sie dann wahrscheinlich auch noch leeren, um sie endgültig loszuwerden.) Das alles geht ganz bestimmt nicht schneller als ein einziges

```
rm music_*.jpg
```

Wenn Unterverzeichnisse dazukommen, wird es in der GUI erst recht knifflig. Da hilft meist nur noch eine Suchfunktion, die hoffentlich eine Liste der gewünschten Dateien liefert. In der Shell genügt dagegen das Hinzufügen eines einzigen Parameters, nämlich `-r` für das »rekursive« Durchsuchen der Unterverzeichnisse:

```
rm -r music_*.jpg
```

Das dürfte für den Anfang genügen, um von den Vorteilen der Shell überzeugt zu werden und hoch motiviert in dieses Kapitel einzusteigen.

4.1 Wichtige Grundbegriffe

Bevor Sie praktisch mit der Shell arbeiten können, benötigen Sie einige Informationen über das Betriebssystem und seine Bestandteile. In diesem Abschnitt werden weitere wichtige Kenntnisse vermittelt. Falls Sie bereits mit Linux oder einem anderen Betriebssystem aus der UNIX-Familie gearbeitet haben, können Sie ihn überspringen oder kurz überfliegen, denn auch hier geht es eher um Grundlagen. Die technischen Details werden in späteren Kapiteln dieses Buches behandelt, besonders in Kapitel 18, »Hinter den Kulissen«.

4.1.1 Dateisysteme und Verzeichnisbaum

Die wichtigste Grundlage für das Verständnis der Shell bildet die Kenntnis der Organisationsform von Dateien und Verzeichnissen, zumal unter Linux – wie auf allen UNIX-artigen Systemen – »alles eine Datei ist«: Geräte, bestimmte Kommunikationskanäle und sogar Informationen über den Systemzustand sind in den Verzeichnisbaum eingebunden und lassen sich wie jede andere Datei behandeln.

Informationen über die Organisation von Dateien in konkreten Dateisystemen erhalten Sie in Kapitel 9, »Datenträger und Dateisysteme«. Hier geht es dagegen um das sogenannte *virtuelle Dateisystem*, das heißt die Art und Weise, wie sich Dateien und Verzeichnisse dem Benutzer ungeachtet der konkreten Dateisystemtreiber und Datenträger präsentieren.

Der Verzeichnisbaum

Auf einem Linux-System existiert nur ein einziger Verzeichnisbaum, unabhängig davon, auf wie viele konkrete Datenträger er verteilt ist. Die Datenträger selbst werden an definierbaren Stellen in den Verzeichnisbaum eingehängt (gemountet). Die Wurzel des gesamten Baums wird als / (Root-Verzeichnis) bezeichnet. Unterhalb dieses obersten Verzeichnisses liegen einzelne Dateien und Unterverzeichnisse; jedes von ihnen kann wiederum in Unterverzeichnisse unterteilt sein.

Die meisten Verzeichnisse, die direkt unterhalb der Wurzel des Dateisystems liegen, haben spezielle Aufgaben, die in allen gängigen Linux-Distributionen identisch oder zumindest ähnlich sind:

► *bin* (binaries)

Dieses Verzeichnis enthält die Systemprogramme, also Standardkonsolenkommandos wie `ls` (Verzeichnisinhalt anzeigen) oder `cat` (Textdateien ausgeben). Viele dieser Dienstprogramme werden im vorliegenden Kapitel besprochen.

- ▶ *sbin* (system oder start binaries)
Dieses Verzeichnis enthält weitere Dienstprogramme. Traditionell handelt es sich vor allem um Initialisierungsprogramme, die beim Systemstart aufgerufen werden.
- ▶ *dev* (devices)
Dort werden die Gerätedateien abgelegt, das heißt Dateien, die auf die einzelnen Hardwarekomponenten verweisen. Der Vorteil dieser Methode ist, dass sich der Zugriff auf Geräte genau wie bei einzelnen Dateien über Benutzerrechte regeln lässt.
- ▶ *usr* (user)
Hier finden Sie diverse Unterverzeichnisse für die Komponenten der wichtigsten Anwendungsprogramme. Beispiele: Unter */usr/bin* befinden sich die ausführbaren Dateien. */usr/lib* ist das Verzeichnis für gemeinsam genutzte Bibliotheken. */usr/include* enthält die C-Header-Dateien des Systems und der Bibliotheken – wichtig für die Kompilierung gängiger Open Source-Software, die im Quellcode geliefert wird. Unter */usr/X11R6* befinden sich manche Anwendungen, die unter der grafischen X-Windows-Oberfläche laufen.
- ▶ *opt* (optional)
Enthält zusätzliche Anwendungen, die weniger häufig benötigt oder aber nachträglich installiert werden.
- ▶ *etc*
Das Verzeichnis *etc* enthält die meisten systemweiten Konfigurationsdateien, sowohl für das Betriebssystem selbst als auch für viele Serverdienste und Anwendungen. Eine besondere Bedeutung haben zum Beispiel die Init-Skripte unter */etc/init.d*, die für den automatischen Start von Programmen beim Booten zuständig sind. Neben den hier gespeicherten globalen Einstellungen gibt es bei vielen Programmen auch benutzerspezifische Konfigurationsdateien. Diese werden in den Home-Verzeichnissen der jeweiligen Benutzer gespeichert; ihre Namen beginnen meist mit einem Punkt.
- ▶ *var*
Dieses Verzeichnis enthält variable Daten, vor allen Dingen Logdateien, in die Fehlermeldungen und Hinweise vom Betriebssystem und aus anderen Quellen eingetragen werden.
- ▶ *home*
Das Hauptverzeichnis *home* enthält für jeden (menschlichen) Benutzer, der dem System bekannt ist, ein *Home-Verzeichnis*. Hier werden alle Anwendungsdaten dieses Benutzers abgelegt, das heißt die Dateien, die aus dessen Arbeit mit den Anwenderprogrammen resultieren. Zusätzlich werden auch die persönlichen Einstellungen der jeweiligen Benutzer für die verschiedenen

Anwendungs- und Systemprogramme in deren Home-Verzeichnissen gespeichert. Gewöhnliche Benutzer haben außerhalb ihres Home-Verzeichnisses in aller Regel keine Schreibrechte, sodass sie weder vorsätzlich noch aus Unkenntnis das System oder die Dateien anderer User beschädigen können.

► *root*

Dieses Verzeichnis ist das spezielle Home-Verzeichnis des gleichnamigen Superusers. Es liegt nicht im Verzeichnis *home* wie die anderen Benutzerverzeichnisse. *home* wird nämlich oft so eingerichtet, dass es auf einem anderen physikalischen Datenträger oder zumindest in einer anderen Partition liegt als der Rest des Betriebssystems. Möglicherweise steht es also nicht zur Verfügung, wenn ein Fehler auftritt, den *root* beheben muss. Da Sie normale Aufgaben am Rechner aus Sicherheitsgründen nicht als *root* erledigen sollten, dient dieses Verzeichnis in der Regel nicht der Speicherung von Arbeitsdateien, sondern beherbergt vor allem die persönlichen Konfigurationseinstellungen von *root*.

Übrigens bricht modernere Software des Öfteren mit diesen klassischen Verzeichniskonventionen: Beim Kompilieren aus dem Quelltext wird selbst Serversoftware wie Apache oder MySQL standardmäßig komplett in ein eigenes Stammverzeichnis unter */usr/local* installiert. Der Vorteil: Beim Umstieg auf eine neue Version (oder ein ganz anderes Produkt) lassen sich solche Programme leichter entfernen oder aktualisieren, als wenn ihre Bestandteile gemäß ihrer jeweiligen Aufgaben im gesamten Verzeichnisbaum verteilt würden. Andererseits befinden sich solche separaten Verzeichnisse in der Regel nicht im Suchpfad, sodass für den Aufruf ihrer Konsolenhilfsprogramme jeweils der gesamte Pfad angegeben werden muss. Beispielsweise müsste bei einer MySQL-Standardinstallation¹ Folgendes eingegeben werden, um den Kommandozeilenclient *mysql* (siehe Kapitel 12, »Der Datenbankserver MySQL«) zu starten:

```
$ /usr/local/mysql/bin/mysql -u username -p
```

In der Regel ist es unnötig, das *bin*-Unterverzeichnis eines solchen Programms permanent zum Suchpfad hinzuzufügen. Stattdessen empfiehlt es sich, einfach einen symbolischen Link zu erzeugen, der das fragliche Programm oder Skript mit einem Standardverzeichnis verknüpft; eine gute Empfehlung ist etwa */usr/local*. Hier die passende Anweisung für den *mysql*-Client, die Sie bei einer MySQL-Standardinstallation als *root* eingeben müssen:

```
# ln -s /usr/local/mysql/bin/mysql /usr/bin/mysql
```

¹ Mit »Standardinstallation« ist hier die Distribution der Firma MySQL gemeint; bei der in openSUSE enthaltenen MySQL-Version brauchen Sie dergleichen natürlich nicht zu tun.

In Zukunft können Sie dann aus jedem beliebigen Arbeitsverzeichnis einfach Folgendes eingeben, um das Programm `mysql` aufzurufen:

```
$ mysql -u root -p
```

Absolute und relative Pfade

Der Pfad zu einer Datei wird von der Wurzel aus angegeben, indem die Namen der entsprechenden Ordner jeweils durch einen Slash voneinander getrennt werden. Das Folgende wäre beispielsweise der Pfad der Datei mit dem aktuellen Kapitel in meinem Home-Verzeichnis:

```
/home/sascha/opensuse11_2/shell.txt
```

Alternativ können Pfade auch relativ angegeben werden, das heißt vom aktuellen Verzeichnis aus. Da in der Shell jeweils ein aktuelles Arbeitsverzeichnis ausgewählt ist (es kann mittels `cd` gewechselt werden – siehe unten), kommt dies häufig vor. Angenommen, das Arbeitsverzeichnis ist zurzeit `/home/sascha/texte`, und es soll auf die Datei `bild.tif` im Verzeichnis `/home/sascha/bilder` zugegriffen werden. Der Pfad dieser Datei kann entweder absolut als `/home/sascha/bilder/bild.tif` oder relativ (von `/home/sascha/texte` aus) als `../bilder/bild.tif` angegeben werden: `..` spricht jeweils das übergeordnete Verzeichnis an; untergeordnete Verzeichnisse werden einfach mit ihrem Namen angegeben.

»Geschwister«-Verzeichnisse, also nebengeordnete – in diesem Fall `texte` und `bilder` – können einander nie direkt ansprechen, sondern müssen mittels `../`-Angaben so weit nach oben wandern, bis ein gemeinsamer Vorfahr gefunden wurde. Im Falle von `texte` und `bilder` brauchen Sie nicht weit nach oben zu wandern; `sascha` ist bereits der direkte, gemeinsame Elternordner beider Verzeichnisse.

Eine Abkürzung für das Home-Verzeichnis des aktuell angemeldeten Benutzers ist übrigens die Tilde (`~`). Sie können von überall aus in Ihr Home-Verzeichnis wechseln, indem Sie `cd ~` eingeben. In der openSUSE-Standard-Shell ist sogar `cd` ohne Parameter zulässig.

Im Übrigen sollten Sie daran denken, dass Linux- und andere UNIX-Systeme bei Datei- und Verzeichnisnamen zwischen Groß- und Kleinschreibung unterscheiden. Die Namen `hallo.txt`, `Hallo.Txt` und `HALLO.TXT` bezeichnen drei verschiedene Dateien, die alle im gleichen Verzeichnis liegen könnten.

Dateinamen dürfen Buchstaben, Ziffern und zahlreiche Sonderzeichen enthalten. Prinzipiell sind auch Umlaute, diakritische Zeichen anderer Sprachen und sogar nichtlateinische Schriftzeichen möglich. Da dies allerdings Probleme bereitet, sobald in einer neuen Umgebung ein anderer Zeichensatz eingestellt wird, ist

davon in der Regel abzuraten. Auch Leerzeichen sollten in Dateinamen eher nicht verwendet werden; wenn Sie es dennoch tun, müssen Sie die entsprechenden Dateinamen in Ihren Kommandos in Anführungszeichen setzen oder den Leerzeichen einen Backslash voranstellen.

Dateinamen, die mit einem Punkt (.) beginnen, werden in der normalen Verzeichnisansicht standardmäßig ausgeblendet (versteckt). Dies wird häufig für die Namen temporärer Dateien oder für lokale Konfigurationsdateien im Home-Verzeichnis eines Benutzers genutzt. Tatsächliches, effektives Verstecken ist auf diese Weise nicht möglich; schon die `ls`-Option `-a` zeigt die »versteckten« Dateien wieder an. Wenn Sie wirklich etwas zu verbergen haben, ist der korrekte Umgang mit Zugriffsrechten gefordert; diese werden weiter unten erläutert.

Intern werden Dateien auf dem Datenträger übrigens nicht durch ihren Namen dargestellt, sondern durch eine ganzzahlige Nummer namens *inode*. Die Dateinamen, das heißt die Einträge in einem Verzeichnis, sind Verweise auf solche *inodes*. Interessanterweise können mehrere Verzeichniseinträge auf dieselbe *inode* zeigen. Ein Verzeichniseintrag wird deshalb auch als *Hard Link* bezeichnet, der fest auf eine bestimmte *inode* verweist. Eine Datei wird auf einem UNIX-System erst gelöscht, wenn Sie *alle* Einträge im Verzeichnisbaum entfernt haben, die auf die entsprechende *inode* zeigen. Daher heißt der Systemaufruf zum Löschen einer Datei (der in vielen Programmiersprachen als Befehl zur Verfügung gestellt wird) auch `unlink` und nicht etwa »delete« oder »remove«.

Im Gegensatz zu den Hard Links werden auch *symbolische Links* oder *Symlinks* unterstützt, die nicht direkt auf eine *inode* zeigen, sondern auf einen anderen Verzeichniseintrag. Anders als die Hard Links können Symlinks auch auf Verzeichnisse verweisen sowie auf Dateien, die in einer anderen Partition oder gar auf einem anderen physikalischen Datenträger liegen.

Die verschiedenen Datenträger und Partitionen können, wie bereits erwähnt, an einer beliebigen Stelle im Verzeichnisbaum eingehängt werden. Dieser Vorgang wird als *Mounten* bezeichnet. Solange ein Datenträger nicht gemountet ist, können die Verzeichnisse und Dateien, die darauf liegen, nicht angesprochen werden. Angenommen, Sie legen eine CD-ROM ein, die eine Datei namens *hallo.txt* in einem Verzeichnis namens *test* enthält. Nach dem Einlegen muss die CD gemountet werden. Bei modernen Linux-Distributionen ist das DVD- oder CD-ROM-Laufwerk für automatisches Mounten konfiguriert; in Ihrer bevorzugten grafischen Oberfläche erhalten Sie dazu meist ein Dialogfeld, das nachfragt, ob und wohin der Datenträger gemountet werden soll (siehe Kapitel 5). Für USB-Sticks oder externen Festplatten gilt in der Regel dasselbe.

Wenn das automatische Mounten nicht aktiviert ist, müssen Sie den Datenträger von Hand mounten. Die Syntax des dafür zuständigen Befehls `mount` wird weiter unten genauer erläutert, im Moment genügt die Angabe, wie Sie im geschilderten Fall ein Verzeichnis namens `cd` anlegen (falls es noch nicht existiert) und die CD-ROM dorthin mounten können. Geben Sie dazu an der Konsole Folgendes ein:

```
$ mkdir /cd
$ mount /dev/hdc /cd
```

Der erste Parameter von `mount` gibt das physikalische Gerät an, das gemountet werden soll – S-ATA und EIDE-Geräte werden unter Linux als `/dev/hda`, `/dev/hdb` und so weiter geführt. `hdc` könnte somit beispielsweise der Secondary Master an einer EIDE/ATAPI-Schnittstelle sein. Wenn Sie nun die genannte Datei `hallo.txt` ansprechen möchten, die sich im Verzeichnis `test` befindet, dann lautet der entsprechende Pfad `/cd/test/hallo.txt`.

4.1.2 Zugriffsrechte

Eine weitere wichtige Eigenschaft der Dateien unter UNIX sind die Benutzerrechte. Jede Datei gehört einem bestimmten Benutzer und einer bestimmten Gruppe; berechnete Benutzer können diese Besitzverhältnisse ändern. Da ein Benutzer beliebig vielen Gruppen angehören kann, lassen sich die Rechte an bestimmten Dateien sehr effizient über das Gruppenzugriffsrecht ändern.

Der Verzeichniseintrag einer Datei enthält die Zugriffsrechte für den Besitzer, für die Gruppe und für alle anderen Benutzer. Die drei möglichen Zugriffsrechte sind Lesen (`r` für read), Schreiben (`w` für write) und Ausführen (`x` für execute). Ein typischer Verzeichniseintrag enthält beispielsweise die folgende Angabe von Zugriffsrechten:

```
-rwxr-xr-x
```

Die erste Stelle gibt den Dateityp an: – für eine gewöhnliche Datei, `d` für ein Verzeichnis oder `l` für einen Symlink. Die neun nachfolgenden Stellen zeigen in Dreiergruppen die Zugriffsrechte an. Ein Buchstabe steht dafür, dass ein Zugriffsrecht gewährt wird, ein Strich bedeutet dagegen, dass es nicht gewährt wird. Im vorliegenden Fall darf der Eigentümer (die ersten drei der neun Zeichen) die Datei lesen, schreiben (dazu gehören in der Regel auch Löschen und Umbenennen) und ausführen. Die Gruppe (die nächsten drei Zeichen) und der Rest der Welt (die letzten drei) dürfen nur lesen und ausführen, aber keinerlei Änderungen durchführen. Das Recht der Ausführung ist nur für Programme und für Verzeichnisse sinnvoll. Letztere lassen sich zwar natürlich nicht wie ein Programm

ausführen, aber wenn das `x` nicht gesetzt ist, können sie nicht als Arbeitsverzeichnis ausgewählt werden.

Intern werden die Zugriffsrechte übrigens als dreistellige Oktalzahl gespeichert. Die erste Stelle enthält die Benutzerrechte des Eigentümers, die zweite die der Gruppe und die dritte die der anderen Benutzer. Der Wert jeder Stelle ist die Summe aus den gewährten Benutzerrechten: 4 steht für Lesen, 2 für Schreiben und 1 für Ausführen. Das Zugriffsrecht `rwxr-xr-x` lässt sich also als `0755` darstellen. Die führende 0 kennzeichnet im Allgemeinen eigentlich eine Oktalzahl. Allerdings kann hier in bestimmten Fällen auch eine andere Zahl als 0 stehen, was auf bestimmte erweiterte Einstellungen hinweist, die weiter unten in diesem Kapitel erläutert werden. Eine einfache Textdatei könnte dagegen beispielsweise die Zugriffsrechte `0640` aufweisen, was `rw-r-----` entspricht – der Eigentümer darf die Datei lesen und schreiben, die Gruppe darf sie lesen und alle anderen dürfen gar nichts.

Neben diesen Grundzugriffsrechten existieren noch einige Spezialrechte, die das konkrete Verhalten mancher Rechte modifizieren. Sie werden statt der 0 an die erste Stelle der Oktalzahl geschrieben. Näheres erfahren Sie weiter unten in diesem Kapitel im Rahmen der Diskussion über das Kommando `chmod`, mit dem die Rechte modifiziert werden.

4.2 Die Shell aufrufen und anwenden

In diesem Abschnitt erhalten Sie einen Überblick über die Verwendung der Konsole: Virtuelle Terminals und GUI-Terminalfenster, Eingabeaufforderung und grundlegende Funktionsweise der Shell.

4.2.1 Virtuelle Terminals und Terminalfenster

Eine openSUSE-Standardinstallation startet mit der grafischen Oberfläche, sodass Sie sich nach der Anmeldung auf dem KDE- oder GNOME-Desktop, einer virtuellen Schreibtischoberfläche, befinden. Hier finden Sie verschiedene Symbole, Menüs und andere Bedienelemente, die in Kapitel 5 behandelt werden. Wechseln Sie an dieser Stelle zunächst in eine Textmodus-Konsole, indem Sie die Tastenkombination `[Strg]+[Alt]+[F1]` betätigen.

Jedes Linux-System bietet mehrere virtuelle Terminals zum Arbeiten an, in jedem von Ihnen können Sie sich unter einem beliebigen Benutzernamen anmelden und jeweils andere Programme ausführen. Standardmäßig sind sechs virtuelle Terminals eingerichtet, die über `[Strg]+[F1]` bis `[Strg]+[F6]` aufgerufen werden können.

Mit `Strg` + `F7` wechseln Sie dagegen wieder zur grafischen Oberfläche, falls sie gestartet wurde. Aus der GUI heraus müssen Sie zusätzlich die `Alt`-Taste festhalten, um wieder in eines der Text-Terminals zu wechseln.

Alternativ können Sie auch innerhalb der grafischen Oberfläche ein Terminalfenster öffnen, beispielsweise das Programm `xterm` oder eine modernere, komfortablere Variante. Halten Sie einfach Ausschau nach einem Icon, das einen schwarzen Textbildschirm zeigt. Das KDE-Standard-Terminalprogramm heißt *Konsole* und wird über `SUSE-MENÜ • SYSTEM • TERMINALS • KONSOLE` gestartet; zusätzlich ist auch das KDE-Panel am unteren Bildschirmrand standardmäßig mit einer Verknüpfung darauf ausgestattet. GNOME besitzt ebenfalls eine eigene Terminalemulation namens *GNOME Terminal*. Die wichtigsten Besonderheiten von KDE-Konsole und GNOME Terminal werden im nächsten Kapitel erläutert; im Moment genügt die Information, dass Sie darin genau wie in den virtuellen Terminals Shell-Befehle eingeben können.

In einem virtuellen Terminal müssen Sie zunächst den Login durchführen (siehe Kapitel 3, »Systemrundgang und Grundkonfiguration«); in einem GUI-Terminalfenster sind Sie dagegen automatisch mit der User-ID angemeldet, unter der die grafische Oberfläche selbst läuft. Sobald Sie angemeldet sind, erhalten Sie eine *Eingabeaufforderung* (englisch `prompt`). Der Prompt kann je nach Konfiguration sehr unterschiedlich aussehen. In der Regel sehen Sie etwa Folgendes:

```
user@rechner: ~ >
```

Statt `user` wird der Benutzername angezeigt, unter dem Sie sich angemeldet haben; hinter dem `@` steht der Name des Rechners, auf dem Sie gerade arbeiten. Auf diese Angaben folgt der Pfad des aktuellen Arbeitsverzeichnisses, im obigen Beispiel befindet sich der Benutzer in seinem Home-Verzeichnis (auf dieses Beispiel bezogen `/home/user`), das durch die Tilde gekennzeichnet wird. Das Kleinerzeichen bildet schließlich den Abschluss; dahinter blinkt der Cursor für die Befehlseingabe. Statt des Kleinerzeichens erscheint bei manchen Shells ein Dollarzeichen (`$`) oder ein anderes Zeichen.

Wenn Sie als `root` angemeldet sind, bekommen Sie einen etwas anderen Prompt zu sehen, beispielsweise folgenden:

```
rechner: ~ #
```

Es wird also kein Benutzername angezeigt, und hinter der Pfadangabe folgt eine Raute (`#`) statt des Dollar- oder Kleinerzeichens. Auch `root` befindet sich in diesem Beispiel in seinem Home-Verzeichnis, standardmäßig `/root`.

In den Beispielen in diesem Kapitel und im Rest dieses Buches wird der Prompt in der Regel einfach als Dollarzeichen dargestellt. Wenn für einen Befehl root-Rechte erforderlich sind, wird dagegen die Raute verwendet. Eine Ausnahme bilden nur Kommandos, die den Prompt ändern oder in denen sein Inhalt eine besondere Bedeutung hat. Benutzereingaben sind in den Beispielen jeweils fett gesetzt, um sie vom Prompt und von der Ausgabe des Systems abzusetzen.

Die Konfiguration, mit der die Shell (und übrigens auch jedes andere Programm) ausgeführt wird, heißt *Umgebung* (englisch Environment). Sie besteht aus der User- und Group-ID, unter der das Programm läuft, aus dem aktuellen Arbeitsverzeichnis sowie aus einer Reihe von *Umgebungsvariablen*, die von dem Programm ausgelesen werden. Die Shell bezieht ihre Umgebung aus diversen Konfigurationsdateien, insbesondere:

- ▶ */etc/profile*: zentrale Konfigurationsdatei für alle Shells und alle User (diese Datei sollte nicht editiert werden; ändern Sie stattdessen *~/.bashrc* oder erstellen Sie eine benutzerspezifische *~/profile.local*)
- ▶ */etc/profile.d/**: zentrale Konfigurationsdateien für bestimmte Aspekte einzelner Shells
- ▶ *~/.bashrc*: bash-spezifische Einstellungen für einen einzelnen Benutzer in dessen Home-Verzeichnis

Sobald die Eingabeaufforderung angezeigt wird, können Sie Befehle eintippen, die durch  abgeschlossen werden. Einige Kommandos sind direkt in die Shell eingebaut; geben Sie als erstes Beispiel Folgendes ein:

```
$ alias
```

Sie erhalten eine Liste der in der aktuellen Shell definierten Aliase, das heißt Kommandoabkürzungen. In openSUSE enthält diese Liste beispielsweise – unter anderem – folgende Zeilen:

```
alias l='ls -lF'
alias la='ls -la'
alias ll='ls -l'
```

Es genügt also, den Buchstaben `l` und ein anschließendes  einzugeben, um eine vollständige und mit Detailinformationen versehene Liste des aktuellen Verzeichnisinhalts zu erhalten. (Wie Sie Aliase selbst definieren, steht in Kapitel 17, »System-Automatisierung«.)

Die meisten Befehle, die Sie eintippen können, sind dagegen externe Dienstprogramme. Geben Sie etwa die folgende Anweisung ein, um das Handbuch (Manual – siehe voriges Kapitel) der `bash` zu lesen:

```
$ man bash
```

Innerhalb des Programms `man` können Sie mit den Pfeiltasten oder der Leertaste blättern; die Taste `Q` beendet es.

4.2.2 Bedeutung der Shell

Das Programm, das innerhalb des virtuellen Terminals oder der Terminalemulation Ihre Befehle entgegennimmt und zu interpretieren versucht, wird *Shell* genannt. Es gibt nicht *die* Linux-Shell, sondern eine Reihe verschiedener Shell-Programme, die sich bis zu einem gewissen Grad voneinander unterscheiden. Auf Linux-Systemen wie openSUSE wird standardmäßig die Shell `bash` ausgeführt. Geben Sie den folgenden Befehl ein, um herauszufinden, welche Shell Sie gerade verwenden:

```
$ echo $0
```

`$0` ist eine spezielle Systemvariable, die jeweils den vollständigen Pfad des zurzeit laufenden Programms enthält. Die Ausgabe dürfte zum Beispiel `/bin/bash` lauten.

Die gängigsten Shells werden in der folgenden Liste aufgeführt:

- ▶ `sh` oder `bsh`, die *Bourne Shell*, benannt nach ihrem Entwickler, war die ursprüngliche Shell des Bell-Labs-UNIX. Sie ist der kleinste gemeinsame Nenner der Fähigkeiten aller anderen Shells.
- ▶ `csh`, die *C-Shell*, und ihre Erweiterung `tcs` enthalten eine Reihe spezieller Funktionen, die besonders den Bedürfnissen von C-Programmierern entgegenkommen. Es handelt sich um die Shell-Eigenentwicklung für das ursprüngliche BSD UNIX.
- ▶ `bash`, die *Bourne Again Shell* (ein nettes Wortspiel) ist die GNU-Weiterentwicklung der ursprünglichen Bourne Shell mit vielen interessanten Zusatzfunktionen. Diese Shell ist in allen Linux-Distributionen als Standard voreingestellt. Trotzdem werden bei openSUSE alle hier Genannten und meist noch weitere mitgeliefert. Sofern nicht anders gekennzeichnet, gelten die Informationen in diesem Kapitel und im Rest dieses Buches für die `bash`.
- ▶ `ksh`, die *Korn Shell*, ist eine andere Neuimplementierung der Bourne-Shell. Sie ist Standard auf vielen kommerziellen UNIX-Systemen. Da die eigentliche Korn Shell proprietär ist, gibt es für freie Systeme die Variante `pdksh` (Public Domain Korn Shell).
- ▶ `sash`, die *Stand-Alone Shell*, ist vor allem ein spezielles Hilfsmittel für Wartungsarbeiten: Zusätzlich zu den üblichen Aufgaben einer Shell, die hier übri-

gens weniger komfortabel als bei *bash* oder *tcsh* erledigt werden, sind eine Reihe von Standard-POSIX-Dienstprogrammen direkt in das Shell-Binary eingebaut. Um sie von den gleichnamigen Systemprogrammen zu unterscheiden, beginnen ihre Namen in der Regel mit einem Minuszeichen, und sie besitzen nicht den vollen Leistungsumfang der GNU-Version dieser Utilities. Für Not-systeme, die vom USB-Stick oder gar (sofern noch vorhanden) von Diskette booten, ist die *sash* allerdings oftmals ideal.

Um Missverständnissen vorzubeugen, sollten Sie zunächst verstehen, dass über 90% der Eingaben, die Sie an der Kommandozeile vornehmen, unter allen Shells identisch sind – es handelt sich bei diesen Eingaben nämlich überhaupt nicht um eingebaute Shell-Kommandos (die bereits erwähnte *sash* bildet hier natürlich eine Ausnahme). Die meisten »UNIX-Befehle« sind separate Systemprogramme, die sich für gewöhnlich im Verzeichnis */bin* befinden und mit der Shell selbst nichts zu tun haben. Die Shells unterscheiden sich insbesondere in der Art und Weise, wie Sie die Funktionen der Systemprogramme durch intelligente Verknüpfungen erweitern können.

4.2.3 Grundfunktionen der Shell

In diesem Abschnitt geht es um die allgemeine Funktionsweise der Shell als Kommando-Interpreter. Falls nicht anders gekennzeichnet, gelten die Informationen für die *bash*; viele von ihnen lassen sich aber auch in anderen Shells einsetzen.

Befehlseingabe

In der Regel bestehen die Befehle, die Sie eingeben, aus dem Namen des gewünschten Systemprogramms und einer durch Leerzeichen getrennten Liste von Parametern. Einige der Parameter sind Optionen, die bei den meisten Befehlen mit einem Minuszeichen beginnen, andere geben dagegen konkrete Werte wie Pfad- oder Dateinamen, Bezeichnungen und Ähnliches an.

Jede Eingabe einer Anweisung wird durch  abgeschlossen und daraufhin sofort verarbeitet; einige Ausnahmen zur Verknüpfung mehrerer Anweisungen lernen Sie in Abschnitt 4.2.4, »Ein-/Ausgabeumleitung und Pipes«, kennen. Sollte eine Eingabe zu lang werden, können Sie jederzeit einen Backslash (\) und  eingeben, um in der nächsten Textzeile weiterzuschreiben. Das folgende Beispiel gibt nacheinander den Inhalt von drei Textdateien aus:

```
$ cat langer_dateiname1.txt langer_dateiname2.txt \  
> langer_dateiname3.txt
```

Wenn Sie eine Anweisung eingeben, sucht die Shell in folgender Reihenfolge nach einer passenden Ausführungsmöglichkeit:

1. Eigene `alias`-Definitionen
2. In das Shell-Programm selbst eingebaute Kommandos (Shell-Builtins)
3. Externe Programme – sie werden in der angegebenen Reihenfolge im *Suchpfad* (dem Inhalt der Umgebungsvariablen `PATH`) gesucht.

Um zu ermitteln, ob eine Anweisung ein Alias, ein Shell-Builtin oder ein externes Programm ist, können Sie das Kommando `type` verwenden.

Beispiele:

```
$ type ls
ls is aliased to `/bin/ls $LS_OPTIONS'
$ type alias
alias is a shell builtin
$ type mkdir
mkdir is hashed (/bin/mkdir)
```

Wenn Sie sich den aktuellen Suchpfad anzeigen lassen möchten, können Sie Folgendes eingeben:

```
$ echo $PATH
/bin:/usr/bin:/usr/sbin:/usr/local/bin:/usr/share/bin
```

Der Befehl `echo` gibt sämtlichen in der Befehlszeile enthaltenen Text aus. Besonders nützlich ist er wie hier für die Ausgabe von Variablen: Das Dollarzeichen sorgt dafür, dass die Shell das nachfolgende Wort als den Namen einer Variablen auffasst, deren Wert ausgegeben werden soll.

Der Wert der Variablen `PATH` besteht aus einer Liste von absoluten Pfadangaben (mit `/` beginnend), die durch Doppelpunkte voneinander getrennt werden. Normalerweise ist die Liste erheblich länger.

Die Verzeichnisse in `PATH` werden der Reihe nach durchprobiert, bis ein Programm mit dem angeforderten Namen gefunden wird. Wenn es nirgendwo gefunden wird, erscheint eine Fehlermeldung.

Hinweis

Falls Ihnen die Windows-Eingabeaufforderung vertraut ist, gibt es einen wichtigen Unterschied zu beachten: Das aktuelle Arbeitsverzeichnis gehört *nicht* explizit zum Suchpfad! Insbesondere beim Test selbst geschriebener Programme oder Skripte ist daher zu beachten, dass Sie dem Kommando ein `./` voranstellen müssen, wenn es im aktuellen, nicht zu `PATH` gehörenden Verzeichnis liegt. Aus Sicherheitsgründen ist dies auch viel besser so, denn auf diese Weise kann es nicht passieren, dass statt eines Systemprogramms versehentlich ein gleichnamiges (und womöglich schädliches) Programm im Arbeitsverzeichnis ausgeführt wird.

Hier zur Verdeutlichung ein Beispiel, das das (fiktive) Ruby-Skript *test.rb* im aktuellen Arbeitsverzeichnis startet:

```
$ ./test.rb
```

Hintergrundprozesse

Eine angenehme Funktion der Shell besteht darin, Programme im Hintergrund zu starten – wenn Sie ein `&`-Zeichen an einen Befehl anhängen, gelangt dessen Standardausgabe nicht auf den Bildschirm, und Sie können sofort den nächsten Befehl eingeben. Es wird beim Aufruf des Befehls lediglich dessen Prozess-ID ausgegeben. Inzwischen bieten fast alle Shells diese Option. Hier sehen Sie ein einfaches Beispiel, in dem die Suche nach Dateien, deren Name mit einem `a` beginnt, in den Hintergrund verbannt wird:

```
$ find . -name "a*" &
[1] 6172
$
```

Die Funktion des Befehls `find` wird weiter unten genauer erläutert.

In eckigen Klammern wird eine Job-Nummer angezeigt; dahinter erscheint die PID. Statt 6172 werden Sie höchstwahrscheinlich eine andere zu sehen bekommen. Mithilfe des Befehls `fg` (für *Foreground*) können Sie die Ausgabe des Befehls im Vordergrund fortsetzen:

```
$ fg
```

Falls mehrere Prozesse in den Hintergrund gestellt wurden, müssen Sie die Job-ID als Parameter angeben. Beispiel:

```
$ fg 1
```

Ebenso können Sie ein bereits laufendes Programm nachträglich in den Hintergrund stellen, indem Sie die Tastenkombination `[Strg]+[Z]` betätigen. Auch in diesem Fall werden Job-Nummer und PID angezeigt, und Sie können das Programm mit `fg` zurückholen.

Besonders nützlich ist dieses Feature für den manuellen Start langlebiger Daemon-Prozesse, also beispielsweise Netzwerk-Servern. Das folgende Beispiel ruft das Standard-Startskript des Datenbankservers MySQL (siehe Kapitel 12, »Der Datenbankserver MySQL«) im Hintergrund auf:

```
$ mysqld_safe &
```

Schließlich dient die Hintergrundfunktion auch dazu, Programme mit grafischer Oberfläche aus der Konsole zu starten, ohne Letztere zu blockieren. Hier ein Bei-

spiel für den Start des Bildbearbeitungsprogramms GIMP (Kapitel 8, »Desktop-Anwendungen«):

```
$ gimp &
```

Wenn Sie herausfinden möchten, welche Programme im Hintergrund ausgeführt werden und wie ihr aktueller Status ist, können Sie das (Shell-interne) Kommando `jobs` verwenden:

```
$ jobs
[1]  Done          find . -name "a*"
[2]  Running       /usr/local/mysql/bin/mysqld_safe
```

Der Status `Done` kennzeichnet einen Job, dessen Ausführung ordnungsgemäß beendet wurde. `Running` steht für einen Job, der noch ausgeführt wird. Ein weiterer gängiger Status ist `Stopped` für Prozesse, die durch den Benutzer angehalten wurden.

Eingabevervollständigung und History

Alle modernen UNIX-Shells beherrschen die sehr bequeme Funktion der *Eingabevervollständigung* (command line completion): Wenn Sie einen Befehl oder den Pfad einer Datei eintippen, können Sie zwischenzeitlich die -Taste betätigen. Wenn der Befehl oder Pfad zu diesem Zeitpunkt bereits eindeutig ist, also nur noch eine Interpretation zulässt, wird er komplett ausgeschrieben. Bei Zweideutigkeiten wird er nur zum Teil ergänzt, und es ertönt ein Warnton. Das folgende Beispiel zeigt, wie Sie aus Ihrem Home-Verzeichnis schnell in das darunter liegende Verzeichnis *dokumente* wechseln können:

```
user@rechner: ~ $ cd do
user@rechner: ~/dokumente $
```

Angenommen, in Ihrem Home-Verzeichnis befindet sich ein weiteres Verzeichnis namens *dokumente2*. In diesem Fall wird durch  zwar das Wort *dokumente* ergänzt, aber die Shell weiß noch nicht, ob Sie wirklich das Verzeichnis *dokumente* meinten oder *dokumente2*. Deshalb wird in vielen Terminalemulationen ein Warnton ausgegeben.

Wenn Sie ein zweites Mal  drücken, erhalten Sie eine Liste der möglichen Alternativen. Angenommen, Sie möchten sich den Inhalt von Dateien oder Verzeichnissen unter dem Konfigurationsverzeichnis */etc* anzeigen lassen, die mit `a` anfangen. Geben Sie also Folgendes ein:

```
$ cat /etc/a 
```

Auf meinem System erhalte ich daraufhin folgende Liste:

```
a2ps.cfg      aliases      apparmor/    autoinstall/
a2ps-site.cfg aliases.d/   armagetronad/ auto.master
aclocal_dirlist aliases.db   asound.state auto.misc
acpi/         alsad/      at.deny      auto.net
adjtime       alternatives/ auditd.conf  auto.smb
aide.conf     apache2/    audit.rules
```

Je nach installierten Paketen dürfte es bei Ihnen ähnlich aussehen. Unter der Liste erscheint wieder der Prompt mit der bisherigen Teileingabe.

Sollte es zu viele Alternativen geben, fragt die Shell vorsichtshalber nach. Geben Sie zum Testen einfach einen einzelnen Kleinbuchstaben, gefolgt von zweimal  ein. Hier als Beispiel das 1:

```
$ 1  
Display all 151 possibilities? (y or n)
```

Drücken Sie die entsprechende Taste, um sich die umfangreiche Tabelle anzeigen zu lassen oder auch nicht.

Ähnlich komfortabel ist die *History* aller bereits eingegebenen Befehle. Mit den Pfeiltasten auf der Tastatur können Sie darin nach oben oder nach unten blättern; die früher beziehungsweise später eingegebenen Befehle werden dadurch wieder angezeigt. Wenn der gewünschte Befehl erscheint, können Sie ihn ändern und anschließend mittels  ausführen.

Bei modernen *bash*-Versionen funktioniert die *History* sogar sitzungsübergreifend: Sobald Sie sich abmelden oder ein Terminalfenster schließen, wird der Inhalt der aktuellen *History* an die Datei *.bash_history* in Ihrem Home-Verzeichnis angehängt. Beim nächsten Login beziehungsweise beim Start einer Terminal-emulation wird der Inhalt der Datei dann wieder als *History* geladen.

Weitere nützliche Tastenkürzel

Noch komfortabler wird die *bash* dadurch, dass sie diverse nützliche Tastenkürzel für erweiterte Funktionen unterstützt. Viele dieser Tastenkombinationen sind von dem bekannten Editor Emacs inspiriert (siehe Kapitel 10, »Klassische Texteditoren«), denn die *bash* befindet sich standardmäßig im Emacs-Modus. Daneben gibt es auch noch einen vi-Modus für die Kompatibilität zu einem anderen weitverbreiteten, ebenfalls in Kapitel 10 beschriebenen Editor. Hier soll allerdings die Beschreibung des Emacs-Modus genügen.

Mit den Tasten  und  können Sie sich um je ein Zeichen nach links oder rechts bewegen; synonym sind die Tastenkürzel + (backward) bezie-

hungsweise `Strg+F` (forward). `Alt+B` bewegt sich um ein ganzes Wort zurück (bis zum nächsten Leerzeichen); `Alt+F` um ein Wort weiter. `Strg+A` springt zum Zeilenanfang, `Strg+E` zum Ende. Auf den meisten Rechnern können Sie dafür auch `Pos1` beziehungsweise `Ende` verwenden.

Mit `Entf` oder `Strg+D` können Sie das Zeichen unter dem Cursor löschen, während `Backspace` beziehungsweise `Strg+H` das Zeichen links vom Cursor entfernt. `Strg+W` löscht ein ganzes Wort. `Strg+K` entfernt alles von der aktuellen Position bis zum Zeilenende, während `Strg+U` die gesamte Eingabezeile löscht.

Für das Durchblättern der History gibt es neben den bereits erwähnten Tasten `↑` und `↓` die Alternativen `Strg+P` (previous) und `Strg+N` (next). Mit `Strg+R` leiten Sie die inkrementelle Suche nach einem Befehl in der History ein, das heißt, jedes weitere Zeichen, das Sie eintippen, führt zum nächstgelegenen Treffer. Drücken Sie gegebenenfalls mehrmals `Strg+R`, um Treffer für die aktuelle Zeichenfolge zu finden, die in der History weiter zurückliegen.

Während GUI-Terminalprogramme in der Regel Rollbalken besitzen, um zu den bereits nach oben weggeschrollten Zeilen des Terminalinhalts zu gelangen, können Sie diesen Puffer in Textterminals mittels `⌘+Bild↑` und `⌘+Bild↓` durchblättern.

Mithilfe von `Strg+L` können Sie den Bildschirm beziehungsweise den Terminalfensterinhalt löschen (genauer gesagt hochscrollen, bis sich der Prompt ganz oben in einem leeren Terminal befindet); viel schneller als mit dem Kommando `clear`.

Auswählen und Kopieren mit der Maus

Eine überaus praktische Funktion moderner Terminalemulationen besteht darin, dass Sie Inhalte mit der Maus markieren und einfügen können. Ziehen Sie dazu einfach mit der linken Maustaste über den gewünschten Textblock. Ein Klick mit der mittleren Maustaste fügt ihn dann an der aktuellen Textcursorposition ein. Als mittlere Maustaste zählt gegebenenfalls auch ein Scrollrädchen. Ist auch dieses nicht vorhanden, können Sie bei einer Standardkonfiguration mit beiden Maustasten gleichzeitig klicken.

Beachten Sie, dass auch Zeilenumbrüche aus dem Auswahltext an der Cursorposition eingefügt werden, sodass eventuelle Kommandos sofort ausgeführt werden. Sollte Text, der mit einem Zeilenumbruch abschließt, keine gültige Anweisung sein, erhalten Sie natürlich auch eine entsprechende Fehlermeldung.

In einer grafischen Benutzeroberfläche unterstützen übrigens auch andere Programme als die Shell dieses Verfahren.

4.2.4 Ein-/Ausgabeumleitung und Pipes

Eine der praktischsten Eigenschaften der UNIX-Shells besteht in der Umleitung von Ein- und Ausgabe sowie ihrer Verkettung. Mit der Ausgabe eines Befehls können Sie mehr tun, als sie einfach auf dem Bildschirm darzustellen, und die Eingabe muss nicht unbedingt von der Tastatur stammen: Sie können die Eingabe für einen Befehl aus einer Datei holen, die Ausgabe in eine Datei schreiben und schließlich die Ausgabe des einen Befehls als Eingabe für den nächsten verwenden. Auf diese Weise können Sie die einfachen Bausteine der Systembefehle zur Erledigung komplexer Aufgaben einsetzen.

Ein- und Ausgabeumleitung

Wie so viele Aspekte von Linux und anderen UNIX-Systemen werden die Ein- und Ausgabefunktionen durch die Standardbibliothek der Programmiersprache C bestimmt. Diese kennt drei Standardkanäle (streams) zur Ein- und Ausgabe (englisch Input/Output oder kurz I/O):

- ▶ *stdin* ist die *Standardeingabe* (standard input). Sie besitzt standardmäßig die Datei- oder Kanalnummer 0 und ist normalerweise mit der Tastatur verknüpft.
- ▶ *stdout*, die *Standardausgabe* (standard output), wird per Voreinstellung auf die Konsole geleitet, das heißt, sie landet bei Vordergrundprogrammen zeilenweise im aktuellen Terminal. Ihre I/O-Kanalnummer ist in der Regel 1.
- ▶ *stderr* schließlich ist die *Standard-Fehlerausgabe* (standard error) – sie besitzt normalerweise die Kanalnummer 2 und landet für gewöhnlich ebenfalls auf der Konsole. Vorteil: Wenn Sie *stdout* in eine Datei umleiten, werden Fehler-, Warn- oder Debug-Meldungen noch immer angezeigt.

Die Ein- und Ausgabeumleitung in Linux basiert auf einer Verknüpfung von *stdin*, *stdout* oder *stderr* mit anderen Dateien oder Geräten.

Hier zunächst ein Beispiel für *Ausgabeumleitung*: Der Befehl `ls` dient dazu, den Inhalt des aktuellen Verzeichnisses auszugeben. Wenn Sie diesen Inhalt lieber in eine Datei als auf die Konsole schreiben möchten, können Sie folgendermaßen vorgehen:

```
$ ls >inhalt.txt
```

In diesem einfachen Beispiel wird der Inhalt des aktuellen Verzeichnisses nicht auf den Bildschirm geschrieben, sondern in die Datei *inhalt.txt*. Diese Datei wird automatisch neu angelegt, falls sie noch nicht existierte, ansonsten wird sie überschrieben. Wenn Sie die Ausgabe eines Befehls stattdessen an eine bestehende Datei anhängen möchten, können Sie statt des einen `>`-Zeichens zwei verwenden:

```
$ ls >>inhalt.txt
```

Auch in diesem Fall wird die Datei neu erzeugt, falls es sie noch nicht gab. Andernfalls erfolgt die Ausgabe unter die bisherige letzte Zeile.

Auf ähnliche Weise können Sie die Eingabe für einen Befehl aus einer Datei lesen. Zum Beispiel gibt der Befehl `grep` alle Zeilen eines eingegebenen Textes zurück, in dem ein Suchmuster (genauer gesagt ein regulärer Ausdruck, siehe unten) vorkommt. Wenn Sie alle Zeilen der Datei *inhalt.txt* suchen möchten, die mindestens ein `a` enthalten, funktioniert das folgendermaßen:

```
$ grep a <inhalt.txt
```

(Eigentlich können Sie das `-`-Zeichen in diesem Fall weglassen, da `grep` das letzte Kommandozeilenargument ohnehin als Dateinamen oder Pfad interpretiert.)

Eine interessante Variante der Eingabeumleitung ist das *HIER-Dokument* (englisch *HERE Document*). Diese Art der Eingabe stammt nicht aus einer Datei, sondern nimmt alle eingegebenen Zeilen bis zu einer speziellen Markierung entgegen. Das folgende Beispiel sucht mittels `grep` nach allen Zeilen in der Eingabe, die mindestens ein `a` enthalten:

```
$ grep a <<ENDE
> Hallo
> liebe
> Welt
> ENDE
```

Die Ausgabe dieser eingegebenen Sequenz lautet folgendermaßen:

```
Hallo
```

Die Markierung `ENDE` bildet das Ende der Eingabe; sie muss allein ohne umgebende Leerzeichen in einer Zeile stehen.

Auch `stderr` lässt sich übrigens umleiten; dazu wird vor das Umleitungszeichen `>` beziehungsweise `>>` die Kanalnummer `2` gesetzt. Das folgende Beispiel sucht im aktuellen Verzeichnis und seinen Unterverzeichnissen nach Dateien, deren Namen mit `b` beginnen, und belässt dabei die Standardausgabe auf dem Bildschirm, während Fehlermeldungen an die Datei *fehler.txt* angehängt werden:

```
$ find . -name "b*" 2>>fehler.txt
```

Auf diese Weise erhalten Sie nur noch die eventuellen Suchergebnisse, aber keine Meldungen wie »Keine Berechtigung« mehr, wenn `find` ein bestimmtes Verzeichnis mit den Rechten des aktuellen Benutzers gar nicht lesen darf. So etwas kann sehr wichtig sein, wenn Sie die Ausgabe des Befehls automatisiert in

einem Shellskript einsetzen möchten, denn übliche Weiterverarbeitungsschritte können natürlich nur mit der regulären Ausgabe etwas anfangen. Auch bei Hintergrundprozessen ist es sinnvoll, *stderr* umzuleiten, da Fehler- und Warnmeldungen sonst – anders als die Ausgabe von *stdout* – doch noch auf der Konsole landen.

Wenn Sie gar keine Fehlermeldungen mehr erhalten möchten, dann können Sie die Fehlermeldungen auch nach */dev/null* – das *Nulldevice*, das UNIX-Äquivalent des absoluten Nichts – umleiten:

```
$ find . -name "b*" 2>/dev/null
```

Um Standardausgabe und Standardfehlerausgabe »über einen Kamm zu scheren«, können Sie die beiden schließlich auch zusammenlegen, das heißt, die eine in die andere umleiten. Die allgemeine Syntax dafür lautet *KanalNr>&KanalNr*. Falls Sie also die Meldungen der Standardfehlerausgabe in die Standardausgabe umleiten möchten, funktioniert das so:

```
$ find . -name "b*" 2>&1
```

Dies lässt sich wiederum einsetzen, um beides in dieselbe Datei umzuleiten:

```
$ find . -name "b*" >gefunden.txt 2>&1
```

Ein Testparcours für die Umleitung

Es zeigt sich immer wieder, dass Konzepte von Linux am besten verstanden werden, wenn die zugrunde liegenden Mechanismen anhand von Programmcode erläutert werden. Deshalb finden Sie an dieser Stelle ein kurzes C-Programm, das Sie selbst kompilieren und ausprobieren können. Geben Sie in Ihrem Home-Verzeichnis den nachfolgenden Code unter dem Dateinamen *outtest.c* in einem Editor (siehe Kapitel 10, »Klassische Texteditoren«) ein oder laden Sie sich die Datei *outtest.c* aus der Abteilung *Listings* von der Website zu diesem Buch herunter:

```
#include <stdio.h>

int main (int argc, char* argv[]) {
    int i;
    char line[255];
    for (i = 1; i < argc; i++) {
        printf ("Ausgabe Argument Nr. %d: %s\n", i, argv[i]);
        fprintf (stderr, "\"Fehler\" Argument Nr. %d: %s\n",
                i, argv[i]);
    }
    while (scanf ("%s", &line) != EOF) {
```

```

    printf ("Ausgabe Benutzereingabe: %s\n", line);
    fprintf (stderr, "\"Fehler\" Benutzereingabe: %s\n",
            line);
}
return 0;
}

```

Hier eine ganz kurze Zeile-für-Zeile-Beschreibung des Quellcodes:

- ▶ `#include <stdio.h>`
 Importiert die Header-Datei *stdio.h* (Standard Input/Output) aus der C-Standardbibliothek. Letztere befindet sich unter openSUSE wie auf jedem normalen Linux-System unter */usr/include*. Die Datei enthält Schnittstellendefinitionen für Ein-/Ausgabefunktionen in einer vorkompilierten Bibliothek, die beim Kompilieren Ihres Programms mit diesem verknüpft (gelinkt) wird. Die *stdio*-Funktionen, die in diesem Programm genutzt werden, sind `printf()`, `fprintf()` und `scanf()`.
- ▶ `int main (int argc, char* argv[]) {...}`
 Dies ist die Deklaration der Hauptfunktion `main()`. C-Programme besitzen kein offizielles Hauptprogramm. Stattdessen verwenden sie die Funktion `main()`, die beim Programmstart automatisch vom Betriebssystem aufgerufen wird. Sie besitzt üblicherweise den Datentyp `int` (Ganzzahl), weil erwartet wird, dass sie einen Wert an die aufrufende Umgebung zurückgibt: bei korrekter Ausführung 0, bei irgendeinem Fehler oder Problem einen anderen Wert. Zudem hat die Funktion zwei Argumente: `argc` ist die Anzahl der Kommandozeilenargumente, `argv[]` dagegen ein Array von Strings für den Inhalt der einzelnen Argumente selbst.
- ▶ `int i;`
 Diese Anweisung deklariert die ganzzahlige Variable `i`, die hier als Schleifen-zähler eingesetzt wird.
- ▶ `char line[255];`
 Dies deklariert eine String-Variable (in C ein Zeichen-Array) mit maximal 255 Zeichen zum Speichern von Benutzereingaben.
- ▶ `for (i = 1; i < argc; i++) {...}`
 Dies ist der Kopf einer Schleife mit determiniertem Ablauf: `i` erhält den Anfangswert 1 (`i = 1`); die Schleife läuft, solange `i` kleiner ist als die Anzahl der Argumente (`i < argc`); `i` wird in jedem Schritt um 1 erhöht (`i++`).
- ▶ `printf ("Ausgabe Argument Nr. %d: %s\n", i, argv[i]);`
 Innerhalb der Schleife gibt diese Anweisung das jeweilige Kommandozeilenargument mit seiner Nummer `i` und seinem Wert `argv[i]` aus. Die *stdio*-Funktion `printf()` dient der Ausgabe auf `stdout`. Ihr erstes Argument ist ein

String mit %-Formatplatzhaltern (zum Beispiel %d für eine Ganzzahl oder %s für einen String). Alle weiteren Argumente sind Ausdrücke, die der Reihe nach an den Platzhalterpositionen eingesetzt werden. Die Anweisung gibt also das jeweilige Kommandozeilenargument auf der Standardausgabe aus.

- ▶ `fprintf (stderr, "\"Fehler\" Argument Nr. %d: %s\n", i, argv[i]);`
`fprintf()` funktioniert im Wesentlichen wie `printf()`, außer dass vor dem Formatstring als allererstes Argument ein Dateideskriptor steht. Das ist die Nummer einer selbst geöffneten Datei oder – wie hier – ein Standardbezeichner für einen Ausgabe Kanal. Die Anweisung gibt dasselbe Kommandozeilenargument also ein zweites Mal auf `stderr` aus.
- ▶ `while (scanf ("%s", &line) != EOF) {...}`
 Diese Bedingung dieser Schleife liest einen String aus der Standardeingabe und vergleicht diesen mit EOF (End Of File, Dateiende). Der Schleifenrumpf, das heißt der Inhalt der geschweiften Klammern, wird nur ausgeführt, wenn die Eingabe ungleich (!=) EOF ist, das heißt, wenn noch Daten folgen. `scanf()` liest einen Wert mit dem angegebenen Format von der Standardeingabe, hier %s für einen String (jeweils durch Leerzeichen oder Zeilenumbruch getrennt). Gespeichert wird er unter der als Zweites angegebenen Speicheradresse, die im Fall einer Variablen über den Dereferenzierungs-Operator & ermittelt wird.
- ▶ `printf ("Ausgabe Benutzereingabe: %s\n", line);`
 Ähnlich wie zuvor die Kommandozeilenargumente wird hier die aktuelle Eingabezeile auf die Standardausgabe geschrieben.
- ▶ `fprintf (stderr, "\"Fehler\" Benutzereingabe: %s\n", line);`
 Auch auf die Standardfehlerausgabe wird die jeweilige Eingabe geschrieben.
- ▶ `return 0;`
 Wie bereits erwähnt, sollte die Funktion `main()` einen Wert zurückgeben. Da in diesem Programm nichts Unvorhergesehenes passieren kann (etwa ungültige Parameter), ist der Rückgabewert grundsätzlich 0 (»keine besonderen Vorkommnisse«).

Nachdem das Programm fertig eingetippt oder geladen ist, müssen Sie es kompilieren. Dazu muss – wie in Kapitel 2 ausdrücklich empfohlen – der C-Compiler GCC installiert sein. Geben Sie Folgendes ein, um aus dem Quellcode `outtest.c` das ausführbare Binärprogramm `outtest` zu machen:

```
$ gcc -o outtest outtest.c
```

Wenn Sie keine Ausgabe erhalten und wieder den Prompt sehen, haben Sie alles richtig gemacht, andernfalls gibt es Kompilierfehler, weil Sie sich vertippt haben. Korrigieren Sie in diesem Fall Ihre Fehler, und versuchen Sie es erneut.

Im Verzeichnis finden Sie nun die neue Datei *outtest*. Nun müssen Sie zu ihren Dateirechten »ausführbar« hinzufügen:

```
$ chmod a+x outtest
```

Wenn Sie das Programm jetzt mit beliebigen Parametern aufrufen, werden diese je zweimal angezeigt: einmal auf `stdout` und einmal auf `stderr`. Danach können Sie beliebig viele Zeilen eingeben; nach jedem `↵` erfolgt wieder die doppelte Ausgabe. Geben Sie zum Beenden einfach `Strg+D` ein, die Tastenkombination für EOF. Hier eine »Beispielsitzung«:

```
$ ./outtest Hallo Welt
Ausgabe Argument Nr. 1: Hallo
"Fehler" Argument Nr. 1: Hallo
Ausgabe Argument Nr. 2: Welt
"Fehler" Argument Nr. 2: Welt
Hallo
Ausgabe Benutzereingabe: Hallo
"Fehler" Benutzereingabe: Hallo
Leute
Ausgabe Benutzereingabe: Leute
"Fehler" Benutzereingabe: Leute
Strg+D
```

Nun können Sie wahlweise einen oder mehrere I/O-Kanäle umleiten. Wenn Sie beispielsweise die Standardausgabe nach `/dev/null` umleiten, sehen Sie nur noch die »Fehlermeldungen«:

```
$ ./outtest Hallo >/dev/null
"Fehler" Argument Nr. 1: Hallo
Welt
"Fehler" Benutzereingabe: Welt
Strg+D
```

Genauso gut können Sie die Eingabe umleiten oder sogar die Ein- und Ausgabe. Dieses Beispiel liest die Quellcodedatei selbst als Eingabe und leitet die Standardfehlerausgabe um:

```
$ ./outtest <outtest.c 2>/dev/null
Ausgabe Benutzereingabe: #include
Ausgabe Benutzereingabe: <stdio.h>
Ausgabe Benutzereingabe: int
Ausgabe Benutzereingabe: main
Ausgabe Benutzereingabe: (int
Ausgabe Benutzereingabe: argc,
Ausgabe Benutzereingabe: char*
```

```
Ausgabe Benutzereingabe: argv[])
Ausgabe Benutzereingabe: {
[...]
```

In diesem Fall brauchen Sie nicht `Strg`+`D` zu drücken, weil die Bedingung »End Of File« – wie der Name schon sagt – am Ende der Eingabedatei eintritt.

Nebenbei zeigt das Programm die beiden Hauptmöglichkeiten der Dateneingabe in einfache Konsolenprogramme: die Übergabe von Kommandozeilenparametern sowie die Shell-artige, zeilenweise interaktive Eingabe bis EOF.

Pipes

Eine weitere Variante der Ein- und Ausgabeumleitung ist die sogenannte *Pipeline* oder kurz *Pipe* (englisch für Röhre). Sie ermöglicht es, die Ausgabe eines Befehls als Eingabe für den nächsten zu verwenden. Eine der gängigsten Kombinationen ist die Weiterleitung der umfangreichen Ausgabe bestimmter Befehle an einen Pager – ein Programm, das Inhalte seitenweise ausgibt. Der ursprüngliche UNIX-Pager wird `more` genannt, die erheblich mächtigere GNU-Alternative heißt `less` (getreu dem Sprichwort »less is more«).

Angenommen, der Inhalt des aktuellen Verzeichnisses ist länger als die Anzahl der Zeilen Ihres Terminals. In diesem Fall können Sie diesen Inhalt an `less` weiterleiten:

```
$ ls |less
```

Das Pipe-Zeichen `|` wird auf einer deutschen Tastatur mit der Tastenkombination `Alt Gr`+`<` erzeugt.

Die Pipe verknüpft die Standardausgabe des ersten Befehls mit der Standardeingabe des zweiten. Im Gegensatz zur Verwendung einer Zwischendatei per Ein- und Ausgabeumleitung fängt der zweite Befehl einer Pipe bereits an zu arbeiten, wenn er die erste Zeile aus der Ausgabe des ersten erhält. Das liegt daran, dass die beiden Befehle als separate Prozesse ausgeführt werden. Näheres zu Prozessen und ihrer Verwaltung erfahren Sie weiter unten in diesem Kapitel.

Eine weitere verbreitete Anwendung für Pipes besteht in der unmittelbaren Filterung einer Ausgabe mithilfe von `grep`. Das folgende Beispiel gibt nur diejenigen Dateien im aktuellen Verzeichnis aus, die die Zeichenfolge `txt` enthalten:

```
$ ls |grep txt
```

Es ist übrigens kein Problem, mehr als zwei Befehle durch Pipes zu verknüpfen; im Grunde können Sie eine beliebig lange Kette anlegen. Angenommen, die Ausgabe der obigen Anweisung `ls |grep txt` ist zu lang für einen Bildschirm. Dann

können Sie sich die Ausgabe von `grep` wiederum durch `less` seitenweise ausgeben lassen:

```
$ ls |grep txt |less
```

Genauso können Sie die oben besprochene Ein- und Ausgabeumleitung beliebig mit Pipes verknüpfen. Das folgende Kommando schreibt die Ergebnisse der Suche nach Dateien, die `txt` enthalten, in die Datei `txtfiles`:

```
$ ls |grep txt >txtfiles
```

Da dieser Befehl seine Standardausgabe nun ohnehin in eine Datei schreibt, gibt es keinen Grund, dass er Ihr aktuelles Terminal blockiert. Führen Sie ihn einfach mittels `&` im Hintergrund aus. Kümmern Sie sich aber zusätzlich um seine Fehlermeldungen, damit diese nicht zwischen Ihre anderweitige Weiterarbeit mit dem Terminal geraten. Das folgende Beispiel leitet sie nach `/dev/null` um, ignoriert sie also:

```
$ ls |grep txt >txtfiles 2>/dev/null &
[1] 22858
```

Eine interessante Alternative zur einfachen Umleitung der Standardausgabe ist übrigens das Dienstprogramm `tee`, das einfach alles, was über die Standardeingabe hereinkommt, in die angegebene Datei schreibt. Der Name steht für den Buchstaben T und damit für ein T-Stück, das eine Leitung verdoppelt. Wenn Sie dieses Programm über eine Pipe an eine beliebige Anweisung anhängen, werden weiterhin alle Meldungen in die Konsole ausgegeben und dennoch zusätzlich in eine Datei geschrieben. Das folgende Beispiel gibt den Inhalt des aktuellen Verzeichnisses aus und schreibt ihn zusätzlich in die Textdatei `inhalt.txt`:

```
$ ls |tee inhalt.txt
```

Eine Sonderform der Pipes sind die sogenannten *Named Pipes*. Es handelt sich um spezielle Verzeichniseinträge (nicht unbedingt »richtige Dateien«), die zur *Interprozesskommunikation* (IPC) eingesetzt werden. Um eine Named Pipe zu erstellen, können Sie das Kommando `mkfifo` einsetzen. Anschließend kann ein Prozess per Ausgabeumleitung in die Named Pipe schreiben, während ein anderer mithilfe der Eingabeumleitung daraus lesen kann. Auf diese Weise können Sie die beiden an der Pipe beteiligten Programme separat starten.

Wenn Sie es ausprobieren möchten, geben Sie in einem beliebigen Terminal(fenster) Folgendes ein, um eine Named Pipe zu erstellen und daraus zu lesen:

```
$ mkfifo mypipe
$ ls -l mypipe
```

```
prw-rw-rw- 1 sascha users 0 Jan 11 01:15 mypipe
$ cat mypipe
```

Das Zeichen `p` ganz am Anfang der Ausgabe von `ls -l` zeigt an, dass es sich bei dem Verzeichniseintrag um eine Named Pipe handelt.

Öffnen Sie anschließend ein anderes virtuelles Terminal oder ein weiteres GUI-Terminalfenster. Wechseln Sie dort in dasselbe Arbeitsverzeichnis. Geben Sie die nachfolgenden Anweisungen an, um einige Zeilen in die Named Pipe zu schreiben:

```
$ cat >mypipe
Hallo
Welt
[Strg]+[D]
```

Wenn Sie nun zurück in das andere Fenster wechseln, werden Sie feststellen, dass die neu eingegebenen Zeilen aus der Named Pipe ausgelesen und angezeigt wurden.

Befehlsverknüpfungen

Es gibt noch einige weitere Möglichkeiten, mehrere Befehle hintereinander in eine Anweisungszeile zu schreiben. Anders als bei einer Pipe werden diese allerdings auf jeden Fall nacheinander ausgeführt, teilweise in Abhängigkeit voneinander.

Die einfachste Verknüpfung besteht darin, zwei Anweisungen durch ein Semikolon getrennt hintereinander zu schreiben. Dies ist im Prinzip gleichbedeutend mit zwei völlig getrennten Anweisungen, die Sie nacheinander eingeben und ausführen, und hat nur den Vorteil, dass sie automatisch hintereinander ausgeführt werden, sodass Sie nach der Eingabe beispielsweise Kaffee trinken gehen können. Das folgende Beispiel schreibt die Liste aller zurzeit laufenden Prozesse in die Datei *prozesse.txt* und zeigt sie anschließend mit `less` an:

```
$ ps aux >prozesse.txt; less prozesse.txt
```

Natürlich bringt dies keinen Zusatznutzen gegenüber dieser Eingabe:

```
$ ps aux |less
```

Viel nützlicher ist eine solche Verknüpfung bei zwei oder mehr Befehlen, deren Ausführung jeweils recht lange dauert. Ein praktisches Beispiel ist die in Kapitel 3 angesprochene Kompilierung gängiger Software, die mittels Automake vorkonfiguriert wurde. Die drei Befehle `configure`, `make` und `make install` könnten Sie auch als Folge eingeben und dann etwa in die Mittagspause gehen:

```
# ./configure [Optionen]; make; make install
```

Da der Puffer der meisten Terminalemulationen für die Ausgabe aller drei Befehle zu kurz ist, lohnt es sich in diesem Fall, die Ausgabe in eine Textdatei umzuleiten, um sie später in Ruhe als Installationsprotokoll lesen zu können. Bitte beachten Sie, dass es drei Befehle sind, sodass Sie sie jeweils einzeln umleiten müssen; zudem muss die Ausgabe des zweiten und dritten Befehls an die Protokolldatei angehängt werden, damit diese nicht überschrieben wird:

```
# ./configure [Optionen] >install_log; \  
> make >>install_log; make install >>install_log
```

Ein kleines Problem löst die Verkettung per Semikolon noch nicht: Wenn eine der Anweisungen nicht ordnungsgemäß ausgeführt werden kann, versucht die Shell, die nachfolgenden Anweisungen trotzdem auszuführen. Wenn Sie den Aufruf des nächsten vom Ergebnis des vorigen Befehls abhängig machen möchten, können Sie statt des Semikolons logische Operatoren zur Verknüpfung verwenden.

Das logische Und (&&) führt zunächst den vorderen Befehl aus. Nur wenn er ohne Probleme durchläuft (sein Rückgabewert oder *Exit-Code* muss 0 sein), dann wird auch die nachfolgende Anweisung ausgeführt. Das liegt an der sogenannten *Short-Circuit-Logik* der logischen Operatoren: Wenn das logische Und schon beim ersten Operanden absehen kann, dass die Gesamtbedingung nicht zu erfüllen ist, bricht es die weitere Überprüfung – und in diesem Zusammenhang die weitere Befehlsausführung – ab. Daher lautet die passendere Schreibweise für die Kompilierbefehle:

```
# ./configure && make && make install
```

Denn wenn bereits `configure` mit einer Fehlermeldung scheitert – meist, weil eine notwendige Bibliothek nicht installiert ist oder in einer inkompatiblen Version vorliegt –, dann ist es völlig sinnlos, `make` aufzurufen; und falls `make` (die eigentliche Kompilierung) nicht fehlerlos durchläuft, dann ist `make install` zwecklos.

Umgekehrt verhält sich die Kombination von Anweisungen durch logisches Oder: Da diese Verknüpfung bereits wahr ist, wenn die erste Teilbedingung erfüllt ist, wird die zweite Anweisung hier nur ausgeführt, falls die erste scheitern sollte. Die Oder-Verknüpfung lässt sich also einsetzen, um im Fall eines Fehlers eine zusätzliche Anweisung auszuführen. Das folgende Beispiel versucht, die Datei `test.txt` im aktuellen Verzeichnis aufzulisten; sollte sie nicht vorhanden sein, wird sie (leer) neu angelegt:

```
$ ls test.txt || touch test.txt  
/bin/ls: test.txt: Datei oder Verzeichnis nicht gefunden
```

Dass die Datei nun existiert, merken Sie durch den zweiten Aufruf desselben Befehls `↑` und `↵`:

```
$ ls test.txt || touch test.txt
test.txt
```

Hinweis

Falls Sie an Programmiersprachen wie C gewöhnt sind, mag es Ihnen ungewöhnlich erscheinen, dass ausgerechnet der Exit-Code 0 »wahr« ist, während von 0 verschiedene Exit-Codes »falsch« sind. Dies ist allerdings – auch aus der Sicht der Shell – eine Besonderheit des Exit-Codes.

Interessant ist zu guter Letzt das Verschachteln von Anweisungen in ``Backticks`` (diese werden durch `⌘`+`⏏`, die Taste rechts neben dem `⏏`, eingegeben). Auf diese Weise wird die Ausgabe der Anweisung in der Umgebung verwendet, beispielsweise in einer Ausgabe mit `echo`. Das folgende Beispiel fügt die Ausgabe des Befehls `whoami` (Username des aktuell angemeldeten Benutzers) in einen vollständigen Satz ein:

```
$ echo Zurzeit ist der Benutzer `whoami` angemeldet
Zurzeit ist der Benutzer sascha angemeldet
```

Nützlicher ist es beispielsweise, die Ausgabe von `whoami` als Filter für `ls` zu verwenden, um sich diejenigen Dateien anzeigen zu lassen, die dem aktuellen Benutzer gehören (oder deren Dateiname seinen Usernamen enthält):

```
$ ls -l |grep `whoami`
```

4.3 Die wichtigen Systembefehle

In diesem Abschnitt lernen Sie die wichtigsten Systemkommandos mit ihren gängigsten Optionen in Form eines umfangreichen Tutorials kennen. Manche Kommandos werden dagegen an passenderen Stellen in anderen Kapiteln dieses Buches eingeführt. Eine praktische Referenz vieler Befehle mit den wichtigsten ihrer zahlreichen Optionen finden Sie schließlich in Kapitel 19, »Referenz der Shell- und Systembefehle«.

4.3.1 Vorbemerkungen zu Shell-Kommandos

Die meisten traditionellen UNIX-Shell-Kommandos haben einige Eigenschaften gemeinsam, die hier vor der Betrachtung der einzelnen Befehle kurz angesprochen werden sollen. Alle Linux-Distributionen enthalten die GNU-Versionen der

Tools; diese bieten üblicherweise mehr Optionen als die Original-UNIX-Dienstprogramme, ihre grundsätzliche Funktionsweise ist aber meist identisch.

Die UNIX-Systemprogramme wurden »für Profis« geschrieben, ihre Autoren gingen also davon aus, dass sie von jemandem benutzt werden, der weiß, was er tut. Das merkt man ihrem Verhalten vor allem im Hinblick auf die folgenden beiden Punkte an:

- ▶ *Keine Meldung bei Erfolg.* Die meisten Systemprogramme geben nur dann eine Rückmeldung, wenn etwas schiefgeht; im Erfolgsfall sehen Sie einfach den Prompt für den nächsten Befehl. Angenommen, Sie möchten die Datei *a.txt* nach *b.txt* kopieren und geben dazu folgende Anweisung ein:

```
$ cp a.txt b.txt
```

Wenn *a.txt* existiert, wird die Kopie kommentarlos erstellt (normalerweise sogar, falls ein vorhandenes *b.txt* dazu überschrieben werden muss – siehe nächsten Punkt). Nur wenn *a.txt* nicht vorhanden ist, erhalten Sie eine Fehlermeldung wie diese:

```
cp: cannot stat `a.txt': No such file or directory
```

Viele Befehle besitzen allerdings eine Option namens *-v* (Langform *--verbose*), die kurz erklärt, was geschieht. Angenommen, Sie benennen mit folgendem Kommando die Datei *b.txt* in *c.txt* um:

```
$ mv -v b.txt c.txt
```

In diesem Fall erhalten Sie folgende Meldung:

```
"b.txt" -> "c.txt"
```

- ▶ *Es wird ohne Rückfrage gelöscht oder überschrieben.* Möglicherweise sind Ihnen die Windows-Konsolenprogramme bekannt. Diese fragen üblicherweise nach, bevor sie beispielsweise mehrere Dateien löschen oder beim Kopieren eine vorhandene Datei überschreiben. Unter Linux und anderen UNIX-Varianten ist das anders; hier werden auch potenziell gefährliche Anweisungen einfach ausgeführt. Andererseits werden die Benutzerrechte streng beachtet, sodass Sie als normaler User zwar mitunter ärgerliche Datenverluste erleiden, aber zumindest nichts für das System Wichtiges zerstören können. Aus diesem Grund sollten Sie im Normalfall nicht als *root* arbeiten, denn dann dürften Sie wirklich alles.

Die meisten Kommandos verarbeiten eine Unmenge verschiedener Optionen. Diese bestehen meist aus einem einzelnen Buchstaben mit vorangestelltem Minuszeichen, wobei zwischen Groß- und Kleinschreibung unterschieden wird.

Mehrere Optionen lassen sich oft hinter einem einzigen Minuszeichen zusammenfassen, solange sie keine eigenen Argumente benötigen – beispielsweise löscht `rm -rf Verzeichnis` genau wie `rm -r -f Verzeichnis` das angegebene Verzeichnis und seinen vollständigen Inhalt.

Bei den GNU-Versionen der Tools gibt es meist auch eine Langform, die aus zwei Minuszeichen, gefolgt von einem ausgeschriebenen Wort besteht (manche seltener genutzte Optionen besitzen dann sogar nur diese Schreibweise und gar keine Kurzform). In der Regel stehen die Optionen hinter dem Befehlsnamen, aber vor den eigentlichen Argumenten wie etwa Datei- oder Verzeichnisnamen.

4.3.2 Arbeiten mit Dateien und Verzeichnissen

Die wichtigste Aufgabe einer Shell besteht darin, Dateien und Verzeichnisse zu verwalten – in einem UNIX-Betriebssystem, in dem »alles eine Datei ist«, erst recht. Im vorliegenden Abschnitt werden die wichtigsten Befehle zur Verarbeitung von Dateien und Verzeichnissen erläutert. Weitere Kommandos kommen in Kapitel 9, »Datenträger und Dateisysteme«, zur Sprache.

Dateimuster

Die Shell bietet die Möglichkeit, Datei- und Verzeichnisnamen in vielen Befehlen durch Muster zu ersetzen, die auf mehrere Dateien passen. In diesen Mustern gibt es die folgenden wichtigen Sonderzeichen (die in Dateinamen verboten oder zumindest problematisch sind):

- ▶ Das Sternchen (*) ersetzt beliebig viele beliebige Zeichen, einschließlich gar keinem Zeichen. `h*o` steht beispielsweise für `hallo`, `hello` oder `ho`.
- ▶ Das Fragezeichen (?) steht für genau ein beliebiges Zeichen. Zum Beispiel bezeichnet `te?t` sowohl `test` als auch `text`, nicht aber `txt` oder `teapot`.
- ▶ Mehrere Zeichen in eckigen Klammern wie `[abc]` bedeuten, dass genau eines dieser Zeichen gemeint ist. Durch einen Bindestrich können Bereiche wie `a-z` gebildet werden; mehrere Listen werden einfach hintereinander geschrieben. Beispielsweise bedeutet die Liste `[a-zA-Z0-9]`, dass alle Kleinbuchstaben, alle Großbuchstaben und alle Ziffern zulässig sind.
- ▶ Ein Ausrufezeichen vor der Liste in den eckigen Klammern bedeutet, dass ein beliebiges Zeichen *außer* dieser Liste zulässig ist. `[!Bb]` bedeutet etwa, dass auf keinen Fall ein `B` erlaubt ist – weder ein groß- noch ein kleingeschriebenes.
- ▶ Eine durch Kommata getrennte Liste von Zeichenketten in geschweiften Klammern bedeutet, dass eine dieser Zeichenketten erwartet wird. Zum Beispiel bedeutet `{info,hinweis,hilfe}.txt`, dass eine der drei Dateien `info.txt`, `hilfe.txt` oder `hinweis.txt` gesucht wird.

- Durch ein Pipe-Zeichen (|) können Sie schließlich mehrere Muster angeben, die durch »Oder« verknüpft werden – trifft eines dieser Muster auf eine Datei zu, dann passt sie zum Gesamtmuster. Der Ausdruck `b*|info*` bedeutet beispielsweise: alle Dateien, die mit `b` oder mit `info` beginnen. Aus Sicherheitsgründen sollten Sie Dateimuster, in denen Sie dieses Zeichen benutzen, in Anführungszeichen setzen, damit die Shell das Zeichen nicht irrtümlich für eine Pipe hält, die den Rest des Musters als eigenständige Anweisung abtrennt.

Falls Sie Dateimuster unter Windows kennen sollten, beachten Sie bitte, dass die »Dateierweiterung« (die Abkürzung hinter dem letzten Punkt wie etwa `txt`) auf UNIX-Systemen ein normaler Bestandteil des Dateinamens ist und kein spezieller Bereich. In einem UNIX-Befehl steht `*` für *alle* Dateien im aktuellen Verzeichnis. Unter Windows ist ein `*` dagegen nur der Platzhalter für Dateien ohne Erweiterung.² Für alle Dateien steht dort `*.*`; dieser Ausdruck würde auf einem UNIX-System nur auf Dateinamen zutreffen, die mindestens einen Punkt enthalten.

Diese einfachen Suchmuster für Dateien werden übrigens nicht mit dem bereits erwähnten Befehl `grep` verwendet. Die dort zulässigen Muster sind reguläre Ausdrücke; diese bieten noch erheblich mehr Möglichkeiten. Eine Einführung finden Sie weiter unten in der Beschreibung von `grep`.

Dateien kopieren mit `cp`

Das Kommando `cp` (Kurzfassung für »copy«) kopiert eine oder mehrere Dateien an den angegebenen Ort. Die Syntax ist grundsätzlich folgende:

```
cp [Optionen] Quelle [Quelle ...] Ziel
```

Die Quelle kann eine einzelne Datei oder ein Muster sein, Sie können alternativ auch einen Pfad angeben. Das Ziel ist entweder ein einzelner Dateiname (falls Sie nur eine Datei kopieren) oder ein Verzeichnis, falls im Zielordner bereits ein Verzeichnis mit diesem Namen existiert oder falls Sie als Quelle keine einzelne Datei, sondern mehrere Dateien oder ein Muster angegeben haben. Das folgende Beispiel kopiert die Datei `hallo.txt` in eine neue Datei namens `hi.txt`:

```
$ cp hallo.txt hi.txt
```

Das nächste Beispiel kopiert alle Dateien aus dem Verzeichnis `briefe` in das Verzeichnis `dokumente`, das im gleichen Verzeichnis liegt wie `briefe`:

```
$ cp briefe/* dokumente
```

² Außer in der relativ neuen Windows Power Shell, in der das Verhalten demjenigen unter UNIX entspricht.

Die Option `-r` (Langform: `--recursive`) kopiert die Dateien, die dem angegebenen Muster entsprechen, aus dem aktuellen Verzeichnis und allen Unterverzeichnissen in das Zielverzeichnis, wobei die entsprechenden Unterverzeichnisse am Zielort angelegt werden. Das folgende Beispiel kopiert das Verzeichnis *texte* mit seinem gesamten Inhalt nach *textkopien*:

```
$ cp -r texte textkopien
```

Standardmäßig überschreibt `cp` vorhandene Zielf Dateien ohne Nachfrage. Wenn Sie dies nur selektiv tun möchten, können Sie die Option `-i` oder `--interactive` verwenden, die vor jedem Überschreiben nachfragt. Beispiel:

```
$ cp -i test.txt test2.txt
cp: "test2.txt" überschreiben?
```

Geben Sie je nach Sprache der Meldung beziehungsweise ein, wenn Sie die angegebene Datei überschreiben möchten, oder , wenn Sie die alte Fassung behalten wollen.

Alternativ können Sie die Option `--backup` verwenden, die von jeder vorhandenen Datei automatisch eine Sicherungskopie anlegt, bevor sie überschrieben wird. Der Name der Backup-Datei entspricht der ursprünglichen Datei, gefolgt von einer Tilde (~). Angenommen, Sie führen folgende Operation durch, während *hi.txt* bereits vorhanden ist:

```
$ cp --backup hallo.txt hi.txt
```

Danach ist *hi.txt* eine Kopie von *hallo.txt*, während *hallo.txt~* eine Sicherungskopie der ursprünglichen Datei *hallo.txt* ist.

Schließlich können Sie auch noch die Option `-u` oder `--update` verwenden, die nur dann eine Kopie anfertigt, wenn die Zielfdatei entweder noch gar nicht existiert oder älter als die Quelldatei ist. Das folgende Beispiel behandelt alle Dateien mit der Endung *.txt* im aktuellen Verzeichnis entsprechend:

```
$ cp -u *.txt textkopien
```

Hinweis

Beachten Sie, dass es sich bei den nachfolgend beschriebenen Aktionen »Umbenennen«, »Verschieben« und »Löschen« um (besondere) Schreibzugriffe handelt, die Sie nur ausführen dürfen, wenn Sie Schreibrechte für die jeweiligen Verzeichnisse und Dateien besitzen. Die Modifikation von Benutzerrechten wird weiter unten beschrieben.

Dateien verschieben und umbenennen mit mv

Der Befehl `mv` (für »move«) dient dazu, Dateien umzubeneden oder in ein anderes Verzeichnis zu verschieben. Die Syntax lautet folgendermaßen:

```
mv [Optionen] Quelle [Quelle ...] Ziel
```

Die Quelle ist wieder eine einzelne Datei beziehungsweise ein Muster, das Ziel ist entweder ein völlig neuer Name oder der Name eines bestehenden Verzeichnisses.

Die folgende Anweisung benennt die Datei *vorher.txt* in *nachher.txt* um:

```
$ mv vorher.txt nachher.txt
```

Wenn Sie als Quelle statt einer einzelnen Datei mehrere oder auch ein Muster angeben, muss das Ziel ein bestehendes Verzeichnis sein – Sie können nicht mehrere Dateien auf einmal umbenennen, sondern nur verschieben.

Wenn Ihre Quelle ein Verzeichnis und das Ziel ein existierendes Verzeichnis ist, wird der gesamte untergeordnete Verzeichnisbaum mitsamt allen enthaltenen Unterverzeichnissen und Dateien an den neuen Ort verschoben. Ist das Ziel dagegen ein neuer Name, so wird einfach das Verzeichnis entsprechend umbenannt. Insofern hat die folgende Anweisung unterschiedliche Auswirkungen, je nachdem, ob das Verzeichnis *dokumente* bereits existiert oder noch nicht:

```
$ mv briefe dokumente
```

Genau wie der Kopierbefehl `cp` kennt auch `mv` die folgenden Optionen:

- ▶ `-i` fragt nach, bevor ein existierendes Ziel überschrieben wird.
- ▶ `-u` ersetzt ein vorhandenes Ziel nur dann, wenn die Quelle neuer ist.
- ▶ `--backup` erstellt Sicherungskopien bereits existierender Zieldateien mit einer angehängten Tilde (~).

Dateien löschen mit rm

Das Kommando `rm` (»remove«) löscht die angegebenen beziehungsweise dem Muster entsprechenden Dateien, und zwar tendenziell *endgültig*. Eine Einrichtung wie der Papierkorb der im nächsten Kapitel behandelten Desktop-Manager ist auf der Konsole nicht vorhanden.

Die Grundsyntax der Anweisung lautet folgendermaßen:

```
rm [Optionen] Dateimuster [Dateimuster ...]
```

Das folgende Beispiel löscht alle Dateien aus dem aktuellen Verzeichnis, deren Name nicht mit `a` beginnt:

```
$ rm [!a]*
```

`rm` löscht Dateien standardmäßig nur im aktuellen beziehungsweise im explizit angegebenen Verzeichnis, aber nicht in dessen Unterverzeichnissen. Wenn Sie auch die Inhalte der Unterverzeichnisse löschen möchten, müssen Sie die Option `-r` oder `--recursive` einsetzen. Noch effizienter (aber auch gefährlicher!) ist die zusätzliche Option `-f` (`--force`), die das Löschen schreibgeschützter Dateien erzwingt. Der folgende Befehl löscht alle Dateien im aktuellen Verzeichnis und alle Unterverzeichnisse und sollte nur mit äußerster Vorsicht eingesetzt werden:

```
$ rm -rf *
```

Weiter oben wurde bereits erwähnt, dass `rm` eine Datei erst tatsächlich löscht, wenn der letzte Hard Link entfernt wird, der auf die entsprechende Inode zeigt. Das liegt daran, dass `rm` intern den Systemaufruf `unlink()` verwendet, der – wie der Name vermuten lässt – einen Link entfernt.

Um herauszufinden, ob mehrere Links auf dieselbe Inode verweisen, können Sie `ls`, den als Nächstes ausführlich besprochenen Befehl für den Verzeichnisinhalt, mit den Optionen `-l` (Details) und `-li` (Inode-Nummern) aufrufen. Im folgenden Beispiel verweisen die beiden Einträge `a.txt` und `b.txt` auf dieselbe Inode (2011), es handelt sich also nur um eine Datei:

```
$ ls -li
insgesamt 2
2011 -rw-r--r-- 2 sascha users 143 2008-04-18 19:52 a.txt
2011 -rw-r--r-- 2 sascha users 143 2008-04-19 19:52 b.txt
```

Diese Datei wird erst tatsächlich entfernt, wenn Sie beide Links löschen:

```
$ rm a.txt b.txt
```

Links erzeugen mit `ln`

Das Kommando `ln` erzeugt einen Link auf eine Datei. Ohne Optionen wird ein Hard Link erstellt; die Quelle (erste Datei) muss eine reguläre Datei sein, das Ziel muss auf derselben Partition liegen wie die Quelle. Beispiel:

```
$ ln text1.txt text2.txt
$ ls -li text*
2046 -rw-r--r-- 2 sascha users 11 2008-04-20 18:26 text1.txt
2046 -rw-r--r-- 2 sascha users 11 2008-04-20 18:26 text2.txt
```

Die mittels `-i` angezeigte inode-Nummer ist bei beiden Einträgen identisch, da es sich um dieselbe Datei handelt.

Ein symbolischer Link (Symlink) verweist – wie bereits erwähnt – nicht auf eine inode, sondern auf einen anderen Verzeichniseintrag. Symlinks können Sie deshalb auch auf Verzeichnisse sowie über Partitions Grenzen hinweg erstellen. Das folgende Beispiel erstellt einen Symlink auf `datei1` namens `datei2`:

```
$ ln -s datei1 datei2
$ ls -li datei*
2047 -rw-r--r-- 1 sascha users 11 2008-04-20 18:32 datei1
2048 lrwxrwxrwx 1 sascha users 6 2008-04-20 18:32 datei2 -> datei1
```

Wie Sie sehen, ist die inode-Nummer unterschiedlich. Zusätzlich besitzt der Symlink die Dateitypangabe `l`, und das Link-Ziel wird angezeigt.

Symlinks werden im Rahmen von UNIX-Standardsoftware sehr häufig eingesetzt, beispielsweise um versionsneutrale Verweise auf Verzeichnisnamen mit Versionsnummer zu erstellen, was etwa die Übernahme von Konfigurationsdateien in eine neue Version erleichtert. Ein Beispiel sind die Kernel-Quellen: `/usr/src/linux` ist ein Symlink auf `/usr/src/linux/versionsnr`; in openSUSE 11.2 ist es beispielsweise `linux-2.6.31`.

Verzeichnisinhalte auflisten mit `ls`

Mit `ls` (»list«) zeigen Sie den Inhalt des aktuellen beziehungsweise des angegebenen Verzeichnisses an, das heißt eine Liste aller enthaltenen Dateien und Unterverzeichnisse. Wenn Sie ein Muster angeben, wird es als Filter verwendet. Existieren Dateien, deren Namen zu diesem Muster passen, dann werden nur diese angezeigt. Andernfalls werden zusätzlich zum aktuellen Verzeichnis auch die Inhalte der Unterverzeichnisse angezeigt, auf deren Namen das Muster passt.

Die folgende Anweisung zeigt beispielsweise alle Dateien an, die mit `b` beginnen – falls es keine gibt, werden alternativ die Inhalte aller Verzeichnisse angezeigt, die mit `b` anfangen:

```
$ ls b*
```

Eine wichtige Option dieses Befehls ist `-l` (»long«), die statt der einfachen Namen ausführliche Informationen über jeden Verzeichniseintrag ausgibt. Eine typische Ausgabe sieht etwa so aus:

```
$ ls -l
insgesamt 3
-rw-r--r-- 1 sascha users 3535 2007-09-16 17:31 draft1.txt
```

```
-rw-r--r-- 1 sascha users 30762 2008-02-20 08:05 shell.txt
-rwxr-xr-x 1 sascha users 253 2008-03-14 21:58 test.pl
```

Die erste Zeile (»insgesamt«) zeigt die Anzahl der Dateien im aktuellen Verzeichnis einschließlich der verschachtelten Unterverzeichnisse an. In jeder darauf folgenden Zeile finden Sie von links nach rechts folgende Informationen über eine Datei:

- ▶ *Dateityp und Benutzerrechte.* Der erste Block besteht aus zehn Zeichen, die sich wie folgt aufschlüsseln:
 - ▶ Dateityp (ein Zeichen): – für eine reguläre Datei, d für ein Verzeichnis, l für einen Symlink, p für eine Named Pipe, b für ein Blockgerät (Gerätedatei in /dev, die wahlfreien Zugriff erlaubt – etwa eine Festplatte), c für ein Zeichengerät (Gerätedatei, die nur sequenzielle Ein- und/oder Ausgabe unterstützt) oder s für ein Socket.
 - ▶ Rechte des Dateieigentümers (drei Zeichen); r/w/x für gewährtes Lesen/Schreiben/Ausführen oder –, falls das entsprechende Recht verweigert wird. Im Beispiel für die beiden Textdateien rw-, weil Ausführen hier sinnlos wäre, aber rwx für das Perl-Skript.
 - ▶ Rechte der Gruppe, der die Datei gehört (drei Zeichen); die Schreibweise entspricht den Eigentümerrechten. Im obigen Beispiel r-- (nur Lesen) für die Textdateien und r-x (Lesen und Ausführen) für das Perl-Skript.
 - ▶ Rechte sonstiger User (drei Zeichen); sind im Beispiel jeweils identisch mit den Rechten der Gruppe.
- ▶ *Anzahl der Hard Links.* Dieser Eintrag gibt an, wie viele Hard Links (Verzeichniseinträge) auf dieselbe inode zeigen.
- ▶ *Eigentümer.* Der Benutzer, dem die Datei gehört (hier jeweils sascha). Auf diesen bezieht sich die erste Gruppe der Rechte im vorigen Abschnitt.
- ▶ *Gruppe.* Die Gruppe, der die Datei gehört: Jede Datei gehört genau einer Gruppe, während jeder User neben seiner primären Gruppe beliebig vielen weiteren Gruppen angehören kann. Die mittleren drei Zeichen aus den Rechten im ersten Block beziehen sich auf die Gruppe.
- ▶ *Größe.* Gibt die Größe der jeweiligen Datei an – standardmäßig in Byte; dies lässt sich allerdings konfigurieren.
- ▶ *Datum der letzten Änderung.* Hier wird angezeigt, wann die Datei zuletzt modifiziert wurde; das Format ist JJJJ-MM-TT.
- ▶ *Uhrzeit der letzten Änderung.* In der nächsten Gruppe sehen Sie die entsprechende Uhrzeit im Format hh:mm.

- *Dateiname*. Die vielleicht wichtigste Information steht als letzte in der Zeile; bei Symlinks wird zusätzlich ein Pfeil und das Ziel des Links angehängt.

Mithilfe der Option `-F` können Sie sich ein Dateitypsymbol hinter dem Dateinamen anzeigen lassen: `*` (ausführbar), `/` (Verzeichnis), `=` (Socket), `@` (Symlink) oder `|` (Named Pipe); normale Dateien werden gar nicht gekennzeichnet.

Auch die Option `-a` (oder lang `--all`) wird relativ häufig verwendet, weil es versteckte Dateien und Verzeichnisse einblendet, das heißt diejenigen, deren Namen mit einem Punkt beginnen. Die intelligentere Variante `-A` (`--almost-all`) lässt die speziellen Einträge `.` für das aktuelle Verzeichnis selbst und `..` für das übergeordnete Verzeichnis aus.

Bereits im Zusammenhang mit dem Löschen von Dateien wurde die Option `-i` angesprochen, die zusätzlich zu den sonstigen ausführlichen Optionen die Inode-Nummer der jeweiligen Datei angibt.

Die Option `-r` zeigt nicht nur die dem Dateimuster entsprechenden Treffer aus dem aktuellen Arbeitsverzeichnis an, sondern durchwandert auch rekursiv alle Unterverzeichnisse.

Das Arbeitsverzeichnis ermitteln und ändern

Wie bereits angesprochen, befinden Sie sich in der Shell stets in einem bestimmten aktuellen Arbeitsverzeichnis. Viele Anweisungen beziehen sich automatisch auf dieses Verzeichnis, beispielsweise zeigt `ls` seinen Inhalt an.

Die Anweisung `pwd` («print working directory») gibt den vollständigen Pfad des aktuellen Arbeitsverzeichnisses an. Dies ist beispielsweise nützlich, um den tatsächlichen Pfad des eigenen Home-Verzeichnisses zu ermitteln, der im Prompt durch `~` abgekürzt wird. In manchen GUI-Terminalemulationen zeigt der Prompt standardmäßig sogar gar nicht das Arbeitsverzeichnis an, sodass diese Anweisung hier oft unerlässlich ist.

`cd` («change directory») wechselt in das angegebene Arbeitsverzeichnis. Sie können den gewünschten Pfad entweder relativ zum aktuellen Arbeitsverzeichnis oder absolut durch einen vorangestellten `/` angeben. Auch die Namen verschachtelter Verzeichnisse werden dabei jeweils durch einen `/` getrennt. Das übergeordnete Verzeichnis erreichen Sie wie bereits erwähnt mit `..`, und wenn Sie mehrere Ebenen nach oben müssen, dann können Sie `../..` und so weiter schreiben. Von übergeordneten Verzeichnissen aus ist dann wiederum der Wechsel in deren andere Unterverzeichnisse möglich, sodass Sie beliebige Hierarchien bei Bedarf auch per relativer Pfadangabe durchwandern können.

Die folgende Anweisung wechselt beispielsweise aus `/home/sascha/opensuse11/kapitel` in das Verzeichnis `/home/sascha/opensuse11/beispiele`:

```
sascha@tux: ~/opensuse11/kapitel > cd ../beispiele
sascha@tux: ~/opensuse11/beispiele >
```

Das folgende Beispiel wechselt dagegen mithilfe einer absoluten Angabe von `/home/sascha/opensuse11/kapitel` nach `/etc`:

```
sascha@tux: ~/opensuse11/kapitel > cd /etc
sascha@tux: /etc >
```

Wie bereits angesprochen, ist die Spezialangabe `~` ein Kürzel für das eigene Home-Verzeichnis. Deshalb können Sie mit `cd ~` stets in Ihr Home-Verzeichnis wechseln. Noch kürzer geht es allerdings mit `cd` ohne Argument.

openSUSE definiert im Übrigen schon seit längerem zwei sehr praktische Aliase: `..` für `cd ..`, also den Wechsel in das übergeordnete Verzeichnis, und `...`, um mittels `cd ../..` noch eine Ebene höher zu wechseln. Beispiel:

```
sascha@tux: ~/opensuse11/stuff/skripte > ...
sascha@tux: ~/opensuse11 >
```

Eine interessante Variante von `cd` ist das Kommando `pushd`: Das Verzeichnis, in das Sie wechseln, wird auf einem Stack oder Stapel (Last in, first out) abgelegt. Mithilfe des Kommandos `popd` (ohne Argumente) können Sie danach wieder in die zuvor per `pushd` auf den Stapel gelegten Verzeichnisse zurückwechseln.

Verzeichnisse erstellen und löschen

Die Anweisung `mkdir` («make directory») legt ein neues Verzeichnis mit dem angegebenen Pfad an. So richtet etwa die folgende Anweisung unterhalb des aktuellen Verzeichnisses das neue Verzeichnis `test` ein:

```
$ mkdir test
```

Beachten Sie, dass bei der Angabe eines mehrgliedrigen Pfads alle Verzeichnisse außer dem letzten bereits existieren müssen. Die Option `-p` («parents») erzeugt dagegen auch verschachtelte Pfade. Das folgende Beispiel erzeugt im aktuellen Verzeichnis die ineinander verschachtelten Verzeichnisse `neu`, `texte`, `briefe`:

```
$ mkdir -p neu/texte/briefe
```

`rmdir` («remove directory») löscht Verzeichnisse, allerdings nur leere. Zum Löschen verschachtelter Verzeichnisbäume wird – wie bereits erwähnt – `rm` mit der Option `-r` verwendet. Der Hauptnutzen von `rmdir` liegt also darin, ein irrtümlich neu erstelltes Verzeichnis sofort wieder zu entfernen.

Dateien suchen mit find

Ein sehr komplexes Kommando mit unzähligen Optionen ist `find`. Es sucht im angegebenen Verzeichnis und all seinen Unterverzeichnissen nach Dateien, deren Name einem bestimmten Kriterium entspricht. Eine ausführliche Liste mit vielen weiteren möglichen Kriterien finden Sie in Kapitel 19; hier sollen einige Beispiele genügen. Die grundlegende Syntax lautet:

```
find Verzeichnis [Optionen] -Kriterium Vergleichswert
```

Das häufigste Kriterium ist `-name` für die Suche nach Dateinamen oder `-mustern`. Das Namensmuster entspricht den weiter oben besprochenen Regeln für Dateimuster. Beachten Sie aber, dass Sie das Muster in Anführungszeichen setzen müssen, wenn es mit einem `*` endet. Das folgende Beispiel sucht im Wurzelverzeichnis und damit im gesamten Verzeichnisbaum des Systems nach Dateien, deren Name den Bestandteil `sys` enthält:

```
$ find / -name "*sys*"
/dev/shm/sysconfig
/etc/X11/twm/system.twmrc
/etc/X11/xdm/sys.xsession
/etc/X11/xsm/system.xsm
```

Die Variante `-iname` ignoriert Groß- und Kleinschreibung in den gefundenen Dateinamen. Die Kriterien `-regex` und `-iregex` suchen mit beziehungsweise ohne Unterscheidung von Groß- und Kleinschreibung nach regulären Ausdrücken (siehe unten in der Beschreibung von `grep`).

4.3.3 Benutzerkonten und Zugriffsrechte modifizieren

Zu den häufigsten Aufgaben der Systemadministration gehören die Verwaltung von Benutzer- und Gruppenkonten sowie die Modifikation der darauf basierenden Dateizugriffsrechte. In diesem Unterabschnitt lernen Sie die wichtigsten Befehle dafür kennen.

Benutzerkonten und Passwörter verwalten

Im vorigen Kapitel wurde bereits gezeigt, wie Sie Benutzerkonten interaktiv mit YaST verwalten können; auch das Konzept der Dateien `/etc/passwd` und `/etc/shadow` wurde dort erläutert. Die hier gezeigte »Handarbeit« geht allerdings oftmals schneller. Um die entsprechenden Befehle verwenden zu dürfen, müssen Sie `root` sein. Sie können sich entweder an einer neuen Konsole als `root` einloggen oder – was viel praktischer ist – mittels `su` (»substitute user« oder auch »super-user«) Ihre Identität wechseln:

```
$ su -
```

Wenn Sie bei `su` keinen Benutzernamen angeben, wird `root` angenommen. Die abgekürzte Option `-` bedeutet, dass eine richtige `root`-Login-Shell zur Verfügung gestellt werden soll. Dadurch befinden Sie sich nach dem Aufruf im Home-Verzeichnis `/root`, und die Einstellungen für `root` werden eingelesen.

Wenn Sie Ihre Verwaltungsaufgaben abgeschlossen haben, geben Sie einfach `exit` ein oder drücken Sie `[Strg]+[D]`, um wieder ein normaler User zu werden. Es kann nicht oft genug betont werden, dass Sie im Alltag nicht als `root` arbeiten sollten.

Das Kommando `useradd` erstellt ein neues Benutzerkonto. Standardmäßig ist das Home-Verzeichnis dieses Benutzers `/home/Username`. Mithilfe der Option `-d` Pfad können Sie ein anderes Verzeichnis angeben. In jedem Fall sorgt die Option `-m` dafür, dass das Home-Verzeichnis neu angelegt ist, falls es nicht existiert.

Die numerische User-ID wird standardmäßig automatisch eingestellt; es wird der erste freie Wert größer oder gleich 100 gewählt. UIDs unter 100 sind für Systemkonten vorgesehen, das heißt für Benutzerkonten, die für die Ausführung von System- und Serverdiensten verwendet werden; die 0 steht für `root`. Wenn Sie die UID manuell angeben möchten, funktioniert das mithilfe von `-u UID`. Falls die angegebene User-ID bereits existiert, erscheint eine Fehlermeldung, es sei denn, Sie geben zusätzlich die Option `-an`, die den bisherigen User mit dieser UID überschreibt.

Mithilfe der Option `-g` Gruppe wird die Gruppe angegeben, der der Benutzer vorrangig angehören soll – ohne Angabe wird automatisch `users` eingestellt. Mittels `-G Gruppe1,Gruppe2,...` können Sie eine durch Kommata getrennte Liste weiterer Gruppen angeben, denen der Benutzer zusätzlich angehören soll. Diese Liste darf keine Leerzeichen enthalten.

Die Option `-s` Pfad gibt die Shell an, die dem User präsentiert werden soll; der Standardwert ist `/bin/bash`.

Das folgende Beispiel zeigt, wie das Verzeichnis `/home/neuer` angelegt wird, und wie Sie anschließend den Benutzer `neuer` mit der Login-Shell `/bin/tcsh` einrichten können:

```
# mkdir /home/neuer
# useradd -g users -s /bin/tcsh neuer
```

Wenn Sie ein Benutzerkonto erstellen möchten, unter dem eine bestimmte Software ausgeführt wird (etwa ein Serverdienst), unter der sich aber niemand persönlich anmelden können soll, dann können Sie das Home-Verzeichnis gezielt

auf `/dev/null` setzen und als Login-Shell `/bin/false` (ein Programm, das nichts tut, außer einen ungültigen Exit-Code zurückzuliefern) angeben. Die folgenden Anweisungen erstellen zuerst eine Gruppe namens `serveruser` und anschließend ein entsprechend eingeschränktes Benutzerkonto, das dieser Gruppe angehört:

```
# groupadd serveruser
# useradd -g serveruser -d /dev/null -s /bin/false serveruser
```

Der Befehl `userdel` löscht das angegebene Benutzerkonto, das heißt die entsprechenden Einträge aus `/etc/passwd` und `/etc/shadow`. Das Home-Verzeichnis des entsprechenden Users wird allerdings nicht gelöscht. Beispiel:

```
# userdel exuser
```

`groupadd` fügt eine neue Gruppe hinzu. Die wichtigste Option ist `-g GID`, mit deren Hilfe Sie die (ansonsten automatisch vergebene) Group-ID manuell einstellen können. Wie bei `useradd` dient in diesem Zusammenhang die Option `-o` dazu, eine eventuell bestehende Gruppe mit derselben Group-ID zu überschreiben. Hier ein einfaches Beispiel:

```
# groupadd druckuser
```

`passwd` dient zum Ändern von Passwörtern. Wenn Sie den Befehl ohne Argument aufrufen, dient er dem Ändern Ihres eigenen Passworts: Sie müssen zunächst Ihr altes und anschließend zweimal hintereinander das neue Passwort eingeben. Die Eingabe wird jeweils nicht angezeigt:

```
sascha@tux:~> passwd

Changing password for sascha.
Altes Passwort:
Neues Passwort:
Geben Sie das neue Passwort erneut ein:
Passwort geändert.
```

Eine weitere Möglichkeit, die `root` vorbehalten ist, besteht darin, `passwd` Username aufzurufen, um das Passwort eines anderen Benutzers zu ändern. Sie müssen das alte Passwort dieses Users nicht kennen, sondern werden nur aufgefordert, zweimal ein neues einzugeben.

Beachten Sie die weiter oben gegebenen Informationen über sichere Passwörter. Der Befehl `passwd` gibt zwar bei vielen zu einfachen Passwörtern die Warnmeldung `Bad Password. Too simple` aus, akzeptiert das Passwort aber trotzdem.

Eigentümer und Gruppe einer Datei ändern

Wie Sie wissen, gehört jede Datei einem Benutzer und einer Gruppe; dies bestimmt die Zugriffsrechte auf die Datei. Mit zwei einfachen Befehlen können Sie die Benutzer- und die Gruppen-ID ändern.

`chown` (»change owner«) weist den angegebenen Dateien einen neuen Eigentümer zu. Die Syntax des Befehls ist folgende:

```
chown User[:Gruppe] Dateimuster [Dateimuster ...]
```

Der User muss ein existierender Benutzer sein; außerdem können Sie diese Änderung nur durchführen, wenn Sie selbst Schreibrechte an dieser Datei haben. Das folgende Beispiel teilt die Datei *info* dem Benutzer `user` zu:

```
$ chown user info
```

Mithilfe der Syntax `User:Gruppe` können Sie den Eigentümer und die Gruppe einer Datei gleichzeitig ändern. Beispiel:

```
$ chown user:users test
```

`chgrp` (»change group«) ändert dagegen nur die Gruppe, zu der eine Datei gehört, und funktioniert ansonsten genau wie `chown`.

Die Option `-R` (großgeschrieben!) führt die gewünschte Änderung nicht nur im aktuellen Verzeichnis durch, sondern auch rekursiv in allen Unterverzeichnissen. Das folgende Beispiel überträgt den Besitz an allen Dateien im aktuellen Verzeichnis und der gesamten Unterverzeichnishierarchie auf den Benutzer `user` und die Gruppe `users`:

```
$ chown -R user:users .
```

Zugriffsrechte ändern

`chmod` (»change mode«) ändert die Zugriffsrechte für Dateien und Verzeichnisse. Das Konzept der Dateizugriffsrechte wurde bereits weiter oben angesprochen. Es gibt grundsätzlich zwei Möglichkeiten, Rechte für die gewünschten Dateien oder Verzeichnisse anzugeben: symbolisch oder numerisch.

Die symbolische Schreibweise verwendet zunächst einen Buchstaben für die Benutzerart, für die ein Recht geändert werden soll: `u` für den Eigentümer (`user`), `g` für die Gruppe (`group`), `o` für andere Benutzer (`others`) und `a` (oder gar nichts) für alle genannten auf einmal. Darauf folgt ein `+`, um ein bestimmtes Recht einzuräumen, ein `-`, um es zu entfernen, oder ein `=`, um *nur* die angegebenen Rechte zu setzen und die nicht erwähnten zu entfernen. Zum Schluss werden die eigent-

lichen Rechte selbst angeben: *r* für Lesen (read), *w* für Schreiben (write) und *x* für Ausführen (execute).

Um mehrere Rechte zu manipulieren, können Sie die entsprechenden Angabenblöcke durch Komma (aber ohne Leerzeichen) getrennt hinschreiben.

Die folgende Anweisung erlaubt beispielsweise allen Benutzern das Lesen der Datei *inhalt.txt*:

```
$ chmod a+r inhalt.txt
```

Numerische Angaben setzen dagegen den gesamten Rechteblock für die Datei auf einmal: Die Stellen einer dreistelligen Oktalzahl (gekennzeichnet durch eine vorangestellte Null, die allerdings optional ist) geben von links nach rechts die Zugriffsrechte für den Besitzer, die Gruppe und die anderen an. Der Wert jeder Stelle ist dabei die Summe der Rechte, die gewährt werden: 4 für Lesen, 2 für Schreiben und 1 für Ausführen.

Das folgende Beispiel erlaubt dem Eigentümer das Lesen, Schreiben und Ausführen, allen anderen nur das Lesen und Ausführen des Verzeichnisses *test*:

```
$ chmod 0755 test
```

Neben den Zugriffsrechten für die drei Benutzerarten gibt es noch drei spezielle Bits, die das allgemeine Verhalten einer Datei betreffen. Diese werden anstelle der führenden Null in der Rechtestelle addiert; es gibt aber auch spezielle symbolische Kürzel dafür. Im Einzelnen handelt es sich um folgende:

- Das *Setuid-Bit* (Wert 4) ändert die User-ID, unter der ein Programm ausgeführt wird – das heißt, die User-ID der Programmdatei und die UID der Ausführung (effektive User-ID oder EUID genannt) unterscheiden sich voneinander. Dies ermöglicht nichtprivilegierten Benutzern den Start eines Programms, das root-Operationen durchführt. Ein bekanntes Beispiel ist das Druckkommando *lpr*: Es kann von jedem Benutzer aufgerufen werden, schreibt aber in die Druckwarteschlangen unter */var/spool*, was eigentlich root vorbehalten ist.

Um das Setuid-Bit zu setzen, können Sie entweder eine Vier zur vordersten Stelle der numerischen Rechte addieren oder aber für den User das Kürzel *s* hinzufügen (*u+s*). Das folgende Beispiel erlaubt dem Besitzer des Programms *hallo* das Lesen, Schreiben und Ausführen, der Gruppe und dem Rest der Welt das Lesen und Ausführen und setzt zusätzlich das Setuid-Bit:

```
$ chmod 4755 hallo
$ ls -l hallo
-rwsr-xr-x 1 sascha users 6987 2008-02-01 21:10 hallo
```

Wie Sie sehen, wird die Anwesenheit des Setuid-Bits in der `ls`-Ausgabe dadurch gekennzeichnet, dass das `x` bei den Rechten des Eigentümers durch `s` ersetzt wird. Hier noch ein Beispiel, das Setuid zu den vorhandenen Rechten hinzufügt:

```
$ chmod u+s hallo
```

- ▶ Das *Setgid-Bit* (Wert 2) führt dieselbe Modifikation durch wie Setuid, allerdings für die Gruppe – hier ergibt sich eine effektive Group-ID (EGID). Gesetzt wird es durch Addieren einer 2 zur führenden Stelle der numerischen Rechteangabe oder mittels `g+s`. `ls -l` zeigt die Anwesenheit von Setgid an, indem das `x` der Gruppe durch ein `s` ersetzt wird. Hier ein Beispiel:

```
$ chmod g+s hallo2
```

```
$ ls -l hallo2
```

```
-rwxr-sr-x 1 sascha users 6987 2008-02-01 21:10 hallo
```

Eine besondere Bedeutung besitzt das Setgid-Bit, wenn es einem Verzeichnis zugewiesen wird: Neu erstellte Dateien gehören in diesem Fall nicht der primären Gruppe des aktuellen Benutzers, sondern der Gruppe des Verzeichnisses.

- ▶ Das *Sticky-Bit* (Wert 1) bedeutet für ausführbare Programme, dass diese nach der Ausführung im Speicher verbleiben dürfen. Das modernere Konzept der Daemons und Hintergrundprozesse ist besser, sodass diese Anwendung des Sticky-Bits heute praktisch keine Rolle mehr spielt. Wichtiger ist dagegen seine Bedeutung für Verzeichnisse: Wenn es für ein allgemein beschreibbares Verzeichnis wie `/tmp` gesetzt ist, kann ein normaler User darin nur noch diejenigen Dateien verschieben oder löschen, deren Eigentümer er ist. Gesetzt wird das Sticky-Bit durch `o+t` oder durch den Wert 1 in der ersten Stelle der numerischen Rechtestelle; in der Ausgabe von `ls -l` erscheint es als `t` anstelle des `x` in den Rechten der sonstigen Benutzer. Beispiel:

```
$ chmod o+t test
```

```
$ ls test
```

```
drwxr-xr-t 3 sascha users 120 2006 - 02 - 19 18:13 test
```

Die Rechte neu erstellter Dateien oder Verzeichnisse werden durch die sogenannte `umask` des aktuellen Benutzers beeinflusst. Ihren Wert können Sie durch folgenden Aufruf ermitteln:

```
$ umask
```

```
0022
```

Die Oktalzahl codiert die Zugriffsrechte im Prinzip genauso wie `chmod`, allerdings dient der Wert nicht dem Hinzufügen, sondern dem Ausschluss von Rechten:

Eine neue Datei erhält alle Rechte *mit Ausnahme* derjenigen, die in der `umask` stehen. Mathematisch-logisch gesehen werden die Rechte für neue Verzeichnisse und Dateien also nach folgender Formel festgelegt:

```
0777 AND (NOT umask)
```

Die für normale Benutzer übliche `umask 0022` setzt also alle Rechte außer `g+w` und `o+w`, mit anderen Worten `755` oder `a=rX,u+w`. Um Ihre `umask` zu ändern, geben Sie einfach den Befehl `umask` und den gewünschten neuen Wert ein. Das folgende Beispiel erstellt neue Dateien künftig mit *allen* Rechten (`0777`):

```
$ umask 0000
```

4.3.4 Kommandos zur Textanzeige und -manipulation

Viele der Arbeiten, die Sie im Betriebssystem durchführen, haben in irgendeiner Weise mit der Manipulation von Zeichenketten und Textdateien zu tun. In diesem Unterabschnitt werden einige der wichtigsten Befehle vorgestellt, die Ihnen die Arbeit mit solchen Dateien ermöglichen. Um die eigentliche Eingabe und Bearbeitung von Textdateien geht es übrigens exklusiv in Kapitel 10, »Klassische Texteditoren«.

Der bereits angesprochene Befehl `echo` gibt den gesamten Text bis zum nächsten Zeilenumbruch aus und eignet sich daher vor allem für Shell-Skripte (siehe Kapitel 17, »System-Automatisierung«). Beispiel:

```
$ echo Hallo Leute
Hallo Leute
```

Wenn Sie einem String in einer `echo`-Anweisung ein Dollarzeichen voranstellen, wird dieser als Variable gewertet. Das folgende Beispiel gibt den Inhalt der Umgebungsvariablen `HOME`, das heißt das Home-Verzeichnis des aktuellen Benutzers, aus:

```
$ echo Mein Home-Verzeichnis ist $HOME
Mein Home-Verzeichnis ist /home/sascha
```

Mithilfe der bereits erwähnten Backticks können Sie auch die Ausgabe von Befehlen in den Ausgabestring aufnehmen. Das folgende Beispiel ermittelt mithilfe der weiter unten näher besprochenen Befehle `grep` und `wc` die Anzahl der direkt im aktuellen Verzeichnis enthaltenen Unterverzeichnisse und gibt sie als benutzerfreundlichen String aus:

```
$ echo `pwd` enthält `ls -l |grep ^d |wc -l` \
> Unterverzeichnisse.
/home/sascha enthält 9 Unterverzeichnisse.
```

`pwd` gibt, wie Sie wissen, den Namen des aktuellen Arbeitsverzeichnisses aus. Der Filter `grep ^d` sucht diejenigen Zeilen in `ls -l`, die mit einem `d` beginnen (das erste Zeichen in der Rechteste, das Verzeichnisse kennzeichnet). `wc -l` zählt schließlich die Anzahl der Zeilen in einer Textdatei oder in der Standardeingabe.

Textdateien ausgeben

`cat` (»concatenate«) zeigt einfach den Inhalt der angegebenen Textdatei oder -dateien an. Wenn Sie eine durch Leerzeichen getrennte Liste von Dateien oder ein Muster angeben, werden die Inhalte aller genannten Dateien hintereinander angezeigt. Dies ist übrigens eine einfache Möglichkeit, mehrere Textdateien in eine einzige zusammenzukopieren. Die folgende Anweisung schreibt die Dateien *teil1* und *teil2* in eine neue Datei namens *kapitel*:

```
$ cat teil1 teil2 > kapitel
```

Sie können `cat` mithilfe der Ausgabeumleitung und mit einem HIER-Dokument sogar als einfachen Editor für eine neue Textdatei verwenden – allerdings können Sie die einzelnen Zeilen nach dem Abschluss durch  nicht mehr ändern. Die folgende Anweisung startet die Eingabe der Datei *neu.txt*, der Text *ENDE* allein in einer Zeile schließt sie ab:

```
$ cat > neu.txt <<ENDE
> Neuer Text
> Noch mehr Text
> ENDE
```

Alternativ können Sie die HIER-Umleitung auch weglassen und die Eingabe mit  (EOF) abschließen.

In vielerlei Hinsicht mächtiger und praktischer als `cat` ist das Kommando `less`, die erweiterte GNU-Version des klassischen UNIX-Pagers `more`. Das Programm gibt seine Eingabedaten bildschirmseitenweise aus. Das folgende Beispiel gibt die Datei *roman* auf diese Weise aus:

```
$ less roman
```

Wenn das untere Ende des Bildschirms beziehungsweise des Terminalfensters erreicht ist, erscheint ein entsprechender Hinweis. An dieser Stelle haben Sie verschiedene Möglichkeiten, unter anderem folgende:

- ▶ Die Leertaste blättert eine ganze Bildschirmseite weiter.
- ▶  blättert nur eine einzige Zeile weiter.
- ▶  blättert eine Bildschirmseite zurück.

- ▶ `/` Suchbegriff `←` sucht vorwärts nach dem angegebenen regulären Ausdruck (siehe die Beschreibung von `grep` weiter unten).
- ▶ `?` Suchbegriff `←` sucht ebenfalls nach einem regulären Ausdruck, allerdings nach oben.
- ▶ `N` findet das nächste Vorkommen des vorherigen Suchbegriffs in der aktuellen Suchrichtung.
- ▶ `↕+N` sucht ebenfalls nach einem weiteren Vorkommen des regulären Ausdrucks, kehrt dabei aber die Suchrichtung um.
- ▶ `Q` beendet `less`.

Weitere Möglichkeiten wurden im vorigen Kapitel für den Hilfe-Reader `man` vorgestellt, dessen Bedienung `less`-kompatibel ist.

Statt eine oder mehrere Dateien als Parameter anzugeben, wird `less` auch häufig über eine Pipe zur seitenweisen Ausgabe der Ergebnisse anderer Befehle eingesetzt. Das folgende Beispiel gibt den Inhalt der ausführlichen Verzeichnisanzeige `ls -l` seitenweise aus:

```
$ ls -l |less
```

Zur schnellen Überprüfung von Textdateien sind die Kommandos `head` und `tail` geeignet. Sie geben die ersten beziehungsweise letzten zehn Zeilen der angegebenen Datei aus. Mit `head` können Sie insbesondere kurz in eine Datei »hineinschnuppern«, um sich einen Eindruck von ihrem Inhalt zu verschaffen. Beispiel:

```
$ head chapters.txt
1. Einführung
2. Installation
3. Grundkonfiguration des Systems
4. Arbeiten mit der Shell
5. Grafische Oberflächen
6. Desktop-Software
7. OpenOffice.org 2.0
8. Bilder und Grafiken
9. Datenträger und Dateisysteme
10. Klassische Texteditoren
```

`tail` dient dagegen vorrangig der Suche nach Fehlern in Logdateien, denn es ist davon auszugehen, dass der Grund für ein neu aufgetretenes Problem ganz unten in einem Protokoll steht. Das folgende Beispiel gibt die letzten zehn Zeilen der Haupt-Systemprotokolldatei `/var/log/messages` aus (das darf übrigens nur `root`):

```
# tail /var/log/messages
Feb 20 12:33:35 tux shadow[26771]: running USERDEL_POSTCMD command -
```

```

script=/usr/sbin/userdel-post.local, account=hanz, uid=1001,
gid=100, home=/home/exuser, by=0
Feb 20 12:50:39 tux syslog-ng[2256]: STATS: dropped 0
Feb 20 12:55:59 tux passwd[26801]: User sascha: Fehler bei
Authentifizierung.
Feb 20 12:55:59 tux passwd[26801]: password change failed, pam error
7 - account=sascha, uid=1000, by=1000
Feb 20 12:56:16 tux passwd[26804]: password changed -
account=sascha, uid=1000, by=1000
Feb 20 13:50:39 tux syslog-ng[2256]: STATS: dropped 0
Feb 20 14:50:40 tux syslog-ng[2256]: STATS: dropped 0
Feb 20 15:50:40 tux syslog-ng[2256]: STATS: dropped 0
Feb 20 16:17:12 tux sudo:  sascha : TTY=pts/4 ; PWD=/home/sascha ;
USER=root ; COMMAND=/usr/bin/tail /var/log/messages
Feb 20 16:17:59 tux sudo:  sascha : TTY=pts/4 ; PWD=/home/sascha ;
USER=root ; COMMAND=/usr/bin/tail /var/log/messages

```

Eine besondere Bedeutung besitzt die `tail`-Option `-f`: Nach der Ausgabe der letzten zehn Zeilen bleibt das Programm aktiv und gibt weiterhin jede neue Zeile aus. Auf diese Weise könnten Sie beispielsweise ein separates Terminalfenster verwenden, um die aktuellen Ereignisse aus der Logdatei jederzeit live lesen zu können.

Wenn Sie mehr oder weniger als zehn Zeilen erhalten möchten, können Sie die Option `-Anzahl` verwenden. Das folgende Beispiel gibt die letzten zwanzig Zeilen der Hauptlogdatei aus; anschließend bleibt das Programm aktiv und zeigt jede weitere Zeile an:

```
# tail -20 -f /var/log/messages
```

Suche nach regulären Ausdrücken mit `grep`

`grep` (»general regular expression print«) sucht in Dateien oder in seiner Eingabe nach Mustern und gibt nur diejenigen Zeilen aus, die das entsprechende Muster enthalten. Bei den verwendeten Mustern handelt es sich um sogenannte reguläre Ausdrücke (regular expressions). Diese mächtige Syntax für die Formulierung von Suchmustern wird in zahlreichen Programmiersprachen, Editoren und Tools verwendet. In Kapitel 17, »System-Automatisierung«, werden die noch mächtigeren `RegExp`-Optionen der Programmiersprache Perl genauer erläutert.

Wenn Sie beispielsweise in der Datei `test` nach Zeilen suchen möchten, die das Wort `hallo` enthalten, funktioniert dies folgendermaßen:

```
$ grep hallo test
```

Alternativ wird `grep` häufig über eine Pipe als Filter eingesetzt. Wenn Sie zum Beispiel alle Dateien im aktuellen Verzeichnis angezeigt haben möchten, deren Name mit `a` beginnt, können Sie Folgendes eingeben:

```
$ ls |grep ^a
```

Beachten Sie, dass es bei den Mustern gewisse Unterschiede zu den Dateimustern der meisten Befehle gibt – beispielsweise steht das `*` allein nicht als Platzhalter für beliebig viele Zeichen, sondern lässt beliebig viele Vorkommen des links daneben stehenden Zeichens oder Teilausdrucks zu. Tabelle 4.1 zeigt eine Übersicht über die wichtigsten verfügbaren RegExp-Bestandteile.

Muster (Beispiel)	Erläuterung
<code>abc</code>	der Text <code>abc</code>
<code>[abc]</code>	Eines der Zeichen <code>a</code> , <code>b</code> oder <code>c</code>
<code>[a-z]</code>	Eines der Zeichen von <code>a</code> bis <code>z</code>
<code>[a-mz0 - 9]</code>	Eines der Zeichen von <code>a</code> bis <code>m</code> oder <code>z</code> oder eine der Ziffern <code>0</code> bis <code>9</code>
<code>[^abc]</code>	Ein beliebiges Zeichen <i>außer</i> den angegebenen
<code>.</code>	Ein beliebiges Zeichen
<code>?</code>	Das davor stehende Muster einmal oder keinmal
<code>*</code>	Das davor stehende Muster beliebig oft
<code>+</code>	Das davor stehende Muster einmal oder öfter
<code>{2}</code>	Das davor stehende Muster genau zweimal
<code>{2, 5}</code>	Das davor stehende Muster mindestens zweimal, höchstens fünfmal
<code>^</code>	Anfang des Ausdrucks
<code>\$</code>	Ende des Ausdrucks

Tabelle 4.1 Die wichtigsten RegExp-Muster für `grep`

Wenn Sie irgendeines der Zeichen aus der Tabelle als Literal benötigen, also als tatsächliches Zeichen in einem Text, müssen Sie ihm einen Backslash (`\`) voranstellen. `\+` steht beispielsweise für ein Pluszeichen und `\.` für einen Punkt. Derartige Konstrukte werden in der Shell und in vielen Programmiersprachen als *Escape-Sequenzen* bezeichnet.

Angenommen, Sie suchen in einem Text nach deutschen Postleitzahlen. Das passende Muster lautet `[0-9]{5}`, weil genau fünf Ziffern (Zeichen zwischen 0 und 9) benötigt werden. Wenn Sie sicher sind, dass die Postleitzahl jeweils am Anfang der Zeile steht, können Sie präziser `^[0-9]{5}` schreiben.

Wichtig ist, dass `*` und `?` nicht dasselbe bedeuten wie bei den Dateimustern: Sie beziehen sich stets auf das links daneben stehende Muster und geben an, wie oft es vorkommen darf. Das folgende Beispiel zeigt, wie Sie in der Datei `adressen.txt` nach Personen suchen können, die »Meier« heißen, und zwar in allen erdenklichen Schreibweisen:

```
$ grep M[ae][iy]e?r adressen.txt
```

Der reguläre Ausdruck bedeutet: Zuerst kommt ein `M`, dann ein `a` oder `e`, anschließend ein `i` oder `y`, dann ein `e` oder auch nicht und zum Schluss ein `r`. Dies findet die Varianten »Maier«, »Mayer«, »Mayr«, »Meier« und »Meyer« sowie drei weitere, die aber wohl nicht häufig in einer Adressliste auftauchen (nämlich »Mair«, »Meir« und »Meyr«).

Vergleichen, sortieren und Statistiken erstellen

Das Kommando `diff` vergleicht die Inhalte zweier Textinhalte miteinander. Als Argumente werden die Namen der beiden Dateien genannt; die Ausgabe besteht aus den Zeilen, die in den beiden Dateien unterschiedlich sind. Dies ermöglicht die Analyse der Unterschiede zwischen verschiedenen Versionen eines Dokuments. Außerdem können Sie eine `diff`-Datei als Update für eine Datei auf eine neuere Version verbreiten. Mit dem Kommando `patch` kann ein Benutzer sie dann auf die alte Datei anwenden.

Betrachten Sie als Beispiel die beiden folgenden Dateien:

```
$ cat text1
Diese Zeile ist in beiden Dateien gleich.
Diese Zeile ist unterschiedlich.
$ cat text2
Diese Zeile ist in beiden Dateien gleich.
Diese Zeile ist anders.
Diese Zeile gibt es nur in einer der beiden Dateien.
```

Der Vergleich zwischen den beiden Dateien liefert folgendes Ergebnis:

```
$ diff text1 text2
2c2,3
< Diese Zeile ist unterschiedlich.
---
```

- > Diese Zeile ist anders.
- > Diese Zeile gibt es nur in einer der beiden Dateien.

Identische Zeilen werden gar nicht aufgeführt. Unterschiedliche Zeilen tauchen zweimal auf, mit vorangestelltem < für die erste und mit > für die zweite Datei. Zeilen, die nur in einer der beiden Dateien stehen, beginnen ebenfalls mit dem entsprechenden Zeichen.

Der Befehl `sort` sortiert die Zeilen einer Textdatei oder der Standardeingabe. Standardmäßig wird in Zeichensatzreihenfolge, das heißt mehr oder weniger alphabetisch mit Unterscheidung von Groß- und Kleinschreibung, sortiert. Eine wichtige Option ist `-d` oder `--dictionary-order`; sie beschränkt die Unterscheidung auf alphanumerische Zeichen und Leerzeichen. `-n` oder `--numeric-sort` sortiert in numerischer Reihenfolge. `-b` (`--ignore-leading-blanks`) schließlich ignoriert führende Leerzeichen.

Das folgende Beispiel sortiert das aktuelle Verzeichnislisting mit Standardoptionen:

```
$ ls |sort
```

In Zusammenarbeit mit `sort` ist manchmal der Befehl `uniq` hilfreich, der bei mehrfach aufeinander folgenden identischen Zeilen nur eine ausgibt. Hier ein Beispiel, das alle unterschiedlichen Zeilen aus der Datei `wortliste.txt` sortiert ausgibt:

```
$ cat wortliste.txt
Haus
Maus
Mann
Buch
Maus
$ sort wortliste.txt |uniq
Buch
Haus
Mann
Maus
```

Das Kommando `wc` (»word count«) zählt standardmäßig die Zeichen, Wörter und Zeilen in einer Textdatei. Wenn Sie nur einen dieser Werte benötigen, geben Sie eine der Optionen `-c` (characters, Zeichen), `-w` (words, Wörter) oder `-l` (lines, Zeilen) an. Hier ein einfaches Beispiel für die oben gezeigte Datei `wortliste.txt`:

```
$ wc wortliste.txt
5 5 25 wortliste.txt
```

Sie können den Befehl auch über eine Pipe auf eine beliebige Ausgabe anwenden. Das folgende Beispiel zählt die Zeilen aus `ls -l` und damit die Anzahl der Einträge im aktuellen Verzeichnis:

```
$ ls -l|wc -l
94
```

4.3.5 Systeminformation und -verwaltung

In diesem letzten Unterabschnitt lernen Sie noch einige wichtige Anweisungen kennen, mit denen Sie Informationen über Ihr Betriebssystem erhalten und einfache Verwaltungsaufgaben durchführen können.

Der Befehl `uname` liefert genaue Informationen über das laufende Betriebssystem. Wichtiger als auf Ihrem eigenen Rechner (wo Sie normalerweise wissen sollten, was Sie installiert haben), ist diese Information, wenn Sie auf einem fremden Computer arbeiten – oder wenn Sie ein Skript schreiben, das sich auf verschiedenen Systemvarianten unterschiedlich verhält. Hier ein Beispiel:

```
$ uname -a
Linux tux 2.6.25.5-1.1-default #1 SMP 2008-06-07 01:55:22 +0200 x86_
64 x86_64 x86_64 GNU/Linux
```

Die folgende Liste beschreibt, welche Informationen Sie dieser Ausgabe entnehmen und mithilfe welcher Optionen Sie sie gegebenenfalls einzeln ermitteln können:

- ▶ *Kernel-Name* (Option `-s`), hier `Linux`
- ▶ *Hostname* (`-n`), im Beispiel `tux`
- ▶ *Kernel-Release* (`-r`), in diesem Fall `2.6.25.5-1.1-default`
- ▶ *Kernel-Version* (`-v`), hier `#1 SMP 2008-06-07 01:55:22 +0200`
- ▶ *Maschine* (`-m`), im vorliegenden Beispiel `x86_64`
- ▶ *Prozessor* (`-p`), hier ebenfalls `x86_64`
- ▶ *Hardwareplattform* (`-i`), in diesem Beispiel `x86_64`
- ▶ *Betriebssystem* (`-o`), hier `GNU/Linux`

`uptime` zeigt an, wie lange der Rechner bereits in Betrieb ist, wie viele Benutzer zurzeit aktiv sind und wie stark das System belastet ist:

```
$ uptime
7:57pm an 6 Tage 12:08, 5 Benutzer, Durchschnittslast: 0,00, 0,00, 0,0
```

`who` zeigt an, welche Benutzer zurzeit an welchen Terminals angemeldet sind:

```
# who
sascha  :0          2009-09-14 07:49 (console)
sascha  pts/0       2009-09-14 07:50
sascha  pts/1       2009-09-14 09:14
sascha  pts/2       2009-09-18 11:30
sascha  pts/4       2009-09-15 11:53
```

Der Befehl `ps` gibt eine Tabelle aller laufenden Prozesse aus. Wenn Sie keine Parameter angeben, werden nur die Prozesse des aktuellen Terminals und Benutzers angezeigt. Für eine vollständige Liste *aller* Prozesse müssen Sie dagegen `ps aux` verwenden. Hier ein kurzer Ausschnitt als Beispiel:

```
$ ps aux
USER  PID %CPU %MEM    VSZ   RSS TTY  STAT START   TIME COMMAND
[... ]
root   3056  0.0  0.1   3024   348 ?    Ss   Feb14   0:00 /opt/kde3/bin/k
root   3081  0.6  6.5  41424 16756 tty7  Ss+  Feb14  57:18 /usr/X11R6/bin/
root   3090  0.0  0.1   3344   420 ?    S    Feb14   0:00 -:0
mdnsd  3422  0.0  0.1   1976   384 ?    Ss   Feb14   0:00 /usr/sbin/mdnsd
sascha 3452  0.0  0.1   3952   372 ?    Ss   Feb14   0:00 /bin/sh /usr/X1
nobody 3466  0.0  0.0   1564   188 ?    Ss   Feb14   0:00 /sbin/portmap
root   3467  0.0  0.1   9856   320 ?    S<s  Feb14   0:00 /sbin/auditd
[... ]
```

Von links nach rechts enthält jede Zeile folgende Informationen:

- ▶ User-ID (USER): Die User-ID, unter der der jeweilige Prozess ausgeführt wird, hier etwa `root`, `sascha` oder `nobody`.
- ▶ Prozess-ID (PID): Die Nummer des Prozesses (Prozess-ID oder kurz PID) ist die wichtigste Information, die Sie der Tabelle entnehmen können, weil Sie sie verwenden können, um einem Prozess mithilfe des Befehls `kill` ein Signal zu senden.
- ▶ Prozentuale CPU-Nutzung (%CPU)
- ▶ Prozentuale Speichernutzung (%MEM)
- ▶ Virtuelle Prozessgröße (VSZ)
- ▶ Residente Speichergröße (RSS)
- ▶ Terminal, von dem der Prozess gestartet wurde (TTY) – oft ? für Daemons oder Systemprozesse
- ▶ Status des Prozesses (STAT); die Hauptabkürzungen bedeuten Folgendes: R – Prozess wird ausgeführt (running), S – Prozess wartet auf Ereignis (sleeping), T – Prozess gestoppt (terminated), D – Prozess wartet auf der Festplatte (disk), W – der Prozess ist ausgelagert (sWap). Die Kleinbuchstaben und Sonderzeichen stehen für die Veränderung des Status' über die Zeit.

- ▶ Startzeitpunkt (START)
- ▶ Verbrauchte CPU-Zeit (TIME)
- ▶ Pfad des ausführbaren Programms (COMMAND)

Eine interessante Variante von `ps` ist `pstree`: Statt ausführlicher Informationen über die einzelnen Prozesse werden nur ihre Namen angezeigt, dafür aber in einem Baumdiagramm. Auf diese Weise können Sie die Hierarchie von Elternprozessen und ihren Kindern ermitteln.

Für Administratoren ist die interessanteste Variante allerdings `top`. Hier wird eine Statistik der aktivsten Prozesse angezeigt. Die Leertaste aktualisiert die Statistik. Die Taste `[H]` zeigt einen Hilfsbildschirm mit allen verfügbaren Kommandos an. Über `[K]` können Sie das im Anschluss besprochene Kommando `kill` aufrufen. `[Q]` verlässt das Programm.

Der Befehl `kill` dient dazu, einem Prozess ein Signal zu senden. Signale sind eine wichtige Möglichkeit der Interprozesskommunikation (IPC); Genauer erfahren Sie in Kapitel 18. Das gewünschte Signal wird dabei hinter der Option `-s` angegeben, entweder numerisch oder mit seinem Namen; noch kürzer geht es mit `-SIGNAL`. Wichtige Signale sind beispielsweise `TERM` zum regulären Beenden, `KILL` für einen erzwungenen Abbruch oder `INT` für eine Unterbrechung (Interrupt). Wenn Sie kein Signal angeben, wird automatisch `TERM` gesendet.

Das folgende Beispiel ermittelt zunächst die PID des Webservers Apache (der eigentlich `httpd` heißt und dessen Hauptprozess als `root` ausgeführt wird) und sendet diesem ein `TERM`-Signal, um ihn zu beenden:

```
# ps aux |grep httpd
root    14870  0.0  0.4 62888 1112 ?        Ss   Feb16   0:00 /usr/
sbin/httpd2-prefork -f /etc/apache2/httpd.conf
wwwrun  14871  0.0  0.0 62888  244 ?        S    Feb16   0:00 /usr/
sbin/httpd2-prefork -f /etc/apache2/httpd.conf
wwwrun  14872  0.0  0.0 62888  224 ?        S    Feb16   0:00 /usr/
sbin/httpd2-prefork -f /etc/apache2/httpd.conf
wwwrun  14873  0.0  0.0 62888  224 ?        S    Feb16   0:00 /usr/
sbin/httpd2-prefork -f /etc/apache2/httpd.conf
wwwrun  14874  0.0  0.0 62888  224 ?        S    Feb16   0:00 /usr/
sbin/httpd2-prefork -f /etc/apache2/httpd.conf
wwwrun  14875  0.0  0.0 62888  224 ?        S    Feb16   0:00 /usr/
sbin/httpd2-prefork -f /etc/apache2/httpd.conf
sascha 28114  0.0  0.2  2816   748 pts/4    S+   19:40   0:00 grep
httpd

# kill -TERM 14870
```

Der Prozess wird durch seine numerische PID angegeben, wie sie von `ps` zurückgegeben wurde. Als gewöhnlicher Benutzer können Sie nur Prozesse beenden, die unter Ihrer User-ID laufen, für alle anderen benötigen Sie `root`-Rechte.

Die meisten Daemons und Systemprogramme besitzen eine sogenannte PID-Datei. Darin steht während der Laufzeit ihre aktuelle PID, und sobald das Programm ordnungsgemäß beendet wird, löscht es die Datei. Sie können diese Datei verwenden, um die PID zu ermitteln und durch Backticks an `kill` weiterzugeben. Im Fall des Apache-Pakets von openSUSE ist die passende Datei `/var/run/httpd2.pid`, sodass der entsprechende Befehl so lautet:

```
# kill -TERM `cat /var/run/httpd2.pid`
```

Die Option `-l` liefert eine Liste aller verfügbaren Signale. Einige von ihnen werden in Kapitel 18, »Hinter den Kulissen«, näher erläutert. Beispiel (hier wurden die Tabs entfernt, damit die Ausgabe in den Satzspiegel passt):

```
$ kill -l
 1) SIGHUP   2) SIGINT   3) SIGQUIT  4) SIGILL
 5) SIGTRAP  6) SIGABRT  7) SIGBUS   8) SIGFPE
 9) SIGKILL 10) SIGUSR1 11) SIGSEGV 12) SIGUSR2
13) SIGPIPE 14) SIGALRM 15) SIGTERM 16) SIGSTKFLT
17) SIGCHLD 18) SIGCONT 19) SIGSTOP 20) SIGTSTP
21) SIGTTIN 22) SIGTTOU 23) SIGURG  24) SIGXCPU
25) SIGXFSZ 26) SIGVTALRM 27) SIGPROF 28) SIGWINCH
29) SIGIO   30) SIGPWR  31) SIGSYS  34) SIGRTMIN
35) SIGRTMIN+1 36) SIGRTMIN+2 37) SIGRTMIN+3 38) SIGRTMIN+4
39) SIGRTMIN+5 40) SIGRTMIN+6 41) SIGRTMIN+7 42) SIGRTMIN+8
43) SIGRTMIN+9 44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12
47) SIGRTMIN+13 48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14
51) SIGRTMAX-13 52) SIGRTMAX-12 53) SIGRTMAX-11 54) SIGRTMAX-10
55) SIGRTMAX-9  56) SIGRTMAX-8  57) SIGRTMAX-7  58) SIGRTMAX-6
59) SIGRTMAX-5  60) SIGRTMAX-4  61) SIGRTMAX-3  62) SIGRTMAX-2
63) SIGRTMAX-1  64) SIGRTMAX
```

Interessant ist zudem das Kommando `killall`. Es sendet ein Signal (standardmäßig ebenfalls `TERM`) an alle Prozesse, die den angegebenen Namen besitzen. Das folgende Beispiel beendet alle Prozesse des Datenbankservers MySQL:

```
# killall mysqld
```

`shutdown` fährt das Betriebssystem herunter oder startet es neu. Für das Herunterfahren wird die Option `-h` (`halt`) verwendet, für einen Neustart `-r` (`restart`). Dahinter wird angegeben, wann die Aktion durchgeführt werden soll. Dafür stehen drei mögliche Optionen zur Verfügung:

- ▶ als Uhrzeit im Format hh:mm, zum Beispiel 13:45
- ▶ als Anzahl von Minuten in der Schreibweise +m, etwa +5
- ▶ now (jetzt sofort)

Für den Befehl benötigen Sie übrigens root-Rechte. Das folgende Beispiel fährt den Rechner sofort herunter:

```
# shutdown -h now
```

Das Kommando `date` dient der Ausgabe beziehungsweise der Änderung von Systemdatum und -uhrzeit.

Wenn Sie Datum und Uhrzeit einfach lesen möchten, genügt die Eingabe von `date`. Alternativ können Sie hinter einem Pluszeichen ein Format angeben, in dem eine Reihe von Formatangaben für die einzelnen Komponenten von Datum und Uhrzeit stehen. Alle diese Angaben entstammen der C-Bibliotheksfunktion `strftime()` und beginnen mit einem Prozentzeichen. Die wichtigsten von ihnen werden in Tabelle 4.2 aufgelistet; in der Kommandoreferenz in Kapitel 19 finden Sie die vollständige Liste.

Formatangabe	Bedeutung
%d	Tag im Monat (01 bis 31)
%m	numerischer Monat
%y	zweistellige Jahreszahl
%Y	vierstellige Jahreszahl
%a	Kurzfassung des Wochentags (zum Beispiel Tue)
%A	Ausgeschriebener Wochentag (etwa Tuesday)
%H	Stunden im 24-Stunden-Format
%I	Stunden im 12-Stunden-Format
%p	AM und PM für das Zwölf-Stunden-Format
%M	Minuten
%S	Sekunden

Tabelle 4.2 Die wichtigsten Formatangaben für den `date`-Befehl

Sehen Sie sich als Beispiel den folgenden Befehl an:

```
$ date +"Es ist jetzt %H:%M:%S Uhr."
```

Die Ausgabe sieht etwa folgendermaßen aus:

```
Es ist jetzt 17:52:31.
```

Sie können die Bestandteile von Datum und Uhrzeit also in einem beliebigen Ausgabebetext verwenden.

Wenn Sie Datum und Uhrzeit ändern möchten, müssen Sie statt des Formats eine Zeitangabe hinzufügen. Eine solche Angabe hat das folgende Format:

```
MMDDhhmm[YY]YY[.ss]
```

Es werden also nacheinander der zweistellige Monat, der zweistellige Tag, die zweistellige Stunde im 24-Stunden-Format, die zweistellige Minute, das zwei- oder vierstellige Jahr und optional die durch einen Punkt abgetrennten zweistelligen Sekunden angegeben. Das folgende Beispiel setzt Datum und Uhrzeit auf den 15. September 2009, 13:47 Uhr:

```
$ date 091513472009
```

4.4 Zusammenfassung

Kein Zweifel: Wer mit der Shell umgehen kann, beherrscht das Betriebssystem. Deshalb legt dieses Buch ein deutliches Gewicht auf dieses Thema: Es wird recht früh im Buch und ziemlich ausführlich behandelt. In späteren Kapiteln lernen Sie weitere Aspekte und Kommandos der Shell kennen.

Um die Shell zu verstehen, müssen Sie zunächst das Linux-Dateisystem durchblicken. Alle Verzeichnisse – egal auf welchem Volume oder auf welcher Partition sie liegen – befinden sich unter der Wurzel /. Die meisten Verzeichnisse haben dabei besondere Aufgaben. Beispielsweise liegen unter /home die Home-Verzeichnisse der normalen Benutzer und unter /usr die normalen Anwendungsprogramme. Jedes Programm und jedes Verzeichnis gehört einem bestimmten Benutzer und einer bestimmten Gruppe; es existieren jeweils getrennte Zugriffsrechte (Lesen, Schreiben und Ausführen) für den Besitzer, die Gruppe und alle anderen Benutzer.

Die Shell können Sie entweder auf einer Textkonsole oder im Terminalfenster einer grafischen Oberfläche aufrufen; im ersteren Fall müssen Sie sich zunächst anmelden. Anschließend steht Ihnen je nach Konfiguration eine der diversen, unterschiedlich leistungsfähigen Shells zur Verfügung, auf einem Linux-System meistens die besonders moderne `bash`. Durch Features wie Befehlsvervollständigung, History und viele nützliche Tastenkürzel erleichtert sie Ihnen die Arbeit enorm.

Ihre eigentliche Stärke entfaltet die Shell aber erst durch externe Befehle, das heißt durch die Systemprogramme. Diese dienen als kleine Einzelbausteine, die sich mithilfe von Befehlsverknüpfungen, Pipes und anderen Hilfsmitteln zu maßgeschneiderten Lösungen kombinieren lassen. In diesem Kapitel haben Sie bereits viele dieser Befehle aus den Bereichen Datei- und Verzeichnisverwaltung, Textbearbeitung und Systemadministration kennengelernt. In späteren Kapiteln dieses Buches werden weitere nützliche Kommandos behandelt; in Kapitel 19 finden Sie schließlich eine »Referenz der Shell- und Systembefehle«.

*Mit unserem Urteil ist es wie mit unseren Uhren. Nicht zwei gehen
genau gleich, und doch glaubt jeder der seinigen.
– Alexander Pope*

5 Grafische Oberflächen

Betriebssysteme wie Windows oder Mac OS X sind jeweils mit genau einer grafischen Benutzeroberfläche (englisch »Graphical User Interface« oder kurz GUI) ausgestattet, die zudem integrativer Bestandteil des Systems ist; substanzielle Änderungen können Sie nur durch Drittanbieter-Erweiterungen vornehmen.

Unter Linux ist die Situation völlig anders. Erstens kann das Betriebssystem selbst auch ohne grafische Oberfläche arbeiten, und zweitens sind die verfügbaren GUIs modular aufgebaut: Auf den sogenannten X-Server (Thema des ersten Abschnitts in diesem Kapitel) setzen viele verschiedene Window-Manager oder Desktops auf, zwischen denen Sie sich entscheiden können. Die beiden wichtigsten – KDE und GNOME – werden relativ ausführlich behandelt; danach folgt ein kurzer Überblick über die restlichen. Und wemgleich mancher mit seinem persönlichen Lieblings-Desktop¹ so umgehen mag wie nach dem Motto dieses Kapitels, schadet es nie, auch mal einen anderen auszuprobieren.

5.1 Das X Window System

Die Grundlage aller Desktops und grafischen Benutzeroberflächen unter Linux und anderen UNIX-Systemen bildet seit nunmehr 25 Jahren ein sogenannter *X Window Server* oder kurz *X-Server*. Dieser sorgt für die Darstellung der grundlegenden Grafikelemente; die Fenster und sonstigen Steuerelemente werden vom Window-Manager oder Desktop bereitgestellt, der auf den X-Server aufsetzt.

Das X Window System wird als Server bezeichnet, weil es seine Dienstleistung lokalen, aber auch entfernten Anwendungsprogrammen zur Verfügung stellen kann: Wenn Sie per Fernzugriff, zum Beispiel über SSH (siehe Kapitel 14, »Datei- und Verzeichnisdienstserver«), auf einem anderen Rechner arbeiten, kann ein

¹ Mein eigener ist seit Jahren KDE (wemgleich Version 4 mich bisher nicht vollends überzeugt hat), aber ohne den besagten Fanatismus.

Programm, das auf dem Fremdrechner läuft, den X-Server Ihres Rechners zur Darstellung seiner Bedienelemente nutzen. Die entfernte Anwendung (aber auch jede lokale, die vom X-Server Gebrauch macht) wird dabei als *X-Client* bezeichnet.

Ausgangspunkt der Entwicklung von X Window war das Projekt Athena, das 1984 als Kooperation von Digital Equipment, IBM und dem MIT gestartet wurde. X11 – die Grundversion, auf die bis heute im Wesentlichen alle X-Server aufbauen – wurde bereits im Herbst 1987 veröffentlicht. Bis 2005 basierten die verbreiteten X Window-Implementierungen auf der Release X11R6 aus dem Jahr 1994; erst Ende 2005 wurde die neue Referenzimplementierung X11R7 veröffentlicht. Die aktuelle Version ist 7.5 von Oktober 2009.

Die Standards für die Weiterentwicklung des X Window Systems setzt eine Organisation namens *X.org Foundation*; ihre Website ist unter <http://www.x.org> zu finden (diese Second-Level-Domain mit nur einem Zeichen ist eine absolute Ausnahme).

Die meisten Linux-Distributionen enthielten früher eine freie X-Implementierung namens XFree86 – insbesondere, weil der Original-X-Server damals kommerziell war. Der Name ist eine Anspielung auf die Intel-Prozessorfamilie »three-eighty-six«, wengleich diese Variante später auch auf andere Prozessorarchitekturen portiert wurde. Im Jahr 2003 wurde die Weiterentwicklung von XFree86 allerdings vorübergehend eingestellt, sodass Linux seitdem auf die Referenzimplementierungen von X.org umgestiegen ist. Daneben gibt es zahlreiche weitere Varianten von X-Servern, sowohl freie als auch kommerzielle.

Eine ganz neue – ebenfalls von X.org (in Kooperation mit Novell) entwickelte – Variante von X Window ist *Xgl*. Dieser X-Server basiert auf der Grafikkbibliothek OpenGL und ist so in der Lage, die in moderne Grafikkarten eingebaute Hardware-Unterstützung für 3-D und andere Spezialeffekte zu nutzen. Dies verbessert die Performance von Spielereien wie halbtransparenten Fenstern oder der Anordnung von Bedienelementen auf virtuellen 3-D-Oberflächen.

5.1.1 Über Grafikhardware

Die Funktion des X Window Systems hängt unmittelbar von den Eigenschaften der verwendeten Grafikkarte und des Monitors ab. Wenn Sie den X-Server manuell konfigurieren, müssen Sie deren Werte eintragen, falls sie nicht automatisch ermittelt werden können (eigentlich ist genau dies der Fall, in dem Sie den X-Server überhaupt selbst konfigurieren müssen). Deshalb liefert dieser Unterabschnitt einige Informationen über diese Geräte, die wichtige Grundbegriffe klären.

Grafikkarten

Die *Grafikkarte* ist dafür zuständig, das Bild zu erzeugen, das auf dem Monitor ausgegeben werden soll. Sie verfügt über einen eigenen Mikroprozessor, der verschiedene Arten der Bildberechnung durchführt, sowie eigenen RAM-Speicher zur Speicherung des Monitorbildes. Alte Grafikkarten konnten lediglich 2-D-Grafik darstellen, also bestimmen, welche Farbe an welcher Pixelposition gesetzt werden soll. Seit etwa 1996 enthalten die meisten Grafikkarten 3-D-Beschleuniger, die das sogenannte Echtzeit-Rendering von 3-D-Szenen unterstützen. Sie sind also in der Lage, die Tiefeninformation der dritten Dimension in unterschiedliche Darstellungsgrößen umzurechnen und Beleuchtungseffekte, gegenseitiges Verdecken »hintereinanderliegender« Objekte oder Transparenzwirkungen zu bestimmen.

Im Laufe der PC-Geschichte wurden viele verschiedene Grafikkartenstandards mit unterschiedlicher maximaler Auflösung und Farbtiefe entwickelt. Diese Werte hängen insbesondere davon ab, wie viel eingebauten RAM-Speicher die Grafikkarte besitzt. Tabelle 5.1 zeigt einige typische Auflösungen und Farbtiefen, ihre klassischen (kaum noch verwendeten) Bezeichnungen und den erforderlichen Mindestspeicherbedarf für die 2-D-Darstellung (für 3-D-Features wird natürlich ein Vielfaches an Speicher benötigt). Der Speicherbedarf wird als technisch realisierbarer Wert angegeben – beispielsweise steht bei VGA 256 KByte statt der rechnerischen 150, die nicht als Speicherbaustein erhältlich sind.

Bezeichnung	Auflösung	Farbtiefe	Speicherbedarf
VGA	640 × 480	4 Bit (16 Farben)	256 KByte
SVGA	800 × 600	8 Bit (256 Farben)	512 KByte
	1.024 × 768	4 Bit (16 Farben)	
XGA	1.024 × 768	16 Bit (65.536 Farben)	2 MByte
SXGA	1.280 × 1.024	24 Bit (16,7 Mio. Farben)	4 MByte

Tabelle 5.1 Einige wichtige Auflösungen und Farbtiefen von Grafikkarten

Grafikkarten können auf verschiedene niedrigere Auflösungen und Farbtiefen heruntergestellt werden, um zum Beispiel ältere Monitore oder alte Software zu unterstützen, die mit den höchsten einstellbaren Werten der Karte nicht zurechtkommen. Tabelle 5.1 ist ein guter Anhaltspunkt für die Werte, die bei den meisten Karten verfügbar sind (allerdings können Sie in der Regel alle genannten Auflösungen und Farbtiefen beliebig mischen). Einige unterstützen noch einen

Zwischenwert von 1.152 x 864, außerdem werden gelegentlich andere Formate wie 16:10 oder gar 16:9 angeboten, die für manche Monitore erforderlich sind.

Die Geschwindigkeit der Grafikkarte hängt von mehreren Faktoren ab:

- ▶ Eine wichtige Rolle spielt die Leistungsfähigkeit des Grafikprozessors selbst.
- ▶ Der RAM-Speicher von Grafikkarten wird durch viele unterschiedliche Speicherbausteintypen (SD-RAM, DDR-RAM oder grafikoptimierte Varianten) realisiert. Je besser die Speichersorte, desto höher die Geschwindigkeit.
- ▶ Der verwendete Anschluss hat ebenfalls Bedeutung: Wird eine Grafikkarte am PCI-Bus betrieben, muss sie sich dessen Datenkanäle mit vielen anderen Geräten teilen. Der neuere AGP-Anschluss steht der Grafikkarte dagegen exklusiv zur Verfügung und ist obendrein schneller und leistungsfähiger als PCI.
- ▶ Zudem ist die Taktfrequenz des RAMDAC wichtig. Es handelt sich um den Chip, der den digitalen Inhalt des Videospeichers in das analoge Bildsignal umwandelt, das am Monitoranschluss ausgegeben wird.

Monitore

Es gibt zwei verschiedene Grundtypen von Computermonitoren: Röhrenmonitore und LCD-Displays (bei Fernsehern kommt als dritte Bauweise der Plasma-Bildschirm hinzu; manche von ihnen besitzen RGB-Eingänge, aber explizit als Computermonitore werden sie nicht verkauft). Hinzu kommen speziellere Anzeigeräte wie beispielsweise Beamer (LCD-Projektoren), die für Vorträge oder im Unterricht verwendet werden, um die Ausgabe eines Computers einer größeren Zuschauergruppe zu zeigen.

Der klassische *Röhrenmonitor* (CRT für »Cathode Ray Tube« oder deutsch Kathodenstrahlröhre) spielt so gut wie keine Rolle mehr; Neugeräte werden praktisch nicht mehr verkauft; weder als Computermonitore noch als Fernseher. Der Röhrenmonitor und der herkömmliche Fernseher funktionieren beide übrigens nach demselben Prinzip: Ein Elektronenstrahl in einer Vakuumröhre stimuliert eine Phosphorschicht, auf die er so das Bild zeichnet, indem er jeden einzelnen Punkt in jeder Zeile nacheinander abtastet – in Wirklichkeit sehen Sie auf einem Röhrenmonitor in einem bestimmten Moment immer nur einen einzigen Pixel! Dass das Bild dennoch kontinuierlich als Ganzes wahrgenommen wird, liegt an der Trägheit des Auges: Sobald etwas schneller als etwa 24-mal pro Sekunde neu gezeichnet wird, erscheint es als fortlaufende Bewegung beziehungsweise als Standbild.

Ein Fernseher arbeitet mit 50 Halbbildern pro Sekunde: In einem Durchgang werden die Zeilen 1, 3, 5 und so weiter gezeichnet, im nächsten Durchgang dann 2, 4, 6 und so fort. Dieses als Interlacing bezeichnete Verfahren erscheint weni-

ger flimmernd als 25 ganze Bilder pro Sekunde. Heutige CRT-Monitore arbeiten mit höheren Bildraten und ohne Interlacing – bei 75 Hz zeichnet ein Monitor beispielsweise 75-mal pro Sekunde den gesamten Bildschirminhalt neu. Ab etwa diesem Wert wird das Bild als vollkommen flimmerfrei wahrgenommen.

Um zu errechnen, welche Bildwiederholrate ein Röhrenmonitor bei einer bestimmten Auflösung erreichen kann, müssen Sie seine *Zeilenfrequenz* kennen. Dieser in kHz angegebene Wert gibt an, wie viele einzelne Zeilen der Kathodenstrahl pro Sekunde zeichnen kann – wenn Sie diesen Wert durch die Zeilenzahl der gewünschten Auflösung teilen, erhalten Sie die maximale Bildrate. Beispielsweise kann bei einer Auflösung von 1.280 × 1.024 Pixeln, also 1.024 Zeilen, und einer Zeilenfrequenz von 80 kHz eine Bildwiederholrate von gut 78 Hz erreicht werden.

Das LCD-Verfahren (Liquid Cristal Display, also Flüssigkristallanzeige) gibt es schon lange. Für Taschenrechner, Digitaluhren und Messgeräte wird es seit Jahrzehnten eingesetzt, seit einigen Jahren auch für Notebooks und Desktop-Monitore.

Das Prinzip funktioniert folgendermaßen: Im Hintergrund leuchtet eine gleichmäßig helle Fläche. Dieses Licht wird durch einen ersten Polarisationsfilter geschickt, der nur noch horizontale Lichtstrahlen durchlässt. Als Nächstes passiert das Licht eine Flüssigkristallschicht. Die Flüssigkristalle können durch Spannung gedreht werden, sodass sie die Polarität des Lichts an manchen Stellen um 90° drehen, an anderen dagegen nicht. Verschiedene Helligkeitswerte werden nun dadurch erreicht, dass das Licht daraufhin durch einen weiteren Polfilter muss – diesmal einen, der nur vertikale Lichtstrahlen passieren lässt. Es kommt also nur an den Stellen Licht an, wo der Flüssigkristall die Polarität gekippt hat. Zu guter Letzt wird diese Hell-Dunkel-Verteilung noch durch eine RGB-Folie gefiltert, die die Grundfarben Rot, Grün und Blau erzeugt, aus denen das Monitorbild zusammengesetzt wird (siehe Kapitel 8, »Desktop-Software«).

Die modernste Form des LCD-Displays, die bei fast allen Notebooks und LCD-Monitoren eingesetzt wird, ist das TFT-Verfahren (Thin Flat Transistor). Hier wird jeder Flüssigkristall durch einen eigenen Transistor angesteuert, wodurch die genauesten Ergebnisse erzielt werden.

Die größten Vorteile von LCD-Displays gegenüber Röhrenmonitoren sind folgende:

- ▶ Ein solches Display kann aufgrund seiner Bauweise nicht flimmern (jeder Pixel erstrahlt so lange gleichmäßig in seiner Farbe, bis diese geändert wird) – aus diesem Grund werden LCDs mit vergleichsweise niedriger Bildrate betrieben, oft zum Beispiel 60 Hz.

- ▶ Die LCD-Anzeige ist absolut flach und rechteckig, während bei Röhren trotz aller Bemühungen noch immer eine kleine Wölbung verbleibt, die einen Teil der sichtbaren Bildfläche abzieht. Auf diese Weise erreicht beispielsweise ein 17-Zoll-TFT-Monitor fast die Anzeigegröße eines 19-Zoll-Röhrengeräts.
- ▶ Die Strahlungsbelastung, der Sie bei der ständigen Arbeit mit einem Röhrenmonitor ausgesetzt sind, ist trotz TCO-2003-Norm für Strahlungsarmut noch immer um ein Vielfaches höher als bei einem Flatscreen. Dasselbe gilt übrigens für den Stromverbrauch, sodass Flachbildschirme auch umweltfreundlicher sind als Röhrengeräte.
- ▶ Schließlich ist auch noch ein wichtiger praktischer Grund ausschlaggebend: Ein Röhrenmonitor benötigt durch die Tiefe der Bildröhre eine ganze Menge Platz auf dem Schreibtisch, während ein LCD-Display so flach ist wie das Telefonbuch einer Kleinstadt und so auch noch auf den kleinsten Schreibtisch passt, ohne Ihnen die Freiheit für Tastatur und Maus zu nehmen.

5.1.2 Den X-Server konfigurieren

Eine der großen Stärken von openSUSE ist das seit einigen Jahren – inzwischen in Version 2 – verfügbare Tool SaX (SUSE Advanced X Configuration Utility). In einer Zeit, als die Systeminstallation selbst noch nicht grafisch durchgeführt wurde, erleichterte es die anschließende Konfiguration des X-Servers entscheidend; in anderen Distributionen musste man die X-Konfigurationsdateien seinerzeit manuell editieren; dort wurden so exotische Werte wie die oben beschriebene Zeilenfrequenz des Monitors abgefragt (über die sich sogar manches Monitor-Handbuch ausschweigt).

Eines vorweg: Wenn die grafische Oberfläche bereits während der in Kapitel 2, »openSUSE installieren«, beschriebenen Installation gestartet und eingerichtet wurde, brauchen Sie die hier gezeigten Arbeitsschritte nicht durchzuführen. Dies ist nur nötig, wenn gar keine grafische Oberfläche zur Verfügung steht (sehr selten) oder wenn Sie wissen, dass Ihre Grafikkarte eine höhere Auflösung oder Farbtiefe erreichen kann, als der Konfigurationsdialog Ihres Desktops anzeigt.

Geben Sie als root Folgendes in ein virtuelles Terminal oder ein Terminalfenster ein, um SaX2 zu starten:

```
# sax2
```

Daraufhin wird SaX2 gestartet; der Startbildschirm sieht so aus wie in Abbildung 5.1. Eine alternative Möglichkeit, SaX2 aufzurufen, ist das Objekt GRAFIKKARTE UND MONITOR in der YaST-Kategorie HARDWARE.

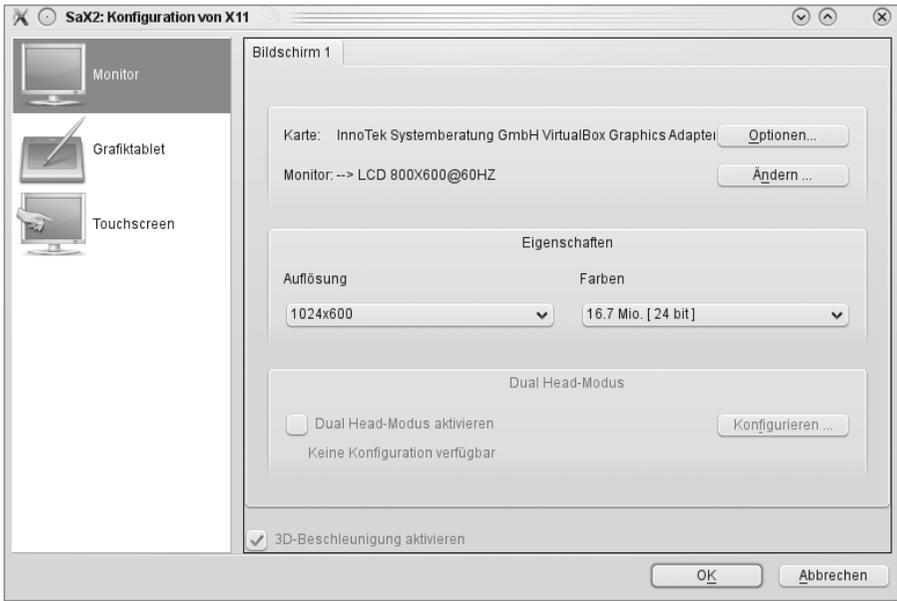


Abbildung 5.1 Der Startbildschirm von SaX2 mit Grafikkarten- und Monitoreinstellungen

Im Einzelnen besitzt der Dialog drei Kategorien, zwischen denen Sie im rechten Feld wechseln können:

- ▶ MONITOR
- ▶ GRAFIKTABLETT
- ▶ TOUCHSCREEN

In den bisherigen openSUSE-Versionen konnten auch Maus und Tastatur über SaX konfiguriert werden. Dies ist jedoch unnötig, da die mit KDE und GNOME gelieferten, weiter unten beschriebenen Konfigurationsprogramme diese Funktionalität enthalten.

Grafikkarte und Monitor konfigurieren

In der Kategorie MONITOR werden die wichtigsten Einstellungen vorgenommen: die Konfiguration von Grafikkarte und Monitor. Im oberen Feld werden KARTE und MONITOR aufgelistet, die der X-Server erkannt hat. Falls als Grafikkarte lediglich einfaches VGA (640 × 480 Pixel, 16 Farben) ermittelt wurde, ist dies ein Zeichen dafür, dass der passende Treiber nicht vorhanden ist. In diesem Fall müssen Sie das betreffende Kernel-Modul laden und zuvor gegebenenfalls separat beschaffen (siehe Kapitel 6, »Wichtige Hardware konfigurieren«).

Wählen Sie **OPTIONEN**, um besondere Einstellungen für Ihre konkrete Grafikkarte vorzunehmen. Hier geht es insbesondere um die Hardwarebeschleunigung und ihre diversen Einzelfeatures. Die Schaltfläche **ÄNDERN** ermöglicht Ihnen dagegen die Auswahl eines anderen Monitors, falls auch dieser nicht korrekt erkannt wurde. Wählen Sie auf der Registerkarte **MONITOR FÜR KARTE ...** links den Hersteller und rechts das Modell. Wenn Ihr Hersteller nicht dabei ist, helfen mitunter die Standardeinstellungen → **VESA** (für Röhrenmonitore) oder → **LCD** (für Flachbildschirme).

Auf der Registerkarte **ANZEIGEGRÖSSE** können Sie Größe, Diagonale und Seitenverhältnis Ihres Monitors einstellen, falls diese nicht automatisch durch den korrekten Treiber bereitgestellt werden konnten.

Mit den **SYNCHRONISATIONSFREQUENZEN** sollten Sie sich nur auseinandersetzen, wenn alle anderen Einstellungsmöglichkeiten keinen Erfolg gebracht haben. **HORIZONTAL** beschreibt die bereits erwähnte Zeilenfrequenz, **VERTIKAL** die Bildwiederholrate des Monitors. Beachten Sie, dass fehlerhafte Einstellungen Monitore (besonders ältere, die keinen diesbezüglichen Schutz besitzen) *irreparabel beschädigen* können! Tragen Sie hier *niemals* auf Verdacht irgendwelche Werte ein, sondern konsultieren Sie das Handbuch des Monitors, den technischen Support des Herstellers oder notfalls andere möglichst seriöse Quellen im Web.

Nachdem Sie die grundlegenden Einstellungen für Monitor und Grafikkarte vorgenommen haben, können Sie im unteren Bereich des Fensters die **AUFLÖSUNG** und die Farbtiefe einstellen, mit denen Sie arbeiten möchten.

Weitere Optionen

Die beiden restlichen Kategorien ermöglichen etwas seltener genutzte Einstellungen:

▶ **GRAFIKTABLETT**

Wenn Sie ein Grafiktablett besitzen, was besonders für Bildbearbeitung und Grafik nützlich ist, können Sie es in dieser Kategorie aktivieren, sofern es zu den unterstützten Modellen kompatibel ist. Auf den beiden zusätzlichen Registerkarten sind weitere Optionen des Tablettts beziehungsweise des drucksensitiven Stifts einstellbar.

▶ **TOUCHSCREEN**

openSUSE unterstützt einige Touchscreen-Systeme, sodass Sie beispielsweise auch Info-Terminals unter Linux betreiben können. In dieser Kategorie lässt sich das entsprechende Modell wählen und dann unter **ANSCHLUSSPORT** festlegen, an welchem Anschluss Ihres Rechners die Eingabeeinheit des Touchscreens angeschlossen ist (heutzutage meist USB).

Abschluss der X-Konfiguration

Klicken Sie auf OK, wenn Sie alle infrage kommenden Kategorien bearbeitet haben. Der Dialog ABSCHLIESSENDE SCHRITTE wird angezeigt. Klicken Sie hier zunächst auf TEST, um den X-Server mit den gewählten Optionen zu testen. Beachten Sie, dass die erste Inbetriebnahme etwas dauern kann; falls der Bildschirm jedoch permanent schwarz bleibt, seltsam flackert oder ähnlichen Ärger bereitet, können Sie den X-Server mit **Strg** + **Alt** + **Backspace** abschießen und neu starten.

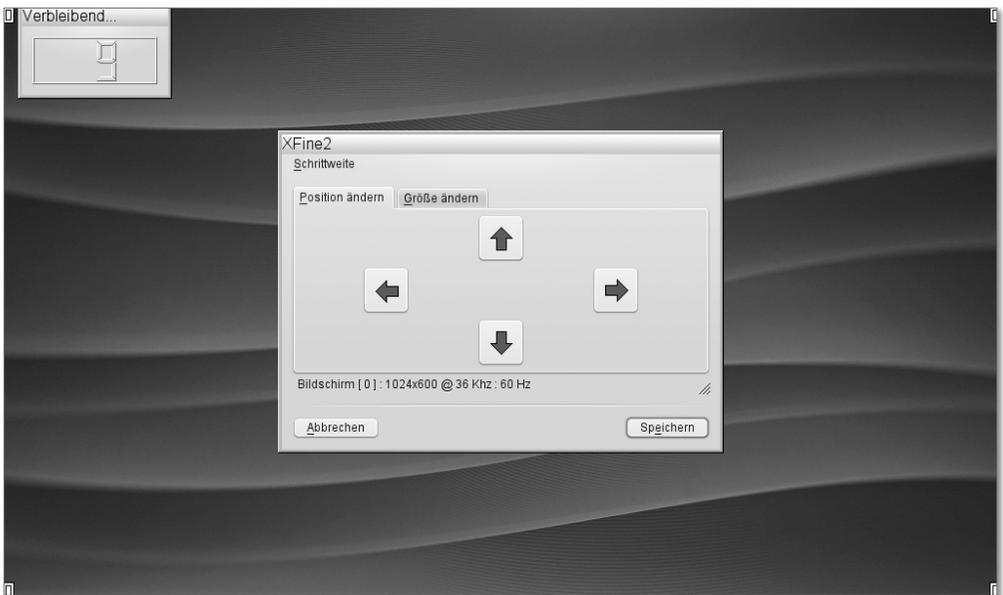


Abbildung 5.2 Der SaX2-Testbildschirm

Wenn alles so weit in Ordnung ist, wird ein Testbildschirm wie in Abbildung 5.2 angezeigt. Falls das Bild noch nicht perfekt auf den Monitor passt (erkennbar am weißen Rand und den kleinen Rechtecken in den vier Ecken), können Sie im Fenster XFINE2 die Position und die Größe anpassen. Klicken Sie auf SPEICHERN, sobald Sie zufrieden sind, oder auf ABBRECHEN, um erneut grundlegende Änderungen der X-Konfiguration vorzunehmen.

Wenn Sie Ihre Konfiguration erfolgreich gespeichert haben, können Sie die anschließende Frage nach dem Beenden von SaX2 mit JA beantworten. Wenn Sie Ihre neuen Einstellungen sofort in Betrieb nehmen möchten, müssen Sie die grafische Oberfläche neu starten. Wählen Sie dazu unter KDE K-MENÜ • ABMELDEN, und klicken Sie auf die Schaltfläche AKTUELLE SITZUNG BEENDEN. Bei GNOME finden Sie den entsprechenden Befehl im SYSTEM-Menü.

Falls Sie Ihr System nicht für den automatischen Start der grafischen Oberfläche konfiguriert haben, können Sie diese aus einer Textkonsole folgendermaßen mit den neuen Einstellungen starten:

```
# startx
```

Falls Sie den automatischen Start eines grafischen Login-Managers nachträglich vornehmen möchten, müssen Sie die Systemkonfigurationsdatei `/etc/sysconfig/displaymanager` konfigurieren. Dies geht ohne Weiteres mit einem beliebigen Texteditor, etwa mit dem grafischen Editor Kate oder mit den in Kapitel 10, »Klassische Texteditoren«, beschriebenen Konsolen-Editoren. Beachten Sie, dass Sie dafür `root` sein müssen, weil Sie die betreffende Datei ansonsten nach der Änderung nicht speichern dürfen. Geben Sie dazu einfach `su` und anschließend das `root`-Passwort ein. Auch Kate können Sie als `root` starten, indem Sie nach dem `su` den folgenden Konsolen-Befehl eingeben (das `&` wie immer, um das Terminal für weitere Anweisungen freizuhalten):

```
# kate &
```

Um Kate als `root` auszuführen, können Sie in KDE alternativ auch **K-MENÜ • RECHNER • BEFEHL AUSFÜHREN** wählen. Geben Sie in das Suchfeld einfach `kate` ein. Klicken Sie in den Suchergebnissen auf den Schraubenschlüssel, der neben dem Eintrag **KATE AUSFÜHREN** angezeigt wird, um die erweiterten Einstellungen zu öffnen. Wählen Sie **ALS ANDERER BENUTZER AUSFÜHREN**, und tippen Sie im Feld **BENUTZERNAME** `root` sowie darunter dessen **PASSWORT** ein. Der entsprechende Dialog wird in Abbildung 5.3 gezeigt. In GNOME müssen Sie dagegen auf die oben beschriebene Konsolen-Methode zurückgreifen, weil der Dialog **ANWENDUNGEN • ANWENDUNG AUSFÜHREN** keine Möglichkeit zum Benutzerwechsel enthält.

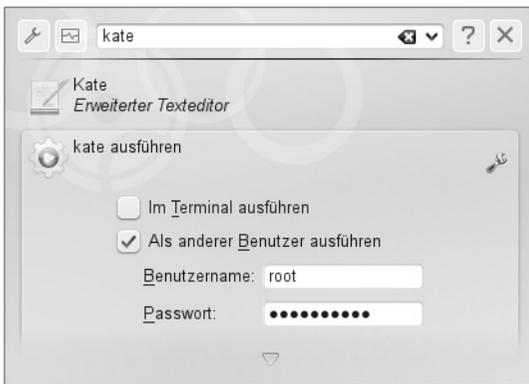


Abbildung 5.3 Den Editor Kate in KDE als `root` ausführen

In Abbildung 5.4 wird gezeigt, wie die Konfigurationsdatei `/etc/sysconfig/displaymanager` im Kate-Dokumentfenster aussieht.

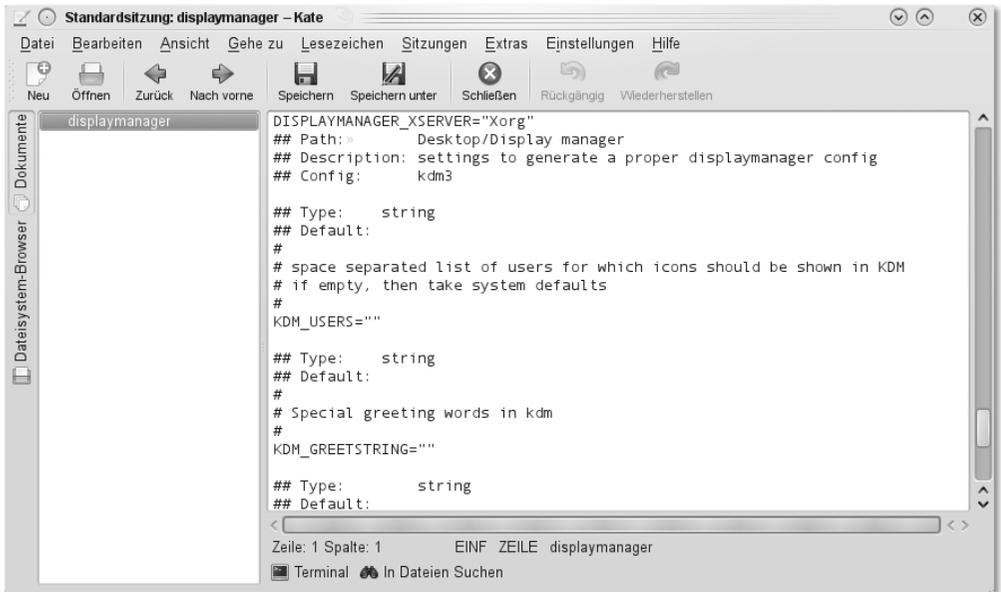


Abbildung 5.4 Die Datei `/etc/sysconfig/displaymanager` im Editor Kate

Noch einfacher ist es, den EDITOR FÜR `/ETC/SYSCONFIG-DATEIEN` in YaST zu verwenden. Er befindet sich im Abschnitt `SYSTEM`. Die meisten Einstellungen, die in `/etc/sysconfig`-Dateien gespeichert werden, können Sie bequem mithilfe benutzerfreundlicherer YaST-Dialoge einstellen. Für Administratoren, die eine unmittelbarere Kontrolle bevorzugen, ist dieser Dialog dagegen sehr praktisch. In der linken Spalte finden Sie eine Baumhierarchie mit den verschiedenen Konfigurationskategorien. Für die Wahl des Login-Verfahrens müssen Sie die Kategorie `DESKTOP` und deren Unterkategorie `DISPLAY MANAGER` aufklappen (siehe Abbildung 5.5). Der erste Eintrag – `DISPLAYMANAGER` – kann einen der folgenden Werte annehmen, die Sie aus dem Pulldown-Menü im rechten Bereich wählen können:

- ▶ `KDM3` – der Display Manager von KDE 3
- ▶ `KDM4` – der neue Display Manager von KDE 4
- ▶ `GDM` – der GNOME Display Manager
- ▶ `XDM` – der klassische X Window Display Manager

- ▶ wdm – der WindowMaker Display Manager (WindowMaker ist ein alternativer Desktop beziehungsweise Window-Manager, der weiter unten kurz angesprochen wird)
- ▶ CONSOLE – es wird gar keine grafische Benutzeroberfläche gestartet.

Beachten Sie, dass die Wahl eines der fünf Display-Managers keineswegs bedeutet, dass Sie nur noch den zugehörigen Desktop starten können. Sie unterscheiden sich vielmehr in einigen Bediendetails bei der Anmeldung, besitzen aber jeweils eine Möglichkeit, den zu startenden Desktop zu wählen. Dies wurde in Kapitel 3, »Systemrundgang und Grundkonfiguration«, am Beispiel von KDM gezeigt.

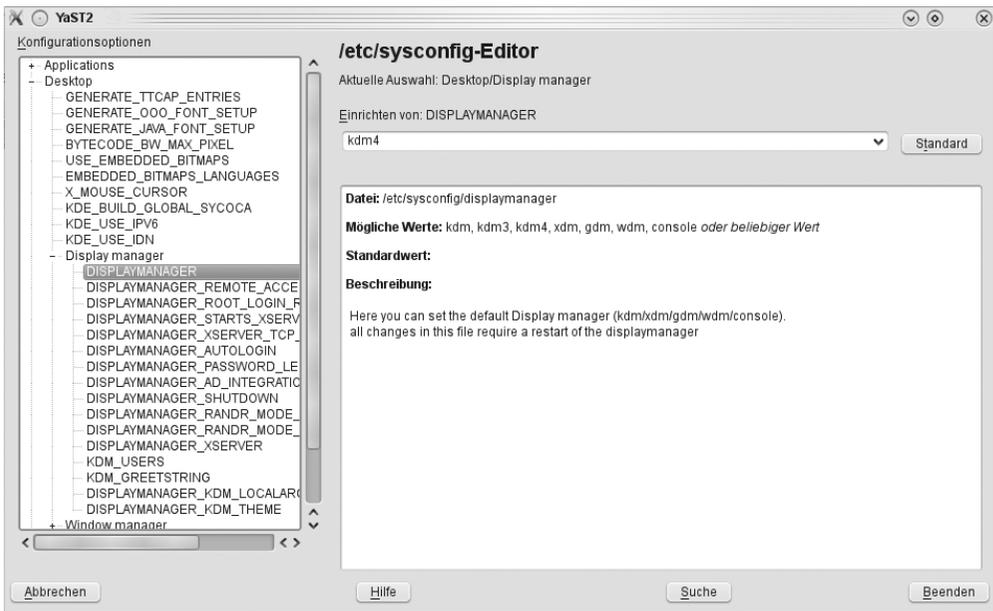


Abbildung 5.5 Auswahl des Display-Managers im /etc/sysconfig-Editor

Wichtig ist zusätzlich, dass der weiter unten zu findende Eintrag `DISPLAYMANAGER_STARTS_XSERVER` den Wert `YES` besitzt – andernfalls können grafische Anmeldungen auf dem Rechner zwar per Fernzugriff (`DISPLAYMANAGER_REMOTE_ACCESS`), aber nicht lokal erfolgen. Die weiteren Optionen in dieser Kategorie wie Autologin und Anmeldung ohne Passwort haben Sie bereits an anderer Stelle komfortabler eingestellt (siehe Kapitel 2 und 3).

Wenn Sie einen Texteditor zum Modifizieren der Konfigurationsdateien verwendet haben, speichern Sie Ihre Änderungen (in Kate etwa `DATEI • SPEICHERN` oder

(Strg + S). Im /ETC/SYSCONFIG-EDITOR von YaST können Sie dagegen einfach auf **BEENDEN** klicken. Beim nächsten Booten müsste die grafische Oberfläche nun automatisch gestartet werden.

5.2 KDE und GNOME – die Desktops

Die grundlegende Funktionsweise der beiden Desktops KDE und GNOME wurde bereits in Kapitel 3 vorgestellt. In diesem Abschnitt lernen Sie einige weitere Aspekte dieser modernen Benutzeroberflächen kennen.

In Tabelle 5.2 sehen Sie zunächst eine Übersicht über einige wichtige Eigenschaften der beiden Desktop-Umgebungen im direkten Vergleich.

Die ursprüngliche Motivation zur Entwicklung von GNOME war übrigens, dass die für KDE eingesetzte Grafikbibliothek Qt von der norwegischen Firma Trolltech damals zwar kostenlos benutzt werden durfte, aber keine Open-Source-Software war. Inzwischen hat sich dies geändert, aber da beide Desktops zwischenzeitlich eine recht große Verbreitung erlangt und auch jeweils eigene Lösungen für die einzelnen Aspekte einer modernen grafischen Benutzeroberfläche gefunden hatten, werden beide weiterentwickelt.

Eigenschaft	KDE	GNOME
vollständiger Name	K Desktop Environment	GNU Network Object Model Environment
aktuelle Version	4.3.4	2.28
Entwicklungsteam	The KDE Project	The GNOME Foundation
Gründungsjahr	1996	1997
Projektgründer	Matthias Ettrich	Miguel de Icaza, Federico Mena
grundlegende Programmiersprache	C++	C
Grafikbibliothek	Qt	GtK+
Lizenz	GPL/LGPL	GPL/LGPL
Dateimanager	Dolphin	Nautilus

Tabelle 5.2 Die wichtigsten Eigenschaften von KDE und GNOME im Vergleich

Eigenschaft	KDE	GNOME
Terminalemulation	Konsole	GNOME Terminal
weitere zugehörige Anwendungen (Beispiele)	Konqueror (Browser) KOffice (Office-Suite) KMail (E-Mail) Krita (Bildbearbeitung) Kate (leistungsstarker Editor) amaroK (Audio-Player)	Epiphany (Browser) Gnumeric (Tabellenkalkulation) und andere Einzel-Office-Komponenten Evolution (E-Mail, Groupware) GIMP (Bildbearbeitung, siehe Kapitel 8, »Desktop-Software«)

Tabelle 5.2 Die wichtigsten Eigenschaften von KDE und GNOME im Vergleich (Forts.)

Verstehen Sie den Punkt »Zugehörige Anwendungen« nicht falsch: Die betreffenden Programme stammen aus dem Umfeld des genannten Projekts, laufen aber prinzipiell auch auf dem jeweils anderen. Das K-MENÜ unter KDE und das GNOME-Menü ANWENDUNGEN (beide ganz unten links im Panel) enthalten bei einer openSUSE-Standardinstallation auch jeweils alle genannten Anwendungen. Vereinzelt kann es allerdings vorkommen, dass einige dieser Programme Gebrauch von speziellen Vorteilen des Desktop-Projekts machen, zu dem sie gehören. Diese stehen dann auf dem anderen Desktop oder in den weiter unten angesprochenen alternativen Window-Managern nicht zur Verfügung.

Ein weiterer wichtiger Unterschied zwischen KDE und GNOME besteht darin, dass der GNOME-Desktop auf einen externen Window-Manager aufsetzt, der die Bedienelemente der Fenster darstellt. Das bemerken Sie erst, wenn die Auswahl des Window-Managers fehlschlägt und die Fenster ohne Dekoration erscheinen, sodass Sie sie nicht mehr verschieben, vergrößern, verkleinern oder schließen können – es geschieht beispielsweise, wenn Sie ohne weitere Konfiguration auf den oben beschriebenen X-Server `xgl` wechseln. Im Normalfall ist der korrekte Window-Manager dagegen voreingestellt.

5.3 KDE

Der grundlegende Aufbau des KDE-4-Desktops wurde bereits in Kapitel 3 beschrieben. Darüber hinaus geht dieses Buch davon aus, dass Sie den grundlegenden Umgang mit irgendeiner grafischen Benutzeroberfläche beherrschen (beispielsweise Windows oder Mac OS X) und daher wissen, was Fenster, Menüs, Icons und so weiter sind.

In diesem Abschnitt werden daher nur einige zusätzliche Aspekte von KDE behandelt:

- ▶ die KDE-Grundkonfiguration
- ▶ der Dateimanager Dolphin
- ▶ die Besonderheiten der Terminalemulation KDE-Konsole
- ▶ ausgesuchte KDE-Anwendungen

5.3.1 KDE 4 konfigurieren

Wenn Sie KDE Ihren Bedürfnissen anpassen möchten, können Sie dies mithilfe der KDE-Systemeinstellungen erledigen. Wählen Sie **K-MENÜ • FAVORITEN • SYSTEMEINSTELLUNGEN**, um sie zu starten. Alternativ können Sie auch **[Alt] + [F2]** drücken und `systemsettings` eingeben (womöglich erscheint »Systemeinstellungen« bereits während der Eingabe in der Vorschlagsliste).

Der Dialog besitzt zwei verschiedene Ansichten, zwischen denen Sie über das Schraubenschlüssel-Icon mit der Beschriftung **EINRICHTEN** wechseln können. Die **SYMBOLANSICHT** unterteilt die Einstellungen in die beiden Registerkarten **ALLGEMEIN** und **ERWEITERT**, während die **KLASSISCHE BAUMANSICHT** alle Optionen in eine Baumstruktur integriert.

Abbildung 5.6 zeigt die Startseite des Dialogs in der Baumansicht. In der Spalte auf der linken Seite können Sie sich zwischen sechs verschiedenen Hauptkategorien entscheiden; jede von ihnen enthält eine verschachtelte Liste von Optionen. Wenn Sie einen Menüpunkt mit Untereinträgen anklicken, erscheinen diese auch im Hauptfenster; sobald Sie einen nicht weiter verschachtelten Menüpunkt auswählen, wird der entsprechende Konfigurationsdialog angezeigt. Die folgenden Unterabschnitte liefern eine Übersicht über die Hauptkategorien, ihre Unterkategorien und deren Bedeutung.

Kategorie ERSCHEINUNGSBILD & VERHALTEN

In diesem Dialog werden alle Einstellungen rund um die Gestaltung und die grundlegende Arbeitsweise des KDE-Desktops vorgenommen.

- ▶ Unterkategorie **ARBEITSFLÄCHE**

In diesem Bereich werden Aussehen und Verhalten des Desktops eingestellt.

- ▶ **ARBEITSFLÄCHEN-EFFEKTE**: Wenn Sie eine hinreichend neue und durch passende Treiber unterstützte Grafikkarte verwenden, erlaubt der X-Server in Zusammenarbeit mit einem sogenannten *Composite-Window-Manager* (zum Beispiel *Compiz*) 3-D-Desktop-Effekte. Diese können Sie mithilfe des

vorliegenden Dialogs einstellen. Auf der Registerkarte ALLGEMEIN geht es vor allem um Animationseffekte, die beim Fenster- und Desktopwechsel angezeigt werden. ALLE EFFEKTE ermöglicht eine genaue Auswahl zahlreicher Einzeleffekte. In mehreren Rubriken können Sie spezielle Animationen und Effekte auswählen, die beispielsweise beim Öffnen und Schließen von Fenstern angezeigt werden sollen. In der letzten Rubrik, ZUGANGSHILFEN, gibt es zudem Hilfsmittel für beeinträchtigte User wie eine Bildschirmleuchte oder einen Scharfzeichnungseffekt. Die dritte und letzte Registerkarte heißt ERWEITERT. Hier wird vor allem der gewünschte COMPOSIT-TYP ausgewählt: OPENGL ist nur für Grafikkarten mit Hardwarebeschleunigung geeignet und sollte, falls verfügbar, gewählt werden. Bei der rein softwarebasierten Lösung XRENDER ist dagegen die Performance schlechter, und es werden nicht alle Effekte unterstützt.

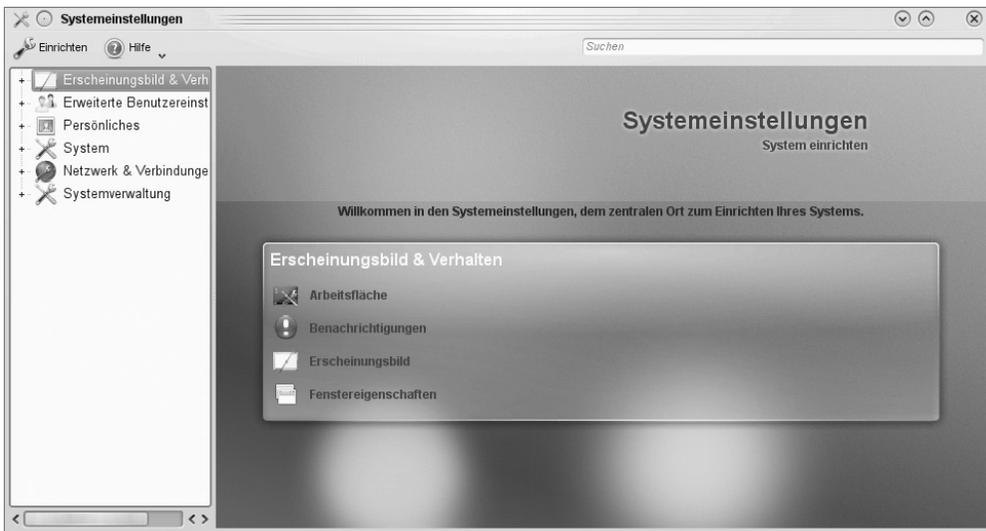


Abbildung 5.6 Startseite der KDE-Systemeinstellungen in der klassischen Baumansicht

- ▶ **VIRTUELLE ARBEITSFLÄCHEN:** Bereits in Kapitel 3 wurde erläutert, dass Sie Ihre Fenster in KDE auf mehreren – standardmäßig vier – virtuellen Desktops verteilen können. In diesem Dialog kann diese Anzahl erhöht oder vermindert werden, und Sie können jeder Arbeitsfläche einen beliebigen Namen zuweisen.
- ▶ **BILDSCHIRMECKEN:** Hier können Sie den vier Ecken und den Mitten der Bildschirmränder Aktionen zuweisen, die nach einer einstellbaren Verzögerung ausgelöst werden, wenn Sie diese Bereiche mit dem Mauszeiger

anfahen. Bei den Aktionen geht es um verschiedene Möglichkeiten des Task- und Fensterwechsels oder darum, das Dashboard anzuzeigen.

- ▶ **Bildschirmschoner:** KDE wird mit zahlreichen Bildschirmschonern geliefert. In diesem Dialog können Sie einen von ihnen auswählen; für manche sind unter **EINRICHTUNG** spezifische Einstellungen verfügbar. Auf einem kleinen Vorschaubildschirm erscheint der jeweils gewählte Bildschirmschoner; mit **TESTEN** können Sie ihn auch im Vollbild ausprobieren. Wenn Sie **RANDOM** auswählen, wird jedes Mal ein anderer Bildschirmschoner gestartet. Unter dem Auswahlmenü finden Sie allgemeine Einstellungen: Nach wie vielen Minuten der Bildschirmschoner aktiviert werden soll, ob zum Beenden eine Passworтеingabe erforderlich ist und ob die in Kapitel 3 beschriebenen Dashboard-Miniprogramme auch im Bildschirmschoner ausgeführt werden dürfen.
- ▶ **PROGRAMMSTARTANZEIGE:** Wenn Sie statt des hüpfenden Cursors anders über einen laufenden Programmstart informiert werden möchten, können Sie dies hier einstellen.
- ▶ **Unterkategorie BENACHRICHTIGUNGEN**
In diesem Bereich werden die Meldungen und Signaltöne von KDE selbst und vielen Anwendungsprogrammen eingestellt.
 - ▶ **SYSTEMNACHRICHTEN:** Hier werden die verschiedenen Meldungen der Anwendungsprogramme konfiguriert. Auf der Registerkarte **ANWENDUNGEN** können Sie in **QUELLE FÜR DAS EREIGNIS** zunächst ein Programm auswählen. In der Box darunter erscheint dann eine Liste von Ereignissen der Anwendung, für die Sie jeweils Meldung und/oder Sound konfigurieren können. Die Registerkarte **WIEDERGABE-EINSTELLUNGEN** dient dazu, das einzusetzende Soundsystem zu wählen. **KDE-SOUNDSYSTEM** ist Standard; falls es Probleme bereitet, können Sie versuchen, unter **EXTERNEN ABSPIELER VERWENDEN** ein anderes Programm zu wählen.
 - ▶ **SIGNALTON:** Wählen Sie hier **SIGNALTON STATT SYSTEMNACHRICHTEN VERWENDEN**, um statt Meldungen einen Ton abzuspielen. Dies ist besonders für sehbehinderte Benutzer geeignet. Sie können Lautstärke, Tonhöhe und Dauer des Tonsignals wählen.
- ▶ **Unterkategorie ERSCHEINUNGSBILD**
Hier können Sie verschiedene visuelle Aspekte der grafischen Oberfläche einstellen, etwa Fensterstil, Farben und Schriftarten.
 - ▶ **STIL:** Ihre Auswahl auf dem gleichnamigen ersten Tab bestimmt das grundlegende Aussehen von Fenstern, Menüs und anderen Bedienelementen. Die Vorauswahl **OXYGEN** ist der aktuelle KDE-Stil; daneben stehen unter

anderem Nachbildungen von CDE und MOTIF (klassische UNIX-GUIs) sowie WINDOWS zur Verfügung. Bei manchen Stilen können Sie mit EINRICHTEN weitere Einstellungen vornehmen. Auf der zweiten Registerkarte, DETAILS, geht es um Auswahl und Positionierung von Text beziehungsweise Symbolen auf Schaltflächen.

- ▶ **FARBEN:** Legen Sie hier das Farbschema der GUI fest. Es gibt insgesamt fünf Registerkarten: SCHEMA für den grundlegenden Farbensatz, OPTIONEN für erweiterte Einstellungen (zum Beispiel das standardmäßig ausgewählte FARBEN AUF NICHT-KDE-4-PROGRAMME ANWENDEN), FARBAUSWAHL für die vom Schema abweichende manuelle Auswahl der einzelnen Farben, INAKTIV für die Darstellung von Fenstern ohne Eingabefokus und DEAKTIVIERT für Objekte, die zurzeit nicht zur Auswahl stehen.
- ▶ **SYMBOLE:** Für allgemeine Icons wie Ordner, Dateien, Papierkorb und so weiter wird ein bestimmter Symbolsatz verwendet; diesen können Sie auf dem Karteireiter DESIGN auswählen. Neben der Vorauswahl OXYGEN können Sie beispielsweise TANGO, ein weit verbreitetes, freies Icon-Set, oder HOHER KONTRAST (gut geeignet für Laptop-Displays bei schlechten Lichtverhältnissen in der Umgebung) verwenden. Auf dem Tab ERWEITERT lassen sich zudem Größen und Effekte für die Icons festlegen.
- ▶ **SCHRIFTARTEN:** Stellen Sie hier ein, welche Schriftarten und -größen für die verschiedenen Beschriftungen in KDE verwendet werden sollen.
- ▶ **FENSTER:** Unter FENSTERDEKORATION wird die Gestaltung der Fenstertitelleisten und -bedienelemente bestimmt; auf der Registerkarte KNÖPFE können Sie die Schaltflächen zum Schließen, Verkleinern und so weiter positionieren.
- ▶ **GTK-STILE UND SCHRIFTARTEN:** In diesem Dialog bestimmen Sie, wie GNOME-Anwendungen (die die Grafikbibliothek GTK verwenden) unter KDE dargestellt werden sollen.
- ▶ **STARTBILDSCHIRM:** Hier können Sie sich ein Design für den Bootscreen von KDE aussuchen.
- ▶ **EMOTICONS:** In diesem Bereich legen Sie fest, durch welche Grafiken die mit Hilfe diverser Tastenkombinationen erzeugten Smileys und anderen Symbole dargestellt werden sollen.
- ▶ **Unterkategorie FENSTEREIGENSCHAFTEN**
Während es bisher eher um das Aussehen der Fenster ging, dienen die beiden Dialoge in dieser Kategorie der Einstellung des Fensterverhaltens.
 - ▶ **FENSTEREIGENSCHAFTEN:** Auf der Registerkarte AKTIVIERUNG kann eingestellt werden, ob die Fenster durch Klick oder – wie in klassischen X-Ober-

flächen – durch einfache Mausberührung den Fokus erhalten sollen. TITELLEISTE bestimmt, was bei Klick auf dieselbe und ähnliche Aktionen geschehen soll. Unter FENSTER können Sie festlegen, welche Auswirkung die diversen Maustasten auf die Fenster haben sollen. Der Tab VERSCHIEBEN enthält Optionen für das Verschieben von Fenstern – unter anderem können Sie hier bestimmen, ob deren Inhalt auch während des Verschiebens angezeigt wird (was bei schwächeren Rechnern möglicherweise deaktiviert werden sollte). ERWEITERT schließlich enthält einen AUTOMATISCHEN FENSTERHEBER, also eine Option, um Fenster nach einer wählbaren Wartezeit in den Vordergrund zu stellen.

- ▶ FENSTERSPEZIFISCHE EINSTELLUNGEN: Hier können Sie bestimmten Fenstern eigene Verhaltensweisen zuordnen, die die im vorigen Dialog eingestellten Optionen für diese überschreiben.

Kategorie ERWEITERTE BENUTZEREINSTELLUNGEN

Hier werden viele Optionen zur KDE-Oberfläche und den mitgelieferten Anwendungsprogrammen zusammengefasst. Anders als bei den weiter oben behandelten Kategorien gibt es hier keine Unterkategorien; die Elemente dieses Menüs sind Einstellungsdialoge.

▶ ARBEITSFLÄCHEN-DESIGN-DETAILS

Während Sie in der soeben behandelten Unterkategorie ERSCHEINUNGSBILD allgemeine Voreinstellungen für das Aussehen von Desktop, Fenstern und anderen Bedienelementen vornehmen können, besteht hier die Möglichkeit, Einfluss auf das Design einzelner Elemente zu nehmen. Für jeden Aspekt wie FARBSHEMA, KICKOFF (K-Menü) oder DIALOG-HINTERGRUND können Sie dazu eine QUELLE auswählen, die deren Aussehen bestimmt. Wenn Ihnen die vorhandenen Quellen nicht reichen, klicken Sie auf NEUES DESIGN HERUNTERLADEN, um im Netz nach weiteren Designs zu suchen.

▶ AUDIO-CDs

In diesem Dialog wird festgelegt, wie Audio-CDs abgespielt und gegebenenfalls digitalisiert werden sollen. Auf der Registerkarte ALLGEMEIN können Sie zunächst das CD-ROM-Laufwerk auswählen und bestimmen, ob eine Fehlerkorrektur für defekte Tracks durchgeführt werden soll. Unter NAMEN stellen Sie ein, wie die resultierenden Audiodateien bei einer Digitalisierung heißen sollen. Auf dem Tab OGG-VORBIS-CODIERER können Sie schließlich den Qualitätsfaktor der Digitalisierung einstellen; standardmäßig wird also das freie Format Ogg Vorbis und nicht das proprietäre MP3 verwendet.

► AUTOSTART

In diesem Bereich können Sie Programme und Skripte wählen, die beim Start von KDE automatisch gestartet werden sollen.

► CDDB-Abfrage

Konfigurieren Sie hier die Abfrage von Audio-CD-Titeln im Internet. Statt des kommerziellen Gracenote-CDDB wird der freie Dienst FreeDB verwendet. Auf der Registerkarte Abfrage wird zunächst der zu verwendende (Spiegel-)Server ausgewählt; die Voreinstellung lautet *freedb.freedb.org*. Zusätzlich können Sie MUSICBRAINZ-ABFRAGE AKTIVIEREN ankreuzen; dabei handelt es sich um einen Community-basierten Dienst zur Erfassung von Titeln, die in FreeDB nicht vorhanden sind. Unter LOKALER SPEICHER wird das Verzeichnis eingetragen, in dem die Titelinformationen zwischengespeichert werden sollen.

Die Registerkarte ÜBERTRAGUNG dient der Auswahl, wie die Titeldaten übertragen werden sollen. Standardmäßig ist HTTP vorausgewählt; optional ist auch SMTP (E-Mail) möglich, was jedoch in der Regel nicht nötig sein dürfte.

► DATEIZUORDNUNGEN

In diesem Bereich wird eingestellt, welche Dateitypen mit welchen Programmen geöffnet werden sollen, beispielsweise beim Anklicken im Dateimanager.

Das Auswahlfeld BEKANNTE TYPEN sortiert sämtliche dem System bekannten Dateitypen nach dem MIME-Schema, das auch für E-Mail und Web verwendet wird: In einem Baum-Menü können Sie die Haupttypen wie *application* (anwendungsspezifische Typen), *audio* (Sounddateien) oder *image* (Bilddateien) aufklappen, um die Untertypen wie *audio/mp3* oder *image/jpeg* zu finden. Wenn Sie einen dieser konkreten Dateitypen anklicken, erscheinen rechts die Einstellungen dafür.

Die Registerkarte ALLGEMEIN enthält die DATEIMUSTER (*.Dateiendung) für den jeweiligen Typ, außerdem die BESCHREIBUNG, die im Dateieigenschaften-Dialog des Dateisystems angezeigt wird, sowie die Liste RANGFOLGE DER ZUGEORDNETEN ANWENDUNGSPROGRAMME. Das oberste Programm in der Liste wird bei einem Linksklick beziehungsweise Doppelklick auf eine Datei des entsprechenden Typs automatisch zum Öffnen verwendet; alle anderen erscheinen bei einem Rechtsklick im Kontextmenü.

Auf dem Tab EINBETTEN können Sie genauer festlegen, was beim Linksklick oder beim Download einer Datei des jeweiligen Typs geschehen soll.

► DESKTOPSUCHE

Dieser Dialog bietet die Möglichkeit, für die – weiter unten im Abschnitt zu GNOME näher beschriebene – Beagle-Desktop-Suche zusätzliche Indexdienste einzuschalten: Auf dem Tab GRUNDEINSTELLUNGEN können Sie NEPOMUK-

SEMANTIK-DIENSTE aktivieren (eine erweiterte Schlagwortsuche) und STRIGI-DATEI-INDEXER aktivieren (schnelle Suche in Dateiinhalten) ankreuzen. Unter ERWEITERTE EINSTELLUNGEN werden die Verzeichnisse für die Indizierung ausgewählt; außerdem können Sie Dateiendungen wählen, für die kein Index erstellt werden soll (typischerweise temporäre Dateien und Backups).

▶ DIENSTEVERWALTUNG

KDE verfügt über eine Reihe von Servern und Diensten, die im Hintergrund ausgeführt werden. Hier können Sie diese Dienste konfigurieren, beispielsweise PLATTENPLATZ-ÜBERWACHUNG, NETZWERK-ÜBERWACHUNG und ähnliches.

▶ DIGITALE BRIEFTASCHE

Die digitale Brieftasche *KWallet* verwaltet Passwörter, Schlüssel und Zertifikate in verschlüsselter Form. Auf diese Weise können Sie all diese Sicherheitsobjekte durch ein einziges Passwort schützen und bei Bedarf freischalten. Wählen Sie auf der Registerkarte EINSTELLUNGEN FÜR DIE DIGITALE BRIEFTASCHE zunächst KDE-BRIEFTASCHENSYSTEM AKTIVIEREN, um *KWallet* zu verwenden. Darunter können Sie unter anderem einstellen, wann die digitale Brieftasche geschlossen werden soll (was bei einer erneuten Aktivierung eine weitere Eingabe des zentralen Passworts erfordert) und wo die Brieftasche gespeichert wird. Auf dem Tab ZUGRIFFSÜBERWACHUNG wird das Protokoll der Brieftaschenzugriffe angezeigt.

▶ DIGITALKAMERA

In diesem Dialog können Sie die Zusammenarbeit von KDE mit einer Digitalkamera konfigurieren. Klicken Sie zunächst auf HINZUFÜGEN, um Ihre Kamera aus einer reichhaltigen Liste vorhandener Treiber auszuwählen. Danach wird der ANSCHLUSS der Kamera (SERIELL oder USB) gewählt; bei so gut wie allen unterstützten Geräten kommt nur USB in Frage. Sie können die neu hinzugefügte oder jede andere Kamera auswählen und dann eine der anderen Schaltflächen in der oberen Leiste anklicken:

- ▶ TESTEN: Das Funktionieren der Verbindung zur Kamera überprüfen
- ▶ ENTFERNEN: Die Konfiguration für diese Kamera löschen
- ▶ EINSTELLUNGEN: Detailoptionen für die ausgewählte Digitalkamera
- ▶ INFORMATIONEN: Daten der Kamera und der darauf gespeicherten Bilder anzeigen

▶ GERÄTE-AKTIONEN

Hier können Sie festlegen, welche Programme oder Befehle beim Anschluss bestimmter Geräte (zum Beispiel beim Anschließen eines USB-Sticks oder dem Einlegen einer CD) ausgeführt werden sollen.

► **HARDWARE**

Hier wird eingestellt, welche Dienste zur automatischen Konfiguration bestimmter Geräte zum Einsatz kommen. Der Dialog besteht aus drei Bereichen, und in jedem von Ihnen können Sie die verfügbaren Dienste nach oben oder nach unten verschieben, um ihre Priorität festzulegen. Die Bereiche sind:

- **POWER MANAGEMENT BACKEND:** Dienste für das Power-Management, also Einstellungen für den Umgang des Rechners mit verfügbarer Energie (zum Beispiel automatischer Ruhezustand nach einer gewissen Zeit der Inaktivität). Hier gibt es meist keine Priorität zu konfigurieren, da nur der Dienst HAL-ENERGIEVERWALTUNG zur Verfügung steht.
- **NETWORK MANAGEMENT BACKEND:** Die Tools zur automatischen Netzwerkkonfiguration. Standardmäßig hat NETWORKMANAGER 0.7 die höchste Priorität, was meist auch in Ordnung ist. Näheres über Netzwerke und auch den NetworkManager erfahren Sie in Kapitel 13, »Netzwerkgrundlagen«.
- **BLUETOOTH MANAGEMENT BACKEND:** Hilfsprogramme für den Kurzstrecken-Datenfunk Bluetooth. Hier steht in der Regel nur die Option BLUEZ zur Auswahl.

► **KDE-RESSOURCEN**

An dieser Stelle werden Ressourcen konfiguriert, die mehrere KDE-Anwendungen gemeinsam nutzen – insbesondere Kontaktdaten (Adressbücher) und Kalendereinträge. Klicken Sie auf HINZUFÜGEN, um eine neue Ressource auszuwählen. Es stehen zahlreiche Alternativen zur Auswahl, von einfachen Dateien – lokal oder im Netzwerk – bis hin zu LDAP-Servern (siehe Kapitel 14, »Datei- und Verzeichnisdienstserver) oder dem freien Groupware-Server OpenXChange.

► **SITZUNGSVERWALTUNG**

In diesem Bereich stellen Sie das Verhalten beim An- und Abmelden am KDE-Desktop ein. Unter ALLGEMEIN können Sie die (standardmäßig aktivierten) Optionen ABMELDUNG BESTÄTIGEN und OPTIONEN FÜR DAS HERUNTERFAHREN ANBIETEN ein- oder ausschalten. Beide dienen dazu, sicherzustellen, dass die Abmeldung von KDE nicht ohne Rückfrage erfolgt. Die STANDARDEINSTELLUNGEN FÜR DAS ABMELDEN regeln, was bei der Abmeldung geschieht, wenn Sie nichts Besonderes auswählen. Zur Auswahl stehen AKTUELLE SITZUNG BEENDEN, RECHNER HERUNTERFAHREN oder RECHNER NEU STARTEN. Im Bereich BEI DER ANMELDUNG legen Sie schließlich fest, ob und wann eine vorherige Sitzung wiederhergestellt werden soll: VORIGE SITZUNG WIEDERHERSTELLEN merkt sich automatisch die geöffneten Programme und Fenster, MANUELL GESPEICHERTE SITZUNG WIEDERHERSTELLEN reaktiviert nur eine aktiv von Ihnen

gespeicherte Sitzung, und MIT LEERER SITZUNG STARTEN beginnt stets mit einer völlig zurückgesetzten Standardsitzung.

Kategorie PERSÖNLICHES

Diese Kategorie enthält Einstellungen zu Land, Region und Sprache, zu Ihren persönlichen Anmeldeinformationen und zu den Standardprogrammen für bestimmte Aufgaben.

▶ Unterkategorie LAND/REGION & SPRACHE

In diesem Dialog können Sie Landeseinstellungen, Tastaturlayout und Rechtschreibprüfung konfigurieren.

- ▶ LAND/REGION & SPRACHE: Hier werden die Regional- und Spracheinstellungen vorgenommen (Locale). Unter REGIONALES können Sie zunächst die installierten Sprachen hinzufügen und ihre Priorität festlegen. Für jede Sprache lassen sich anschließend diverse Aspekte konfigurieren, auf Wunsch auch abweichend von den Standardeinstellungen für diese Sprache: ZAHLEN (zum Beispiel Tausender- und Dezimaltrennzeichen), WÄHRUNG, ZEIT & DATUM sowie SONSTIGE (Papierformat und Maßsystem).
- ▶ TASTATURLAYOUT: Dieser Dialog ermöglicht die Auswahl der Tastenbelegung. Auf der Registerkarte LAYOUT ist zunächst die Option TASTATURLAYOUTS DEAKTIVIEREN aktiv, so dass nur die bei der Installation gewählte Tastaturbelegung zur Verfügung steht. Wählen Sie TASTATURLAYOUTS AKTIVIEREN, um dies zu ändern; anschließend können Sie alternative Layouts hinzufügen. Auf dem Tab UMSCHALT-EINSTELLUNGEN geht es um Tastenkombinationen oder Panel-Schaltflächen zum Wählen der jeweils zu verwendenden Belegung, und ERWEITERT enthält detaillierte Einstellmöglichkeiten für das Verhalten einzelner Tasten.
- ▶ RECHTSCHREIBPRÜFUNG: Hier können Sie die über alle KDE-Programme hinweg zu verwendende Rechtschreibprüfung konfigurieren. Zunächst gibt es Optionen für das Ignorieren von Zusammensetzungen und Akronymen; darunter können Sie in ein Textfeld neue Wörter eintippen und mittels HINZUFÜGEN in das Wörterbuch aufnehmen.

▶ Unterkategorie PERSÖNLICHE INFORMATIONEN

Diese Abteilung enthält Dialoge zum Einstellen Ihrer Benutzerinformationen und Ihres Passworts sowie der Pfade Ihrer persönlichen Verzeichnisse.

- ▶ PASSWORT & BENUTZERZUGANG: Hier können Sie Ihr eigenes Passwort ändern und zusätzliche Benutzerinformationen wie Ihren vollständigen Namen, Ihre E-Mail-Adresse oder ein Bild editieren. Alles Wissenswerte zur Benutzerverwaltung wurde bereits in den beiden vorigen Kapiteln

erläutert; dieser Dialog ist allerdings weniger detailliert als die in Kapitel 3 beschriebene YaST-Benutzerverwaltung.

- ▶ **PFADE:** Stellen Sie hier Ihre persönlichen Verzeichnisse für Desktop, Downloads, Dokumente, Musik, Video und so weiter ein. Als Voreinstellung werden Unterverzeichnisse Ihres Home-Verzeichnisses verwendet, was in der Regel auch so bleiben sollte.

▶ STANDARD-KOMPONENTEN

Dies ist keine Unterkategorie, sondern ein einfacher Dialog. Er dient zur Festlegung Ihrer persönlichen Standardprogramme für diverse Aufgaben, zum Beispiel E-Mail, Dateimanager oder Webbrowser.

▶ ZUGANGSHILFEN

Formal handelt es sich um eine Unterkategorie, sie enthält jedoch nur einen gleichnamigen Dialog. Hier können Sie Hilfen für Benutzer mit Behinderungen einstellen. Unter **SIGNALE** können Sie optische und akustische Meldungen konfigurieren. **KLEBENDE TASTEN** ermöglicht das Einrasten von Modifikations-tasten wie **[Shift]** oder **[Strg]**. **VERLANGSAMTE TASTEN** drosselt auf Wunsch die Tastaturwiederholrate. **AKTIVIERUNGSGESTEN** schließlich ermöglicht das Einschalten der Eingabehilfen durch bestimmte Mausbewegungen.

Kategorie SYSTEM

Dieser Block enthält keine Unterkategorien, sondern sechs Dialoge für verschiedene Systemaspekte. Für einige von Ihnen müssen Sie jeweils das root-Passwort eingeben.

▶ ANMELDUNGSMANAGER

Wenn Sie den KDE-Displaymanager KDM verwenden (siehe Abschnitt 5.1), dann können Sie ihn hier genauer konfigurieren. Auf dem Tab **ALLGEMEIN** gibt es zunächst die Option **DESIGN FÜR DEN ANMELDUNGSMANAGER VERWENDEN**, und Sie können hier die zu verwendenden **SCHRIFTARTEN** einstellen. Die Tabs **DIALOG** (Text des Anmeldedialogs) und **HINTERGRUND** (Auswahl des Hintergrundbilds) stehen nur zur Verfügung, wenn Sie das Design abwählen, der Reiter **DESIGN**, auf dem Sie sich eines von mehreren vorgefertigten Designs aussuchen können, dagegen im umgekehrten Fall. Die Registerkarte **HERUNTERFAHREN** bestimmt, ob nur root oder alle Benutzer den Rechner vom Displaymanager aus herunterfahren dürfen. Auf dem Tab **BENUTZER** können Sie bestimmte Benutzer von der Anmeldung ausschließen und die Quelle für Benutzerbilder auswählen, und **VEREINFACHUNG** dient der Konfiguration einer automatischen Anmeldung.

- ▶ **AUFGABENPLANER:** Hier können Sie die Ausführung von sogenannten *Cron-jobs*, das heißt automatisierten Aufgaben, die zu bestimmten Zeitpunkten ausgeführt werden, komfortabel einstellen. Näheres über Cronjobs erfahren Sie in Kapitel 17, »System-Automatisierung«.
- ▶ **ENERGIEVERWALTUNG:** Ein ausführlicher Dialog für Energiesparoptionen. Im linken Bereich des Fensters können Sie drei verschiedene Bereiche auswählen, die dann rechts jeweils angezeigt werden:
 - ▶ **ALLGEMEINE EINSTELLUNGEN:** Im oberen Bereich gibt es einige Grundoptionen wie **BILDSCHIRM NACH DEM RUHEZUSTAND SPERREN**; unten wird verschiedenen Betriebsarten (Netzstecker, Akku, niedriger Akkuladestand) jeweils ein Profil zugeordnet.
 - ▶ **PROFILE ÄNDERN:** In jedem Profil wird eingestellt, nach wie vielen Minuten der Rechner in den Ruhezustand wechseln, wann er den Bildschirm abdunkeln soll und so weiter. Die vorgefertigten Profile reichen vom äußerst sparsamen **XTREME POWERSAVE** (vorausgewählt für kritischen Akkuladestand) bis zu **PRESENTATION**, bei dem – ideal für Präsentationen – niemals der Bildschirm abgedunkelt oder gar der Ruhezustand aktiviert wird. Sie können die vorhandenen Profile modifizieren, aber auch neue anlegen.
 - ▶ **FÄHIGKEITEN:** Hier sind keine Einstellungen möglich, sondern es wird eine Zusammenfassung angezeigt, über welche Energiesparmaßnahmen Ihr System überhaupt verfügt.
- ▶ **POLICYKIT-BERECHTIGUNGEN**

Das PolicyKit ist eine moderne Methode, Programme, für die eigentlich root-Berechtigungen erforderlich sind, durch nichtprivilegierte Benutzer steuern zu lassen. Alle bisherigen Methoden wie `setuid` (siehe Kapitel 4, »Mit der Shell arbeiten«) oder `sudo` (Näheres in Kapitel 17, »System-Automatisierung«) erlauben dem angemeldeten Benutzer nämlich möglicherweise zu viel, so dass er auch allgemein root-Rechte erlangen könnte. Im vorliegenden Dialog finden Sie eine umfangreiche Baumstruktur, in der Sie für verschiedene Aufgaben wie Drucker- und Netzwerkverwaltung konfigurieren können, welche Benutzer diese ausführen dürfen, ohne root-Rechte zu besitzen.
- ▶ **SAMBA**

Hier können Sie Freigaben für Windows-Clients über den Samba-Server verwalten. Näheres zu Samba steht in Kapitel 14, »Datei- und Verzeichnisdienst-Server«.
- ▶ **YAST**

Eine alternative Möglichkeit, das in Kapitel 2 und 3 ausführlich vorgestellte Systemverwaltungstool YaST aufzurufen.

Kategorie NETZWERK UND VERBINDUNGEN

In den beiden Unterkategorien FREIGABE und NETZWERKEINSTELLUNGEN können Sie Einstellungen für Netzwerkfreigaben und -verbindungen vornehmen. Da Netzwerke ausführlich ab Kapitel 13, »Netzwerkgrundlagen«, behandelt werden, bringt eine detaillierte Beschreibung an dieser Stelle noch nichts.

Kategorie SYSTEMVERWALTUNG

In diesem Bereich können Aspekte der Betriebssystemoberfläche konfiguriert werden, die nicht allzu tief ins Systemgeschehen eingreifen (was bei Laufwerken oder Netzwerkgeräten der Fall wäre) – es geht insbesondere um die Ein- und Ausgabegeräte.

▶ Unterkategorie ANZEIGE

Diese Unterkategorie enthält verschiedene Einstellmöglichkeiten für den Monitor.

- ▶ **GRÖSSE UND ORIENTIERUNG:** Stellen Sie in diesem Dialog Auflösung, Orientierung und Position des Monitorbildes ein. Voraussetzung ist die korrekte Konfiguration des X-Servers, die im ersten Abschnitt dieses Kapitels beschrieben wurde – Anzeigemodi, die aufgrund des verwendeten Bildschirmtreibers nicht zur Verfügung stehen, können Sie hier nicht auswählen.
- ▶ **ENERGIEKONTROLLE:** Sie können wählen, nach wie vielen Minuten der Inaktivität Ihr Monitor in den Bereitschaftsmodus wechseln oder gar ausgeschaltet werden soll.
- ▶ **GAMMA:** Die Gammakorrektur legt Farbtemperatur und Weißpunkt des Monitors fest. Je nach Monitorbauart und den Lichtverhältnissen der Umgebung können verschiedene Einstellungen angebracht sein.
- ▶ **MEHRERE MONITORE:** An dieser Stelle lässt sich nur etwas einstellen, wenn Sie zwei oder mehr Monitore parallel betreiben – dies gilt auch für einen Laptop mit einem zusätzlich angeschlossenen Monitor oder für die Arbeit mit einem Beamer. Grundsätzlich besteht die Möglichkeit, entweder das Monitorbild zu duplizieren, oder den Desktop über mehrere Bildschirme zu verteilen.

▶ DATUM & ZEIT

In diesem Dialog lassen sich Datum, Uhrzeit und Zeitzone einstellen, ähnlich wie in dem entsprechenden Dialog während der Installation, der in Kapitel 2 beschrieben wurde.

► MULTIMEDIA

Der Dialog kümmert sich um die Audio- und Videoeinstellungen. Der Tab GERÄTEPRIORITYÄT legt fest, in welcher Reihenfolge Geräte für die AUDIO-AUSGABE und die AUDIO-AUFNAHME verwendet werden sollen. Der Reiter BACKEND regelt dagegen die zu verwendenden Audio- und Video-Ausgabesysteme.

► SCHRIFTARTENINSTALLATION

In diesem Bereich können Sie die installierten Schriftarten verwalten (zum Beispiel ENTFERNEN) oder neue HINZUFÜGEN. Wenn Sie eine Schrift mit der rechten Maustaste anklicken, steht die Option IM SCHRIFARTENBETRACHTER ÖFFNEN zur Verfügung.

► Unterkategorie TASTATUR UND MAUS

Dieser Bereich stellt verschiedene Dialoge zur Tastatur- und Mauskonfiguration bereit.

► TASTATUR: Hier werden Wiederholrate und Verhalten der Num-Lock-Taste beim Systemstart eingestellt. Die Tastaturbelegung finden Sie dagegen im bereits erläuterten Dialog PERSÖNLICHES • LAND/REGION & SPRACHE • TASTATURLAYOUT.

► MAUS: Diverse Einstellungen für die Mausfunktionalität. Auf der Registerkarte ALLGEMEIN wird zunächst einmal grundsätzlich die RECHTSHÄNDIGE oder LINKSHÄNDIGE BEDIENUNG, das heißt die Belegung der Maustasten, eingestellt. Zudem haben Sie die Wahl zwischen den Optionen DOPPELKLICK oder EINZELKLICK ZUM ÖFFNEN VON DATEIEN/ORDNERN, wobei Letzteres bei KDE Standard ist. Die restlichen Tabs sind ZEIGERDESIGN für das Aussehen der Mauszeiger, ERWEITERT zum Festlegen von Geschwindigkeiten und Abständen sowie MAUSNAVIGATION zum Bewegen des Mauszeigers per Tastatur.

► JOYSTICK: Falls Sie Joysticks besitzen, können Sie sie an dieser Stelle konfigurieren.

► STANDARD-KURZBEFEHLE: Hier können Sie systemweite Tastenkürzel wie **Strg**+**C** zum Kopieren in die Zwischenablage oder **Strg**+**A** für »Alles auswählen« umkonfigurieren.

► GLOBALE KURZBEFEHLE: Dieser Dialog betrifft dagegen Tastenkürzel für bestimmte KDE-Komponenten, etwa die Lautstärkeregelung mit **KMix** oder das Ausführen von Befehlen mit **Alt**+**F2**.

► Unterkategorie TASTENKOMBINATIONEN

Und noch einmal Tastenkürzel – in dem einzigen, gleichnamigen Dialog dieser Unterkategorie können Sie zusätzliche Tastaturkurzbefehle einrichten.

5.3.2 Der Dateimanager Dolphin

Dolphin ist eine Neuentwicklung für KDE 4; bis zur Version 3 kam der auch als Webbrowser verwendbare Konqueror zur Verfügung. Während Letzterer durch immer neue Features recht überladen wirkte, ist Dolphin ziemlich schlank und übersichtlich.

Sie öffnen Dolphin am schnellsten durch die gleichnamige Schaltfläche im Panel. In der Standardansicht wird links der Dialog ORTE angezeigt; hier können Sie die verschiedenen Laufwerke, das Netzwerk oder den Mülleimer auswählen. Rechts erscheinen INFORMATIONEN zur jeweiligen Auswahl oder – falls Sie nichts ausgewählt haben – zum aktuellen Verzeichnis. Beide Dialoge lassen sich auf Wunsch durch Ziehen mit der Maus abtrennen und wieder andocken.

In der Mitte wird der eigentliche Dateimanager angezeigt, das heißt der Inhalt des Ordners, zu dem Sie gerade navigiert sind. Es gibt drei verschiedene Ansichtsmodi, die Sie über das Menü ANSICHT • ANSICHTSMODUS, mit Hilfe der entsprechenden Schaltflächen in der Symbolleiste oder per Tastenkombination auswählen können:

- ▶ SYMBOLE (**Strg**+**1**): Jedes Unterverzeichnis und jede Datei wird als Symbol angezeigt.
- ▶ DETAILS (**Strg**+**2**): Es erscheint eine Liste der Verzeichnisse und Dateien; welche Detailinformationen in den verschiedenen Spalten angezeigt werden, können Sie unter ANSICHT • ZUSATZINFORMATIONEN einstellen. Neben Verzeichnissen erscheint ein kleines Pluszeichen; wenn Sie darauf klicken, wird der Inhalt des Verzeichnisses eingerückt in der Liste angezeigt.
- ▶ SPALTEN (**Strg**+**3**): Diesen Modus haben die KDE-Entwickler bei Apples Betriebssystem Mac OS X abgeschaut. Die Inhalte des aktuell ausgewählten Verzeichnisses erscheinen in einer schmalen Spalte. Sobald Sie ein Unterverzeichnis anklicken, öffnet sich rechts daneben eine weitere Spalte, in der dessen Inhalt erscheint. Diese Vorgehensweise lässt sich für beliebig tief verschachtelte Verzeichnishierarchien fortsetzen. Abbildung 5.7 zeigt Dolphin in der Spaltenansicht.

Zusätzlich zu diesen grundlegenden Modi können Sie noch zwei Zusatzoptionen aktivieren: VORSCHAU vergrößert in jedem Modus die Symbole und zeigt den Inhalt von Bilddateien in Form von Thumbnails an. TEILEN unterteilt das Hauptfenster in zwei unabhängige Bereiche, in denen Sie verschiedene Verzeichnisse öffnen können – ideal zum Kopieren oder Verschieben von Dateien.

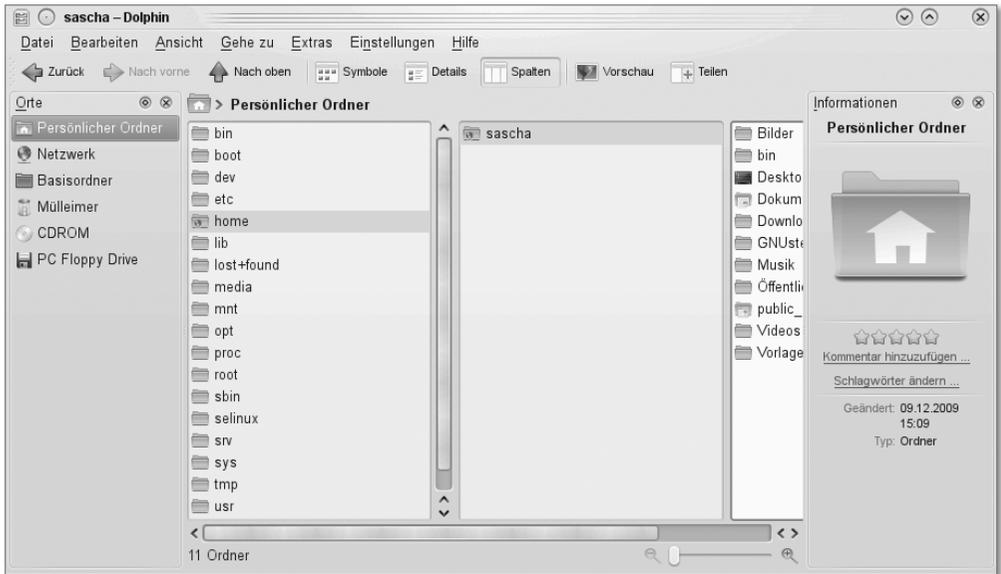


Abbildung 5.7 Der KDE-Dateimanager Dolphin in der Spaltenansicht

5.3.3 Die Terminalemulation KDE-Konsole

Eine recht genaue Vorstellung davon, was Sie alles in ein virtuelles Terminal oder ein Terminalfenster eingeben können, haben Sie im vorigen Kapitel erhalten. An dieser Stelle geht es lediglich um einige Besonderheiten der KDE-spezifischen Implementierung KONSOLE.

Im Hauptfenster wird normalerweise die Bash ausgeführt. Als zusätzliche Komfortfunktion stehen Rollbalken zur Verfügung, mit denen Sie die vorherigen Ausgabebildschirme einfacher als per + Bild ↑ und + Bild ↓ durchblättern können. Links unten finden Sie eine Schaltfläche, mit der sich eine weitere Instanz der Konsole auf einer separaten Registerkarte öffnen lässt. Falls Sie für den neuen Tab spezielle Eigenschaften wünschen, können Sie stattdessen die diversen Kommandos im Menü SITZUNG verwenden – etwa NEUE SECURE SHELL für eine SSH-Sitzung auf einem entfernten Rechner (siehe Kapitel 14, »Datei- und Verzeichnisdienstserver«), NEUES BEFEHLSFENSTER (SYSTEMVERWALTUNGSMODUS) für eine root-Shell, die natürlich die entsprechende Passwortheingabe erfordert, oder NEUER MIDNIGHT COMMANDER, um den beliebten Konsolen-Dateimanager mc (siehe Kasten weiter unten) zu öffnen, einen Nachbau des DOS-Klassikers Norton Commander. Abbildung 5.8 zeigt ein Konsole-Fenster mit geöffnetem Midnight Commander; auf den anderen Registerkarten befinden sich eine normale Shell, eine root-Shell sowie eine SSH-Verbindung.

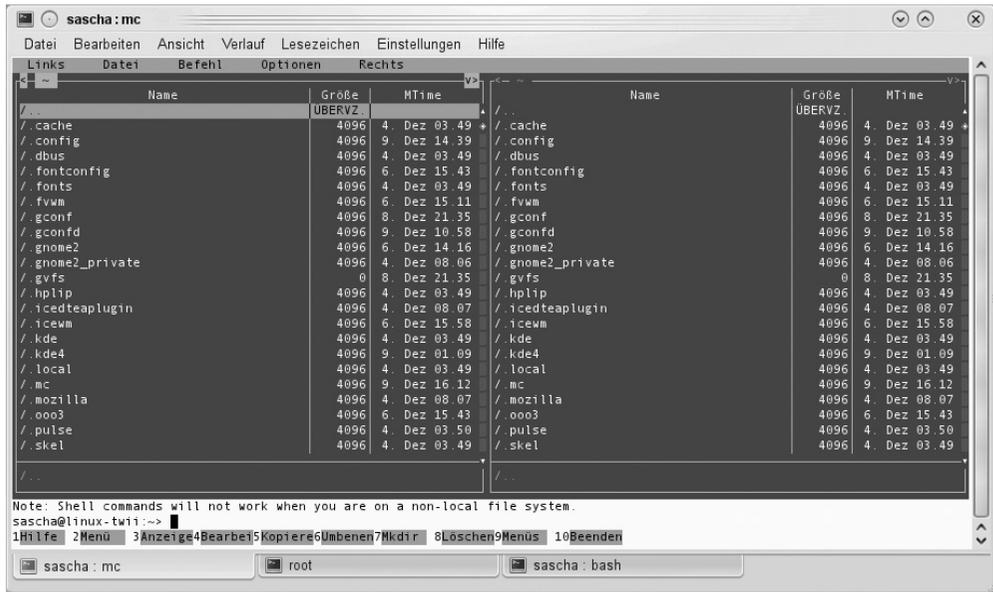


Abbildung 5.8 Der Midnight Commander und einige andere Terminalsitzungen in der KDE-Konsole

Um einen Karteireiter von Hand umzubenennen (im Beispiel nur die `user_shell`), klicken Sie ihn mit der rechten Maustaste an und wählen **SITZUNG UMBENENNEN**. Eine weitere interessante Option an dieser Stelle ist **SITZUNG VERSELBSTÄNDIGEN**, die aus der Registerkarte ein eigenes Fenster macht.

Sehr interessant ist die Funktion **LESEZEICHEN**: Hier können Sie sich jeweils das aktuelle Arbeitsverzeichnis merken, indem Sie **LESEZEICHEN HINZUFÜGEN** anklicken. Um später zum entsprechenden Verzeichnis zurückzukehren, ohne `cd` einzugeben, wählen Sie den betreffenden Eintrag einfach aus der Liste im **LESEZEICHEN**-Menü aus.

Einige weitere wichtige Optionen bietet das Menü **EINSTELLUNGEN**: Die **KODIERUNG** legt den jeweiligen Zeichensatz fest, was mitunter praktisch ist, falls Sie Textdokumente mit anderer Kodierung verwenden. Standard ist UTF-8, also zu ASCII abwärtskompatibler Unicode. Das **FARBSCHEMA** bestimmt, wie Hintergrund und Text dargestellt werden sollen. Die **GRÖSSE** schließlich legt die sichtbare Anzahl der Zeichen pro Zeile beziehungsweise der Zeilen (jeweils gemessen in Zeichen) fest.

Der Midnight Commander (mc)

Wenn Sie den Midnight Commander manuell starten möchten, können Sie übrigens in jedem beliebigen Terminal(fenster) `mc` eingeben. Die Bedienung ist sehr einfach: Zusätzlich zur manuellen Eingabe beliebiger Shell-Kommandos am Prompt, der fortwährend in der vorletzten Zeile angezeigt wird, können Sie mit den Pfeiltasten die Verzeichnisansichten durchblättern. `←` öffnet das jeweilige Unterverzeichnis beziehungsweise das übergeordnete Verzeichnis `..`; mit `F1` bis `F10` können Sie die ganz unten angezeigten Funktionen aufrufen. Hier einige Beispiele:

- ▶ `F3` zeigt Dateien an (standardmäßig Textdateien, aber mithilfe weiterer Funktionstasten etwa auch Hexadezimal-Ansichten von Binärdateien).
- ▶ `F4` öffnet Dateien zum Bearbeiten mit dem einfach zu bedienenden `mc`-Editor.
- ▶ `F6` ermöglicht das Umbenennen der aktuellen Datei.
- ▶ `F7` wird zum Erzeugen neuer Verzeichnisse verwendet.
- ▶ `F8` dient zum Löschen der jeweiligen Auswahl.
- ▶ `F10` beendet den `mc` beziehungsweise die jeweilige Unterauswahl.

5.3.4 Weitere KDE-Anwendungen

Zum Abschluss möchte ich Ihnen noch einige der vielen Hundert KDE-Programme und -Progrämmchen ans Herz legen, die zum Lieferumfang dieser GUI sowie zu openSUSE gehören. KDE-Software erkennen Sie in der Regel daran, dass ihr Name mit einem großen K beginnt. Hier also eine kurze Liste:

- ▶ Die Office-Suite *KOffice* ist eine Alternative zu dem in Kapitel 7, »OpenOffice.org«, vorgestellten Paket OpenOffice.org. Sie enthält etwas weniger Optionen, ist dafür aber auch ein wenig leichter bedienbar. Die wichtigsten Bestandteile sind die Textverarbeitung *KWord*, die Tabellenkalkulation *KSpread* sowie das Präsentationsprogramm *KPresenter*. Sie finden diese Einzelprogramme in den jeweiligen Kategorien des Menüs BÜROPROGRAMME unter dem K-MENÜ. Genau wie OpenOffice.org verwendet übrigens auch KOffice die offenen OASIS-Dateiformate.
- ▶ Das Bildbearbeitungsprogramm *Krita* – das im weiteren Sinne ebenfalls zu KOffice gehört – bietet zwar nicht ganz den Leistungsumfang von GIMP (siehe Kapitel 8, »Desktop-Software«), ist aber dafür ein wenig übersichtlicher.
- ▶ *Kate* wurde bereits mehrfach erwähnt. Es handelt sich um einen besonders leistungsfähigen Texteditor. Praktische Funktionen sind etwa das Suchen und Ersetzen mit regulären Ausdrücken oder die Syntaxhervorhebung für unzählige Programmier- und Skriptsprachen, Datenbanken oder Konfigurationsdateien.

- ▶ *Kontakt* ist ein Persönlicher Informations-Manager, der Funktionen wie Kalender, E-Mail (*KMail*) und Kontaktverwaltung enthält. Zudem ist ein Abgleich mit Groupware-Servern sowie mit Palm-PDAs (Komponente *KPilot*) möglich.
- ▶ *Krusader* ist ein sogenannter Twin-Panel-Dateimanager, der einer grafischen Variante des Midnight Commanders (siehe oben) ähnelt.
- ▶ Mit *KRename* können Sie beliebig viele Dateien unter Verwendung regulärer Ausdrücke nach einem gemeinsamen Schema umbenennen. Stellen Sie sich beispielsweise vor, Sie bereiten eine große Menge Dateien für den Einsatz auf einer Website vor. Groß- und Kleinschreibung sind problematisch, wenn sie in Links verwendet werden (weil Sie sich vertun könnten, sodass die verknüpfte Datei nicht gefunden wird); auch Leerzeichen sollten Sie vermeiden. In *KRename* ist nur ein einziger Arbeitsvorgang erforderlich, um beispielsweise alles kleinzuschreiben und alle Leerzeichen in (Unterstriche) umzuwandeln.
- ▶ *amaroK* ist ein benutzerfreundlicher Sound-Player. Beachten Sie, dass Sie zur MP3-Wiedergabe zusätzliche Pakete aus dem YaST-Schema VERSCHIEDENE PROPRIETÄRE PAKETE installieren müssen.

5.4 GNOME



Abbildung 5.9 Das GNOME-Kontrollzentrum

GNOME wird von manchen Benutzern als übersichtlicher empfunden als KDE. Jedenfalls präsentiert dieser Desktop sich etwas schlichter und schmuckloser. Einen ersten Überblick finden Sie in Kapitel 3, »Systemrundgang und Grundkonfiguration«. In diesem Unterabschnitt werden zusätzlich folgende GNOME-Aspekte besprochen:

- ▶ GNOME konfigurieren
- ▶ der Dateimanager Nautilus
- ▶ Besonderheiten des GNOME Terminals

5.4.1 GNOME konfigurieren

Wenn Sie Design und Verhalten von GNOME einstellen möchten, wählen Sie zunächst SYSTEM • KONTROLLZENTRUM aus dem Panel-Menü. Die Hauptseite dieses Dialogs wird in Abbildung 5.9 gezeigt. Doppelklicken Sie auf das jeweilige Icon, um etwas einzustellen. Im Einzelnen stehen folgende Kategorien (Überschriften) mit ihren jeweiligen Icons zur Verfügung:

- ▶ Kategorie AUSSEHEN UND VERHALTEN
Stellen Sie hier ein, wie der Desktop sowie die Fenster und Steuerelemente beschaffen sein sollen:
 - ▶ BILDSCHIRMSCHONER: GNOME bietet zwar nicht annähernd so viele verschiedene Bildschirmschoner wie KDE, aber die wenigen sind recht ansehnlich (mein persönlicher Favorit ist der WELTRAUM). Konfigurieren Sie in diesem Dialog, welcher Bildschirmschoner nach wie vielen Minuten aktiviert werden soll. Standardmäßig lässt sich der Bildschirmschoner nur durch Passwordeingabe beenden; auch dies können Sie hier ändern.
 - ▶ DESKTOP-HINTERGRUND: An dieser Stelle können Sie sich ein Hintergrundbild aussuchen. Wählen Sie alternativ KEIN HINTERGRUND, und stellen Sie stattdessen eine Hintergrundfarbe oder einen Farbverlauf ein – auf schlechter ausgestatteten Rechnern schont dies Ressourcen.
 - ▶ FENSTER: Legen Sie hier grundlegende Verhaltensweisen der Fenster fest. FENSTER AKTIVIEREN, WENN SICH DIE MAUS DARÜBER BEFINDET ist das klassische X Window-Fensterverhalten; zusätzlich kann das jeweilige Fenster auch nach einer definierbaren Zeitspanne in den Vordergrund geholt werden. Außerdem geht es darum, was bei einem Doppelklick auf die Titelleiste geschehen soll (Maximieren oder Einrollen), und welche Taste Sie gedrückt halten können, um das Fenster zu verschieben (abgesehen vom immer möglichen Ziehen an der Titelleiste).

- ▶ **FONTS:** In diesem Dialog werden die Schriftarten zur Verwendung in GNOME ausgewählt.
- ▶ **MENÜS UND WERKZEUGLEISTEN:** Einige Einstellungen für das Verhalten der wichtigsten Bedienelemente. Wählen Sie unter anderem, ob die Menüs mit oder ohne Symbole angezeigt werden sollen, ob sich Symbolleisten abtrennen lassen oder ob und wie diese beschriftet werden sollen.
- ▶ **THEMA:** Ein Thema ist eine Kombination aus Farben und Formen für die Fensterdekoration. Wählen Sie eines der vorgefertigten Themen, oder stellen Sie sich die einzelnen Aspekte unter **THEMENDetails** selbst zusammen. Mittels **THEMA INSTALLIEREN** können Sie auch ein heruntergeladenes Thema hinzufügen.
- ▶ **Kategorie HARDWARE**
Hier können Sie Einstellungen für diverse Geräte vornehmen; die meisten Optionen wurden weiter oben als SaX2-Einstellungen erklärt oder werden im nächsten Kapitel beziehungsweise in Kapitel 13, »Netzwerkgrundlagen«, (jeweils mit YaST) erläutert. Im Einzelnen stehen folgende Dialoge bereit:
 - ▶ **3D DESKTOP SETTINGS:** Falls Sie Xgl installiert haben (siehe oben), können Sie hier die einzelnen 3-D-Effekte konfigurieren.
 - ▶ **BILDSCHIRMAUFLÖSUNG:** Stellen Sie hier Auflösung und Wiederholrate des Monitors ein.
 - ▶ **BLUETOOTH:** Fragt Sie nach dem `root`-Passwort und öffnet den gleichnamigen YaST-Dialog.
 - ▶ **DRUCKER:** Öffnet die GNOME-Druckerverwaltung. Wenn Sie darin auf **NEUER DRUCKER** doppelklicken, können Sie über das GNOME-CUPS-Interface einen Drucker installieren, nachdem Sie sich als `root` angemeldet haben. Näheres zu CUPS und zur Druckerkonfiguration steht im nächsten Kapitel.
 - ▶ **DSL:** Startet den YaST-Dialog für DSL-Geräte (siehe Kapitel 13).
 - ▶ **GRAFIKKARTE UND MONITOR:** Öffnet den im ersten Abschnitt dieses Kapitels besprochenen X-Konfigurator SaX2.
 - ▶ **MAUS:** Konfigurieren Sie hier die Maustastenbelegung (links- oder rechts-händig), das Aussehen der Mauszeiger sowie die Mausgeschwindigkeit.
 - ▶ **MODEM:** Startet die entsprechende YaST-Komponente (siehe Kapitel 13).
 - ▶ **NETZWERKKARTE:** Auch dieser YaST-Aspekt wird ausführlich in Kapitel 13 behandelt.

- ▶ **NETZWERK-PROXIES:** Stellen Sie hier den Zugriff auf Internetdienste über Proxyserver ein (siehe Kapitel 14, »Datei- und Verzeichnisdienstserver«).
- ▶ **SCANNER:** Wieder eine YaST-Komponente (siehe nächstes Kapitel)
- ▶ **TASTATUR:** Hier können Sie allerlei Einstellungen für die Tastatur vornehmen – die Registerkarte TASTATUR regelt die Wiederholrate, BELEGUNGEN ist für die Sprache zuständig, BELEGUNGSEINSTELLUNGEN steuert weitere Details der Tastaturbelegung (etwa Caps-Lock, Euro-Zeichen oder Ähnliches). Die TIPPPAUSE schließlich (siehe Abbildung 5.10) erzwingt nach einem bestimmten Intervall durch Tastatursperrung, dass eine Zwangspause eingelegt werden muss – eine praktische Möglichkeit, Vieltipper wie den Autor dieses Buches vor einer chronischen Sehnenscheidenentzündung zu bewahren.

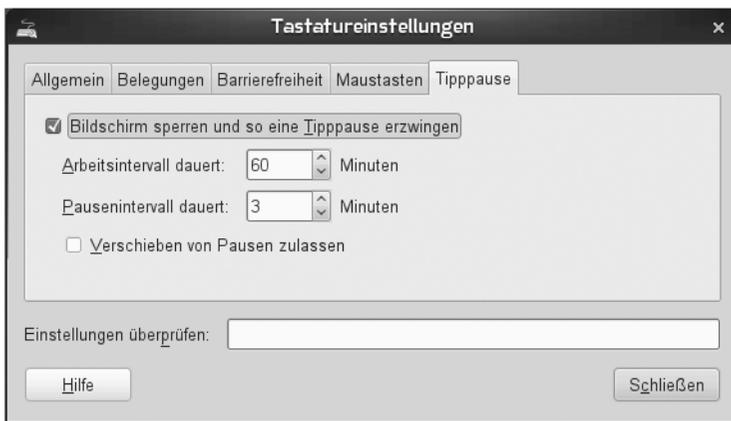


Abbildung 5.10 Manchmal sollte man sich zur Vernunft zwingen – kein Problem mit der GNOME-Tipppause.

- ▶ **WECHSELDATENTRÄGER UND -MEDIEN:** In diesem Dialog lässt sich detailliert konfigurieren, was mit CDs, DVDs, angeschlossenen Digitalkameras oder Ähnlichem geschehen soll. Unter SPEICHERORT können Sie zunächst auswählen, ob Wechselmedien automatisch gemountet werden sollen. UNTERHALTUNGSMEDIEN regelt, wie Audio-CDs, Video-DVDs oder angeschlossene mobile Musikplayer behandelt werden sollen. KAMERAS bestimmt, mit welchen Programmen die Daten von Digitalkameras beziehungsweise Digital-Videokameras importiert werden sollen. Die restlichen Registerkarten beschäftigen sich mit weiteren hot-plugging-fähigen USB-Geräten wie PDAs, Druckern und Scannern sowie Tastaturen und Mäusen.

► Kategorie PERSÖNLICH

Hier werden verschiedene benutzerspezifische Einstellungen vorgenommen:

- EINGABEHILFEN: Verschiedene Erleichterungen für die Tastatur- und Mausbenutzung wie »klebende« **[Strg]**- oder **[Alt]**-Tasten oder Tastenverzögerungen, die weiter oben bereits für KDE beschrieben wurden.
- HILFSTECHNOLOGIEN: Einige erweiterte Hilfselemente für Barrierefreiheit. Die BILDSCHIRMTASTATUR stellt beispielsweise alle Menüs und Schaltflächen in Textansicht dar und zeigt auf Wunsch auch eine komplette, mit der Maus bedienbare Tastatur auf dem Desktop an, und die LUPE vergrößert den jeweils aktuellen Bildschirmabschnitt.
- KURZBEFEHLE: Konfigurieren Sie hier diverse Tastenkürzel für GNOME-Befehle und zugehörige Komponenten.
- PASSWORT ÄNDERN: Hier können Sie Ihr eigenes Passwort ändern.
- SCHLÜSSELBUND VERWALTEN: Ähnlich wie KDE-Brieftasche ist der GNOME-Schlüsselbund ein Hilfsmittel zur Verwaltung verschiedener Passwörter unter einem gemeinsamen Dach.
- SPRACHE WÄHLEN: Öffnet den YaST-Dialog zur Sprachenauswahl (siehe Kapitel 2, »openSUSE installieren«).

► Kategorie SYSTEM

In dieser letzten Kategorie lassen sich diverse systemrelevante Einstellungen vornehmen:

- AUDIO: Konfigurieren Sie hier Klänge, die bei verschiedenen Ereignissen abgespielt werden sollen, sowie den Systemwarnton.
- BEAGLE-EINSTELLUNGEN: Beagle ist eine fortgeschrittene, indizierte Desktop-Suche von Novell. Sie können damit nach Text suchen, der (beispielsweise) in diversen Arten von Dateien, im Webcache und -verlauf oder in E-Mails vorkommt. An dieser Stelle lässt sich zunächst festlegen, dass die Such- und Indizierungsdienste automatisch gestartet werden sollen (dies ist bei openSUSE Voreinstellung). Zusätzlich können Sie wählen, mit welcher Taste oder Tastenkombination das Beagle-Suchfenster angezeigt werden soll (standardmäßig **[F12]**). Die Registerkarte INDIZIERUNG regelt, welche Verzeichnisse indiziert werden sollen und ob es Bereiche gibt, die aus Datenschutzgründen von der Indizierung ausgenommen werden. Letzteres ist auf Rechnern in öffentlichen Umgebungen wichtig, weil Sie sicher nicht möchten, dass andere Benutzer etwa Ihre Mails durchsuchen können. In Abbildung 5.11 sehen Sie das Ergebnis der Suche nach »Linux«. Wie dort gezeigt, lässt sich jede Ergebniskategorie separat durchblättern. Im Kasten weiter unten finden Sie einen kurzen Überblick über die Eingabe von Suchbegriffen in der Beagle-Suche.

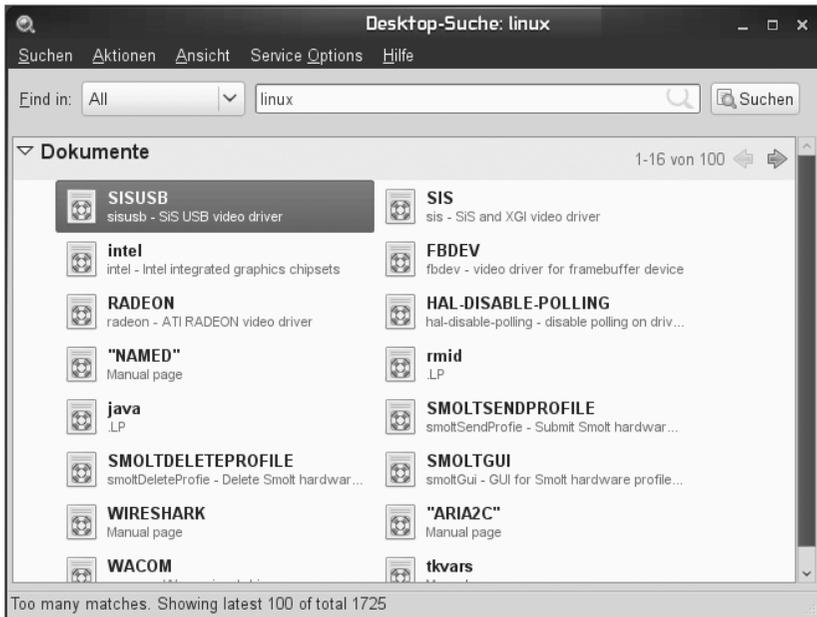


Abbildung 5.11 Ergebnis einer Suche nach »Linux« mit Beagle

- ▶ **BENUTZER BEARBEITEN UND ANLEGEN:** Öffnet den in Kapitel 3, »Systemrundgang und Grundkonfiguration«, ausführlich besprochenen YaST-Dialog zur Benutzerverwaltung.
- ▶ **BEVORZUGTE ANWENDUNGEN:** Wählen Sie in diesem Dialog, welche Programme Sie für bestimmte Verwendungszwecke nutzen möchten – im Einzelnen WEBBROWSER, E-MAIL-BETRACHTER, FTP, NEWS und TERMINAL.
- ▶ **DATUM UND ZEIT:** Auch dieses Icon verweist auf YaST, und zwar auf den in Kapitel 2 beschriebenen Dialog zur Einstellung von Datum, Uhrzeit und Zeitzone.
- ▶ **ENERGIEVERWALTUNG:** Hier können Sie verschiedene Energiespar-Optionen einstellen, unter anderem das Abschalten des Bildschirms sowie den Schlafmodus des Computers.
- ▶ **GSTREAMER-EIGENSCHAFTEN:** GStreamer ist der Multimedia-Layer von GNOME. Stellen Sie hier die Optionen zum Abspielen von Audio und Video ein. Sie können sie auch jeweils an Ort und Stelle testen.
- ▶ **SITZUNGEN:** Konfigurieren Sie in diesem Dialog, welche Programme und Dienste beim GNOME-Start automatisch aktiviert werden sollen.

Beagle-Suchbegriffe

Beagle Desktop Search ist nicht auf GNOME beschränkt, sondern gehört zum allgemeinen Leistungsumfang von openSUSE. Wie bereits erwähnt, wird die Suche in GNOME standardmäßig mit `F12` aktiviert. KDE enthält ein eigenes Interface zur Beagle-Suche namens Kerry Beagle. Es wird über `K-MENÜ • SYSTEM • DATEISYSTEM • KERRY BEAGLE` gestartet. Die Ergebnisse werden standardmäßig anders sortiert als in GNOME, nämlich nach Relevanz (prozentuale Gewichtung des Suchbegriffs im Gesamtdokument).

Egal, welche der beiden Oberflächen Sie verwenden – die Optionen für die eingegebenen Suchbegriffe sind stets identisch:

- ▶ Beagle Search unterscheidet nicht zwischen Groß- und Kleinschreibung.
- ▶ Wenn Sie mehrere Suchbegriffe eingeben, wird nach Dokumenten gesucht, die alle Begriffe enthalten (Und-Verknüpfung). Wenn Sie beispielsweise `Linux Windows` eingeben, erhalten Sie alle Ergebnisse, die »Linux« und »Windows« enthalten.
- ▶ Möchten Sie dagegen auch nach einzelnen Begriffen suchen, verknüpfen Sie diese einfach durch `OR`. Beispiel: `Linux OR Windows`.
- ▶ Für die Suche nach kompletten Ausdrücken werden Anführungszeichen verwendet, zum Beispiel `"GNU GPL"` oder `"Have a lot of fun"`.
- ▶ Wenn Sie nur nach bestimmten Dateitypen suchen möchten, können Sie mit `ext:Endung` eine Dateierweiterung angeben, etwa `ext:txt` für einfache Textdateien oder `ext:pdf` für PDF-Dokumente.
- ▶ Um Dokumente auszuschließen, die einen bestimmten Begriff enthalten, stellen Sie diesem ein Minuszeichen voran. Das folgende Beispiel findet Dokumente, in denen »Bill«, aber nicht »Gates« vorkommt: `Bill -Gates`.

5.4.2 Der GNOME-Dateimanager Nautilus

Nautilus ist kein Universalwerkzeug wie der alte KDE-3-Konqueror, sondern ein reiner Dateimanager wie *Dolphin* – er enthält keine Browser-Komponente.² Auch ansonsten gilt für den Unterschied zwischen den beiden Dateimanagern, was generell über die beiden Desktops gesagt wurde: Die GNOME-Variante besitzt weniger Features, erscheint aber etwas übersichtlicher. Sie öffnen *Nautilus* am einfachsten, indem Sie auf das Desktop-Symbol `PERSÖNLICHER ORDNER` doppelklicken oder aber eines der Verzeichnisse im Panel-Menü `ORTE` wählen. In Abbildung 5.12 sehen Sie *Nautilus* in der Listenansicht.

Unter dem Menü befindet sich eine einfache Leiste mit Standardkommandos – Vor- und Zurückblättern, eine Ebene höher, Laden abbrechen und so weiter. Darunter sehen Sie eine Leiste mit der aktuellen Verzeichnishierarchie in Form von Schaltflächen, damit Sie alle höher gelegenen Verzeichnisse erreichen können.

² Einen einfachen, robusten Browser besitzt GNOME ebenfalls; er heißt *Epiphany*. Aber auch für GNOME ist in openSUSE *Firefox* als Standardbrowser vorkonfiguriert.

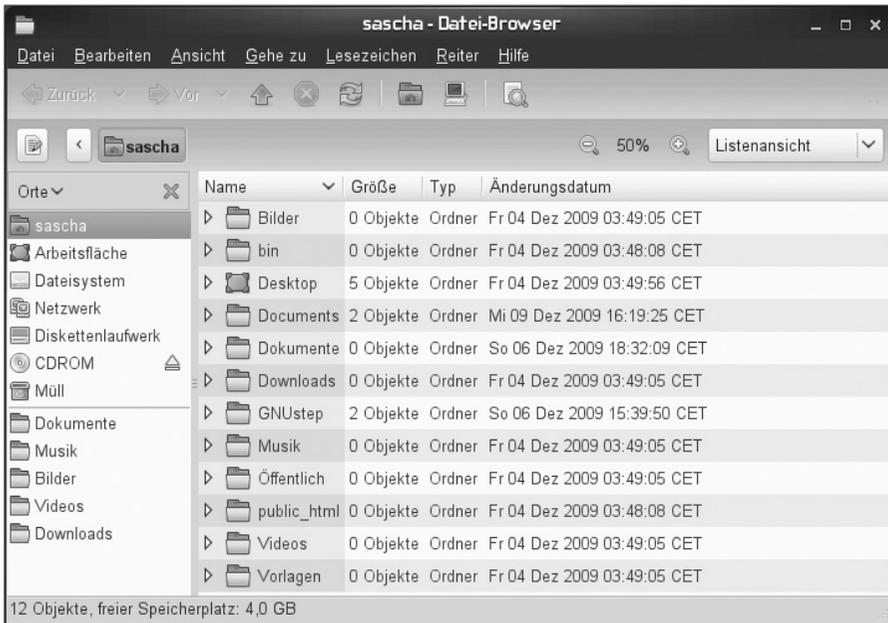


Abbildung 5.12 Der GNOME-Datei-Manager Nautilus im Listenmodus

Links finden Sie optional eine Seitenleiste (falls sie ausgeblendet sein sollte, wählen Sie ANSICHT • SEITENLEISTE). Mit dem Pulldown-Menü an ihrem Kopf können Sie sich für verschiedene Inhalte entscheiden:

- ▶ **ORTE:** Die wichtigsten Standardverzeichnisse wie Home-Verzeichnis, Wurzelverzeichnis und diverse Wechseldatenträger; entspricht dem Inhalt des gleichnamigen Panel-Menüs.
- ▶ **INFORMATIONEN:** Zeigt Infos über das jeweils ausgewählte Objekt sowie eine Liste von Programmen, mit denen es sich öffnen lässt.
- ▶ **BAUM:** Blendet eine aufklappbare Verzeichnishierarchie ein.
- ▶ **CHRONIK:** Eine chronologische Liste der zuletzt betrachteten Ordner.
- ▶ **NOTIZEN:** An dieser Stelle können Sie sich Anmerkungen zum aktuellen Ordner aufschreiben.
- ▶ **EMBLEME:** Eine Unmenge von Symbolen, die Sie auf beliebige Dateien und Ordner ziehen können, um diese speziell zu kennzeichnen. Um ein Emblem wieder loszuwerden, klicken Sie das jeweilige Objekt mit der rechten Maustaste an, wählen **EIGENSCHAFTEN** und wählen das betreffende Emblem ab.

Nautilus bietet nur zwei grundlegende Ansichten: ANSICHT • SYMBOLANSICHT (Tastenkürzel `[Strg] + [1]`) zeigt große Symbole und nur den Dateinamen an, während ANSICHT • LISTENANSICHT (`[Strg] + [2]`) eine Liste mit diversen Detailinformationen anzeigt. Die darin enthaltenen Ordner können Sie jeweils aufklappen, um eine verschachtelte Hierarchie zu öffnen. In beiden Ansichten öffnet ein Doppelklick auf einen Ordner exklusiv dessen Inhalt; die Symbolleiste oder Hierarchieleiste über der eigentlichen Verzeichnisansicht ermöglichen die Rückkehr zum übergeordneten Verzeichnis.

5.4.3 Das GNOME Terminal

Genau wie die KDE-Konsole besitzt auch das GNOME Terminal einige Komfortfunktionen. Beispielsweise können Sie auch hier mehrere Tabs öffnen, indem Sie DATEI • REITER ÖFFNEN • <PROFILNAME> wählen. Profile können verwendet werden, um unterschiedliche Kombinationen aus Schriften, Farben und weiteren Arten von Einstellungen zu speichern. Verwenden Sie dazu die Option BEARBEITEN • PROFILE. Dort können Sie mit NEU ein Profil erstellen oder ein vorhandenes BEARBEITEN. Der Menüpunkt TERMINAL • PROFIL WECHSELN ermöglicht es Ihnen, das Profil der aktuellen Registerkarte nachträglich auszutauschen. In Abbildung 5.13 sehen Sie ein GNOME Terminal mit drei Reitern, die jeweils per TERMINAL • TITEL FESTLEGEN umbenannt wurden.

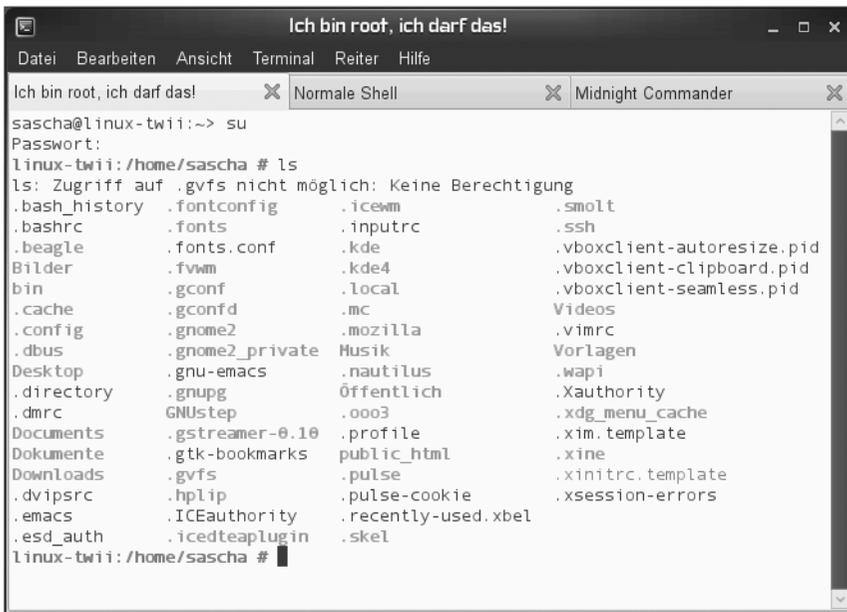


Abbildung 5.13 Ein angepasstes GNOME Terminal mit drei Registerkarten

Den Zeichensatz für den aktuellen Tab können Sie mit `TERMINAL • ZEICHENCODIERUNG FESTLEGEN` einstellen. Zunächst stehen nur UTF-8 (Standard) und ISO-8859-15 (die um ein Euro-Zeichen ergänzte Variante des westeuropäischen Zeichensatzes ISO-8859-1) zur Verfügung. Wählen Sie `HINZUFÜGEN / ENTFERNEN`, um weitere Zeichensätze zu verwalten.

5.4.4 GNOME-Anwendungen

Zum Umfeld von GNOME gehört eine Reihe von Anwendungen und Hilfsprogrammen. Hier eine kleine Auswahl im Schnellüberblick:

- ▶ *GNOME Office* ist kein in sich geschlossenes Office-Paket, sondern eine Sammlung zuvor unabhängig entwickelter Einzelprogramme. Es handelt sich um die Textverarbeitung *AbiWord*, die Tabellenkalkulation *Gnumeric* sowie die grafische Datenbankschnittstelle *GNOME-DB*.
- ▶ *The GIMP*, das mächtige Bildbearbeitungsprogramm, wird detailliert in Kapitel 8, »Desktop-Software«, besprochen. Dieser Anwendung hat GNOME seine Grafikbibliothek zu verdanken.
- ▶ *Evolution* ist die GNOME-Komponente für E-Mail-, Information-Management- und Groupware-Funktionen.
- ▶ *Banshee* ist der GNOME Audio-Player.

5.5 Weitere Window-Manager

Neben KDE und GNOME stellt openSUSE noch diverse andere Window-Manager beziehungsweise Desktops bereit. In diesem Abschnitt erhalten Sie einen kurzen Überblick über einige von ihnen, allerdings können diese hier nicht wie KDE und GNOME vertieft werden. Beachten Sie, dass die meisten dieser Desktops standardmäßig nicht installiert werden, sondern nur dann, wenn Sie in YaST gezielt danach suchen und sie manuell hinzufügen.

5.5.1 FVWM – der Klassiker

Der *FVWM* (ursprünglich *Feeble Virtual Window Manager*, später dann nur noch *F Virtual Window Manager*) wurde ab 1993 von *Robert Nation* und *Chuck Hines* entwickelt. Die aktuelle Version wurde zur Abgrenzung lange Zeit als *FVWM2* bezeichnet; seit die Vorversion praktisch keine Bedeutung mehr hat, ließ man die 2 in der alltäglichen Bezeichnung fallen. Bevor KDE und GNOME populär wurden, war *FVWM* der Standard-Window-Manager unter Linux. Die zurzeit stabile Version ist 2.4.16. *FVWM* steht unter der GNU GPL.

Die Ausstattung ist relativ spartanisch, aber dafür speicherschonend und doch funktional. FVWM führte als einer der ersten Window-Manager das Konzept der virtuellen Desktops ein; standardmäßig stehen sechs Stück zur Verfügung, zwischen denen Sie mithilfe des Feldes ganz links in der Bedienleiste am oberen Bildschirmrand wechseln können.



Abbildung 5.14 FVWM – ein speicherschonender Windows-Manager

Einige Besonderheiten von FVWM sind dem »unbekleidet« durchscheinenden X Window System zu verdanken. Beispielsweise erscheint ein neu geöffnetes Fenster zunächst als frei schwebendes Gitternetz, bis Sie es durch einen Mausklick an der gewünschten Stelle platzieren. Standardmäßig werden die einzelnen Fenster auch durch einfaches Darüberfahren mit der Maus aktiviert.

In Abbildung 5.14 sehen Sie FVWM; geöffnet sind die der Webbrowser Firefox sowie die ebenfalls von Rob Nation geschriebene Terminalemulation `rxvt`.

5.5.2 WindowMaker

Der *GNU WindowMaker* basiert auf einigen Design-Ideen von NeXTStep (NeXT war eine später von Apple aufgekaufte Firma, auf deren Betriebssystemideen Mac OS X basiert). Wesentliche Eigenschaften sind die WindowMaker-Applets, kleine dockbare Anwendungsboxen, sowie eine beliebige Anzahl virtueller Arbeitsflächen, die Sie öffnen und wieder schließen können. Die meisten Befehle

werden aufgerufen, indem man mit der rechten Maustaste auf den leeren Desktop klickt und sich durch die verschachtelten Menüs navigiert. Der WindowMaker ist hochgradig konfigurierbar. In openSUSE sind bereits zahlreiche Themen und Stile enthalten; im Internet können Sie zahlreiche weitere herunterladen.

Abbildung 5.15 zeigt den WindowMaker mit einigen Applets (links unten), dem geöffneten Menü für weitere Applets sowie seinem eigenen Einstellungsdialog.



Abbildung 5.15 Der GNU WindowMaker

5.5.3 IceWM

Der Desktop *IceWM* wurde von *Marko Macek* entwickelt. Das Leitmotiv dieses Window-Managers ist Benutzerfreundlichkeit durch vornehme Zurückhaltung (»not getting in the user's way«), was mit einer guten Performance einhergeht. Auch IceWM lässt sich durch zahlreiche Themen und Stile anpassen; mitgeliefert werden unter anderem einige, die das Aussehen von Windows 95, OS/2 Warp 4 oder dem UNIX-Klassiker Motif nachahmen. Standardmäßig stehen vier virtuelle Arbeitsflächen zur Verfügung; am unteren Rand befindet sich ein Panel, das der Windows-Taskleiste ähnelt.

In Abbildung 5.16 sehen Sie IceWM mit dem Webbrowser Opera, einem `ml` Terminalfenster und geöffnetem Startmenü.

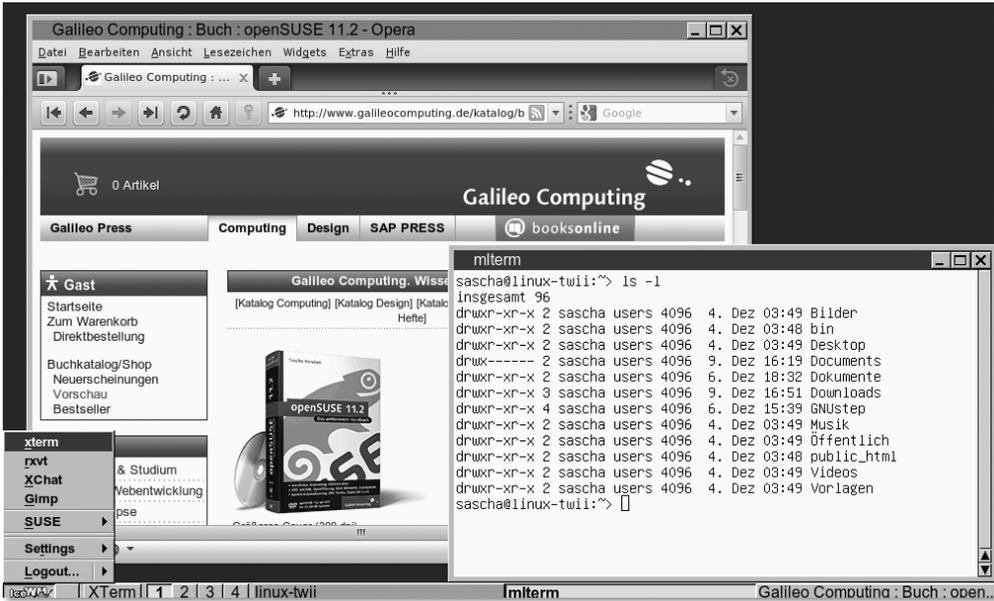


Abbildung 5.16 Der Window-Manager IceWM

5.6 Zusammenfassung

Während die im vorigen Kapitel vorgestellte Shell besonders effizient ist, dienen grafische Benutzeroberflächen dem komfortablen Arbeiten. Zudem gibt es zahllose Computeranwendungen, die in einem textbasierten Terminal gar nicht möglich sind, beispielsweise Bildbearbeitung, WYSIWYG-Textverarbeitung und Multimedia.

Das Herzstück der grafischen Oberfläche auf jedem UNIX-artigen System ist der X Window Server oder kurz X-Server. Dieser stellt die grundlegende Grafikfunktionalität bereit – sowohl für lokale als auch für entfernte Anwendungen (daher eben »Server«). Unter Linux wird heute in der Regel der X.org-Server verwendet, die Referenzimplementierung des X-Konsortiums.

Mit SaX2 steht unter SUSE Linux beziehungsweise openSUSE seit einigen Jahren ein besonders benutzerfreundliches X-Konfigurationstool zur Verfügung, zudem wird der X-Server heutzutage in aller Regel bereits während der Installation korrekt eingerichtet.

Um tatsächlich mit fensterbasierten Anwendungen arbeiten zu können, benötigen Sie außer dem X-Server einen Window-Manager oder Desktop. Die beiden weitverbreiteten, exzellent ausgestatteten Desktops KDE und GNOME wurden in

diesem Kapitel relativ ausführlich vorgestellt (die jeweilige Einführung für die ersten Schritte finden Sie allerdings in Kapitel 3, »Systemrundgang und Grundkonfiguration«). Daneben wurden einige alternative Window-Manager kurz angesprochen.

*Gegenwärtig konkurriert die Maschine mit dem Menschen.
Unter den richtigen Umständen wird sie ihm dienen.
– Oscar Wilde*

6 Wichtige Hardware konfigurieren

Wie Sie in Kapitel 2, »openSUSE installieren«, bemerkt haben dürften, erkennt die Distribution zahlreiche Hardwaregeräte bereits bei der Installation. Sollten einige Ihrer Geräte nicht dazugehören, erhalten Sie in diesem Kapitel die wichtigsten Informationen darüber, wie Sie diese dennoch aktivieren können. Beachten Sie, dass Netzwerkgeräte exklusiv in Kapitel 13, »Netzwerkgrundlagen«, besprochen werden. Einiges Hintergrundwissen über die diversen Geräteanschlüsse erhalten Sie zudem in Kapitel 18, »Hinter den Kulissen«.

6.1 Kernel-Module

Bereits in Kapitel 1, »Einführung«, wurde erwähnt, dass der Linux-Kernel die meisten Gerätetreiber in Form dynamisch ladbarer Module bereitstellen kann. Dies ist besonders nützlich, wenn es um moderne *Hot-Plugging*-Geräte geht, die über USB oder ähnliche Schnittstellen im laufenden Betrieb angeschlossen und bei Bedarf wieder entfernt werden können. In der Regel werden die betreffenden Kernel-Module sogar vollautomatisch geladen. Um die Informationen in diesem Abschnitt brauchen Sie sich also in der Regel erst dann zu kümmern, wenn ein Gerät nicht auf Anhieb funktioniert. Im zweiten Unterabschnitt erfahren Sie dann, wie Sie selbst einen eigenen, angepassten Linux-Kernel kompilieren können. Dies ist heutzutage eigentlich nur noch in Ausnahmefällen nötig.

6.1.1 Module laden und entladen

Bevor Sie erwägen, ein Kernelmodul manuell zu laden, sollten Sie sich einen Überblick über die bisher aktiven Module verschaffen. Dazu wird folgende Konsolenanweisung verwendet:

```
$ lsmod
```

Das Ergebnis ist eine Liste mit Einträgen wie diesen:

```
parport_pc          43492  1
lp                  15652  0
parport             40648  2 parport_pc,lp
```

Die erste Spalte enthält den Namen des jeweiligen Moduls. Es handelt sich um den relevanten Teil des jeweiligen Dateinamens. Die eigentlichen Kernelmodul-Dateien befinden sich in Verzeichnissen nach folgendem Schema:

```
/lib/modules/Kernelversion/kernel/drivers/Kategorie/Modul.ko
```

Die Dateiendung *.ko* steht für »Kernel Object«; dies zeigt, dass das Konzept der Kernelmodule den *Dynamic Shared Objects* (DSO) ähnelt, die die Funktion von Anwendungen modular ergänzen und die Endung *.so* besitzen. Der oben gezeigte Eintrag `parport` (es handelt sich um den Treiber für den klassischen, kaum noch genutzten Parallelport) besitzt in openSUSE 11.2 beispielsweise folgenden Dateipfad:

```
/lib/modules/2.6.31.5-default/kernel/drivers/parport/parport.ko
```

In der zweiten Spalte steht die Größe des jeweiligen Moduls in Byte. Die dritte enthält eine Information darüber, von welchen anderen Treibern und Diensten das Modul genutzt wird. Die Nummerierung ist dabei eine Art Rangfolge: Treiber der Stufe 0 werden unmittelbar vom Kernel geladen; die Stufe 0 lädt die Module mit der Stufe 1, diese lädt wiederum die Stufe 2 und so weiter. Dahinter stehen mitunter die Namen von Modulen, die das aktuelle Modul benötigen.

Falls ein Kernelmodul beim Systemstart oder beim Anschluss eines Geräts nicht automatisch geladen wird, können Sie es folgendermaßen dynamisch laden (dazu müssen Sie `root` sein):

```
# insmod Modul
```

Dies scheitert allerdings, wenn andere Module nicht geladen sind, von deren Funktionalität das neu zu ladende Modul abhängt. Um dieses Problem zu umgehen, können Sie `modprobe` statt `insmod` verwenden – diese Anweisung löst alle Modulabhängigkeiten auf, indem sie automatisch zuerst die benötigten Module lädt. Die Syntax lautet in den meisten Fällen einfach:

```
# modprobe Modul
```

Bei beiden Kommandos können Sie optionale Parameter hinzufügen, was aber meist nur für veraltete ISA-Karten und ähnliche antike Hardware erforderlich ist. Die beiden wichtigsten Parameter sind:

- ▶ `irq=Nummer` – ein IRQ (Interrupt Request) ist ein Kommunikationskanal für ein bestimmtes Gerät beziehungsweise einen Anschluss.
- ▶ `io=Adresse` – die I/O-Basisadresse ist eine virtuelle Speicheradresse, über die ein Gerät mit dem Prozessor kommuniziert.

Näheres über diese Hardware-Ressourcen erfahren Sie in Kapitel 18, »Hinter den Kulissen«.

Das folgende Beispiel lädt den Treiber für eine altmodische SoundBlaster-16-ISA-Soundkarte mit dem IRQ 5 und der I/O-Adresse 0x0220 (einen solchen Rechner, Baujahr bis ca. 1996, werden Sie in der Praxis kaum noch irgendwo finden):

```
# modprobe snd-sb16 irq=5 io=220
```

Wenn Sie ein Modul wieder entladen möchten, weil Sie das betreffende Gerät nicht mehr benötigen, können Sie das Kommando `rmmmod` verwenden. Für die oben gezeigte alte Soundkarte sieht dies beispielsweise folgendermaßen aus:

```
# rmmmod snd-sb16
```

6.1.2 Einen eigenen Kernel kompilieren

In alten Kernel-2.0-Zeiten war das Kompilieren eines eigenen Linux-Kernels an der Tagesordnung, wenn es um die Verwendung exotischerer Hardware, beispielsweise fast jede Art von Multimedia-Unterstützung, ging. Heute ist das Kernel-Kompilieren eine interessante Option, die aber in den meisten Fällen nicht benötigt wird.

Falls Sie dennoch einen neuen Kernel kompilieren möchten (oder in Ausnahmefällen müssen), geht dies mit folgenden Arbeitsschritten:

1. Überprüfen Sie, ob die Kernelquellen installiert sind. Das betreffende YaST-Schema heißt *Linux-Kernel-Entwicklung* (siehe Kapitel 2, »openSUSE installieren«); das eigentliche Paket `kernel-source`. Bitte beachten Sie, dass die Kernelquellen Ihrer Distribution um einige Funktionen ergänzt wurden; die offiziellen Kernel-Archive von <http://www.kernel.org> besitzen nicht denselben Leistungsumfang.
2. Wechseln Sie als `root` in das Verzeichnis `/usr/src/linux` (ein Symlink auf `/usr/src/linux/Aktuelle_Kernel_Version`):

```
# cd /usr/src/linux
```

3. Geben Sie Folgendes ein, um das Kernel-Konfigurationsprogramm zu starten:

```
# make xconfig
```

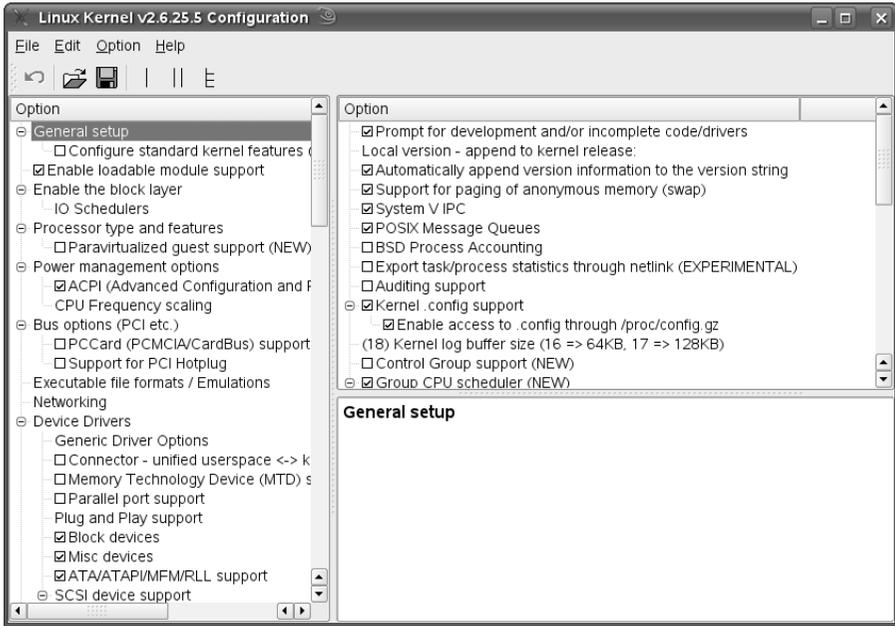


Abbildung 6.1 »xconfig« – X-basiertes Tool zum Konfigurieren der Kernel-Kompilierung

xconfig ist ein GUI-basiertes Konfigurations-Tool, dessen Startbildschirm in Abbildung 6.1 gezeigt wird. Sie können auch die folgende Alternative verwenden, wenn Sie keine grafische Oberfläche nutzen können oder wollen:

```
# make menuconfig
```

Das Ergebnis dieses Kommandos sehen Sie in Abbildung 6.2. Beide Tools besitzen dieselben Optionen. menuconfig wird etwas schneller geladen, weil es textbasiert ist; xconfig ist dafür ein wenig komfortabler.

1. Gehen Sie die einzelnen Kategorien der Konfiguration durch, und überprüfen Sie, was für Sie infrage kommt. Falls sich Treiber als Module laden lassen, sollten Sie diese Option im Allgemeinen der statischen Kompilierung vorziehen. Einen Überblick über die einzelnen Kategorien finden Sie in Tabelle 6.1.
2. Beenden Sie das jeweilige Konfigurationsprogramm, wobei Sie wählen müssen, dass die Einstellungen gespeichert werden sollen.
3. Geben Sie nun nacheinander die folgenden Befehle ein, um den Kernel zu kompilieren:

```
# make all
# make modules_install
```

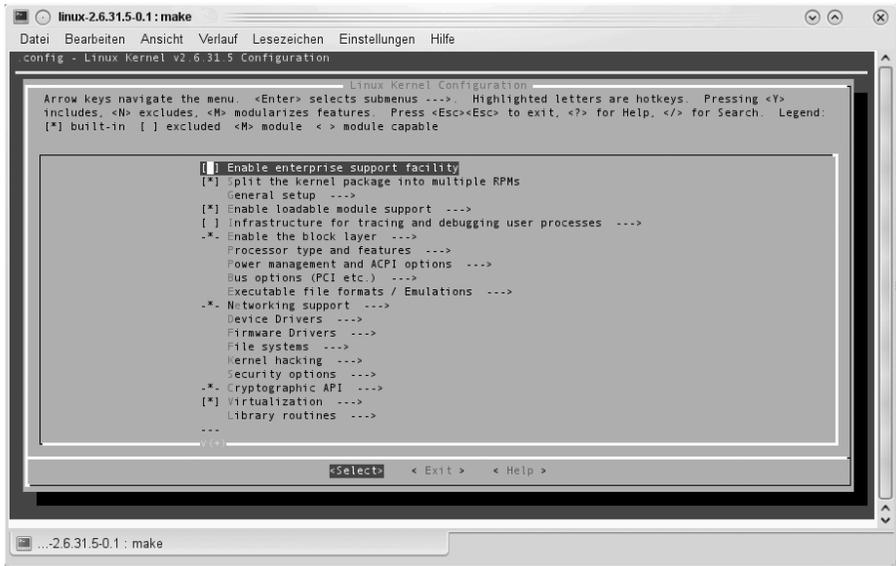


Abbildung 6.2 »menuconfig« – das textbasierte Tool zur Kernel-Konfiguration

Da dies recht lange dauern kann, lohnt es sich, folgende Kombination einzutippen und anschließend zum Beispiel in die Mittagspause zu gehen:

```
# make all && make modules_install
```

4. Wenn alles korrekt gelaufen ist, befindet sich der neu kompilierte Kernel in der Datei `/usr/src/linux/arch/i386/boot/bzImage`. Kopieren Sie ihn unter einem neuen Namen nach `/boot`. Beispiel:

```
# cp /usr/src/linux/arch/i386/boot/bzImage \
```

```
> /boot/kernel-2.6.31-new
```

Zu guter Letzt müssen Sie Ihren Bootmanager dazu bringen, den neuen Kernel zum Booten anzubieten. Wie Sie GRUB oder Lilo entsprechend konfigurieren können, steht in Kapitel 9, »Datenträger und Dateisysteme«).

Ältere Kernel kompilieren

Falls Sie jemals in die Verlegenheit geraten sollten, einen Kernel der Versionen 2.0 bis 2.4 kompilieren zu müssen, lauten die Befehle nach dem Durchlauf von `make xconfig` beziehungsweise `make menuconfig` etwas anders. Im Einzelnen müssen Sie Folgendes eingeben:

```
# make dep
# make clean
```

```
# make modules
# make modules_install
# make bzImage
```

Danach geht es auch hier mit dem Kopieren des neuen Kernels nach `/boot` und der Konfigurationsänderung des Bootmanagers weiter.

Kategorie	Bedeutung
CODE MATURITY LEVEL OPTIONS	Warnmeldung, bevor in Entwicklung befindliche oder nicht ausreichend getestete Bestandteile eingefügt werden
GENERAL SETUP	Grundbestandteile des Kernels wie System V IPC, Sysctl usw. Schalten Sie ohne guten Grund <i>niemals</i> eine dieser Optionen aus.
LOADABLE MODULE SUPPORT	Unterstützung für dynamisch ladbare Module (siehe oben). Sollte in aller Regel ebenfalls nicht deaktiviert werden.
ENABLE THE BLOCK LAYER	I/O-Scheduler für besonders große Block-Devices (Festplatten und Ähnliches)
PROCESSOR TYPE AND FEATURES	Wählen Sie hier den Typ und die besonderen Funktionen Ihres Prozessors, um den Kernel dafür zu optimieren.
POWER MANAGEMENT OPTIONS (ACPI, APM)	Unterstützung für Power Management und Energiesparoptionen
BUS OPTIONS (PCI, PCMCIA, EISA, MCA, ISA)	Treiber für die diversen Bus- und Anschluss-Systeme des Rechners
EXECUTABLE FILE FORMATS	Die verschiedenen Binärformate für ausführbare Programme. ELF ist Linux-Standard, a.out und ECOFF sind andere Formate, und MISC-Binaries ermöglichen die Behandlung von Programmen, die zur Ausführung einen Interpreter oder eine virtuelle Maschine benötigen.
NETWORKING	Treiber für Netzwerkgeräte und -protokolle
DEVICE DRIVERS	Die diversen Klassen von Gerätetreibern
FILE SYSTEMS	Dateisysteme (Ext2, Ext3, ReiserFS, ISO, NTFS und viele andere)
INSTRUMENTATION SUPPORT	Profiling-System (experimentell)
KERNEL HACKING	spezielle Kernel-Features

Tabelle 6.1 Kategorien der Kernel-Konfiguration

Kategorie	Bedeutung
SECURITY OPTIONS	Rechner- und Netzwerk-Sicherheitseinstellungen
CRYPTOGRAPHIC OPTIONS	Unterstützung zahlreicher Kryptografie-Algorithmen wie MD5, SHA1 oder Blowfish
LIBRARY ROUTINES	diverse Prüfsummenalgorithmen (CRC16 und CRC32)

Tabelle 6.1 Kategorien der Kernel-Konfiguration (Forts.)

6.2 Drucker

Drucker gehören zu den verbreitetsten Peripheriegeräten überhaupt. Sie ermöglichen die dauerhafte Ausgabe von Computerdaten für den Postversand, zur Archivierung, zum bequemeren Lesen oder für Fotos, die Sie sich an die Wand hängen können. In diesem Abschnitt erfahren Sie, wie Sie Ihren Drucker unter openSUSE konfigurieren können, was seit der Einführung des CUPS-Standards recht einfach geworden ist.

6.2.1 Druckerarten

Von den zahllosen Drucker-Bauarten, die im Lauf der letzten vierzig Jahre entwickelt wurden, interessieren heute im Grunde nur noch zwei: Tintenstrahldrucker und Laserdrucker.

Tintenstrahldrucker schießen durch feine Düsen winzige Tintentropfen auf das Papier. Die ältesten von ihnen druckten nur mit schwarzer Tinte, die durch unterschiedlich große Rasterpunkte verschiedene Graustufen erzeugen konnte. Später kamen die ersten Farbtintenstrahldrucker auf den Markt. Viele alte Modelle waren nur mit den drei subtraktiven Grundfarben Cyan, Magenta und Gelb ausgestattet. Sie stellten das fehlende Schwarz durch Übereinanderdrucken der drei anderen Farben her. Dieser Vorgang ergibt zwar rein rechnerisch Schwarz, in der Praxis war das Ergebnis aber zu blass und zu kontrastarm. Aus diesem Grund sind moderne Farbtintenstrahldrucker zusätzlich mit Schwarz als vierter Druckfarbe ausgestattet und entsprechen damit dem Vierfarbdruck. Einige hochwertige Fotodrucker verfügen sogar über sechs bis acht Farben. Dies ermöglicht einen farbtreueren Ausdruck bestimmter heller Farben und Pastelltöne. Die meisten Modelle besitzen eine Patrone für Schwarz und eine für die anderen Farben. Nur die Farbtintenstrahler der Firma Canon sind seit jeher mit vier einzelnen Tintenkartuschen versehen.

Die Auflösung von Tintenstrahldruckern beträgt zwischen 300 und 1.440 dpi (Punkte pro Zoll). Die besonders hohe Auflösung mancher Drucker wird vor allem für den Fotodruck eingesetzt.

Es gibt zwei etwas unterschiedliche Verfahren für den Tintenstrahldruck: die Bubble-Technik und die Piezo-Technik. Bei einem Bubble-Drucker wird ein kleiner Draht sehr stark erhitzt und erzeugt dadurch eine Tintendampfblase, die durch die Düse auf das Papier »geschossen« wird. Beim Piezo-Verfahren wird dagegen ein sogenannter Piezo-Kristall verwendet, der sich durch elektrische Spannung ausdehnt und auf diese Weise einen Tintentropfen auf das Papier drückt.

Eine noch höhere Druckqualität als der Tintenstrahldrucker bietet der *Laserdrucker*. Dieses Gerät arbeitet mit einer Technik, die vom Fotokopierer stammt (Xerografie): Eine rotierende, elektrisch leitfähige Trommel wird nach und nach an verschiedenen Stellen durch einen Laserstrahl belichtet, der ihre elektrische Ladung an den entsprechenden Stellen ändert. Anschließend wird die Trommel mit Toner bedeckt, der jedoch nur an den geladenen Stellen haftet und von anderen wieder abfällt. Daraufhin wird die Trommel auf einen Bogen Papier abgerollt. Anschließend wird der Toner durch Hitze fixiert: Die enthaltenen Kunststoffanteile schmelzen und verbinden sich mit dem Papier. Deshalb ist das Papier so warm, wenn es aus dem Laserdrucker kommt. Bei Farblaserdruckern wird der ganze Vorgang für die vier Druckfarben insgesamt viermal wiederholt; jedes Mal wird Toner einer anderen Farbe aufgetragen.

Unabhängig von der Bauart werden verschiedene Drucker übrigens durch unterschiedliche *Druckersprachen* angesteuert. Die Lösung mit dem hochwertigsten Ergebnis ist *Adobe PostScript*; dies beherrschen allerdings nur die teuersten Drucker. Die meisten Drucker verwenden ihre eigenen Sprachen; ein Beispiel ist etwa *HPGL* für Drucker von Hewlett-Packard. Es gibt zudem einige besonders preisgünstige Drucker, die das sogenannte *GDI*-Verfahren verwenden: Sie implementieren gar keine eigene Steuerungssprache und verlassen sich darauf, dass das Betriebssystem (traditionellerweise Microsoft Windows) ihnen fertig aufbereitete Bitmaps liefert. Inzwischen werden solche Drucker auch leidlich von Linux unterstützt, wenngleich ihr Einsatz aus Qualitätsgründen nicht zu empfehlen ist.

6.2.2 CUPS

Das Drucken unter Linux und anderen UNIX-Varianten gehörte in früheren Zeiten zu den komplizierteren Themen der Administration. Seit einigen Jahren steht mit dem Common UNIX Printing System (CUPS) eine sehr praktische Schnitt-

stelle zu vielen verschiedenen Druckern, Druckertreibern und Seitenbeschreibungssprachen zur Verfügung. Es handelt sich um ein GNU-basiertes Open-Source-Projekt, das von der Firma Easy Software Products entwickelt wird.

Systeme wie der BSD-`lpd` (Line Printer Daemon) oder ein ähnliches Paket aus System V wurden in den 70er-Jahren entworfen und dienen lediglich der zeilenweisen Ausgabe von Text. Für Grafik, PostScript-Druck und so weiter wurden in der Folgezeit zahlreiche proprietäre Lösungen eingeführt. CUPS fasst die Funktionen der klassischen UNIX-Druckumgebungen zusammen und fügt allgemeingültige Lösungen für den Grafikdruck hinzu.

Falls Ihnen die im Folgenden verwendeten Netzwerkbeuriffe nichts sagen, können Sie diese ab Kapitel 13, »Netzwerkgrundlagen«, oder im Glossar in Anhang A nachschlagen.



Abbildung 6.3 Die Drucker-Konfigurationsseite der webbasierten CUPS-Administrationsoberfläche

Die Schaltzentrale von CUPS ist der Scheduler, der die verschiedenen Druckjobs entgegennimmt und in die Warteschlangen (Queues) der verschiedenen Drucker stellt. Der Scheduler ist ein kleiner HTTP-Server und kann Druckjobs deshalb sowohl vom lokalen Rechner als auch aus dem Netzwerk entgegennehmen; er lauscht am TCP-Port 631. Die Druckjobs werden als HTTP-PUT-Anfragen in einem bestimmten Format übermittelt, das als Internet Printing Protocol (IPP) in RFC

2911 und anderen definiert ist. Außerdem dient der Scheduler als gewöhnlicher Webserver, was genutzt wird, um eine webbasierte CUPS-Konfiguration zu ermöglichen. Geben Sie in einen Browser die Adresse `http://localhost:6311` ein, um die CUPS-Startseite anzuzeigen. Abbildung 6.3 zeigt die Seite zur Konfiguration von Druckern (`http://localhost:631/printers`).

Das wohlbekannte, weitverbreitete HTTP-Protokoll sorgt dafür, dass es sehr leicht ist, CUPS-Clients zu schreiben. Aus diesem Grund breiten sie sich schnell in der UNIX-Welt aus – innerhalb weniger Jahre hat sich CUPS zu einer der führenden Drucklösungen entwickelt.

Wenn an Ihren Rechner ein Drucker angeschlossen ist, der sich über CUPS ansteuern lässt (das System ist mit Treibern für fast alle gängigen Druckermodelle ausgestattet), können Sie den CUPS-Server `cupsd` starten, um selbst darauf zu drucken oder anderen Benutzern im Netzwerk den Zugriff auf den Drucker zu erlauben. Auch für den Zugriff auf die CUPS-Drucker anderer Hosts müssen Sie den `cupsd` auf Ihrem eigenen Rechner einrichten.

Die Konfiguration des Daemons steht normalerweise in der Konfigurationsdatei `/etc/cups/cupsd.conf`. Sie wurde – bis auf die explizit druckerbezogenen Konfigurationseinstellungen – weitgehend der Konfigurationsdatei des Webserver Apache (`httpd.conf`, siehe Kapitel 15, »LAMP [Linux, Apache, MySQL und PHP]«) nachempfunden. Die wichtigsten Befehle sind `Allow` und `Deny`, mit denen Sie den jeweils angegebenen Hosts den Zugriff gewähren beziehungsweise verweigern können. `BrowseAllow` und `BrowseDeny` ermöglichen dagegen die Angabe von Hosts, deren Druckerinformationen Sie erhalten oder nicht erhalten möchten.

6.2.3 Drucker mit YaST konfigurieren

Auf einem SUSE-System werden Sie CUPS in aller Regel nicht manuell konfigurieren, sondern stattdessen das betreffende YaST-Modul verwendet. Wählen Sie dazu `DRUCKER` im Bereich `HARDWARE`. Nach dem Laden der Treiberdatenbank, das etwas dauern kann, wird der Dialog `DRUCKERKONFIGURATION` angezeigt. Falls bereits Drucker konfiguriert sind oder automatisch erkannt wurden, werden diese in der Liste angezeigt. Diese können Sie mithilfe des Buttons `BEARBEITEN` konfigurieren.

Falls Ihr Drucker nicht automatisch gefunden wurde, klicken Sie stattdessen auf `HINZUFÜGEN`. Wählen Sie zuerst zwischen den Optionen `DIREKT ANGESCHLOSSE-`

1 Anders als bei »normalen« Webadressen (ihre Standard-Portnummer 80 braucht nicht angegeben zu werden) müssen Sie den Vorspann »http://« mit eingeben, weil er für den Port 631 nicht Standard ist.

NER DRUCKER oder NETZWERKDRUCKER, je nachdem, ob der Drucker an Ihrem Rechner oder irgendwo im Netzwerk (direkt oder an einem anderen Computer) angeschlossen ist. Die nachfolgende Betrachtung bezieht sich zunächst auf lokale Drucker.

Wählen Sie für einen direkt angeschlossenen Drucker als Nächstes den Anschlusstyp; bei aktuellen Modellen wird es sich mehrheitlich um USB-DRUCKER handeln. Parallele und serielle Drucker für die traditionellen Schnittstellen sind recht selten geworden; dagegen nehmen die drahtlosen Lösungen IrDA (Infrarot) und Bluetooth (Kurzstrecken-Datenfunk) allmählich immer mehr zu.

Im nächsten Schritt wird der konkrete Anschluss ausgewählt, falls es mehrere von derselben Sorte gibt. Klicken Sie gegebenenfalls auf DRUCKERANSCHLUSS TESTEN, wenn Sie sich nicht sicher sind. Wenn es der richtige Anschluss ist, wird der Drucker eine kurze Textnachricht ausdrucken (die klassische Meldung »Hello World«).

Wenn Sie das nächste Mal auf WEITER drücken, wird der NAME DER WARTESCHLANGE gewählt. Eine *Druckwarteschlange* (»printing queue«) ist ein First-in-First-out-Speicher (FIFO), der die an den Drucker gesendeten Daten puffert und dann in der passenden Geschwindigkeit an den Drucker weiterleitet, damit der Computer sich währenddessen anderen Aufgaben widmen kann. Dieser Vorgang wird auch als *Spooling* bezeichnet.

Geben Sie unter NAME FÜR DEN DRUCKER idealerweise eine Bezeichnung ein, die dem Druckermodell entspricht (zum Beispiel »hp_laserjet« oder »kyocera«). Falls es sich um den einzigen Drucker weit und breit handelt, können Sie auch einfach den Vorgabenamen »printer« beibehalten. Als weitere Informationen, die beispielsweise durch den CUPS-Servecienst publiziert werden, können Sie eine DRUCKERBESCHREIBUNG (etwa »Farblaserdrucker« oder »Fotodrucker«) und den STANDORT (beispielsweise »Flur 3. Etage«) eintragen. Wenn die aktuelle Konfiguration nicht dem Drucken auf Ihrem eigenen Rechner dient, sondern als Netzwerkfreigabe des Druckers gedacht ist, kreuzen Sie zusätzlich LOKALES FILTERN DURCHFÜHREN an.

Wenn das Druckermodell gar nicht oder fehlerhaft erkannt wurde, erscheint als Nächstes der Dialog zur Auswahl des Druckermodells (siehe Abbildung 6.4). Wählen Sie in diesem Fall den HERSTELLER aus der Liste im linken Fensterbereich; anschließend können Sie rechts das konkrete Modell auswählen. Angesichts der gebotenen Vielfalt dürfte es kaum schwerfallen, zumindest einen Drucker zu finden, der kompatibel mit Ihrem Gerät ist.

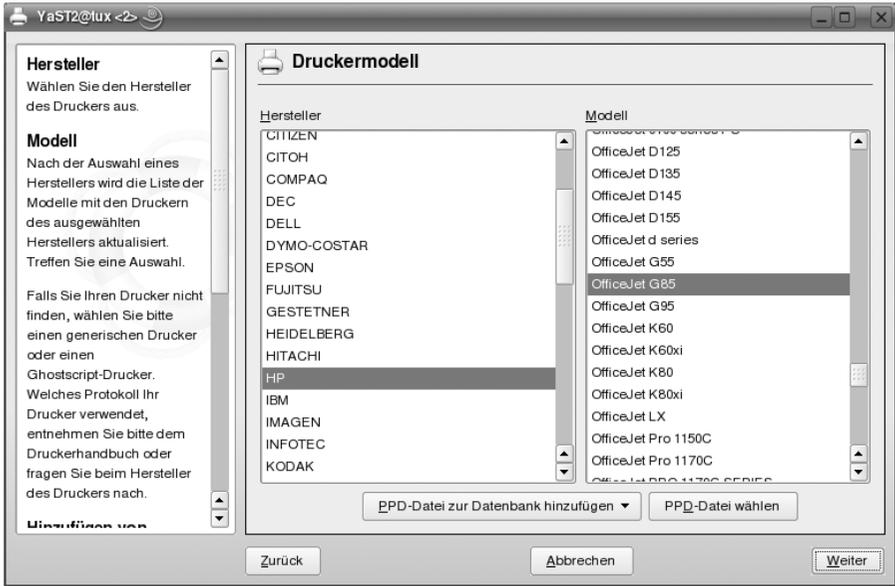


Abbildung 6.4 Auswahl eines Druckermodells in der YaST-Druckerkonfiguration

Falls Sie ausnahmsweise kein Glück haben, begeben Sie sich auf die Website des Herstellers, und suchen Sie nach einer *PPD*-Datei (PostScript Printer Description) für Ihren Drucker. Diese können Sie dann mittels *PPD-DATEI ZUR DATENBANK HINZUFÜGEN* in die Liste importieren und anschließend auswählen.

6.2.4 Drucken

Nachdem der Drucker eingerichtet ist, können Sie ihn nutzen. Anwendungen mit grafischer Oberfläche besitzen jeweils ihren eigenen Druckdialog, in dem Sie detaillierte Einstellungen vornehmen können. Es würde zu weit führen, hier auf mehrere davon einzugehen. In Abbildung 6.5 sehen Sie exemplarisch den KDE-Druckdialog, nachdem eine PDF-Datei im PDF-Viewer »Okular« geöffnet und die Auswahl *DOKUMENT • DRUCKEN* getroffen wurde. Die Einstellungen dürften im Wesentlichen selbsterklärend sein.

Aber auch auf der Konsole können Sie drucken. Zuständig ist das Kommando `lpr` (Line Printer). Geben Sie dazu einfach `lpr Dateiname` ein, zum Beispiel:

```
$ lpr test.txt
```

`lpr` besitzt eine Reihe traditioneller Kommandozeilenoptionen. Die meisten von ihnen sind allerdings heutzutage unwirksam, weil sie zum Drucksystem CUPS, das alle wichtigen Aspekte des Druckens kontrolliert, inkompatibel sind.

Alternativ können Sie `lpr` auch über eine Pipe verwenden, um die Ausgabe eines Systemprogramms auszudrucken. Das folgende Beispiel druckt den Inhalt des aktuellen Verzeichnisses:

```
$ ls -l |lpr
```



Abbildung 6.5 Der Druckdialog im KDE-Dolphin

6.3 Weitere Hardware mit YaST einrichten

Der YaST-Dialog **HARDWARE** enthält zahlreiche weitere Kategorien. In Tabelle 6.2 sehen Sie zunächst eine Übersicht. Anschließend wird exemplarisch die Konfiguration von Soundkarten und Scannern erläutert.

Kategorie	Bemerkungen
BLUETOOTH	Nahbereichsdatenfunk für den drahtlosen Anschluss diverser Peripheriegeräte (z. B. Bluetooth-Handys zur Datensynchronisation, aber beispielsweise auch Tastaturen, Mäuse)
DRUCKER	Konfiguration von Druckern (siehe oben)

Tabelle 6.2 Die verfügbaren Einstellungen in der YaST-Kategorie »Hardware«

Kategorie	Bemerkungen
FESTPLATTEN-CONTROLLER	Konfiguration der Kernel-Module für die Festplatten-Controller (ergibt manchmal Performance-Verbesserungen)
GRAFIKKARTE UND MONITOR	startet SaX2 (siehe voriges Kapitel).
HARDWARE-INFORMATIONEN	Liefert eine umfangreiche, aufklappbare Baumansicht über sämtliche installierte Hardware, die Sie mittels IN DATEI SPEICHERN exportieren können, etwa um auf dieser Basis technischen Support anzufordern.
IDE DMA-MODUS	steuert die Zugriffsmethode auf die EIDE-Schnittstellen für Festplatten
INFRAROT-GERÄT	Konfiguration der Infrarot-Schnittstelle IrDA
JOYSTICK	Einstellungen für Joysticks oder Gamepads, die bei PCs standardmäßig mit dem betreffenden Anschluss der Soundkarte verkabelt werden.
MAUS	öffnet das Modul MAUS von SaX2 (siehe voriges Kapitel).
SCANNER	Konfiguration von Scannern
SOUND	Einstellungen für Soundkarten (siehe unten)
TV-KARTE	Konfiguration einer TV-Karte
TASTATURBELEGUNG AUSWÄHLEN	startet das SaX2-Modul TASTATURBELEGUNG (siehe voriges Kapitel)

Tabelle 6.2 Die verfügbaren Einstellungen in der YaST-Kategorie »Hardware« (Forts.)

6.3.1 Soundkarten

Früher war es recht schwierig, Soundkarten unter Linux zu aktivieren – ihre Inbetriebnahme war einer der wichtigsten Gründe, den Kernel neu zu kompilieren. Dank des modernen Soundsystems ALSA werden die meisten Soundkarten heute bereits während der Installation erkannt. Sollte dies bei Ihnen nicht der Fall sein, erfahren Sie hier, was Sie tun können.

Allgemeines

So gut wie jeder Rechner ist heute in der Lage, Audiodaten zu verarbeiten. Dies beinhaltet das Abspielen von Sounddateien, die Digitalisierung aufgenommener Töne sowie einen per Hard- oder Software realisierten eingebauten Synthesizer. Immer mehr Mainboards enthalten integrierte Sound-Unterstützung in ihrem Chipsatz. Wenn Sie ein Mainboard besitzen, das nicht damit ausgestattet ist, oder erweiterte Soundfähigkeiten benötigen, müssen Sie sich eine Soundkarte anschaf-

fen. Sie werden in vielen verschiedenen Qualitäts- und Preisklassen angeboten. Achten Sie darauf, ob die von Ihnen benötigten Optionen von der gewählten Karte angeboten werden. Außerdem ist es zu empfehlen, Markenware zu kaufen, da die Treiberunterstützung bei No-Name-Karten oft mangelhaft ist.

Eine Soundkarte (oder die Onboard-Sound-Hardware) ist mit einer unterschiedlichen Anzahl von Anschlüssen für die Audio-Ein- und -Ausgabe ausgestattet. Beim Eingang ist zwischen Line-in und Microphone-in zu unterscheiden. Ein Line-in-Eingang dient der Eingabe vorverstärkter Töne, beispielsweise aus einer Stereoanlage oder dem Verstärker eines elektronischen Musikinstruments. Microphone-in leitet die über ein Mikrofon hereingekommenen Töne dagegen zunächst an den internen Verstärker weiter. Umgekehrt sieht es bei den Ausgängen aus: Der Line-out-Ausgang dient dem Anschluss an einen externen Verstärker oder Aktivboxen, während Speaker-out vorverstärkten Sound an einen Kopfhörer ausgibt.

Um Audio-CDs über die Ausgänge der Soundkarte verstärken und abspielen zu können, benötigen Sie ein spezielles Audiokabel vom CD-ROM-Laufwerk zur Soundkarte. Bei den meisten externen Soundkarten wird es mitgeliefert, während es bei vielen Komplettrechnern mit Onboard-Sound bereits vorinstalliert ist.

Viele aktuelle Mainboards oder Soundkarten unterstützen mehr als bloßen Stereo-Sound: Sie können ein sogenanntes 5.1-Boxenset anschließen, das mit einem Subwoofer, einem Mittellautsprecher und vier Satelliten ausgestattet ist. Dieser Surround-Sound ist beispielsweise von Vorteil, wenn Sie auf dem Computer mit entsprechender Software Video-DVDs ansehen möchten, da diese mit solchen Tonspuren ausgestattet sind.

Höherwertige Soundkarten sind nicht nur mit analogen Klinkenbuchsen versehen, sondern zusätzlich mit sogenannten SP-DIF-Ein- und -Ausgängen. Diese ermöglichen den Anschluss an moderne digitale Audiogeräte.

Eine weitere Komponente von Soundkarten ist ein eingebauter MIDI-Synthesizer. MIDI ist ein in den 80er-Jahren definierter Standard zur Steuerung elektronischer Musikinstrumente. Ältere Soundkarten verwendeten zur Erzeugung von MIDI-Klängen die sogenannte FM-Synthese, eine durch und durch künstliche Klangerzeugung. Das Ergebnis war wenig überzeugend; egal, ob Klassik, Rock oder Jazz abgespielt wurde, es klang alles wie frühe 80er-Jahre-Synthesizer. Bei neueren Soundkarten wird die sogenannte Wavetable-Synthese verwendet: Ein eingebauter Speicher enthält digitalisierte Klänge von Originalinstrumenten, die zum Abspielen der MIDI-Daten verwendet werden.

Außerdem sind die meisten externen Soundkarten und manche Onboard-Lösungen mit einem externen MIDI-Anschluss ausgestattet, der dem Austausch von

MIDI-Informationen mit einem externen Gerät dient. Auf diese Weise können Sie MIDI-Klänge über ein Keyboard einspielen, um sie in einem Sequencer-Programm weiterzuverarbeiten, oder aber einen hochwertigeren externen Synthesizer zum Abspielen von MIDI-Dateien verwenden. Derselbe Anschluss wird übrigens auch für Joysticks verwendet.

Soundkarten-Konfiguration

Wählen Sie SOUND im YaST-Bereich HARDWARE, um Ihre Soundkarte einzurichten. Klicken Sie auf HINZUFÜGEN, um Ihre Karte manuell auszuwählen, falls sie nicht bereits automatisch erkannt wurde. Im Auswahldialog können Sie wieder links den Hersteller und rechts das Modell wählen. Wenn Sie das Kontrollkästchen LISTE DER KERNEL-MODULE ANZEIGEN ankreuzen, wird die rechte Liste jeweils deutlich kürzer, da für mehrere Soundkarten eines Herstellers dasselbe Modul verwendet wird.

Wenn Sie Ihre Karte ausgewählt haben oder falls sie bereits automatisch erkannt wurde, können Sie auf BEARBEITEN klicken, um weitere Einstellungen vorzunehmen. Je nach verwendetem Modul stehen unterschiedliche Optionen zur Verfügung; bei vielen kann lediglich der Joystick-Anschluss konfiguriert werden.

Um auszuprobieren, ob die Konfiguration erfolgreich war, sollten Sie versuchen, eine Sounddatei abzuspielen. Zu empfehlen ist zunächst eine .wav-Datei – MP3 kann openSUSE nur mit (mitgelieferten) kommerziellen Tools abspielen.

6.3.2 Scanner

Ein Scanner dient dazu, eine Bildvorlage abzutasten und in digitale Daten umzurechnen. Zu diesem Zweck wird zeilenweise jeder einzelne Punkt der Vorlage mit einem Lichtstrahl beleuchtet. Die Stärke des reflektierten Lichts wird von einer Einheit namens CCD (Charged Coupled Device) gemessen und in ein entsprechendes Digitalsignal umgewandelt.

Im Gegensatz zu diesem sogenannten Auflicht-Scan wird der Durchlicht-Scan für transparente Vorlagen verwendet: Diese werden nicht von unten mit Licht angestrahlt, das reflektiert wird, sondern von oben durchleuchtet. Das hindurchscheinende Licht wird daraufhin wiederum von einer CCD-Einheit ausgewertet. Beachten Sie, dass nur teurere Modelle mit einer Durchlichteinheit versehen sind; für manche Scanner können Sie diese auch separat nachkaufen.

Wenn im Alltagsgebrauch von Scannern die Rede ist, geht es so gut wie immer um *Flachbettscanner*. Die Scan-Fläche ist eine flache Glasplatte, auf die die Vorlage gelegt wird, und besitzt eine Größe zwischen A4 und A3. Das Scannen erfolgt zeilenweise durch einen beweglichen Schlitten, auf den Lampe und Spiegel montiert sind.

Die mögliche Auflösung liegt üblicherweise zwischen 1.200 und 2.400 Pixeln pro Zoll. Die Farbtiefe beträgt bis zu 16 Bit für Graustufen (65.536 Abstufungen) beziehungsweise 48 Bit Farbe (je 16 Bit für die primären Lichtfarben Rot, Grün und Blau).

Neben der angegebenen optischen Auflösung kann ein Flachbettscanner höhere Auflösungen durch Interpolation (Berechnung von Farbdurchschnittswerten) bilden. Beachten Sie, dass dadurch nicht mehr Details in das Bild aufgenommen werden, es handelt sich um eine rein rechnerische Erhöhung der Auflösung.

Die meisten Scanner werden heutzutage per USB angeschlossen; früher waren vor allem SCSI-Scanner verbreitet, vereinzelt auch Parallelport-Scanner. Unter Linux war Treiberunterstützung für Scanner früher nicht selbstverständlich. Inzwischen steht mit SANE (Scanner Access Now Easy) allerdings eine leistungsfähige Schnittstelle zur Verfügung, die mit Scannern der meisten Hersteller zusammenarbeiten kann.

Wählen Sie **SCANNER** in der YaST-Kategorie **HARDWARE**, um Ihren Scanner zu konfigurieren. Wie immer kann es vorkommen, dass der Scanner automatisch erkannt wird. Andernfalls müssen Sie auf **HINZUFÜGEN** klicken, um ihn manuell einzurichten. In der Liste finden Sie zahlreiche Scannermodelle zur Auswahl (siehe Abbildung 6.6).

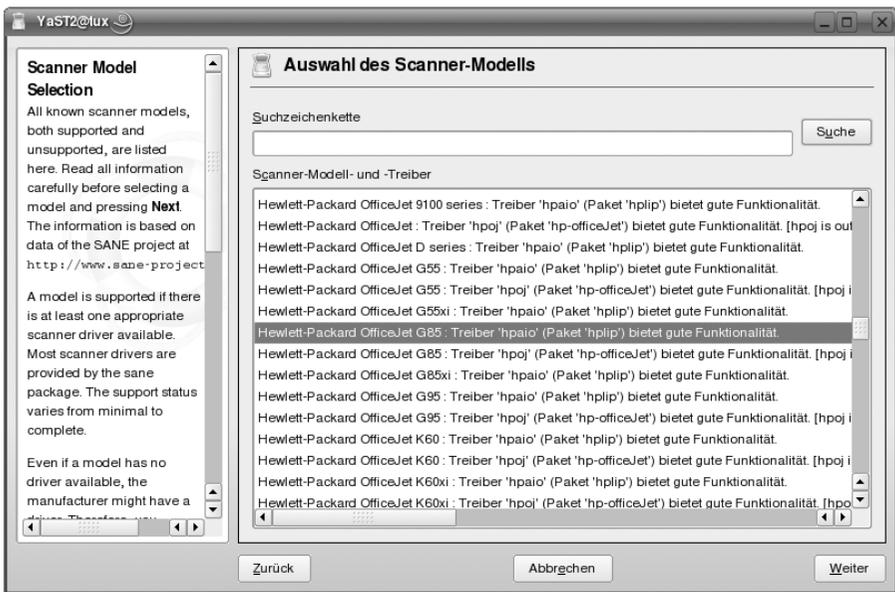


Abbildung 6.6 Manuelle Auswahl eines Scanners in YaST

Beachten Sie, dass für manche Scanner mehrere Treiber zur Verfügung stehen, die mitunter einen leicht unterschiedlichen Funktionsumfang oder andere Vor- und Nachteile bieten. Für mein Kombigerät HP OfficeJet G85 kommen beispielsweise grundsätzlich die beiden Treiber `hpaio` (basiert auf der Bibliothek `hplip`) oder `hpoj` (verwendet `ptal`) infrage. Da `hplip` und `ptal` einander ausschließen, `hplip` aber von CUPS verwendet wird, um mit diesem Gerät zu drucken, reduziert sich die Auswahl in der Praxis auf das Modul `hpaio`. Erfreulicherweise informiert Sie der Einrichtungsdialog detailliert über solche Probleme, installiert die benötigten Bibliotheken und aktiviert ihren automatischen Start beim Booten.

Wenn der Scanner eingerichtet ist, haben Sie die Wahl zwischen zwei Scan-Programmen: *xsane* ist eine grafische Oberfläche für die SANE-Schnittstelle und verhältnismäßig spartanisch ausgestattet, funktioniert dafür aber mit jedem SANE-fähigen Scanner. *Kooka* aus dem KDE-Projekt bietet mehr Optionen und enthält zusätzlich OCR-Funktionalität (Texterkennung in eingescannten Schriftstücken), ist aber etwas weniger kompatibel. Der Scan-Arbeitsablauf ist immer gleich:

1. Legen Sie die gewünschte Vorlage auf die Scanner-Oberfläche, und führen Sie zunächst einen Vorschau-Scan durch.
2. Markieren Sie im Vorschaubild den tatsächlich zu scannenden Bereich.
3. Stellen Sie die Parameter ein (Auflösung, Farbtiefe und so weiter).
4. Geben Sie den Befehl zum eigentlichen Scannen.
5. Öffnen Sie das fertig gescannte Bild in einem Bildbearbeitungsprogramm wie GIMP (siehe Kapitel 8, »Desktop-Software«), um weitere Feineinstellungen vorzunehmen.

6.4 Zusammenfassung

In diesem Kapitel haben Sie erfahren, wie Sie mit den wenigen Geräten umgehen können, die bei der Systeminstallation nicht automatisch erkannt werden. Zunächst wurde erläutert, wie Sie Kernelmodule manuell laden und wieder entladen können. Anschließend haben Sie erfahren, wie Sie einen eigenen, angepassten Kernel kompilieren können, was heutzutage allerdings nur noch in Ausnahmefällen erforderlich ist.

Im zweiten Abschnitt ging es um das wichtige Thema Druckerkonfiguration. Nachdem kurz das Funktionsprinzip der beiden wichtigsten Druckertypen – Tintenstrahl- und Laserdrucker – beschrieben wurde, haben Sie einiges über das Drucksystem CUPS erfahren. Die Druckerkonfiguration mit YaST, unter openSUSE selbstverständlich die bevorzugte Methode, wurde ebenfalls erläutert.

Abgerundet wurde der Abschnitt durch einige Anmerkungen zum Thema Drucken, damit Sie den fertig konfigurierten Drucker auch gleich praktisch einsetzen können.

Der dritte Abschnitt erläuterte nach einem Überblick über sämtliche Konfigurationsdialoge in der YaST-Kategorie `HARDWARE` exemplarisch die Einrichtung von Soundkarten und Scannern, die ebenfalls zu den wichtigsten Hardwarekomponenten gehören. Beachten Sie, dass einige weitere wichtige Geräte in anderen Kapiteln behandelt werden:

- ▶ Grafikkarte und Monitor sowie Tastatur und Maus im Rahmen der X-Konfiguration in Kapitel 5, »Grafische Oberflächen«.
- ▶ Laufwerke und ihre Datenträger in Kapitel 9, »Datenträger und Dateisysteme«.
- ▶ Netzwerkhardware wie Netzwerkkarten, WLAN und DSL in Kapitel 13, »Netzwerkgrundlagen«.

Es ist gefährlich, einen extrem fleißigen Büroangestellten einzustellen, weil die anderen ihm dann ständig zuschauen.
– Henry Ford

7 OpenOffice.org

Als Hauptargument gegen den Einsatz von Linux auf dem Desktop von Firmenrechnern galt früher, dass Büroarbeit ohne das Standardpaket Microsoft Office kaum möglich sei. Das hat sich geändert: Mit OpenOffice.org steht ein alternatives Office-Paket zur Verfügung, dessen Leistungsumfang den Microsoft-Programmen mindestens ebenbürtig ist. Einige Features – etwa der eingebaute PDF-Export – sind der kommerziellen Konkurrenz sogar überlegen. Zudem können Sie die allgegenwärtigen MS-Office-Dateiformate problemlos importieren und exportieren, sodass auch die elektronische Zusammenarbeit mit anderen, traditioneller orientierten Firmen möglich bleibt.

OpenOffice.org blickt auf eine recht lange Geschichte zurück. Sie begann im Jahr 1984, als der Hamburger *Marco Börries* von einem Schüleraustausch in den USA zurückkehrte und die Softwarefirma *Star Division* gründete. Das erste, bereits einigermaßen erfolgreiche Produkt war die Textverarbeitung *StarWriter*, die im Laufe der Jahre um andere Komponenten – etwa die Tabellenkalkulation *StarCalc* – zu dem vollwertigen Office-Paket *StarOffice* ergänzt wurde.

Im Jahr 1999 wurde *Star Division* von *Sun Microsystems* aufgekauft. Um Microsofts Marktmacht im Office-Bereich etwas entgegenzusetzen, gab Sun *StarOffice* bald darauf zum kostenlosen Download, aber noch nicht als Open-Source-Software frei. Doch bereits Mitte 2000 wurde auch der Quellcode – bereinigt um proprietäre Drittanbieterkomponenten – unter dem neuen Projektnamen *OpenOffice.org* freigegeben (ursprünglich war der Name einfach *OpenOffice*, aber dies scheiterte am Markenrecht). Das nach wie vor weitergeführte, kommerzielle *StarOffice* bezieht seine Codebasis aus der Weiterentwicklung von *OpenOffice.org* und wird von Sun um einige nicht freigegebene Komponenten ergänzt; im Vergleich zu *Microsoft Office* ist aber auch diese Variante sehr preisgünstig.

Ende 2005 erschien die Final Release der *OpenOffice.org*-Version 2.0. Die wichtigste Verbesserung war der Umstieg auf die neuen *OASIS OpenDocument*-Dateiformate. Diese XML-basierten Office-Dokumentformate wurden von einem

unabhängigen Gremium (Organization for the Advancement of Structured Information Standards) entwickelt und im Mai 2006 als ISO-Standard anerkannt. Neben OpenOffice.org nutzen auch StarOffice 8.0 und das KDE-Office-Paket KOffice 1.4 diese Formate. Abgesehen davon wurde OpenOffice.org 2.0 um einige neue Komponenten erweitert, vor allem um das Datenbankmodul Base.

openSUSE 11.2 enthält die nochmals in vielen Details verbesserte Version 3.

Sie hat insbesondere folgende Neuerungen zu bieten:

- ▶ Unterstützung für das OpenDocument-Format 1.2 mit diversen neuen Funktionen
- ▶ Importfilter für Microsoft Office 2007: In der aktuellen Version seines Office-Pakets hat Microsoft ebenfalls ein XML-basiertes Dokumentformat namens OpenXML eingeführt. Der neue Importfilter in OpenOffice.org 3 kann dieses Format erstmals lesen.
- ▶ Verbesserte Diagrammfunktionen
- ▶ Verbesserte Funktionen zum Zuschneiden von Bildern in Draw und Impress
- ▶ 1.024 Spalten pro Kalkulationstabelle (statt zuvor 256)
- ▶ Anzeige mehrerer Writer-Seiten während der Eingabe (bisher nur bei der Druckvorschau möglich)
- ▶ Verbesserte Unterstützung für XML und XSLT (siehe Kapitel 11, »Textbasierte Auszeichnungssprachen«)
- ▶ VBA-Unterstützung: OpenOffice.org verfügt über einen eigenen Basic-Dialekt zur Makroprogrammierung. Dies ist an sich recht nützlich, verhindert aber den Import von Makros aus Microsoft-Office-Dateien. Deshalb enthält auch OpenOffice.org nun eine (eingeschränkte) Unterstützung für Microsofts Visual Basic for Applications (VBA).

7.1 Textverarbeitung mit Writer

Der Writer ist die älteste und am weitesten entwickelte Komponente von StarOffice beziehungsweise OpenOffice.org. Es handelt sich um ein komfortabel ausgestattetes Textverarbeitungsprogramm. In diesem Abschnitt werden seine wichtigsten Funktionen vorgestellt; viele davon werden auch für die anderen Teilprogramme von OpenOffice.org genutzt.

Starten Sie Writer, indem Sie im K-MENÜ oder im GNOME-ANWENDUNGEN-Menü den Punkt BÜROPROGRAMME • TEXTVERARBEITUNG • OPENOFFICE.ORG WRITER wählen. In Abbildung 7.1 sehen Sie das Writer-Hauptfenster, wie es sich nach

dem Start mit einem einfachen Dokument präsentiert. Sie können zunächst recht intuitiv Text eingeben und mithilfe der Format-Symbolleiste oberhalb des Dokumentfensters formatieren. Bevor weiter unten genauer darauf eingegangen wird, sollen an dieser Stelle einige wichtige Grundgedanken der Typografie präsentiert werden. Diese spielen nicht nur bei der Arbeit mit Writer eine Rolle, sondern beispielsweise auch beim Erstellen von Webseiten. Die vollkommenste Verwirklichung typografischer Ideale auf Computersystemen stellt die Textsatz-Software LaTeX dar. Beide Themen werden in Kapitel 11, »Textbasierte Auszeichnungssprachen«, eingeführt.

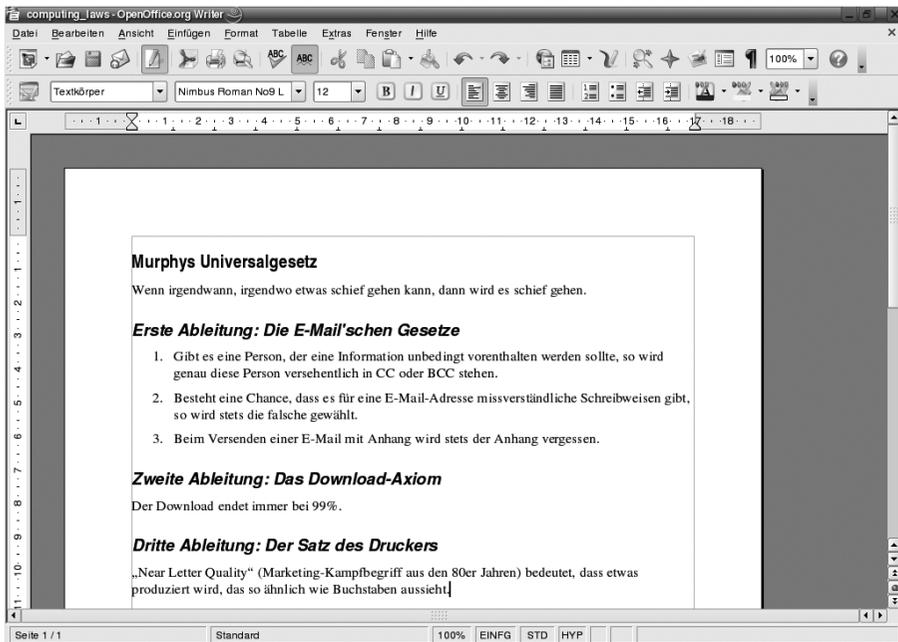


Abbildung 7.1 Die Grundansicht von OpenOffice.org Writer mit geöffnetem Dokument

7.1.1 Grundlagen der Typografie

Bereits vor fünf- bis sechstausend Jahren haben die Menschen aus vereinfachten, stilisierten Bildern die ersten Schriften entwickelt – Bilder- und Silbenschriften wie die noch heute gebräuchliche (allerdings stark abstrahierte) chinesische Schrift.

Erst die Phönizier entwickelten vor gut 3.000 Jahren eine Schrift, die nicht mehr Gegenstände abbildete, sondern Laute wiedergab – die 22 Konsonanten der phönizischen Sprache. Die Schrift der Phönizier ist der gemeinsame Vorfahr der hebräischen, arabischen und griechischen Schrift. Dies bemerkt man heute am

ehesten an den Namen und der Reihenfolge der Buchstaben; die ersten vier griechischen Buchstaben (Alpha, Beta, Gamma, Delta) ähneln beispielsweise den hebräischen Bezeichnungen (Aleph, Beth, Gimel, Daleth). Die Formen dagegen haben sich recht weit auseinanderentwickelt.

In der griechischen Schrift wurden von Anfang an gleichberechtigte Zeichen für die Vokale hinzugefügt. Sie wird noch heute für die griechische Sprache und zum Teil in der Wissenschaft verwendet. Aus ihr wurde auch die lateinische Schrift entwickelt, die ihre heutige Form im Prinzip seit der Antike beibehalten hat.

Genauer gesagt wurden damals nur die Großbuchstaben eingesetzt; diese in Stein gehauenen Capitalis Monumentalis können bis heute in Museen bewundert werden. Die Kleinbuchstaben entstanden erst viel später, als sich bis zum 9. Jahrhundert nach unserer Zeitrechnung diverse Kursiv- und Handschrift-Schnitte zur Karolinger Minuskel weiterentwickelten. Die heutige Art der Verwendung der beiden Alphabete, nämlich die Stellung der römischen Kapitale am Wort- beziehungsweise Satzanfang und der Minuskel im Wortinneren, entstand zur Zeit der Renaissance. Da diese Kombination eine optimale Lesbarkeit von Text ergibt, hat sie sich – mit gewissen Variationen – bis heute gehalten.

Typografische Begriffe und Maßsysteme

Die meisten Begriffe der Typografie entstammen der Tradition des Bleisatzes, bei dem die einzelnen Zeichen als spiegelverkehrtes Gepräge aus einem eckigen Metallstab, dem Kegel, herausragen. Diese einzelnen Lettern werden auf eine Platte montiert und bilden so eine Druckseite, die eingefärbt und unter einer Druckerpresse auf eine Seite gedrückt wird. Echter Bleisatz wird heutzutage nur noch sehr selten eingesetzt. Bücher, die so entstehen, sind teure, aufwendig gestaltete Werke mit meist sehr niedriger Auflage.

Die eigentliche Schriftgröße ist die *Kegelhöhe*, das heißt die Gesamthöhe der Bleilettern. Gemessen wird sie in *Punkt*, allerdings gibt es verschiedene Versionen dieser Maßeinheit, die im Laufe der Typografiegeschichte als Bruchteile jeweils gängiger Maßeinheiten definiert wurden. Klassisch ist der sogenannte *Didot-Punkt* (dd), 1784 von dem Franzosen *François Didot* definiert. 1 dd beträgt 0,376 mm (1973 abgerundet auf 0,375 mm). Heute wird – insbesondere im Zusammenhang mit der Arbeit am Computer – der *DTP-Punkt* (pt) verwendet. Er ist definiert als 1/72 Inch (1 Inch = 2,54 cm) und misst demnach knapp 0,353 mm. Leicht abweichend ist der traditionelle amerikanische *Pica-Point* (ebenfalls pt), der 1/96 von 35 cm und damit etwa 0,352 mm beträgt. Natürlich fallen diese geringen Unterschiede nicht am einzelnen Zeichen oder an einer Zeile auf, können bei einer ganzen Seite aber durchaus eine Differenz von mehreren Zeilen ergeben.

Zu allem Überfluss gibt es auch noch traditionelle Namen für die wichtigsten Schriftgrößen, die im Bleisatz verwendet wurden und unter gestandenen Schriftsetzern noch heute gelten – allerdings werden sie in der Regel nicht mehr auf den Didot-Punkt, sondern auf den DTP-Punkt angewendet. In Tabelle 7.1 werden einige Beispiele für diese Maße zur besseren Übersicht im direkten Vergleich aufgeführt.

DTP	Pica-Point	Didot (neu)	Didot (alt)	mm	Name
09	09,03	08,47	08,45	3,177	Bourgeois
10	10,03	09,41	09,39	3,53	Korpus
11	11,03	10,35	10,33	3,883	Rheinländer
12	12,03	11,3	11,27	4,236	Cicero
14	14,04	13,18	13,14	4,942	Mittel
09,97	10	09,39	09,36	3,52	–
11,97	12	11,26	11,23	4,224	1 Pica
10,62	10,65	10	09,97	3,75	–
12,75	12,78	12	11,97	4,5	–
10,65	10,68	10,03	10	3,76	–
12,78	12,81	12,03	12	4,512	–

Tabelle 7.1 Vergleich verschiedener typografischer Maßsysteme

Die optische Größe, in der die Buchstaben verschiedener Schriften bei Einstellung einer Schriftgröße erscheinen, variiert beträchtlich, weil je nach Schrift bestimmte Bestandteile der Buchstaben mehr oder weniger betont werden. Maßeinheiten für die einzelnen Größenbestandteile sind folgende:

- ▶ Die bereits genannte *Kegelhöhe* definiert den gesamten Raum, den Zeichen einer Schrift überhaupt einnehmen könnten, einschließlich Ober- und Unterlängen.
- ▶ Die *Versalhöhe* bezeichnet die Höhe der Großbuchstaben.
- ▶ Die *Mittellänge* oder *x-Höhe* ist die Höhe eines Kleinbuchstabens ohne Unterlänge, insbesondere des kleinen x, da es in keiner Schrift Elemente besitzt, die über die Mittellänge hinausragen. Den Fuß der Mittellänge bildet die sogenannte *Grundlinie*.

- ▶ Die *Oberlänge* ist der Raum von der Oberkante der Mittellänge bis zum höchsten Punkt von Kleinbuchstaben wie k oder d – in manchen Schriften ragen diese noch über die Versalhöhe hinaus.
- ▶ Die *Untерlänge* bezeichnet den Raum von der Grundlinie bis zur Unterkante von Kleinbuchstaben wie g oder y.
- ▶ Die *Dicke* ist die gesamte Breite, die das Druckbild eines Buchstabens einnimmt.

Weitere wichtige Maßangaben betreffen die Abstände zwischen den Buchstaben, Wörtern und Zeilen:

- ▶ Die *Laufweite* bezeichnet die Abstände der Buchstaben im Wort. Der standardmäßige Abstand wird als 0 bezeichnet; eine geringere oder höhere Laufweite wird in Prozent angegeben.
- ▶ Ausgleichungen der Laufweite bestimmter Buchstabenkombinationen werden als *Unterschneidung* (englisch *Kerning*) bezeichnet. Die Bezeichnung stammt daher, dass man früher an den Bleiblöcken ein Stück unter der eigentlichen Letter wegschneiden musste, um die Buchstaben näher aneinanderrücken zu können. Hochwertige Computerschriften werden mit sogenannten Kerning-Tabellen für alle denkbaren Verbindungen ausgeliefert.
- ▶ Um den Abstand zwischen Wörtern und anderen Elementen innerhalb der Zeile zu messen, wird das sogenannte *Geviert* zugrunde gelegt: ein Quadrat, bei dem die Breite gleich der Kegelhöhe der aktuellen Schriftgröße ist. In professionellen DTP-Programmen können Abstände bis auf ein Tausendstelgeviert genau eingestellt werden; bei Writer geht die Freiheit nicht so weit.
- ▶ Der eigentliche Zeilenabstand (der *numerische Zeilenabstand*) wird von Grundlinie zu Grundlinie gemessen.
- ▶ Der *optische Zeilenabstand* ist der Abstand von der Grundlinie einer Zeile bis zur Mittellänge der darunterliegenden Zeile.
- ▶ Als *Durchschuss* bezeichnet man den Leerraum zwischen der Untерlänge einer Zeile und der Oberlänge der nächsten. Satz ohne Durchschuss wird als *kompress* bezeichnet; der normale Zeilenabstand (auch *splendid* genannt) beträgt etwa 120% des Kompress-Zeilenabstands.

Die Unterscheidung dieser Bestandteile ist wichtig, um die besonderen Merkmale der verschiedenen Schriftfamilien und Schriften beschreiben zu können.

Buchstaben und Alphabete

Es ist wichtig zu verstehen, dass Druckschriften aufgrund ihrer Geschichte aus zwei separaten Alphabeten bestehen. Die *Großbuchstaben*, auch *Kapitale*, *Majus-*

keln oder *Versalien* genannt, sind durch einfache Grundformen gekennzeichnet – sie alle lassen sich auf Kreis, Dreieck und Quadrat zurückführen. Text, der nur in Versalien gesetzt ist, wirkt aufgrund seiner Monotonie ermüdend und ist schlecht lesbar. Die *Kleinbuchstaben*, die auch *Minuskeln* oder *Gemeine* heißen, weisen abwechslungsreichere, komplexere Formen auf. Erst die Mischung der beiden Alphabete im Schriftbild sorgt für optimale Lesbarkeit.

Eine weitere spezielle Schriftform bilden die *KAPITÄLCHEN*. Es handelt sich nicht einfach um kleiner gesetzte Versalien, sondern um ein eigenes Alphabet aus Zeichen, die etwas breiter sind als gleich große Großbuchstaben und deren Strichstärken an die Minuskeln der entsprechenden Schriftart angepasst werden. Kapitälchen werden vor allem für Einleitungszeilen in Buchkapiteln oder zur Auszeichnung von Namen und anderen speziellen Begriffen eingesetzt.

Auch der *kursive* Schriftschnitt ist in Wirklichkeit eine separat entwickelte Schrift. Die Unterschiede können Sie vor allem am kleinen *f* oder am *ß* bemerken: Diese Zeichen besitzen in der kursiven Form vieler Schriften eine Unterlänge. Wenn von einer Schrift keine kursive Variante vorhanden ist, stellen manche Textverarbeitungs- oder Layoutprogramme einfach die »normale« Schrift schräg. Diese Schrift wird als *oblique* bezeichnet und sollte nur in begründeten Ausnahmefällen verwendet werden.

Eine Gruppe von Sonderzeichen innerhalb der einzelnen Schriftschnitte bilden schließlich die *Ligaturen*: Es handelt sich um die Zusammenziehung zweier hintereinanderliegender Buchstaben. Einige historische Ligaturen sind inzwischen zu eigenständigen Zeichen geworden, beispielsweise das *ß* (ursprünglich eine Ligatur aus *S* und *Z*), das *&* (eine verschnörkelte Form des lateinischen Wortes *et*) oder Zeichen wie *Æ* und *æ*.

Einige andere Ligaturen sind weniger auffällig, waren aber dennoch schon in den Setzkästen der Bleisetzer als eigenständige Lettern vorhanden: Es handelt sich um die Kombinationen *ff*, *fi* und *ft*. Wichtig ist allerdings, dass diese Zeichen nur dann als Ligaturen geschrieben werden, wenn sie zur gleichen Silbe gehören. Beispielsweise wird *ff* in dem Wort »schaffen« als Ligatur geschrieben, in »Auffahrt« dagegen nicht. Gute Layoutprogramme besitzen entweder einen integrierten Wortkatalog zur Unterscheidung dieser Fälle oder eine Funktion zur manuellen Umstellung zwischen Ligatur und zwei separaten Buchstaben.

In allen fließtexttauglichen Druckschriften weisen die Buchstaben unterschiedliche Breiten auf – der Fachbegriff für eine solche Schrift heißt *Proportionalschrift*. Das Gegenstück bilden die nichtproportionalen Festbreiten- oder Schreibmaschinenschriften. Der manchmal verwendete Ausdruck »dicktengleiche Schrift« ist

nicht ganz korrekt, weil zwar jedes Zeichen denselben Platz auf dem Druckbogen einnimmt, die Dicken selbst aber durchaus verschieden sein können. Schreibmaschinenschrift sieht in den meisten Fällen unharmonisch aus und ist schlechter lesbar als Proportionalchrift. Dennoch wird sie gerade in letzter Zeit gern in der Werbung eingesetzt. In diesem Buch und in anderen EDV-Fachpublikationen wird Nichtproportionalchrift für Programmlistings eingesetzt, weil das korrekte Untereinanderschreiben der Zeichen bei ihnen eine Bedeutung hat und weil manchmal auch Whitespace von Bedeutung ist.

Die nächste Gruppe von Zeichen einer Schrift bilden die Ziffern. Es gibt zwei verschiedene Arten von Ziffern in Druckschriften:

- ▶ Die *Majuskelziffern* (z.B. 0123...) beginnen alle auf der Grundlinie und weisen dieselbe Höhe auf wie Großbuchstaben. Anders als die Buchstaben besitzen sie in jeder Schrift identische Breiten, weil sie in Rechnungen oder Tabellen oft exakt ziffernweise untereinander gesetzt werden.
- ▶ Die selteneren *Mediäval-* oder *Minuskelziffern* (z.B. 0123...) werden dagegen mit Ober- und Unterlängen versehen, weisen verschiedene Höhen auf und besitzen unterschiedliche Breiten. Sie sind besonders gut für den Satz von Zahlen im Fließtext geeignet, weil sie sich harmonischer in das Schriftbild einfügen. Mediävalziffern sind nicht für alle Schriften verfügbar, meist werden sie nur mit den sogenannten Expert-Zeichensätzen besonders hochwertiger und teurer Schriftarten geliefert.

Die verschiedenen Schriftfamilien

Da Drucksachen zu vielen verschiedenen Anlässen angefertigt werden müssen, ist es praktisch, dass es unzählige unterschiedliche Schriften gibt, die jeweils für einen anderen Zweck geeignet sind und völlig verschiedene Stimmungen oder Werte zum Ausdruck bringen können. Grundverschiedene Schriftstücke wie eine Boulevardzeitung, eine Werbeanzeige für einen Lifestyle-Artikel oder eine Todesanzeige können ihre volle Wirkung erst entfalten, wenn das Gesamtlayout durch die passende Schrift unterstrichen wird. Es gehört zum Handwerk eines Schriftsetzers, die geeignete Schrift für die jeweilige Gelegenheit auszuwählen und in den Rahmen eines passenden Layouts zu setzen.

Die Schriftfamilien unterscheiden sich aufgrund vieler formaler und ästhetischer Kriterien voneinander. In Deutschland existiert eine DIN-Norm, die alle verfügbaren Schriften in elf Gruppen unterteilt. Abbildung 7.2 zeigt ein Beispiel für jede der ersten zehn Gruppen.



Abbildung 7.2 Beispiele für die ersten zehn DIN-Schriftklassen

Ein Open-Source-Office-Paket auf einem freien Betriebssystem bietet leider nicht für jede Klasse eine hochwertige Schrift – diese sind kommerziell und unterliegen den Copyrights von Herstellern wie Adobe oder Monotype. Dennoch ist open-SUSE mit einer relativ beeindruckenden Schriftenvielfalt ausgestattet, sodass Sie in der Regel für jeden Anlass etwas Passendes finden sollten.

- ▶ *Gruppe I: Venezianische Renaissance-Antiqua*
Die Schriften dieser Gruppe sind das Vorbild für alle modernen Druckschriften. Sie wurden im 15. Jahrhundert entwickelt und sehr früh in Bleiletern für den Buchdruck gegossen. Gekennzeichnet ist diese Gruppe durch große Ober- und Unterlängen, ausgeprägte Serifen und geringe Unterschiede der Strichstärke; die Achsen der Buchstaben sind insgesamt leicht nach links geneigt. In den meisten Schriften der Gruppe I besitzt das kleine e einen schrägen Querstrich. Beispiele für Schriften dieser Gruppe sind Trajanus oder Guardi.
- ▶ *Gruppe II: Französische Renaissance-Antiqua*
Diese etwas später entwickelten Schriften fallen vor allem dadurch auf, dass die Oberlängen der Kleinbuchstaben meist etwas höher ragen als die Versalien. Anders als bei der venezianischen Renaissance-Antiqua sind die Serifen etwas abgerundet. Zu dieser Gruppe gehören Schriften wie Garamond, Palatino oder Bembo.
- ▶ *Gruppe III: Barock-Antiqua*
Bei diesen Schriften sind die Strichstärken unterschiedlicher als bei den Varianten der Renaissance-Antiqua. Die Serifen der Kleinbuchstaben sind oft unten senkrecht und oben schräg. Zur Gruppe III zählen beispielsweise die Schriften Baskerville oder Concorde.
- ▶ *Gruppe IV: Klassizistische Antiqua*
Diese Gruppe zeigt besonders starke Unterschiede in den Strichstärken, die Serifen sind fast waagrecht. Zu dieser Gruppe gehören etwa Walbaum oder Bodoni.
- ▶ *Gruppe V: Serifenbetonte Linear-Antiqua*
Diese Gruppe bezeichnet modernere Antiqua-Formen, die seit dem 19. Jahrhundert entwickelt wurden. Bei diesen Schriften sind alle Strichstärken so gut wie identisch; die Serifen sind gerade und nicht abgerundet. Da sie dieselbe Stärke aufweisen wie die Grundstriche, erscheinen sie besonders betont. Zu dieser umfangreichen Schriftgruppe zählen unter anderem die Schriften Rotation, Clarendon oder Serifa.
- ▶ *Gruppe VI: Serifenlose Linear-Antiqua*
Die Gruppe fasst alle serifenlosen Antiqua-Schriften zusammen. Serifenlose Schriften wurden aus verschiedenen Formen der klassischen Antiqua abgeleitet, sodass die Schriften dieser Gruppe recht unterschiedlich aussehen. Elegante und gut lesbare Schriften wie Frutiger, Syntax oder Meta gehören ebenso dazu wie die etwas langweilige Helvetica oder die besonders geradlinige Futura.
- ▶ *Gruppe VII: Antiqua-Varianten*
Diese Gruppe versammelt alle Antiqua-artigen Schriften, die in keine andere

Gruppe passen. Es handelt sich vornehmlich um Schmuck- und Headline-Schriften. Die meisten Schriften der Gruppe sind nicht für Fließtext geeignet. Es gibt unzählige Beispiele für diese Gruppe, bekannt sind etwa die Schriften Arnold Böcklin (die Lieblingsschrift aller Antiquitätenhändler) oder Stencil.

► *Gruppe VIII: Schreibschriften*

Diese Gruppe umfasst sämtliche Schreibschriften, die jemals in Bleiletttern gegossen oder zu Computerzeichensätzen gemacht wurden. Beispiele sind KünstlerScript oder Zapf Chancery.

► *Gruppe IX: Handschriftliche Antiqua*

Diese Gruppe versammelt Antiqua-Schriften, die nicht ganz Schreibschrift, aber auch keine richtigen Druckschriften sind; sie liegt etwa zwischen den Formen der Gruppen VII und VIII. Beispiele sind Comic Sans, Momberg oder Mistral.

► *Gruppe X: Gebrochene Schriften*

In dieser Gruppe sind sämtliche Varianten der Frakturschrift und verwandte Formen wie die Gotische und die Schwabacher versammelt. Während die Fraktur in der ersten Hälfte des 20. Jahrhunderts als »Paralleluniversum« der Schriftkultur existierte, ließ ihre Bedeutung nach dem 2. Weltkrieg stark nach. Heute werden fast nur noch die Titel mancher Tageszeitungen in solchen Schriften gesetzt. Einige Schriften der Gruppe X sind Fraktur, Old English Text oder San Marco.

► *Gruppe XI: Fremde Schriften*

Die letzte Gruppe umfasst sämtliche nicht lateinischen Schriftzeichen wie Chinesisch, Arabisch, Hebräisch, Kyrrillisch oder Griechisch. Die Gruppe bemüht sich nicht um eine Unterscheidung der verschiedenen Schriftsysteme oder Schriftarten, sondern steht einfach für »Sonstiges«.

Ein typischer Anfängerfehler beim Erstellen von Dokumenten ist die Verwendung zu vieler verschiedener Schriften oder von Schriften, die nicht zusammenpassen. Es gibt zwei gängige Möglichkeiten, ein Dokument typografisch vernünftig zu setzen:

► *Eine Schriftfamilie verwenden*

Wenn Sie sich für eine gut ausgebaute Schriftfamilie entscheiden, stehen Ihnen genügend verschiedene Schnitte für die verschiedenen Textauszeichnungen wie Überschriften, Fließtext, betonte Wörter oder Zitate zur Verfügung. Es besteht keine Notwendigkeit, mehrere Schriftarten zu verwenden. Zu einer voll ausgestatteten Schriftfamilie gehören mindestens drei verschiedenen fette Zeichensätze in normal und kursiv, darüber hinaus sollten auch echte Kapitalchen verfügbar sein.

► *Eine Schrift für Überschriften, eine weitere für Fließtext*

Diese Vorgehensweise kommt relativ häufig zum Einsatz. Besonders oft wird eine Serifenschrift für Fließtext und eine serifenlose für Überschriften verwendet. Wenn Sie diese Variante wählen, müssen Sie darauf achten, dass die gewählten Schriften sich genügend stark voneinander unterscheiden. Zwei Schriften aus derselben Gruppe oder mit sonstigen auffälligen Ähnlichkeiten werden im direkten Nebeneinander als unharmonisch empfunden.

Beachten Sie aber, dass dies lediglich Faustregeln sind. Selbstverständlich gibt es begründete Ausnahmen, in denen zum Beispiel drei verschiedene Schriften verwendet werden können. Im Grunde ist es mit der Typografie wie mit jedem Handwerk: Wenn Sie es neu erlernen, müssen Sie zunächst sämtliche wichtigen Regeln studieren und sich daran halten. Wenn Sie später mit sehr viel praktischer Übung genügend Erfahrung gesammelt haben, können Sie sich im Einzelfall Gedanken darüber machen, ob das Brechen einer bestimmten Regel sinnvoll sein könnte.

Kürzer gesagt: Das solide Handwerk hält sich an die Regeln, das echte Kunstwerk interpretiert sie dagegen freier oder definiert sie bei Bedarf neu. Das ist etwas ganz anderes als das Verhalten von Anfängern, die Regeln missachten, weil sie ihnen nicht bekannt oder nicht geläufig sind.

Mikrotypografie

An dieser Stelle werden einige wichtige Regeln zusammengefasst, die beim Setzen von Text zu beachten sind. Die Vorgaben für den Satz einzelner Zeichen werden als *Mikrotypografie* bezeichnet. Viele dieser Regeln sind DIN-genormt, andere sind dagegen einfach Konvention. In der vorliegenden Form gilt die Liste vor allem für deutschen Text, für andere lateinisch geschriebene Sprachen können leicht abgewandelte Regeln gelten.

- Die Satzzeichen *Punkt* und *Komma* stehen unmittelbar am vorigen Wort. Dahinter folgt ein normaler Wortabstand.
- Andere Satzzeichen wie *Doppelpunkt* oder *Fragezeichen* halten ein wenig Abstand zum vorigen Zeichen (allerdings viel weniger als einen ganzen Wortabstand); hinter diesen Satzzeichen folgt ein gewöhnlicher Wortabstand.
- Der *Trennstrich (Divis)* ist ein sehr kurzer Strich. Bei Wort- oder Silbentrennungen steht er eng am vorderen Wort. Als Bindestrich verbindet er beide Wörter ohne Abstand.

Beispiele: Trenn- und Bindestriche; Friedrich-Schiller-Straße

- Der *Gedankenstrich* ist in deutschen Texten ein Halbgeviert lang, sodass er auch als *Halbgeviertstrich* bezeichnet wird. Die Abstände zu den Wörtern

davor und dahinter sind etwas geringer als der normale Wortabstand. Derselbe Strich kann auch bei Zahlen als Ersatz für das Wort »bis« stehen und wird dann ohne Abstand geschrieben.

Beispiele: einerseits – andererseits; 1855–1930.

Unter Linux lässt sich der Gedankenstrich leider nicht ohne Weiteres über die Tastatur eingeben; in OpenOffice.org können Sie sich entweder mit der Option EINFÜGEN • SONDERZEICHEN behelfen oder unter EXTRAS • AUTOKORREKTUR ein eigenes Kürzel zur automatischen Ersetzung definieren. Die Standardeinstellung – dass ein doppeltes Minus (--) durch einen Halbgeviertstrich ersetzt werden soll – müssen Sie zumindest dann aufgeben, wenn Sie Computerliteratur mit ihren zahlreichen --Kommandozeilenoptionen schreiben.

- ▶ *Anführungszeichen* werden im Deutschen durch „Gänsefüßchen“ dargestellt (‹99› unten, ›66› oben). Manchmal – wie in diesem Buch – werden auch »französische Anführungszeichen« gesetzt, weil sie sich in gedruckte Texte harmonischer einfügen. Die Bezeichnung »französische Anführungszeichen« ist insofern verkehrt, als sie in deutschen Texten mit der Spitze nach innen stehen, in französischen dagegen andersherum: « Qu'est-ce que c'est? ».

Die "geraden Anführungszeichen", die in diesem Buch in Programmlistings eingesetzt werden, haben in normalen Texten übrigens nichts zu suchen.

- ▶ Bei *Zahlen* können je drei Stellen von rechts an durch einen Punkt voneinander getrennt werden. Dezimalstellen werden durch ein Komma abgetrennt.

Beispiel: 3.157.250 oder 2.123,99 €.

- ▶ *Telefonnummern* werden laut DIN-Norm von rechts an in Zweiergruppen geschrieben, der Abstand zwischen den Gruppen kann dabei reduziert werden. Dasselbe gilt für die Vorwahl. Die Vorwahl wird von der Telefonnummer durch einen Schrägstrich getrennt oder steht in Klammern.

Beispiele: 5 67 89 01; 0 40 / 2 34 56 78 oder 01 72 / 6 12 34 56.

- ▶ *Postleitzahlen* und *Jahreszahlen* dürfen nicht unterteilt werden.

Beispiele: 50668 Köln oder 2009.

Zeile und Absatz

Der gesamte Bereich, den der Text auf einer Druckseite einnimmt, wird als *Satzspiegel* bezeichnet. Wie der Satzspiegel auf der Seite platziert werden kann, wird im nächsten Unterabschnitt erläutert. An dieser Stelle geht es zunächst einmal um den Textfluss selbst, um das Setzen von Zeilen und Absätzen.

Jedes Textverarbeitungs- oder Layoutprogramm beherrscht heutzutage den automatischen Zeilenwechsel, der immer dann stattfindet, wenn ein Wort über den

rechten Rand des Satzspiegels hinausragen würde. Wahrscheinlich erscheint Ihnen dieser Umstand selbstverständlich, weil Sie es gewöhnt sind, mit solcher Software zu arbeiten.

Früher war das aber völlig anders. Bei der Schreibmaschine bestand die einzige Kontrolle für das Zeilenende aus der »Glocke«, die einige Zeichen vor dem Erreichen des rechten Randes ertönte, sodass man sich meist noch rechtzeitig für einen Zeilenumbruch oder eine Silbentrennung entscheiden konnte. Und in klassischen Texteditoren scrollte der Text üblicherweise einfach nach links weg, wenn man rechts zu viele Zeichen eingab.

In modernen Programmen gibt es dagegen drei verschiedene Arten, eine neue Zeile zu beginnen:

- ▶ Der bereits erwähnte automatische Wortumbruch findet statt, wenn der rechte Zeilenrand überschritten wird.
- ▶ Der Absatzwechsel, in Writer und den meisten anderen Textverarbeitungs- und Layoutprogrammen hervorgerufen durch die große -Taste im alphanumerischen Block der Tastatur, erzeugt einen neuen Absatz.
- ▶ Der einfache Zeilenwechsel innerhalb eines Absatzes (auch weiche Zeilenschaltung genannt) wird in Writer und anderer Software durch die Tastenkombination  +  erzeugt.

Typische Fehler bei Zeilen und Absätzen

Sie sollten sich vor der unter Anfängern weitverbreiteten Textverarbeitungsunsitte hüten, einen einfachen Zeilenwechsel durch einmaliges Drücken von  und einen neuen Absatz durch zweimaliges Drücken zu erzeugen. Der volle doppelte Zeilenabstand ist als Absatztrennung zu groß (Faustregel: höchstens das Anderthalbfache des normalen Zeilenabstands). Die unterschiedlichen Abstände, die für gewöhnliche Absätze und vor oder nach Überschriften benötigt werden, werden in Layoutprogrammen über die Definition von Absatzformaten eingestellt. Auch andere Absatzkennzeichnungen wie zum Beispiel Einzüge können dort definiert werden.

Für ein und denselben Absatz sollte übrigens auch dann eine gemeinsame Einstellung möglich sein, wenn ein erzwungener Zeilenwechsel darin vorkommt. Da  aber einen neuen Absatz erzeugt, können die beiden getrennten Absätze nicht mehr automatisch in einem formatiert werden.

Ein weiterer, sogar noch schlimmerer Fehler besteht darin, Fließtextzeilen durch manuelle Zeilenwechsel abzuschließen. In diesem Fall spielt es keine Rolle, ob Absatzwechsel oder weiche Zeilenschaltungen verwendet werden, es ist in beiden Fällen verkehrt. Das merken Sie aber möglicherweise erst viel später beim Hinzufügen oder Entfernen von Text, vielleicht auch erst dann, wenn Sie zufällig die Breite des Satzspiegels oder die Schriftgröße ändern. Plötzlich sitzen die erzwungenen Umbrüche nicht mehr am jeweiligen Zeilenende, sondern irgendwo mitten in den jeweiligen

Zeilen, weil zusätzlich automatische Zeilenwechsel hinzugekommen sind. Die einzigen Textarten, bei denen jede Zeile durch einen manuellen Zeilenwechsel abgeschlossen wird, sind Gedichte oder schmucklose Auflistungen.

Eine wichtige Eigenschaft von Absätzen ist die *Ausrichtung*. Insgesamt lassen sich vier verschiedene Arten der Absatzausrichtung unterscheiden:

- ▶ *Linksbündige Ausrichtung*
Das erste Zeichen jeder Zeile beginnt auf der linken Seite exakt an derselben Stelle, die Zeichen sind bündig am linken Rand des Satzspiegels ausgerichtet. Auf der rechten Seite gibt es einen wechselnden Zeilenfall (Flattersatz).
- ▶ *Rechtsbündige Ausrichtung*
Bei dieser Variante stehen die Zeichen am rechten Rand bündig auseinander, auf der linken Seite ist der Zeilenanfang dagegen variabel.
- ▶ *Zentrierte Ausrichtung*
Beim zentrierten Satz werden die Zeilen an einer mittleren Achse untereinander ausgerichtet. Von der Mittelachse aus gemessen besitzt also jede Zeile dieselbe Länge nach links wie nach rechts.
- ▶ *Blocksatz*
Alle Zeilen eines Absatzes besitzen dieselbe Länge. Dies wird durch einen Ausgleich der Zeichen- und Wortabstände ermöglicht. Die Silbentrennung sorgt dafür, dass es weniger Zeilen mit allzu großen Lücken gibt.

Die beiden gängigsten Arten der Absatzausrichtung sind linksbündiger Satz und Blocksatz. Dabei sollte die linksbündige Ausrichtung vor allem für Briefe und andere kürzere Dokumente verwendet werden. Auch bei einem besonders kurzen Satzspiegel, zum Beispiel bei mehrspaltigem Satz, kann linksbündiger Satz angebracht sein, weil sich sehr kurze Zeilen trotz effizienter Silbentrennung nicht gut aufteilen lassen.

Blocksatz ist die bevorzugte Satzvariante für Bücher, Zeitschriften und andere Massendrucksachen. In Tageszeitungen wird trotz der recht kurzen Spalten ebenfalls Blocksatz eingesetzt. Dies ermöglicht relativ eng nebeneinanderstehende Spalten, weil die Spaltengrenzen auf beiden Seiten gut erkennbar sind.

Damit Blocksatz vernünftig aussieht und keine allzu großen Lücken aufweist, ist bei größeren Textmengen eine gut funktionierende automatische Silbentrennung erforderlich. Inzwischen verfügen die verschiedenen Programme über Trennalgorithmien, die nur sehr wenige Fehler machen. Dennoch gehört es zu Ihrer Sorgfaltspflicht, selbst bei umfangreichen Texten alle durchgeführten Silbentrennungen auf Korrektheit zu überprüfen, denn immer wieder gibt es Fremdwörter,

besonders seltene Wörter oder einzelne Sätze in Fremdsprachen, für die in einem Layoutprogramm keine Trennregeln vorhanden sind.

Sie sollten auch darauf achten, dass nicht zu viele aufeinanderfolgende Zeilen durch Trennstriche abgeschlossen werden. Die meisten Programme können Sie so einstellen, dass sie automatisch einen vernünftigen Wert für maximal erlaubte aufeinanderfolgende Trennungen einhalten; eine sehr häufige Wahl sind drei Zeilen.

Im Übrigen ist es eine gute Idee, in längere Wörter von vornherein optionale Trennstriche einzufügen. Dieses spezielle Zeichen ist nur dann sichtbar, wenn es gerade tatsächlich als Trennstrich verwendet werden muss. In Writer wird der optionale Trennstrich durch die Tastenkombination + erzeugt.

Beim linksbündigen Flattersatz müssen Sie darauf achten, dass der Zeilenfall (die unterschiedlichen Zeilenlängen) sich an bestimmte Vorgaben hält:

- ▶ Mehr als zwei aufeinanderfolgende Zeilen sollten nach Möglichkeit nicht gleich lang sein.
- ▶ Eine über mehrere Zeilen gleichmäßig anwachsende oder abfallende Zeilenlänge (Treppeneffekt) sollte vermieden werden. Überhaupt sollten Sie alle rhythmischen Abfolgen vermeiden, die beabsichtigt aussehen könnten – solche Zeilenfolgen sind der Lyrik beziehungsweise der konkreten Poesie vorbehalten.
- ▶ Viel zu kurze Zeilen, die dadurch entstehen, dass ein sehr langes Wort nicht mehr in die Zeile passt, sollten ebenfalls nicht verwendet werden. Hier schafft wie beim Blocksatz die Silbentrennung Abhilfe.

Rechtsbündiger und zentrierter Satz sollten nur in Ausnahmefällen eingesetzt werden. Rechtsbündig werden in normalen Dokumenten nur ganz wenige, spezielle Informationen gesetzt: Ort und Datum zu Beginn eines Briefes oder Orientierungshilfen in Form der aktuellen Absatzüberschrift am Kopf oder am Fuß einer rechten Druckseite. Ansonsten ist rechtsbündiger Text eigentlich nur für Schriften geeignet, die von rechts nach links geschrieben werden, zum Beispiel Hebräisch oder Arabisch.

Zentrierter Satz kann vereinzelt auf speziellen Schmuckseiten mit wenig Text verwendet werden, zum Beispiel für ein Motto oder eine Widmung in einem Buch. Diese Satzform wurde in den Anfangstagen des World Wide Web inflationär auf Webseiten eingesetzt, gehört aber auch dort nur in Ausnahmefällen hin. In manchen Zeitschriften oder Büchern werden Absatzüberschriften zentriert gesetzt – zu manchen, in der Regel eher traditionellen Layouts passt das gut.

Im Zusammenhang mit Absätzen müssen Sie zusätzlich auf ein ganz anderes Problem achten: Es sieht sehr unästhetisch aus, wenn die erste Zeile eines neuen Absatzes am Fuß einer Spalte oder Seite steht (im Setzerjargon ein »Schusterjunge«) oder wenn die letzte Zeile des Absatzes allein in der nächsten Spalte oder auf der nächsten Seite landet (»Hurenkind« genannt).

Dieses Problem ist so verbreitet, dass die meisten Programme einen speziellen Schutz davor anbieten: Sie können bei der Definition von Absatzformaten dafür sorgen, dass eine bestimmte Mindestzahl von Zeilen eines Absatzes automatisch zusammengehalten wird. Der Absatz wandert also im Zweifelsfall eher komplett in die nächste Einheit, als dass weniger Zeilen als die angegebene Mindestzahl allein bleiben.

Eine ähnliche Schwierigkeit bereiten übrigens Überschriften. Natürlich gehört eine Überschrift über den Absatz, den sie betitelt, und nicht für sich allein auf die vorige Seite. In Writer und anderen guten Layout- oder Textverarbeitungsprogrammen können Sie deshalb auch einstellen, dass ein Absatz mit dem darauffolgenden oder mit dem vorigen zusammenbleiben soll.

Seitengestaltung

Neben der bisher behandelten korrekten Verwendung von Zeichen und Abständen sowie dem richtigen Umgang mit Schrift gibt es eine Reihe von Grundregeln für die *Gestaltung* von Dokumenten und anderen Drucksachen. Diese Regeln betreffen unter anderem die Flächenaufteilung, die Abgrenzung verschiedener Bereiche voneinander oder den Umgang mit Farben, Rahmen und Linien. Selbstverständlich gibt es auch bei Beachtung der wichtigsten Gestaltungsprinzipien viele unterschiedliche kreative Möglichkeiten.

Beim Platzieren von Text, Bildern und Schmuckelementen auf einer Druckseite ist es sehr wichtig, sie entsprechend der gewünschten Aussage zu verteilen. Spannung oder Ruhe, Dynamik oder Statik werden durch unterschiedliche Flächenaufteilungen erzielt. Von Bedeutung sind dabei nicht nur die Bereiche, die mit Inhalten gefüllt werden, sondern auch diejenigen, die aus Kontrastgründen frei bleiben.

Die Aufteilung der Fläche beginnt bereits bei der Auswahl des Papierformats: Verschiedene Blattgrößen und -proportionen und die Entscheidung zwischen Hoch- und Querformat transportieren schon für sich eine bestimmte Aussage oder Stimmung und bilden vor allem den Rahmen für die spätere Platzierung der Inhaltselemente.

Für Briefe und andere Bürodokumente gibt es eigentlich nur eine Wahl: das Standardformat DIN A4. Seine Größe ist 210 × 297 mm; das Seitenverhältnis aller

Formate der DIN-Formatreihe A beträgt $1:\sqrt{2}$ (etwa 1:1,4143), was als recht harmonisch und ausgewogen empfunden wird. Diese Dokumentformate basieren zwar auf der deutschen Norm DIN 476, sind aber inzwischen international gebräuchlich. Die Formatreihe A beginnt mit A0, das 841×1.189 mm groß ist. Jedes weitere Format ist genau ein halber Bogen des vorherigen; bei A8 beträgt die Größe nur noch 52×74 mm. Weitere DIN-Formatreihen sind übrigens B für Umschläge und C für Verpackungen. Tabelle 7.2 zeigt eine Übersicht über die einzelnen Formate der DIN-Reihen.

Format	A (Papierbögen)	B (Umschläge)	C (Verpackungen)
0	841×1.189	1.000×1.414	917×1.297
1	594×0.841	707×1.000	648×0.917
2	420×0.594	500×0.707	458×0.648
3	297×0.420	353×0.500	324×0.458
4	210×0.297	250×0.353	229×0.324
5	148×0.210	176×0.250	162×0.229
6	105×0.148	125×0.176	114×0.162
7	74×0.105	88×0.125	81×0.114
8	52×00.74	62×00.88	57×00.81

Tabelle 7.2 Übersicht über die wichtigsten DIN-Papierformate in mm

Andere Papierformate, die nicht in das DIN-Schema passen, sind zum Beispiel das traditionelle amerikanische Briefformat US-Letter (etwas kürzer als A4) oder das international gebräuchliche Scheckkartenformat 86×54 mm.

Auch für Tageszeitungen gibt es einige festgelegte Formate:

- ▶ Berliner Format: 315×470 mm
- ▶ Rheinisches Format: 360×530 mm
- ▶ Nordisches Format: 400×570 mm

Wenn Sie mit einer Textverarbeitung wie Writer und einem gewöhnlichen Heim- oder Bürodrucker arbeiten, werden Sie in den meisten Fällen wahrscheinlich A4-Seiten erstellen und ausdrucken; andere Formate bringen technische Schwierigkeiten mit sich: Größere Formate müssten auf mehrere A4-Seiten aufgeteilt und dann zusammengeklebt werden. Kleinere lassen sich dagegen nur schwer korrekt in einen Standarddrucker einlegen; oft ist es einfacher, die Inhalte wiederum auf A4-Papier auszudrucken und anschließend zu schneiden.

Als sehr harmonisch werden nach der klassischen Gestaltungslehre alle Formate empfunden, die auf dem *Goldenen Schnitt* basieren, dem bereits seit der Antike bekannten 5:8-Format (umgerechnet 1:1,618). Dieses Verhältnis wird nicht besonders häufig für Seitenformate eingesetzt, aber dafür umso öfter für die Proportionen verschiedener Elemente auf der Druckseite.

Gleiche Elemente, die auf den verschiedenen Bereichen einer Druckseite angeordnet werden, werden von Betrachtern nicht an jeder Stelle als gleich groß empfunden. Das rechte untere Viertel wird als besonders »schwer« oder betont wahrgenommen, das linke obere dagegen als »leicht«. Der Grund für diese spezielle Wahrnehmungsverschiebung ist die seit frühester Kindheit eingeübte Gewohnheit, Zeilen von oben nach unten und Buchstaben von links nach rechts zu lesen; der Text »fällt« förmlich nach rechts unten.

Der Kontrast wird deutlich, wenn Sie sich gedruckte Dokumente aus anderen Schreibkulturen ansehen: Eine arabische oder japanische Zeitschrift wirkt nicht nur wegen der völlig unterschiedlichen Schriftzeichen »anders«, sondern auch wegen der Schreibrichtung von rechts nach links bzw. oben nach unten, die zu einer umgekehrten Flächenaufteilung führt. Für westliche Leser wirkt eine Anordnung von Elementen auf der Diagonalen von links oben nach rechts unten harmonisch. Die umgekehrte Diagonale wird eher als spannungsgeladen wahrgenommen. Für Leute, die das Lesen von rechts nach links gewöhnt sind, ist es dagegen genau umgekehrt.

Aus demselben Grund wird bei einer Seite zwischen der tatsächlichen und der *optischen Mitte* unterschieden. Letztere liegt etwa 10% höher als die tatsächliche Mitte. Auch der Satzspiegel wird meistens etwas weiter oben auf der Seite platziert, der obere Rand ist niedriger als der untere.

Wie die einzelnen Bestandteile auf der Seite verteilt werden, hängt letztlich von ihrer absoluten und relativen Größe und vom Verwendungszweck ab. Die folgende Aufzählung enthält einige grundlegende Angaben für wichtige Kategorien von Printmedien:

- ▶ Bei *Romanseiten* ist es verhältnismäßig einfach – es müssen lediglich der Satzspiegel und eine Seitenzahl platziert werden. Aber bereits dafür gibt es viele verschiedene Möglichkeiten: Die Seitenzahl kann oben oder unten stehen, jeweils in der Seitenmitte oder bündig am Außenrand des Satzspiegels und so weiter.
- ▶ *Fachbücher* enthalten neben dem Satzspiegel auch Abbildungen, Tabellen und (im speziellen Fall eines EDV-Buchs wie diesem) Programmlistings. Strukturhilfen wie hierarchisch gegliederte Überschriften und Marginalien sorgen für mehr Übersicht. Die Aufteilung der Elemente kann wie im vorliegenden Band

ziemlich geradlinig erfolgen. Für manche Fachgebiete bietet sich dagegen eine aufgelockertere Verteilung an – beispielsweise für die Bände der Reihe Galileo Design, die aufgrund ihrer Thematik schon fast Bildbände sind.

- ▶ *Zeitschriften* sind je nach Themenbereich und Zielgruppe sehr unterschiedlich ausgestattet. Beispielsweise ist der Satz von Nachrichtenmagazinen und Fachzeitschriften relativ schlicht und übersichtlich, während es in Unterhaltungszeitschriften oft ziemlich turbulent zugeht – der Satzspiegel wird häufig durch außer der Reihe angeordnete Elemente aufgelockert.
- ▶ *Tageszeitungen* besitzen ein sehr klares und übersichtliches Satzbild, das nur wenige Überraschungen birgt. In der Regel sind sie mit sechs bis acht Spalten ausgestattet; Überschriften überspannen die gesamte Spaltenzahl eines Artikels, Abbildungen oft mehrere Spalten. Bestimmte Sonderartikel werden durch Rahmen oder farbliche Hinterlegung hervorgehoben. Diese Grundregeln gelten sowohl für seriöse Tageszeitungen als auch für Boulevardblätter; Letztere sind lediglich erheblich bildlastiger.
- ▶ *Kataloge, Broschüren und Verkaufsprospekte* liegen etwa in der Mitte zwischen Zeitschrift und Werbeanzeige. Es sollte darauf geachtet werden, den Betrachter nicht durch zu viel Text zu erschlagen; eine aufgelockerte und abwechslungsreiche Anordnung von Text und Bildern ist wichtig. Dies wird allerdings bei schnell produzierten Massenwerbezetteln als Aufforderung missverstanden, ein völliges Durcheinander von Bildern, schlecht zuzuordnenden Beschriftungen und oft unpassenden grafischen Schmuckelementen zu produzieren.
- ▶ *Werbeanzeigen* müssen sich von allen Print-Layouts die meiste Mühe geben: Alle anderen Drucksachen werden zunächst mehr oder weniger freiwillig wegen ihres Inhalts gelesen, die Werbung möchte dagegen auch dann auf sich aufmerksam machen, wenn der Inhalt Sie eigentlich nicht interessiert. Deshalb ist der Anspruch bei der Auswahl passender Bilder oder Schriften und bei der Aufteilung der verschiedenen Bestandteile auch viel größer als bei den restlichen Printmedien. Hinzu kommt, dass auf einer relativ großen Fläche relativ wenig Inhalt harmonisch und geschmackvoll verteilt werden muss.

Bei der Anordnung von Bildern und Schmuckelementen, die nicht die gesamte Spalten- oder Satzspiegelbreite einnehmen, ist die Frage wichtig, welche Art des *Umfließens* gewählt werden soll. Dieser Begriff bezeichnet das Verhalten von Text, der um ein Bild herum platziert wird. Der Text kann entweder das Begrenzungsrechteck des Bildes oder dessen Kontur umfließen, und das einseitig oder beidseitig. Writer akzeptiert in diesem Zusammenhang übrigens sowohl weiße als auch transparente Bildbereiche.

Sie können das Umfließen in Writer folgendermaßen einstellen:

1. Positionieren Sie den Cursor an der Stelle, an der das Bild positioniert werden soll.
2. Wählen Sie EINFÜGEN • BILD • AUS DATEI, und suchen Sie sich das gewünschte Bild aus.
3. Klicken Sie das eingefügte Bild an, und wählen Sie FORMAT • BILD, um die Eigenschaften der Bildeinbettung zu steuern.
4. Das Umfließen wird auf der Registerkarte UMLAUF eingestellt. Unter VORGABEN können Sie sich für verschiedene Typen entscheiden: KEIN platziert das Bild allein in seinen Zeilen; VOR, HINTER und PARALLEL sorgen dafür, dass es links, rechts beziehungsweise beidseitig von Text umflossen wird; DURCHLAUF schließlich stellt das Bild in den Hintergrund, sodass der Text einfach darüber hinwegfließt (dies geht nur mit sehr behutsam eingefärbten Bildern oder aber mit speziellen Textfarben, sonst können Sie – erst recht im Ausdruck mit einem normalen Bürodrucker – nichts lesen).

Eine besondere Variante ist das Einfügen eines Bildes in den Textfluss eines Absatzes, was beispielsweise nützlich ist, um Bedienelemente oder komplexe mathematische Formeln in den Text zu integrieren. Wechseln Sie dazu auf die Registerkarte TYP, und wählen Sie unter VERANKERUNG die Option ALS ZEICHEN. In diesem Fall können Sie ganz unten auf derselben Registerkarte unter VERTIKAL einstellen, auf welcher Höhe der aktuellen Zeile das Bild eingepasst werden soll.

Eine weitere interessante Variante für die Platzierung von Bildern oder Grafiken ist der *Anschnitt*: Das Bild beginnt ohne Beachtung des Seitenrandes unmittelbar an der Kante einer Seite, ein Teil des Inhaltes wird dabei weggelassen. Diese Art der Bilddarstellung sorgt für eine gewisse Dynamik und Auflockerung des Layouts, ist aber nicht für alle Arten von Bildern geeignet: Ein Personenporträt oder die explizite Abbildung eines Objekts, von dem ein Artikel handelt, sollten in der Regel nicht im Anschnitt stehen. Eine gute Wahl ist der Anschnitt dagegen etwa für weitläufige Panoramen, die Darstellung dynamischer Vorgänge oder für manche grafische Schmuckelemente wie Pfeile und Linien.

Neben den bereits beschriebenen Texten und Bildern kann eine weitere Art von Objekten in einem Print-Layout platziert werden. Rahmen dienen der Abgrenzung von Bereichen mit besonderer Bedeutung. Eine vergleichbare Art der Hervorhebung ist übrigens die farbliche Hinterlegung, die an dieser Stelle ebenfalls behandelt wird. Linien können vielfältige Beziehungen zwischen Objekten herstellen, den Raum aufteilen, Spannung, Dynamik oder Ordnung schaffen. Pfeile sind eine Steigerung der Linie, weil sie in eine bestimmte Richtung weisen.

Schließlich gibt es noch zahlreiche Arten von Schmuckelementen wie Punkte, Sterne oder Ornamente. Auch Initialen, also stark betonte Anfangsbuchstaben von Absätzen, gehören dazu.

Bei *Rahmen* ist besonders darauf zu achten, dass die Form des gewählten Rahmens zum Inhalt passt. Allzu leicht erreichen Sie einen Todesanzeigen- oder Ausschneidecoupon-Effekt. In den meisten Fällen ist ein dünner, gerader, einfacher Rahmen völlig ausreichend. Spezielle Schmuckrahmen passen nur für besondere Dokumente wie zum Beispiel Urkunden. In einem durchschnittlichen Layout sollten Sie ohnehin eher einen farbig hinterlegten Kasten bevorzugen. Bei diesem müssen Sie allerdings wiederum auf genügend Kontrast zur Schriftfarbe achten, damit die Lesbarkeit nicht beeinträchtigt wird.

Linien werden oft regulär im Layout eingesetzt, um den oberen oder unteren Rand des Satzspiegels von Kopf- oder Fußzeilen abzutrennen oder um Zwischenüberschriften nach oben und unten abzugrenzen. Darüber hinaus können sie auf vielfältige Weise außer der Reihe benutzt werden, um auf spezielle Inhalte hinzuweisen oder um besondere Elemente zu verbinden oder zu trennen. Eine noch stärkere Wirkung erzielen in diesem Zusammenhang Pfeile, mit denen Sie allerdings sparsam umgehen sollten.

Die Bedeutung von *Ornamenten* ist im modernen Layout erheblich zurückgegangen. Funktionale Schmuckelemente wie Punkte oder besondere Aufzählungszeichen sind dagegen recht häufig anzutreffen. Beachten Sie, dass solche besonderen Elemente möglicherweise zu viel Aufmerksamkeit auf sich lenken; sie sollten also nur dort eingesetzt werden, wo etwas speziell betont wird. Ähnliches gilt für Initialen: Sie passen gut in kunstvoll aufgemachte Bücher, beispielsweise bestimmte Romane oder Bildbände, oder in das Satzbild hochwertiger, seriöser Zeitschriften. In schlichten, schmucklosen, funktionalen Layouts sind sie dagegen fehl am Platz.

7.1.2 Grundlegende Textformatierung

Die grundlegendste Aufgabe eines Textverarbeitungsprogramms besteht darin, Zeichen und Absätze zu formatieren. In diesem Unterabschnitt werden die grundlegenden Werkzeuge und Dialoge vorgestellt.

Die Format-Symboleiste

Für die Formatierung von Texten ist zunächst einmal die Symbolleiste **FORMAT** zuständig, die normalerweise direkt über dem Dokumentfenster angezeigt wird. Falls sie gerade ausgeblendet sein sollte, können Sie sie mittels **ANSICHT • SYMBOLLEISTEN • FORMAT** anzeigen. In Abbildung 7.3 sehen Sie die Symbolleiste mit einer

Beschriftung gemäß den Tooltips, die beim Darüberfahren mit der Maus angezeigt werden.

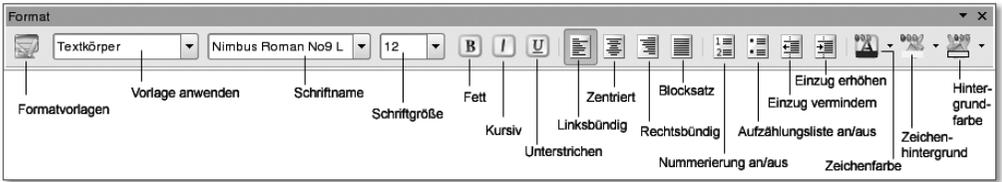


Abbildung 7.3 Die »Writer«-Symbolleiste »Format« und ihre Bedienelemente

Mithilfe der Schaltfläche ganz links können Sie das im nächsten Unterabschnitt besprochene Bedienfeld **FORMATVORLAGEN** öffnen; das Pulldown-Menü daneben ermöglicht es Ihnen, einige besonders häufig genutzte Vorlagen auszuwählen.

Die nachfolgenden Bedienelemente, von **SCHRIFTNAME** bis **UNTERSTRICHEN**, sind die wichtigsten *Zeichenformate* – sie können einzelnen Zeichenfolgen innerhalb eines Absatzes zugewiesen werden. Dies gilt nach der Auswahl eines bestimmten Formats für alle nachfolgenden Zeichen, kann aber auch nachträglich allen markierten Zeichen zugewiesen werden. Zum Markieren können Sie entweder mit gedrückter linker Maustaste ziehen oder beliebige Richtungstasten (Pfeiltasten, **Pos1** und **Ende**, **Bild ↑** und **Bild ↓** etc.) zusammen mit **⇧** verwenden.

Die nächste Schaltfläche dient der Auswahl der Schriftart. Wie bereits im **Typografie-Abschnitt** erwähnt, sollten Sie in einem Dokument niemals zu viele Schriften miteinander kombinieren. Eine relativ gut ausgebaute Schriftfamilie, die mit openSUSE geliefert wird, ist beispielsweise die Nimbus; sie besitzt eine serifenlose und eine serifenbetonte Variante (Nimbus Roman beziehungsweise Nimbus Sans) und sogar eine Nichtproportionalschrift (Mono) für Listings und Ähnliches.

Als Nächstes können Sie die Schriftgröße einstellen. Wählen Sie für Fließtext im Normalfall 10–12 Punkt, für Hauptüberschriften nicht mehr als 18 Punkt und für die untergeordneten Überschriften die entsprechenden Zwischenwerte.

Die nachfolgenden drei Schaltflächen dienen der Hervorhebung einzelner Zeichen: **FETT**, **KURSIV** und **UNTERSTRICHEN** (Letzteres sollten Sie im Normalfall nicht einsetzen; es hat in gedruckten Texten seit dem Ende des Schreibmaschinenzeitalters nichts mehr verloren). Sie können die drei Optionen auch schneller per Tastaturkürzel ein- und ausschalten:

- ▶ **Strg** + **⇧** + **F** für fett
- ▶ **Strg** + **⇧** + **K** für kursiv
- ▶ **Strg** + **⇧** + **U** für unterstrichen

Die nächsten Formatierungen – bis einschließlich EINZUG ERHÖHEN – betreffen Absätze; wenn Sie nichts markieren, wirken sie in jedem Fall auf den ganzen aktuellen Absatz sowie bis auf Weiteres auf alle danach neu eingegebenen Absätze. Alternativ lassen sich natürlich auch mehrere Absätze markieren, um diesen jeweils dieselben Formate zuzuweisen.

Die ersten vier Absatzformate betreffen die *Absatzausrichtung* – sie schließen einander logischerweise aus; jeder Absatz kann nur eine einzige Ausrichtung besitzen. Wie oben erwähnt, kommt in normalen Dokumenten kaum je etwas anderes als linksbündiger Satz (für Geschäftsbriefe) oder Blocksatz (für Artikel und Ähnliches) infrage. Lediglich die Zeile mit Ort und Datum wird in Briefen rechtsbündig ausgerichtet.

Es folgen zwei Schaltflächen für nummerierte Listen beziehungsweise Aufzählungen (Bullet-Listen). Wenn Sie eine von ihnen betätigen, werden alle folgenden Absätze Teil einer entsprechenden Liste, bis Sie denselben Button erneut drücken. Die standardmäßig aktiven Autokorrektur-Optionen sorgen auch dafür, dass Listen erstellt werden, wenn Sie einen Absatz mit »*« oder »-« beziehungsweise mit »1.« beginnen.

Es folgen zwei Schaltflächen, mit deren Hilfe sich Absätze jeweils schrittweise ein- oder wieder ausrücken lassen. Die letzten drei Buttons der Leiste schließlich sind für Textfarbe, Zeichenhintergrundfarbe sowie Absatzhintergrundfarbe zuständig.

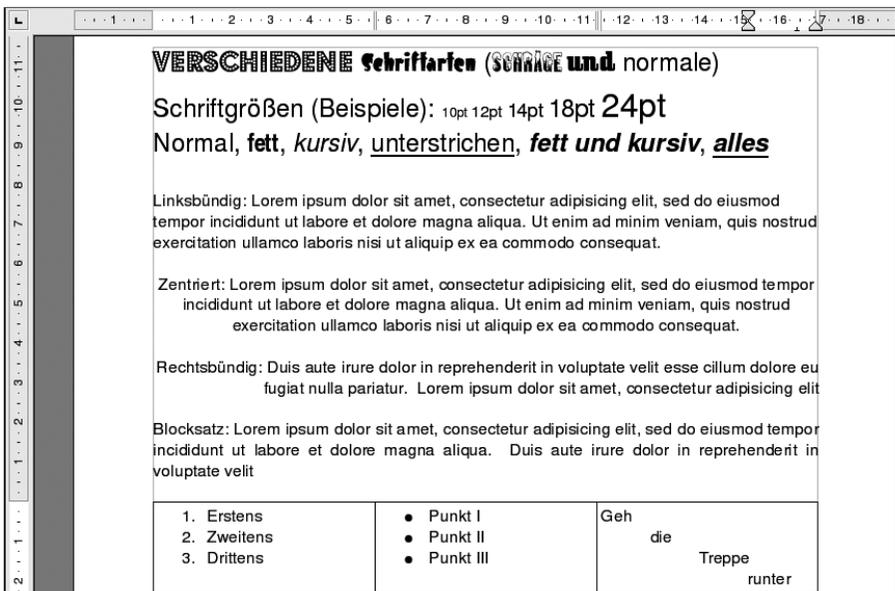


Abbildung 7.4 Beispiele für die einfachen Formatierungen aus der Symbolleiste »Format«

Abbildung 7.4 zeigt einige Formatierungsbeispiele in der Reihenfolge, in der sie in der Symbolleiste vorkommen. Der für die Absatzformate verwendete Text »Lorem ipsum ...« ist übrigens ein wortwörtlich seit dem 16. Jahrhundert gebräuchlicher sogenannter *Blindtext* in verballhorntem Latein. Blindtexte werden verwendet, um realistisch wirkendes Textmaterial zum Testen von Layouts zur Verfügung zu haben. Die Originalversion von »Lorem ipsum« und diverse Abwandlungen können Sie beispielsweise unter <http://www.lipsum.com> herunterladen.

Writer und die anderen OpenOffice.org-Programme besitzen allerdings auch eingebauten Blindtext (einen kurzen, spannenden Krimi-Text). Tippen Sie an der gewünschten Stelle **BT** ein, und drücken Sie danach **[F3]**, um ihn einzufügen.

Zeichenformate

Weitere Formatierungen können Sie unter **FORMAT • ZEICHEN** beziehungsweise **FORMAT • ABSATZ** vornehmen. Der Zeichenformate-Dialog verteilt sich über insgesamt fünf Registerkarten, auf denen Sie im Einzelnen Folgendes einstellen können:

► SCHRIFT

Hier werden die **SCHRIFTART**, der **SCHRIFTSCHNITT** (normal, fett, kursiv und so weiter) sowie der **SCHRIFTGRAD** eingestellt. Zusätzlich können Sie die **SPRACHE** des aktuellen Textes festlegen, was Rechtschreibprüfung und Trennregeln beeinflusst.

► SCHRIFTEFFEKT

Diese Registerkarte bietet zahlreiche Zusatzoptionen für die Wirkung der Schrift. Zunächst können Sie viele verschiedene Arten der **UNTERSTREICHUNG** und **DURCHSTREICHUNG** wählen. Die **SCHRIFTFARBE** steht standardmäßig auf **AUTOMATISCH**; das bedeutet, dass sie durch das übergeordnete Absatz- oder Dokumentformat bestimmt wird.

Unter **AUSZEICHNUNGEN** können Sie sich neben dem Normalzustand für komplette Großschreibung (**VERSALIEN**), vollständige Kleinschreibung (**GEMEINE**), alle Anfangsbuchstaben groß (**TITELSCHRIFT**) oder die oben beschriebenen **KAPITÄLCHEN** entscheiden. Unter **RELIEF** stehen die beiden gegensätzlichen 3-D-Effekte **ERHABEN** und **VERTIEFT** zur Verfügung.

Die vier Kontrollkästchen schließlich bieten diverse Spezialeffekte: **KONTUR** zeigt nur die Umrisse der Zeichen an; **SCHATTEN** fügt einen Schlagschatten hinzu (nur sinnvoll, wenn Sie eine andere Schriftfarbe als Schwarz einstellen); **BLINKEND** funktioniert nur auf dem Bildschirm (und ist somit, wenn überhaupt, vor allem für die Vorführung von Präsentationen mit Impress geeignet).

► POSITION

Dieser Teil des Dialogs enthält zunächst die Positionen HOCH und TIEF – geeignet etwa für mathematische ($a^2 + b^2 = c^2$) beziehungsweise chemische Formeln (H_2SO_4). Die genaue Position und Schriftverkleinerung werden normalerweise automatisch berechnet. Sollte dies für eine bestimmte Schriftart ungünstig sein, dann können Sie notfalls von Hand nachkorrigieren.

Im Bereich ROTATION/SKALIERUNG können Sie vertikale Schrift einstellen (90 GRAD bedeutet von unten nach oben, 270 GRAD von oben nach unten). BREITE SKALIEREN ermöglicht es, die Schrift bei gleichbleibender Höhe um einen prozentualen Wert breiter oder schmaler darzustellen; dies sollten Sie nur in Ausnahmefällen tun, da es aus typografischer Sicht ähnlich kritisch zu bewerten ist wie unechter Kursivsatz oder falsche Kapitalchen.

Die LAUFWEITE schließlich regelt den Zeichenabstand; in manchen Texten werden Wörter auf diese Weise durch Sperrung (leicht erhöhte Laufweite) hervorgehoben; das Gegenteil sollten Sie kaum je in Erwägung ziehen. Die zusätzliche Option PAARWEISES KERNING beachtet trotz Sperrung oder verringertem Abstand die Unterschneidungsregeln der jeweiligen Schrift.

► HYPERLINK

Auf dieser Registerkarte können Sie einen Text-Hyperlink erstellen; dies ist nützlich, wenn Sie Ihre Dateien als HTML- oder PDF-Dokumente exportieren, denn die betreffenden Links bleiben darin stehen und sind aktiv. Im Einzelnen sind folgende Felder und Optionen verfügbar:

- ▶ Unter URL wird zunächst die Adresse eingegeben, die beim Klick auf den Hyperlink geöffnet werden soll – zum Beispiel *http://buecher.lingo-world.de*.
- ▶ Mithilfe der Schaltfläche AUSWÄHLEN können Sie sich alternativ eine lokale Datei aussuchen, die verlinkt werden soll.
- ▶ In das Feld TEXT können Sie den Linktext eintragen, falls Sie beim Aufruf des Dialogs nicht bereits Text markiert hatten.
- ▶ Der NAME ist eine eindeutige Bezeichnung des Links, die vor allem für JavaScript-Verhalten benötigt wird.
- ▶ Unter FRAME können Sie auswählen, wohin die verknüpfte Seite beim Anklicken des Hyperlinks geladen werden soll. Falls Sie eine Webseite mit der heutzutage aus der Mode gekommenen Frame-Technik verwenden, können Sie hier den Namen eines Frames eingeben. Andernfalls können Sie beispielsweise _BLANK aus dem Pulldown-Menü wählen, um den Link in einem neuen Browserfenster zu öffnen. Wenn der Link ganz normal im aktuellen Fenster geöffnet werden soll, brauchen Sie hier nichts anzugeben.

- ▶ Mithilfe der Schaltfläche EREIGNISSE können Sie einstellen, dass der Hyperlink ein OpenOffice.org-Makro (gespeicherte Befehlsfolge) ausführen soll.
- ▶ Unter ZEICHENVORLAGEN können Sie bestimmen, wie der Link in besuchter beziehungsweise unbesuchter Form dargestellt werden soll, indem Sie ihm entsprechende Formatvorlagen (siehe den nächsten Unterabschnitt) zuweisen.
- ▶ HINTERGRUND
Hier können Sie die Hintergrundfarbe für einzelne Zeichen einstellen; dies ist in Einzelfällen nützlich, um diese hervorzuheben.

Absatzformate

Unter FORMAT • ABSATZ können Sie detaillierte Einstellungen für die Formatierung von Absätzen vornehmen. Der Dialog besitzt acht Registerkarten, auf denen Sie folgende Optionen einstellen können:

▶ EINZÜGE UND ABSTÄNDE

Hier wird die Positionierung des Absatzes in seiner Umgebung gesteuert. Zunächst können drei Arten von Einzügen gewählt werden: VOR TEXT bestimmt den Abstand zum Satzspiegelanfang auf der linken Seite, HINTER TEXT legt den rechten Abstand fest, und ERSTE ZEILE sorgt für eine hängende oder eingerückte Startzeile des Absatzes.

Die Abstände ÜBER ABSATZ beziehungsweise UNTER ABSATZ bestimmen, wie weit der Absatz von den umgebenden Absätzen entfernt sein soll. Wenn Sie den Wert in Bezug auf die aktuelle Schriftgröße festlegen möchten, können Sie als Maßeinheit PT statt CM wählen – geben Sie im Zusammenhang mit 12-Punkt-Schrift beispielsweise 6PT als Abstand UNTER ABSATZ ein, um eine halbe Zeile Platz bis zum nachfolgenden Absatz zu lassen. Beachten Sie in diesem Zusammenhang, dass die beiden Teilabstände sich bei aufeinanderfolgenden Absätzen mit identischem Format natürlich addieren. Bei Fließtextabsätzen sollten Sie sich deshalb auf eine der beiden Abstandsarten (vorzugsweise den unteren) beschränken; nur bei Überschriften, auf die üblicherweise ein anderes Absatzformat folgt, sind beide gleichzeitig sinnvoll.

Wenn Sie den ZEILENABSTAND erhöhen (normalerweise maximal auf 1,5-ZEILIG), kann dies in manchen Dokumenten die Lesbarkeit verbessern. Denken Sie daran, dass Sie in diesem Zusammenhang auch den Abstand zwischen den Absätzen erhöhen sollten, damit Absätze als solche erkennbar bleiben.

Die Option REGISTERHALTIGKEIT BERÜCKSICHTIGEN sorgt dafür, dass sich alle Zeilen an einem absoluten, vertikalen Register orientieren. Dies ist besonders nützlich, wenn Sie relativ dünnes Papier beidseitig bedrucken, damit durch-

scheinende, versetzte Zeilen nicht die Lesbarkeit behindern. Dies hebt einige der obigen Einstellungen auf, weil die Registerhaltigkeit Vorrang vor den diversen Abständen erhält. Beachten Sie, dass die eigentliche Registerhaltigkeit in den Seiteneigenschaften eingestellt werden muss.

► **AUSRICHTUNG**

Auf dieser Registerkarte können Sie zunächst die bereits besprochenen Absatzausrichtungen LINKS, RECHTS, ZENTRIERT oder BLOCKSATZ einstellen. Wenn Blocksatz gewählt wird, können Sie zusätzlich das Verhalten der letzten Zeile einstellen: LINKS (Standard und für alle von links nach rechts geschriebenen Sprachen zu empfehlen), RECHTS oder BLOCKSATZ. Letzteres ist der sogenannte erzwungene Blocksatz, der bei kurzen Schlusszeilen recht lächerlich aussieht und deshalb in der Regel vermieden werden sollte.

Die zusätzliche Option TEXT-AN-TEXT bestimmt gegebenenfalls, wie Zeichen unterschiedlicher Schriftgröße innerhalb einer Zeile vertikal aneinander ausgerichtet werden sollen.

► **TEXTFLUSS**

Hier wird im Einzelnen festgelegt, wie die Zeilen und Absätze umbrochen werden und in die nächste logische Einheit (Spalte oder Seite) überfließen. Die erste Kategorie ist die SILBENTRENNUNG. Wenn Sie AUTOMATISCH ankreuzen, wird nach den Regeln der zurzeit eingestellten Sprache getrennt (Sie sollten das Ergebnis allerdings sorgfältig begutachten und bei komplizierten Wörtern selbst die oben empfohlenen optionalen Trennstriche einfügen). Zusätzlich werden die an dieser Stelle konfigurierbaren Mindest-Zeichenzahlen in der oberen und unteren Zeile sowie die maximale Anzahl direkt aufeinanderfolgender Trennzeilen beachtet.

Im Bereich UMBRÜCHE können Sie festlegen, dass vor oder nach dem aktuellen Absatz ein Spalten- oder gar Seitenumbruch stattfinden soll. Das ist beispielsweise für Überschriften einer hohen Hierarchiestufe (etwa Kapitelüberschriften) nützlich, die eine neue Seite einleiten sollen.

Unter ZUSÄTZE werden diverse Einstellungen für den Zusammenhalt der Absätze über Spalten- oder Seitengrenzen hinweg vorgenommen: ABSATZ NICHT TRENNEN hält alle Zeilen des Absatzes zusammen (notfalls auf der nächsten Seite). ABSÄTZE ZUSAMMENHALTEN sorgt dafür, dass der aktuelle und der nachfolgende Absatz nicht voneinander getrennt werden (praktisch etwa für Überschrift und nachfolgenden Fließtextabsatz). Die SCHUSTERJUNGENREGELUNG und die HURENKINDERREGELUNG schließlich verhindern die oben beschriebene Einzelzeilenproblematik, indem sie eine Mindestzahl von Zeilen definieren, die am Seiten- oder Spaltenende beziehungsweise -anfang stehen sollen.

► **NUMMERIERUNG**

An dieser Stelle können Sie festlegen, dass der Absatz Beginn oder Teil einer Aufzählung oder einer nummerierten Liste sein soll. Detailliertere Einstellungen können Sie über den Menübefehl **FORMAT • NUMMERIERUNG UND AUFGÄH- LUNGSZEICHEN** vornehmen.

► **TABULATOR**

Diese Registerkarte dient der Festlegung von Tabstopps, die eine saubere spaltenweise Formatierung von Inhalten ermöglichen. Wenn Sie einen neuen Tabstopp erstellen möchten, geben Sie zunächst unter **POSITION** einen Wert ein; die Positionen werden vom Beginn des Satzspiegels an gerechnet. Wählen Sie im Bereich **TYP** die Ausrichtung des neuen Tabulators: Der Standardtyp **LINKS** setzt den linken Rand des neuen Textes an den Tabstopp, sobald Sie  drücken; bei **RECHTS** wird entsprechend der rechte Rand bündig ausgerichtet – die passende Wahl für eine Auflistung von Zahlen wie etwa Stückzahlen oder Beträgen. **ZENTRIERT** ist nur in Ausnahmefällen sinnvoll. **DEZIMAL** schließlich setzt die Stelle, an der ein bestimmtes Zeichen (vorzugsweise Punkt oder Komma) vorkommt, genau auf den Tabstopp – dies ist insbesondere für Fließkommazahlen geeignet, bei denen die Anzahl der Nachkommastellen variiert. Im Bereich **FÜLLZEICHEN** können Sie zusätzlich festlegen, dass der Leerraum bis zum Tabstopp mit bestimmten Zeichen wie etwa Punkten aufgefüllt werden soll – dies ist beispielsweise bei Inhaltsverzeichnissen üblich.

► **INITIALEN**

In manchen Layouts erstreckt sich der erste Buchstabe bestimmter Absätze, vorzugsweise nach der Hauptüberschrift, über mehrere Zeilen. Dieses Verhalten können Sie hier detailliert einstellen.

► **UMRANDUNG**

Die Einstellungen auf dieser Registerkarte ermöglichen es Ihnen, einen sichtbaren Rahmen um Absätze zu ziehen. Dies ist eine häufige Art der Hervorhebung für Absätze mit spezieller Bedeutung. Wählen Sie **MIT DEM NÄCHSTEN ABSATZ VERSCHMELZEN**, wenn sich mehrere Absätze denselben Kasten teilen sollen.

► **HINTERGRUND**

Eine alternative (oder auch zusätzliche) Art der Hervorhebung für Absätze ist die hier einstellbare Hintergrundfarbe.

Seitenformate

Die nächste Formatierungseinheit eines Textverarbeitungsdokuments nach Zeichen und Absatz ist die Seite. Unter **FORMAT • SEITENEINSTELLUNGEN** können Sie

festlegen, wie sie beschaffen sein soll. Auch dieser Dialog besteht aus acht Registerkarten:

▶ **VERWALTEN**

Legen Sie hier einen Namen für das aktuelle Seitenformat fest, und wählen Sie, welches als Nächstes folgen soll (etwa die Abfolge Titelseite – Standardseite).

▶ **SEITE**

Wählen Sie unter **PAPIERFORMAT** zunächst die gewünschte Seitengröße; voreingestellt ist selbstverständlich A4. Sie können sich zwischen **HOCHFORMAT** und **QUERFORMAT** entscheiden; Ersteres ist Standard. Auf Wunsch können Sie die **BREITE** und **HÖHE** Ihres Formats auch manuell angeben.

Die **SEITENRÄNDER** bestimmen letztlich die Positionierung und Größe des Satzspiegels. Unter **SEITENLAYOUT** wird festgelegt, ob Sie Doppelseiten (rechts und links) oder nur Einzelseiten einer bestimmten Sorte erzeugen möchten – bei Doppelseiten werden Elemente wie Satzspiegel und Seitennummerierung jeweils gespiegelt platziert. Zusätzlich können Sie an dieser Stelle die oben erwähnte **REGISTERHALTIGKEIT** einschalten.

▶ **HINTERGRUND**

Analog zum Zeichen- und Absatzhintergrund können Sie auf dieser Registerkarte eine Hintergrundfarbe für die Seite einstellen.

▶ **KOPFZEILE**

Wenn Sie für die aktuelle Seitenvorlage eine Kopfzeile festlegen möchten, etwa zur Wiederholung der aktuellen Kapitel- beziehungsweise Absatztitel, kreuzen Sie an dieser Stelle **KOPFZEILE EINSCHALTEN** an, und stellen Sie deren Eigenschaften ein.

▶ **FUSSZEILE**

Analog können Sie auch eine Fußzeile anlegen, beispielsweise für die Seitennummerierung.

▶ **UMRANDUNG**

Falls Sie der gesamten Seite (beziehungsweise dem Satzspiegel) einen sichtbaren Rahmen zuweisen möchten, finden Sie auf dieser Registerkarte die notwendigen Hilfsmittel.

▶ **SPALTEN**

An dieser Stelle können Sie Einstellungen für Mehrspaltensatz vornehmen. Geben Sie die Anzahl der **SPALTEN** ein, oder wählen Sie eines der vorgefertigten Spaltenlayouts. In aller Regel (besonders bei Blocksatz) sollten Sie zusätzlich einen **ABSTAND** eingeben, der zwischen den Spalten einzuhalten ist – normalerweise sollte ein Wert zwischen 4 und 5 mm in Ordnung sein.

► FUSSNOTE

Legen Sie auf der letzten Registerkarte fest, wie Fußnoten auf Ihren Seiten formatiert werden sollen. Um später tatsächlich eine Fußnote in den Text einzufügen, wird der Menüpunkt EINFÜGEN • FUSSNOTE verwendet.

7.1.3 Formatvorlagen verwenden

Es wäre überaus mühselig und fehlerträchtig, wenn Sie die oben beschriebenen Formatierungen für jeden Zeichenbereich und jeden einzelnen Absatz durchführen müssten. Damit dies nicht nötig ist, bietet Writer – genau wie viele andere Textverarbeitungs- und Layoutprogramme – sogenannte **FORMATVORLAGEN**. Eine solche Vorlage enthält einen gespeicherten Satz von Zeichenformaten oder Absatzformaten, die Sie Ihren Textblöcken nachträglich per einfachem Mausklick zuordnen können. Ein weiterer Vorteil besteht darin, dass nachträgliche Änderungen einer Vorlage auch automatisch die Formatierung von sämtlichem Text modifizieren, der auf der entsprechenden Vorlage basiert.

Öffnen Sie das Dialogfeld **FORMATVORLAGEN** mithilfe der bereits gezeigten Schaltfläche in der Symbolleiste **FORMAT**, über den Menübefehl **FORMAT • FORMATVORLAGEN** oder die Taste **[F11]**. Der Dialog wird in Abbildung 7.5 gezeigt.

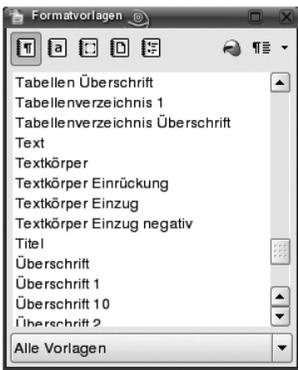


Abbildung 7.5 Die Formatvorlagen in OpenOffice.org Writer

In der oberen Reihe sehen Sie links zunächst fünf Schaltflächen, um zwischen den verschiedenen Arten von Vorlagen zu wechseln:

- Absatzformate
- Zeichenformate
- Rahmenformate
- Seitenformate
- Aufzählungs- und Listenformate

Wählen Sie die passende Kategorie, und doppelklicken Sie auf das gewünschte Format, um es den zurzeit ausgewählten Zeichen, dem aktuellen Absatz oder Ähnlichem zuzuweisen. Alternativ können Sie den sogenannten *Gießkannenmodus* einschalten (oben, zweite Schaltfläche von rechts). Er erlaubt das Zuweisen von Formaten durch einfaches Klicken auf Absätze beziehungsweise Ziehen über einzelne Zeichen.

Wenn Sie ein Format bearbeiten möchten, klicken Sie es mit der rechten Maustaste an. Wenn Sie aus dem Kontextmenü **ÄNDERN** wählen, können Sie das Originalformat editieren, was auch alle Vorkommen des Formats im aktuellen Dokument entsprechend modifiziert. Wählen Sie dagegen **NEU** (entspricht der Schaltfläche ganz rechts oben), um eine neue Vorlage als Kopie der ausgewählten zu erstellen.

Beim Bearbeiten eines Formats gelangen Sie zunächst auf die Registerkarte **VERWALTEN**. Im Feld **NAME** können Sie eine beliebige Bezeichnung für die Vorlage eingeben; das Kontrollkästchen **AUTOM. AKTUALISIEREN** daneben ändert die Vorlage, sobald Sie einzelne Verwendungen derselben mit den oben beschriebenen normalen Formateinstellungen modifizieren. Bei Absatzvorlagen kommt als Nächstes das Feld **FOLGEVORLAGE** – praktisch, um etwa auf eine Überschrift automatisch einen normalen Fließtextabschnitt oder auf einen Definitionstitel (ausgerückt) die eingerückte Definition folgen zu lassen. **VERKNÜPFT MIT** ist ebenfalls interessant: Die Vorlage »erbt« alle Einstellungen von der hier ausgewählten Formatvorlage; Sie brauchen dann nur die Unterschiede neu einzustellen.

Eine Besonderheit bei Absatzvorlagen ist zudem die Registerkarte **BEDINGUNG**: Hier können Sie festlegen, dass die aktuelle Vorlage in einem bestimmten Kontext andere Attribute besitzen soll als im Normalfall. Der Kontext wird dabei jeweils durch eine **VERWENDETE VORLAGE** bestimmt; unter **ABSATZVORLAGEN** können Sie dann wählen, welche Vorlage in Wirklichkeit aktiv werden soll, wenn die betreffende Bedingung zutrifft. Die Einstellungen für die Formate selbst entsprechen im Wesentlichen den im vorigen Unterabschnitt vorgestellten einfachen Formaten. Beachten Sie, dass auch Absatzvorlagen die Registerkarten für das Zeichenformat besitzen, damit Sie die Voreinstellungen für alle Zeichen eines entsprechenden Absatzes vornehmen können.

7.1.4 Weitere Features

OpenOffice.org Writer ist mit unzähligen Funktionen ausgestattet; wenn Sie sie alle kennenlernen möchten, können Sie die umfangreiche Hilfefunktion konsultieren (**HILFE • OPENOFFICE.ORG HILFE**) oder sich ein OpenOffice.org- beziehungsweise Writer-Buch besorgen (siehe Anhang B). Hier werden exemplarisch einige

Zusatzfunktionen angesprochen: Suchen und Ersetzen, Tabellen und Zeichenfunktionen.

Suchen und Ersetzen

Funktionen zum automatischen Suchen und Ersetzen von Text gehören seit jeher zum Repertoire von Textverarbeitungsprogrammen. Writer ist in dieser Hinsicht besonders komfortabel ausgestattet; unter anderem werden auch reguläre Ausdrücke unterstützt.

Sie können den Dialog SUCHEN & ERSETZEN über BEARBEITEN • SUCHEN & ERSETZEN oder mithilfe der Tastenkombination **[Strg] + [F]** aufrufen. In Abbildung 7.6 wird das Dialogfeld gezeigt.

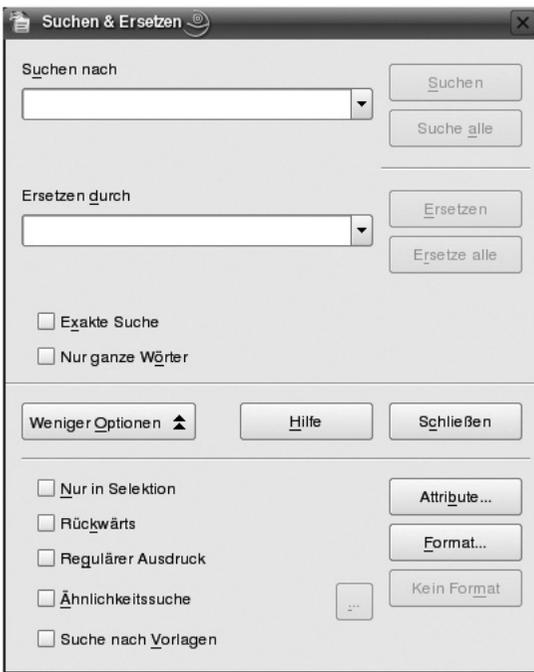


Abbildung 7.6 Der Writer-Dialog »Suchen & Ersetzen«

Die Bedienung ist im Grunde recht intuitiv: Geben Sie zunächst in das Feld SUCHEN NACH eine Zeichenfolge ein, nach der Sie suchen möchten. Wenn Sie das nächste Vorkommen ab der aktuellen Cursorposition suchen möchten, klicken Sie einfach auf SUCHEN. Die Schaltfläche SUCHE ALLE markiert jedes Vorkommen der gesuchten Zeichenfolge. Rechts unten, unmittelbar unterhalb des vertikalen Rollbalkens für das Dokumentfenster, finden Sie übrigens zwei Pfeil-Schaltflä-

chen, mit denen Sie zum vorigen beziehungsweise nächsten Vorkommen des Suchbegriffs klicken können; der kleine Punkt-Button dazwischen öffnet die NAVIGATION, mit der Sie gezielt zu einzelnen Elementen wie Absätzen, Überschriften, Tabellen, Abbildungen und so weiter springen können.

Wenn Sie den gefundenen Text ersetzen möchten, geben Sie den gewünschten Ersatztext in das Feld ERSETZEN DURCH ein. Drücken Sie danach ERSETZEN, um das nächste Vorkommen zu ersetzen, oder ERSETZE ALLE, wenn Sie alle Vorkommen im gesamten beziehungsweise aktuell markierten Text ersetzen möchten.

Die Option EXAKTE SUCHE beachtet ausnahmsweise Groß- und Kleinschreibung, während NUR GANZE WÖRTER die Suche durch Beschränkung auf Wortgrenzen beschleunigt.

Klicken Sie auf MEHR OPTIONEN, um die restlichen Einstellungen zu öffnen:

- ▶ NUR IN SELEKTION beschränkt das Suchen und Ersetzen auf die aktuelle Auswahl.
- ▶ RÜCKWÄRTS sucht ab der Cursorposition nach oben.
- ▶ REGULÄRER AUSDRUCK ermöglicht die Suche nach regulären Ausdrücken; die Syntax entspricht im Wesentlichen `grep` (siehe Kapitel 4). Geben Sie beispielsweise `[aeiou]` ein, um nach beliebigen Vokalen zu suchen.

Hier ein weiteres Beispiel für eine RegExp-Suche: Angenommen, Ihr Dokument enthält zahlreiche ISBN-Nummern (internationale Buchnummern) – mal mit und mal ohne die Trennstriche (Letzteres ist beispielsweise bei Amazon üblich). Die (eigentlich veraltete) zehnstellige ISBN des vorliegenden Buches wäre ohne Trennstriche beispielsweise 383621497-1; mit Trennstrichen ergibt sich dagegen die Schreibweise 3-8362-1497-1. Geben Sie dazu in das Feld SUCHE NACH den folgenden regulären Ausdruck ein:

```
[0-9]\-?[0-9]{3}\-?[0-9]{5}\-?[0-9X]
```

Von links nach rechts sind die folgenden Bestandteile zu unterscheiden:

- ▶ `[0-9]`: genau eine Ziffer
- ▶ `\-?`: optionaler Bindestrich (muss durch einen Backslash eingeleitet werden, wie alle Zeichen, die innerhalb regulärer Ausdrücke eine besondere Bedeutung haben)
- ▶ `[0-9]{3}`: genau drei Ziffern
- ▶ `\-?`: optionaler Bindestrich
- ▶ `[0-9]{5}`: genau fünf Ziffern
- ▶ `\-?`: optionaler Bindestrich

- ▶ [0-9X]: genau eine Ziffer oder ein X (da es sich bei dem letzten Element um eine Modulo-11-Prüfziffer handelt, codiert das X den Divisionsrest 10)
Eine besondere Option sind die Kürzel \< und \>, die für Wortanfang beziehungsweise Wortende stehen. Die bereits im Zusammenhang mit grep erwähnten Zeichen ^ und \$ betreffen in OpenOffice.org dagegen Absatzanfänge beziehungsweise -enden.
- ▶ **ÄHNLICHKEITSSUCHE**
Um leicht abweichende Schreibungen von Wörtern zu finden, bietet sich diese Option an. Die Schaltfläche ... ermöglicht in diesem Fall die Festlegung einer maximalen Anzahl unterschiedlicher Zeichen.
- ▶ **SUCHE NACH VORLAGEN**
In den Feldern SUCHEN NACH und ERSETZEN DURCH wird die Texteingabe deaktiviert; stattdessen erscheinen Pulldown-Menüs. Wählen Sie aus diesen Formatvorlagen aus, um Text mit einer bestimmten Vorlage zu suchen und diese gegebenenfalls durch eine andere zu ersetzen.
- ▶ **ATTRIBUTE**
Hier finden Sie eine Liste besonderer Zustände (Silbentrennung, Umbrüche und mehr), nach denen Sie ebenfalls suchen können.
- ▶ **FORMAT**
Wählen Sie beliebige Formatierungen, denen der Suchtext zusätzlich entsprechen soll.
- ▶ **KEIN FORMAT**
Klicken Sie diese Schaltfläche an, um zuvor gewählte Formatierungen für eine neue Suche komplett auszuschalten.

Tabellen

Viele Textdokumente werden mit Tabellen ausgestattet, um Informationen übersichtlicher darzustellen. Die schnellste Methode zum Einfügen einer Tabelle besteht in Writer darin, die Schaltfläche TABELLE in der Symbolleiste STANDARD gedrückt zu halten und in dem sich öffnenden Feld die Anzahl der gewünschten Zeilen und Spalten zu markieren. Die Zellen der obersten Zeile werden automatisch mit der Formatvorlage TABELLEN ÜBERSCHRIFT formatiert (im Wesentlichen zentriert, fett und kursiv), die restlichen Zellen mit TABELLEN INHALT.

Sobald die Tabelle auf diese Weise eingefügt wurde, können Sie in das erste Feld klicken und Text eingeben. Mit  gelangen Sie in das nächste Feld. Falls Sie im letzten Feld der Tabelle erneut  drücken, wird automatisch eine neue Zeile eingefügt. Sollten Sie innerhalb einer Tabellenzelle einen Tabulator benötigen, wird dieser ausnahmsweise mit  +  erzeugt.

Weitere Bearbeitungsbefehle bieten der Hauptmenüpunkt TABELLE sowie die Symbolleiste TABELLE (ANSICHT • SYMBOLLEISTEN • TABELLE). Beide enthalten Funktionen zum Hinzufügen oder Löschen von Tabellenteilen sowie diverse Formatierungsbefehle.

Zeichenfunktionen

Mithilfe der Symbolleiste ZEICHNEN können Sie *Vektorgrafiken* (Definition siehe nächstes Kapitel) erstellen; dieses Arbeitswerkzeug steht in allen OpenOffice.org-Programmen zur Verfügung. Wählen Sie ANSICHT • SYMBOLLEISTEN • ZEICHNEN, um es einzublenden. Sie können dieselben Funktionen wahlweise auch als eigenes Programm namens OpenOffice.org Draw starten (normalerweise unter GRAFIK • VEKTORGRAFIK im Menü des jeweiligen Desktops).



Abbildung 7.7 Die OpenOffice.org-Symbolleiste »Zeichnen«

In Abbildung 7.7 sehen Sie die gesamte Zeichnen-Symbolleiste. Sie enthält von links nach rechts folgende Arbeitswerkzeuge und Befehle:

► AUSWAHL

Das Mauszeigerwerkzeug dient der Auswahl bereits erstellter Zeichnungen, um diese nachträglich zu formatieren, zu ändern oder mithilfe von Zwischenablage-Operationen zu bearbeiten. Sie können ausgewählte Elemente auch über die Pfeiltasten verschieben; jeder Klick bewegt sie um einen Zentimeter (bezogen auf die Druckgröße); mit gedrückter **[Alt]**-Taste können sie auch um einzelne Pixel verschoben werden.

Wenn Sie mehrere Objekte auswählen möchten, halten Sie ab dem zweiten die **[⇧]**-Taste gedrückt, um diese zur Auswahl hinzuzufügen beziehungsweise davon abzuziehen.

Wenn Sie an den Anfasserpunkten außerhalb eines ausgewählten Objekts ziehen, können Sie es skalieren. Halten Sie zusätzlich die **[⇧]**-Taste gedrückt, um proportional zu skalieren, das heißt das Höhen- und Breitenverhältnis beizubehalten.

► LINIE

Ziehen Sie mit diesem Werkzeug beliebige Linien auf. Wenn Sie **[⇧]** festhalten, wird der Winkel auf ein Vielfaches von 45° beschränkt.

- ▶ **RECHTECK**
Dieses Werkzeug zeichnet Rechtecke; mit gedrückter -Taste sind es Quadrate.
- ▶ **ELLIPSE**
Mithilfe dieser Option können Sie Ellipsen oder – mit  – Kreise zeichnen.
- ▶ **FREIHANDLINIE**
Dieser Befehl dient dem freihändigen Zeichnen beliebiger Kurven oder Linien. Wenn Sie Linienzüge schließen, werden die Innenräume mit der aktuellen Füllfarbe gefüllt.
- ▶ **TEXT**
Wenn Sie diese Schaltfläche betätigen, können Sie ein frei positionierbares Textfeld aufziehen. Beachten Sie, dass Sie auch alle geschlossenen Zeichnungen mit Text versehen können, indem Sie diese mit dem Auswahlwerkzeug doppelklicken.
- ▶ **LEGENDE**
Erzeugt ein Textfeld mit einem anhängenden Pfeil, das Sie zur Beschriftung von Zeichenelementen verwenden können.
- ▶ **STANDARDFORMEN**
Öffnet eine Liste mit diversen geometrischen Figuren, die Sie in beliebiger Größe aufziehen können.
- ▶ **SYMBOLFORMEN**
Dieses Pulldown-Menü enthält einige verspieltere Formen.
- ▶ **BLOCKPFEILE**
In diesem Bereich stehen diverse gefüllte Pfeile zur Verfügung.
- ▶ **FLUSSDIAGRAMME**
Dieses Menü enthält die wichtigsten Elemente, die Sie zum Zeichnen von Flussdiagrammen für die Softwareentwicklung benötigen.
- ▶ **LEGENDEN**
Enthält einige weitere Auswahlmöglichkeiten für Legenden sowie Sprechblasen.
- ▶ **STERNE**
Hier sind diverse Sternformen zu finden.
- ▶ **PUNKTE**
Bietet Zugriff auf die einzelnen Eckpunkte beziehungsweise Kurvenankerpunkte einer Freihandzeichnung. Wenn Sie einen von ihnen anklicken, können Sie ihn mit der Maus verschieben oder den Punkteverlauf über die Anfasserpunkte der angezeigten Hilfslinie modifizieren.

- ▶ **FONTWORK GALLERY**
Eine Galerie für Schmucktexteffekte
- ▶ **AUS DATEI**
Ermöglicht das Einfügen einer Grafik- oder Bilddatei.
- ▶ **EXTRUSION**
Diese Schaltfläche aktiviert beziehungsweise deaktiviert einen 3-D-Effekt für das aktuelle Objekt – die Form wird zu einer Box in eine bestimmte Richtung erweitert. Die normalerweise automatisch angezeigte Symbolleiste 3-D-EINSTELLUNGEN ermöglicht genaue Einstellungen für Richtung, Tiefe und Beleuchtung der Extrusion.



Abbildung 7.8 Das Dialogfeld »Zeichnungsobjekt-Eigenschaften«

Die genauen Einstellungen für Aussehen und Positionierung Ihrer Zeichenobjekte können Sie mithilfe der Symbolleiste ZEICHNUNGSOBJEKT-EIGENSCHAFTEN (siehe Abbildung 7.8) vornehmen. Hier stehen von links nach rechts folgende Optionen zur Verfügung:

- ▶ **LINIE**
Öffnet einen Dialog mit detaillierten Einstellungen für Linienstärke, -farbe und -stil.
- ▶ **LINIENENDSTIL**
Hier stehen zahlreiche Pfeilspitzen zur Auswahl.
- ▶ **LINIENSTIL**
Dieses Pulldown-Menü enthält viele verschiedene Linienstile (durchgezogen, gestrichelt, gepunktet, doppelt und so weiter).
- ▶ **LINIENBREITE**
In dieses Eingabefeld können Sie die gewünschte Linienstärke eingeben oder mithilfe der danebenliegenden Pfeil-Schaltflächen schrittweise steigern beziehungsweise vermindern.
- ▶ **LINIENFARBE**
Das Menü listet die wichtigsten Linienfarben auf; mehr Auswahl liefert der oben erwähnte Dialog LINIENSTIL.
- ▶ **FLÄCHE**
Ein Dialog, in dem Sie sämtliche Aspekte der Objektfüllung einstellen können – Farbe oder Farbverlauf, Schlagschatten, Transparenz, Schraffierung und so weiter.

- ▶ **FLÄCHENSTIL/-FÜLLUNG (ART)**
Schnellzugriff auf die Füllungsart (KEINE, FARBE, VERLAUF, SCHRAFFUR oder BITMAP).
- ▶ **FLÄCHENSTIL/-FÜLLUNG (AUSWAHL)**
Die wichtigsten vorgefertigten Auswahloptionen der gewählten Füllungsart. Wenn Sie mehr Kontrolle benötigen, können Sie wie angesprochen den Dialog FLÄCHE öffnen.
- ▶ **DREHEN**
Diese Option ermöglicht es Ihnen, ein zuvor ausgewähltes Objekt durch Ziehen an den vier umgebenden Anfasserpunkten zu drehen. Zusätzlich können Sie den Mittelpunkt verschieben, was die Drehrichtung modifiziert.
- ▶ **IN VORDERGRUND**
Wenn mehrere Objekte übereinander platziert werden, verschiebt dieser Befehl das ausgewählte Objekt um eine Ebene nach vorn.
- ▶ **IN HINTERGRUND**
Verschiebt das aktuelle Element entsprechend eine Ebene nach hinten.
- ▶ **GANZ NACH VORN**
Platziert das aktuelle Objekt ganz im Vordergrund.
- ▶ **GANZ NACH HINTEN**
Positioniert das markierte Element am weitesten hinten.
- ▶ **AUSRICHTUNG**
Mit diesem Befehl können Sie mehrere ausgewählte Objekte horizontal oder vertikal aneinander ausrichten, und zwar jeweils entweder an einem der Außenränder oder in der Mitte.
- ▶ **VERANKERUNG WECHSELN**
Bestimmt, mit welchem Textelement die aktuelle Zeichnung verknüpft werden soll, was letztlich seine Positionierung im Dokument bestimmt. Die besondere Verankerung ALS ZEICHEN wurde bereits weiter oben im Zusammenhang mit importierten Bildern erwähnt.
- ▶ **AUFHEBEN**
Macht eine Gruppierung (siehe nächstes Steuerelement) rückgängig.
- ▶ **GRUPPIEREN**
Diese Option behandelt mehrere ausgewählte Elemente bezüglich Skalierung, Anordnung, Ausrichtung und so weiter künftig wie ein Einzelelement.

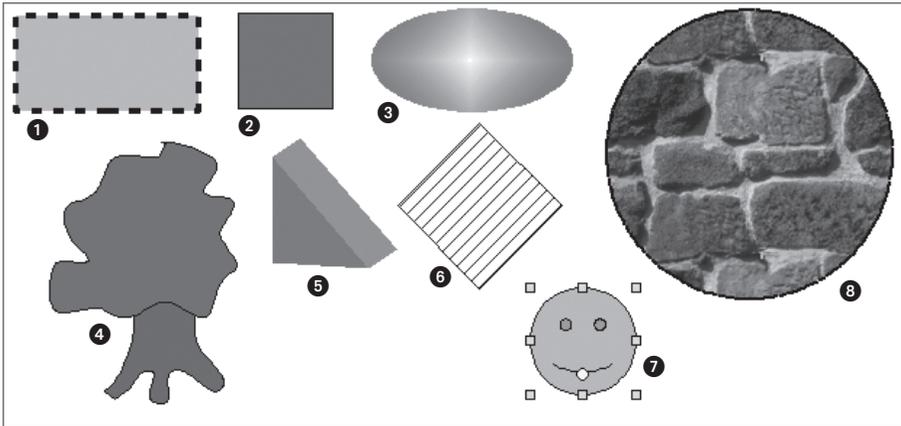


Abbildung 7.9 Einige mit den OpenOffice.org-Zeichnenfunktionen erstellte Grafiken

In Abbildung 7.9 sehen Sie einige Beispiele für die oben beschriebenen Werkzeuge und Formatierungen im Überblick. Die nummerierten Objekte wurden wie folgt erstellt:

1. Rechteckwerkzeug, Linienform fein gestrichelt, Linienbreite 0,10 cm
2. Quadrat (Rechteckwerkzeug mit )
3. Ellipsenwerkzeug, Füllungsart VERLAUF, Linienform unsichtbar
4. zwei geschlossene Freihandzeichnungen
5. Dreieck aus dem Pulldown-Menü STANDARDFORMEN; 3-D-Effekt mittels EXTRUDIEREN hinzugefügt
6. Raute aus dem Pulldown-Menü STANDARDFORMEN, Füllungstyp SCHRAFFUR
7. Smiley aus dem Pulldown-Menü SYMBOLFORMEN; *markiert*, sodass die Anfasserpunkte zum Skalieren sichtbar sind. Mit dem runden Punkt auf dem Mund (im Original gelb) können Sie den Smiley auch zum Frownie :-(- umfunktionieren; viele der Symbolformen sind mit solchen »magischen« Anfasserpunkten ausgestattet.
8. Kreis (Ellipsenwerkzeug mit ) , Füllungsart BITMAP, Standardauswahl MAUER.

7.2 Tabellenkalkulation mit Calc

Eine *Tabellenkalkulation* (englisch Spreadsheet) ermöglicht die Anfertigung von Berechnungen in Tabellenform, wie sie unter anderem für Bilanzen, Finanzie-

rungskalkulationen, Monatsabrechnungen und viele andere Aufgaben eingesetzt werden. Der entscheidende Fortschritt ist dabei nicht die Tabellenform selbst; diese wurde bereits vor Hunderten von Jahren von Kaufleuten in großen Tabellenbüchern eingesetzt. Das Praktische an einem elektronischen Spreadsheet ist vielmehr, dass Sie bei einem Fehler oder einer Änderung in den zugrundeliegenden Daten nicht die gesamten darauf aufbauenden Berechnungen neu durchführen brauchen – eine Zelle in einer Kalkulationstabelle kann neben einem festen Wert auch eine Formel enthalten, die dynamisch aus den umgebenden Werten berechnet wird.

OpenOffice.org Calc ist ein besonders leistungsfähiges Tabellenkalkulationsprogramm, dessen Funktionsumfang höchstens von dem Klassiker Microsoft Excel übertroffen wird. Neben den eigentlichen Kalkulationen können Sie Ihre Tabellen beliebig formatieren, zudem können Sie sich alle Daten auch in Form verschiedenster Diagramme aufbereiten lassen.

Da das Programm so viele verschiedene Optionen zu bieten hat, lernen Sie hier einige der wichtigsten in Form eines kurzen Schritt-für-Schritt-Tutorials kennen: Sie erstellen eine Klimatablelle für Köln; daraus soll anschließend ein Klimadiagramm erstellt werden.

7.2.1 Eine Tabelle erstellen

In Abbildung 7.10 sehen Sie zunächst, wie sich das Programm Calc nach dem Start (BÜROPROGRAMME • TABELLENKALKULATION • OPENOFFICE.ORG CALC im PROGRAMME-Menü des jeweiligen Desktops) präsentiert. Sie befinden sich auf dem ersten von drei Kalkulationsblättern, zwischen denen Sie durch die drei Karteireiter TABELLE 1 und folgende unterhalb des eigentlichen Tabellenbereichs wechseln können. Dort können Sie auch mit der rechten Maustaste klicken, um weitere Tabellenblätter einzufügen.

Weiterhin können Sie sehen, dass die Spalten mit A, B, C ... und die Zeilen mit 1, 2, 3 ... beschriftet sind. In Formeln können Sie sich mithilfe von Ausdrücken wie A1 oder C7 auf bestimmte Zellen beziehen, um deren Werte in die Berechnung einzubeziehen. Die stark umrandete Zelle A1 ist die aktuelle Arbeitszelle; sobald Sie etwas eintippen, erscheint es in dem langen Textfeld über der Tabelle sowie in der Zelle selbst. Schließen Sie eine Eingabe ab, indem Sie das Häkchen ÜBERNEHMEN links neben dem Eingabefeld betätigen. Alternativ können Sie  drücken, um gleichzeitig in die benachbarte Zelle zu wechseln, oder  für die nächste Zeile.

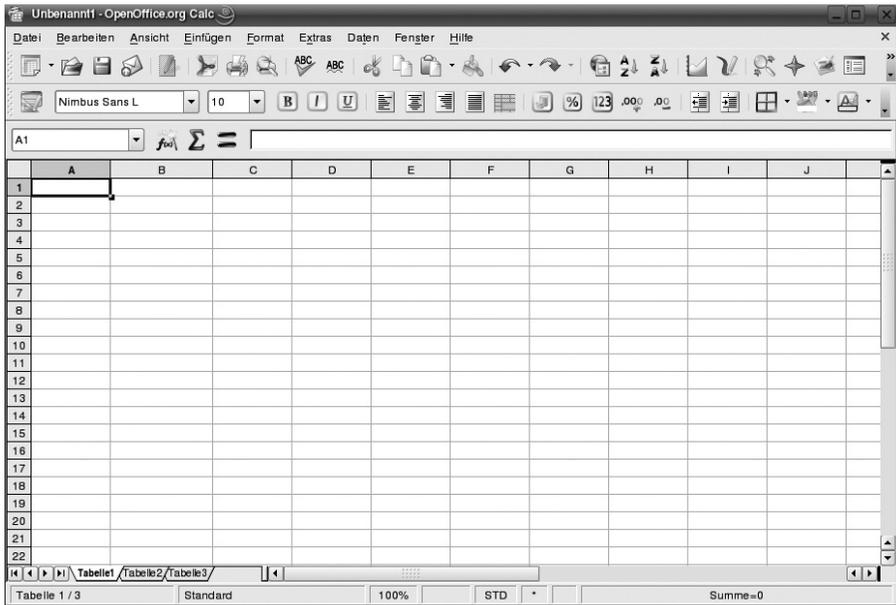


Abbildung 7.10 Die Arbeitsoberfläche von OpenOffice.org Calc

Sie können in eine Zelle unter anderem folgende Arten von Werten eingeben:

► *Zahlen*

Geben Sie beliebige Zahlen ein; anders als in Programmiersprachen ist das Dezimaltrennzeichen in der deutschen OpenOffice.org-Version tatsächlich ein Komma. Zahlen werden automatisch rechtsbündig ausgerichtet.

► *Text*

Sobald Sie das erste Zeichen eingeben, das nicht zu einer Zahl gehören kann, wird der Inhalt der aktuellen Zelle als Text interpretiert und linksbündig angeordnet.

► *Datum und Uhrzeit*

Eingaben wie »2.4.06« oder »1.1.70« werden als Datum interpretiert. Intern wird die Jahreszahl stets vierstellig gespeichert; wenn Sie sie zweistellig eingeben, werden 00–69 als 2000 bis 2069 interpretiert, 70–99 dagegen als 1970 bis 1999. Die Bestandteile von Uhrzeiten werden durch Doppelpunkte voneinander getrennt.

► *Formeln*

Wenn Ihre Eingabe mit einem Gleichheitszeichen beginnen, werden sie als Formeln interpretiert; im Eingabefeld erscheint jeweils die Formel, in der Tabelle selbst dagegen ihr augenblicklicher Wert.

Die automatische Formatierung der verschiedenen Arten von Werten gelingt erstaunlich gut; falls Sie trotzdem nachbessern möchten, können Sie eine oder mehrere Zellen markieren, mit der rechten Maustaste anklicken und dann ZELLEN FORMATIEREN wählen. Die Option FORMAT • ZELLEN im Hauptmenü bewirkt dasselbe. Hier können Sie etwa die Anzahl der Nachkommastellen, das Tausendertrennzeichen, Schriftart und -Farbe und vieles andere einstellen.

Eine besonders praktische Option ist das Auto-Ausfüllen: Wenn Sie in eine Zelle einen Wert eingetragen haben, können Sie an deren linkem unterem Rand ziehen (je nach Bedarf nach rechts oder nach unten), um die nachfolgenden Zellen mit logisch passenden Werten zu füllen. Mitunter müssen Sie auch zwei aufeinanderfolgende Zellen beschreiben und dann beide markieren, um der Zähl-Logik von Calc auf die Sprünge zu helfen. Hier einige Beispiele für Einträge in der Startzelle und die Folgen:

- ▶ 1 – 2, 3, 4, 5, 6, ...
- ▶ Januar – Februar, März, April ...
- ▶ 1. Kalenderwoche – 2. Kalenderwoche, 3. Kalenderwoche ...
- ▶ 2, 4 (zwei Zellen) – 6, 8, 10, 12 ...

Damit steht bereits das erste Hilfsmittel für die Klimatabelle fest: Geben Sie in die Zelle A2 den Text »Januar« ein, und bestätigen Sie Ihre Eingabe mit dem Häkchen. Ziehen Sie an der rechten unteren Ecke der Zelle nach unten, bis Sie A13 und damit den Dezember erreicht haben. Falls die Spalte zu schmal für manche Monatsnamen ist, können Sie auf die Trennung zwischen den Spaltenbeschriftungen A und B doppelklicken (oder ziehen, um eine bestimmte Breite festzulegen).

Geben Sie nun in die Zelle A14 »Gesamt« ein. Als Nächstes soll die Zelle B1 mit »°C« (Temperatur in Grad Celsius) und C1 mit »mm« (Niederschlagsmenge in mm) beschriftet werden.

Markieren Sie nun die Zellen A1 bis C1, und nehmen Sie mit gedrückter Strg-Taste A2 bis A13 hinzu. Da diese Zellen die Beschriftungen bilden, sollten Sie die Schaltfläche B in der Symbolleiste FORMAT betätigen, um die Zellen fett zu formatieren. Wenn Sie möchten, können Sie danach auch durch Ziehen den gesamten Block von A1 bis C14 markieren, um mithilfe der passenden Format-Buttons beispielsweise eine Umrandung und/oder einen farbigen Hintergrund auszuwählen.

Nach diesen Vorbereitungen können Sie die Klimawerte eingeben, wie sie in Abbildung 7.11 zu sehen sind. Wenn Sie möchten, können Sie auch die Werte für Ihren eigenen Heimatort oder eine ganz andere Stadt suchen und diese stattdessen verwenden.

	A	B	C
1	KLIMA KÖLN	°C	mm
2	Januar	1,8	62
3	Februar	2,4	48
4	März	5,2	63
5	April	8,7	55
6	Mai	13,2	74
7	Juni	16,2	86
8	Juli	17,8	84
9	August	17,4	77
10	September	14,2	61
11	Oktober	10,3	55
12	November	5,6	65
13	Dezember	2,9	72
14	Gesamt		

Abbildung 7.11 Die Klimawerte für Köln in der Kalkulationstabelle

Danach geht es an die eigentliche Kalkulation: Die Gesamtwerte sollen berechnet werden. Dies ist bei den Temperaturen das arithmetische Mittel – mathematisch ausgedrückt:

$$(t1 + t2 + \dots + t12) : 12$$

Bei den Niederschlägen wird stattdessen die Summe berechnet.

Klicken Sie für den Mittelwert der Temperaturen in das Feld B14, und geben Sie Folgendes ein:

```
=MITTELWERT(B2:B13)
```

Die Eingabe von `=MITTELWERT()` können Sie in der Regel schon frühzeitig durch  bestätigen, sobald der entsprechende Toollipp angezeigt wird. Statt den Zellbereich B2:B13 manuell einzutippen, können Sie ihn auch mit der Maus aufziehen, sobald die Funktion selbst geschrieben oder automatisch ergänzt wurde. Über Zellangaben in Formeln sollten Sie noch wissen, dass der Doppelpunkt für einen fortlaufenden Bereich steht (hier B2–B13), während ein Semikolon einzelne Zellen auflistet: B2;B13 stünde beispielsweise für die beiden Einzelzellen B2 und B13.

Als Nächstes soll im Feld C14 die Summe der mittleren Niederschläge aus den Zellen C2–C13 berechnet werden. Dafür genügt es in diesem Fall, die Ergebniszelle anzuklicken und das Summenzeichen links neben dem Eingabefeld zu markieren. Alternativ können Sie die Formel auch manuell eintippen:

```
=SUMME(C2:C13)
```

7.2.2 Ein Diagramm erstellen

Aus den Werten des Klimadiagramms soll nun ein Diagramm erstellt werden. In der Regel werden 10 °C auf der rechten Y-Achse mit 20 mm auf der linken Y-Achse zueinander in Beziehung gesetzt. Die Monate, in denen die Niederschlagslinie in dieser Darstellung die Temperaturlinie überschreitet, werden als humid (ausreichend feucht) bezeichnet, die anderen als arid (trocken). Wie Sie in dem Klimadiagramm für Köln sehen werden, besitzt diese Stadt ein voll humides Klima.

Markieren Sie den gesamten Bereich A1:C13 (ohne die Ergebniszeile), und wählen Sie dann EINFÜGEN • DIAGRAMM. Der Dialog AUTOFORMAT DIAGRAMM wird angezeigt. Achten Sie darauf, dass unter BEREICH die korrekten Werte eingestellt sind. Die Dollarzeichen stehen übrigens für absolute Angaben, die in der Kalkulationstabelle auch dann nicht geändert werden, wenn Sie eine Formel per Auto-Ausfüllen vervielfältigen (normale Bezüge ohne \$ werden automatisch angepasst). Da die Beschriftungen bereits im gewählten Tabellenbereich enthalten sind, wählen Sie ERSTE ZEILE ALS BESCHRIFTUNG sowie ERSTE SPALTE ALS BESCHRIFTUNG. Klicken Sie danach auf WEITER.

Nun geht es um die Auswahl des Diagrammtyps. Wählen Sie FLÄCHEN, das zweite Objekt von links in der obersten Zeile. Die bereits korrekt gewählte Option DATENREIHEN IN: SPALTEN bestimmt, dass die beiden Diagrammflächen den Datenspalten der Tabelle entnommen werden sollen. Auf der nächsten Seite können Sie sich für verschiedene Untertypen entscheiden; die Vorauswahl ist in Ordnung.

Als Nächstes geht es darum, eine zweite Y-Achse einzufügen und jeden Wert an einer von ihnen zu orientieren. Wählen Sie EINFÜGEN • ACHSEN, und entscheiden Sie sich für eine SEKUNDÄRE Y-ACHSE. Verwenden Sie nun den Menübefehl FORMAT • ACHSE • Y-ACHSE beziehungsweise FORMAT • ACHSE • SEKUNDÄRE Y-ACHSE, um die Wertebereiche der neuen Achsen anzupassen. Wechseln Sie dazu auf die Registerkarte SKALIERUNG, und deaktivieren Sie die AUTOMATISCH-Kontrollkästchen neben MINIMUM beziehungsweise MAXIMUM. Tragen Sie als MINIMUM jeweils 0 und als MAXIMUM 50 für die primäre und 100 für die sekundäre Y-Achse ein.

Zum Schluss muss die Datenreihe mm an der sekundären Y-Achse orientiert werden. Doppelklicken Sie diese dazu irgendwo innerhalb des eigentlichen Diagramms. Wählen Sie auf der Registerkarte OPTIONEN unter DATENREIHE AUSRICHTEN AN den Wert SEKUNDÄRER Y-ACHSE. Zusätzlich können Sie die Datenreihen hier umfärben – traditionell die Temperaturfläche rot und die Niederschlagsfläche blau.

In Abbildung 7.12 sehen Sie das fertige Diagramm, in dem zusätzlich der Haupttitel geändert wurde.

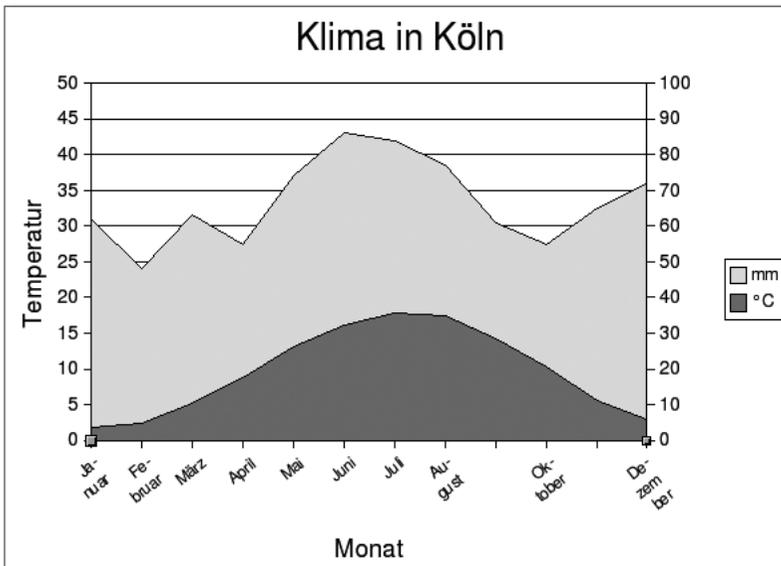


Abbildung 7.12 Das mit Calc erstellte Klimadiagramm für Köln

7.2.3 Formeln und Funktionen

Die Arbeit mit Formeln und den darin enthaltenen Operationen und Funktionen ist im Grunde die wichtigste Aufgabe in Calc. Daher werden an dieser Stelle einige weitere Informationen über Formeln sowie mehr Funktionen als `SUMME()` und `MITTELWERT()` vorgestellt.

Prinzipiell können Formeln mit konkreten Werten (Literalen) sowie mit Referenzen (Bezügen) auf Tabellenzellen arbeiten. Mögliche Literale sind Ganzzahlen, Fließkommazahlen (bei der deutschen Calc-Version gemäß Locale tatsächlich mit Komma, nicht mit Punkt wie in den meisten Programmiersprachen) und Strings in doppelten Anführungszeichen.

Für Zellreferenzen gelten folgende Regeln:

- ▶ Einzelne Tabellenzellen werden durch Spaltenbuchstabe und Zeilennummer ohne Abstand gekennzeichnet, zum Beispiel A1 oder AB199.
- ▶ Wenn Formeln per Auto-Ausfüllen nach rechts oder nach unten übernommen (oder über die Zwischenablage kopiert) werden, dann werden die Bezüge automatisch angepasst. Angenommen, in einer Zelle befindet sich ein Bezug

auf B3. Wird die Formel um ein Feld nach rechts verschoben, dann lautet der Bezug C3; bei einer Verschiebung nach unten wird er zu B4. Zum Schutz vor dieser Anpassung sind absolute Bezüge möglich, indem ein Dollarzeichen vor dem jeweiligen Element notiert wird: \$B3 würde nur bei einer Verschiebung nach unten, aber nicht nach rechts geändert; bei B\$3 wäre es umgekehrt, und \$B\$3 würde stets genau gleich bleiben.

- ▶ Manche Funktionen – beispielsweise die bereits vorgestellten Funktionen SUMME() und MITTELWERT() – arbeiten mit Gruppen von Zellen. Bezüge auf einzelne Zellen werden dabei durch Doppelpunkt getrennt. A1:A4 verweist beispielsweise auf die Zellen A1, A2, A3 und A4, während A1:B2 für A1, A2, B1 und B2 steht. Solche Bezüge lassen sich während der Formeleingabe durch Ziehen mit der Maus erstellen oder indem Sie zuerst die erste gewünschte Zelle und dann mit gehaltener -Taste die letzte anklicken.
- ▶ Nicht nebeneinanderliegende Zellen oder Zellfolgen werden durch Semikolon voneinander getrennt. A1:A3;A5 steht beispielsweise für die drei angegebenen einzelnen Zellen. A1:A4;C1:C4;D1 bezeichnet entsprechend die Bereiche A1 bis A4, C1 bis C4 sowie die einzelne Zelle D1. Bei der Formeleingabe können Sie Zellen mit gedrückter -Taste anklicken oder ziehen, um solche getrennten Zellen beziehungsweise Bereiche zu markieren.
- ▶ Zellreferenzen können sich auf andere Tabellen in derselben Datei beziehen. Dazu wird der Tabellenname – der sich per Doppelklick auf den jeweiligen Karteireiter am Fuß der Tabelle ändern lässt – durch einen Punkt getrennt vor die Referenz geschrieben. Tabelle1.A1 steht zum Beispiel für die Zelle A1 in Tabelle1; Jahresabschluss.A1:A10 für den angegebenen Bereich in der (umbenannten) Tabelle Jahresabschluss. Wenn Sie mehrere, durch Semikolon getrennte Bezüge verwenden, müssen Sie den Tabellennamen vor jeder von ihnen wiederholen. Beispiel: Tabelle2.A1:A5;Tabelle2.C1:C5.
- ▶ In der Symbolleiste über der Tabelle wird ganz links die Bezeichnung der Zelle oder des einzelnen Bereichs angezeigt, die zuletzt markiert wurden. Bei einer bestehenden Markierung können Sie hier einen beliebigen Namen eingeben, unter dem der Zellbezug künftig tabellenübergreifend angesprochen werden kann. Markieren Sie beispielsweise die Felder B2 bis B13, und nennen Sie den Bereich »Monatsumsatz«. Dann könnten Sie mithilfe der Formel

=SUMME(Monatsumsatz)

die Summe dieses Bereichs berechnen. Das Namenseingabefeld dient zusätzlich als Pulldown-Menü, um alle bereits verwendeten Bezeichner später wieder markieren oder für Formeln auswählen zu können.

Beachten Sie, dass benannte Bereiche in Formeln stets absolute Bezüge darstellen; wenn Sie diese Formeln verschieben oder kopieren, werden solche Bezüge also niemals angepasst.

Die offensichtlichsten Elemente für Formeln sind die arithmetischen Operatoren: +, -, * und /. Zusätzlich wird ^ als Potenz-Operator verwendet. Für die Modulo-Operation (Rest der Division) steht dagegen kein spezieller Operator zur Verfügung, sondern die Funktion `REST(Dividend; Divisor)`. Die Formel `=REST(7;3)` ergibt beispielsweise 1. Das Gegenstück ist die Funktion `QUOTIENT(Dividend; Divisor)`; sie errechnet im Gegensatz zu / kein Fließkomma-Ergebnis, sondern liefert lediglich den ganzzahligen Anteil. `=QUOTIENT(7;3)` ergibt somit 2.

Die Rangfolge arithmetischer Operatoren entspricht dem mathematischen Standard: Zuerst werden Potenzen ausgewertet, dann Multiplikation und Division und zuletzt Addition und Subtraktion. Sie können jederzeit Klammern verwenden, um die Rangfolge zu modifizieren. So ergibt die Formel `=5+6*7` den Wert 47, während `=(5+6)*7` das Ergebnis 77 hat.

Außer den Operatoren stehen in Calc Unmengen von Funktionen zur Verfügung. Einige der wichtigsten und interessantesten sind – neben den bereits genannten – folgende:

- ▶ `PRODUKT(Wert[;Wert;...])` berechnet das Produkt aller angegebenen Werte beziehungsweise Zellbezüge.
- ▶ `MAX(Wert[;Wert; ...])` liefert den maximalen Wert aus der angegebenen Gruppe von Werten (von Literalen und/oder Zellbezügen).
- ▶ `MIN(Wert[;Wert; ...])` gibt entsprechend den minimalen Wert aus der untersuchten Gruppe zurück.
- ▶ `WENN(Testwert[;Dann-Wert[;Sonst-Wert]])` ist die wichtigste aus einer Reihe von Funktionen, die programmiersprachenähnliches Verhalten zu Calc hinzufügen. Die Funktion berechnet zunächst den `Testwert`. Wenn er zutrifft (jeder Wert außer 0, dem leeren String oder einer leeren Zelle gilt als wahr), nimmt die Zelle den `Dann-Wert` an; andernfalls – falls vorhanden – den `Sonst-Wert`. Wurde kein `Dann-Wert` angegeben und der `Testwert` ist wahr, dann erhält die Zelle den speziellen Wert `WAHR`; bei falschem `Testwert` und fehlendem `Sonst-Wert` ist der Wert der Zelle entsprechend `FALSCH`.

Im Zusammenhang mit `WENN()` ist es wichtig, die Vergleichsoperationen zu kennen. In OpenOffice.org sehen sie – anders als in vielen Programmiersprachen – wie folgt aus: `=` (gleich) ist wahr, wenn die beiden Operanden gleich sind; `<>` (ungleich) dagegen genau dann, wenn sie verschieden sind. Daneben

stehen < (kleiner als), <= (kleiner oder gleich), > (größer als) und >= (größer oder gleich) zur Verfügung.

Um zu überprüfen, ob eine Zelle leer ist, können Sie sie mit dem leeren String (zwei unmittelbar aufeinanderfolgende Anführungszeichen) vergleichen. Das folgende Beispiel fügt in die aktuelle Zelle den String »voll« ein, falls die Zelle B3 einen Inhalt hat, und ansonsten den String »leer«:

```
=WENN(B3="";"leer";"voll")
```

Noch einfacher funktioniert die Prüfung, ob eine Zelle leer ist, mithilfe der Funktion ISTLEER(Zelle). Entsprechend gibt es beispielsweise ISTZAHL(Wert) und ISTTEXT(Wert), um zu überprüfen, ob der angegebene Wert eine Zahl beziehungsweise ein String ist.

- ▶ Bei der Verwendung von WENN() und ähnlichen Testfunktionen ist es manchmal notwendig, mehrere Vergleiche miteinander zu verknüpfen. UND(Wert[; Wert; ...]) ist nur dann wahr, wenn alle Einzelausdrücke zutreffen. Das folgende Beispiel liefert »bestanden«, wenn die Zellen B2, B4 und B6 jeweils mindestens den Wert 50 enthalten (stellen Sie sich eine Prüfung vor, in der die Teilnehmer in jedem Teilbereich mindestens 50 Punkte erzielen müssen):

```
=WENN(UND(B2>=50;B4>=50;B6>=50);"bestanden";"durchgefallen")
```

- ▶ ODER(Wert[; Wert; ...]) trifft dagegen zu, wenn mindestens einer der Teilausdrücke wahr ist. Das folgende Beispiel kehrt die obige Überprüfung um – sobald auch nur ein Prüfungsergebnis unter 50 Punkten liegt, wird »durchgefallen« angezeigt:

```
=WENN(ODER(B2<50;B4<50;B6<50);"durchgefallen";"bestanden")
```

- ▶ VERWEIS(Testwert;Suchvektor;Ergebnisvektor) sucht den Testwert in den Zellen des Suchvektors, der aus mehreren untereinanderstehenden Werten besteht. Wenn der Testwert gefunden wird, liefert die Funktion den Wert aus der entsprechenden Zeile des Ergebnisvektors. Damit kann VERWEIS() für Zuordnungen bestimmter Werte zu anderen Werten verwendet werden. Deutlicher wird dies durch ein Beispiel: Angenommen, die Zellen B2 bis B7 enthalten die Werte 1 bis 6, während C2 bis C7 die Strings »sehr gut« bis »ungenügend« enthalten – es geht also darum, den numerischen Schulnoten ihre Text-Pendants zuzuordnen. Wenn sich nun beispielsweise in Zelle B9 eine numerische Note befindet, kann die folgende Formel verwendet werden, um die zugehörige Bezeichnung zu finden:

```
=VERWEIS(B9;$B$2:$B$7;$C$2:$C$7)
```

Wie Sie sehen, werden für den Such- und den Ergebnisvektor absolute Bezüge verwendet, um die Formel gefahrlos verschieben zu können. Für `VERWEIS()` und ähnliche Funktionen bieten sich daher benannte Bereiche an.

Wenn der Testwert nicht im Suchvektor vorkommt, erhalten Sie übrigens das Fehler-Ergebnis `#NV` (ungültiger Bezug). Dies können Sie vermeiden, indem Sie den `VERWEIS()`-Aufruf durch ein `WENN()` umschließen, das zunächst den Wertebereich überprüft. Für das obige Beispiel könnte dies beispielsweise wie folgt funktionieren (um es ganz genau zu nehmen, müssten Sie auch noch die Überprüfung `ISTZAHL(B9)` in das `UND()` einfügen):

```
=WENN(UND(B9>=1;B9<=6);VERWEIS(B9;$B$2:$B$7;$C$2:$C$7);"ungültig")
```

7.3 Zusammenfassung

Das freie Office-Paket OpenOffice.org ist für sämtliche Standard-Büroaufgaben gut gerüstet. In Ausstattung, Funktionsumfang und Bedienkomfort kann es gut mit dem verbreiteten Microsoft Office-Paket mithalten. In diesem Kapitel haben Sie einen ersten Einblick in seine beiden wichtigsten Teilprogramme (sowie die in allen Teilprogrammen verfügbare Grafik-Komponente Draw) erhalten.

Writer ist eine moderne Textverarbeitung, mit der Sie durchaus auch typografisch korrekte Layout-Aufgaben erstellen können, die zumindest für Standard-Bürodrucker gut geeignet sind. In diesem Kapitel wurden die wichtigsten Aufgaben der Formatierung und die praktischere Arbeit mit Formatvorlagen vorgestellt. Zusätzlich haben Sie einen Einblick in die Funktionen Suchen und Ersetzen, Tabellen sowie das in allen OpenOffice.org-Teilprogrammen verfügbare Zeichenmodul erhalten.

Mit der Tabellenkalkulation Calc können Sie beliebige Berechnungen in Tabellenform erstellen; anschließend können Sie diese beispielsweise auch zu Diagrammen verarbeiten. Im zweiten Abschnitt dieses Kapitels haben Sie den praktischen Arbeitsablauf mit den wichtigsten Funktionen anhand der Klimatablelle und des zugehörigen Diagramms für Köln kennengelernt; anschließend haben Sie Genaueres über Formeln und Funktionen erfahren.

Der Menschengeist hat keinen Halt, wenn er sich in der Unbegrenztheit gestaltloser Gedanken bewegt; er muss sie zu bestimmten Bildern verdichten, die seiner Welt entnommen sind.

– Michel de Montaigne

8 Desktop-Software

Neben der im vorigen Kapitel vorgestellten Familie von Büroprogrammen gibt es zahlreiche weitere Anwendungen mit grafischer Benutzeroberfläche. Dieses Kapitel stellt einige von ihnen vor:

- ▶ das Bildbearbeitungsprogramm GIMP
- ▶ das Vektorzeichenprogramm Inkscape
- ▶ den Webbrowser Firefox

8.1 Bildbearbeitung und Grafik

Das Erstellen und Bearbeiten von Bildern und Grafiken gehört heute zu den selbstverständlichen Bestandteilen aller Betriebssysteme. Linux ist da natürlich keine Ausnahme. In diesem Abschnitt erfahren Sie nach einem kurzen Einstieg in die theoretischen Hintergründe das Wichtigste über freie Grafiksoftware, Bildbearbeitung mit GIMP sowie Zeichnen mit Inkscape.

8.1.1 Theoretische Grundlagen

Zunächst sollten Sie wissen, dass es zwei grundsätzlich verschiedene Arten von Computergrafik gibt:

- ▶ Die *Pixelgrafik*, auch *Bitmap-Grafik* genannt, speichert Bilder in Form einzelner farbiger Bildpunkte oder Pixel. Bitmaps besitzen eine bestimmte Auflösung, die ihre Darstellungsqualität und ihren Detailreichtum bestimmt. Pixelgrafik wird vor allem für sämtliche Arten digitalisierter Bilder verwendet, also für Scans und Fotos aus Digitalkameras.
- ▶ Bei der *Vektorgrafik* werden die Kurven und Linien durch mathematische Formeln beschrieben und abgespeichert. Das führt dazu, dass Vektorgrafik auflö-

sungsunabhängig ist, weil sich die Kurven und Linien auf jede beliebige Anzahl von Pixeln umrechnen lassen. Diese Art der Grafik ist gut für Zeichnungen oder Diagramme geeignet. Im übernächsten Unterabschnitt wird dazu das Vektorzeichenprogramm Inkscape beschrieben, und bereits im vorigen Kapitel wurden die Vektorgrafikfunktionen von OpenOffice.org angesprochen.

In diesem kurzen Abschnitt werden einige wichtige Grundbegriffe der Grafik erläutert, deren Kenntnis für das Verständnis von Bildbearbeitungs- und Zeichenprogrammen sinnvoll ist. Es geht vor allem um Farben und um die Bildauflösung.

Licht und Farbe

Damit Sie die Funktionsweise von Computergrafik verstehen können, folgt hier als Erstes eine kurze Einführung in die Phänomene Licht und Farbe, auf denen die grafischen Darstellungen basieren, die Sie auf dem Bildschirm und später im Ausdruck sehen.

Elektromagnetische Wellen eines bestimmten Wellenlängenbereichs können vom menschlichen Auge wahrgenommen werden; diese Wellen werden als *Licht* bezeichnet. Je nach Wellenlänge werden diese Lichtwellen als verschiedene Farben gesehen. Die längsten sichtbaren Lichtwellen haben eine Länge von etwa 800 nm (Nanometer; $1 \text{ nm} = 10^{-9} \text{ m}$) und werden als rot wahrgenommen, die kürzesten sind ungefähr 400 nm lang und erscheinen violett. Dazwischen liegt das sogenannte Farbspektrum mit der vom Regenbogen bekannten Farbfolge Rot, Orange, Gelb, Grün, Cyan, Blau und Violett. Wellen, die länger sind als rotes Licht, heißen infrarot, sind sie kürzer als violett, heißen sie ultraviolett. Sehr kurze Wellen sind beispielsweise Röntgen- oder Gammastrahlen; verhältnismäßig lang sind dagegen etwa Radiowellen.

Weißes Licht, wie es etwa die Sonne abstrahlt, ist die gleichmäßige Mischung des gesamten Farbspektrums. Durch ein Prisma lässt sich dieses Licht in die einzelnen Farben des Spektrums zerlegen. Genau dieser Effekt ist für den Regenbogen verantwortlich, denn die Regentropfen dienen dabei als Prisma.

Lichtstrahlen sind übrigens das Einzige, was Sie wirklich sehen können. Wenn Sie einen Gegenstand, eine Person oder ein Bild betrachten, sehen Sie eigentlich nur das Licht, das von ihnen reflektiert wird. Die verschiedenen Farben von Objekten kommen dadurch zustande, dass verschiedene Oberflächen nur einzelne Teile des Farbspektrums reflektieren, andere aber absorbieren.

Wenn Sie also zum Beispiel eine Pflanze als grün wahrnehmen, liegt das daran, dass sie nur grünes Licht reflektiert und alle anderen Farben des Spektrums absorbiert. Genau dasselbe gilt für ein Bild, das auf Papier gedruckt wurde.

Betrachten Sie dagegen ein Bild auf einem Monitor oder Fernseher, dann nehmen Sie direkt die Lichtstrahlen wahr, die das Gerät aussendet.

Diese unterschiedliche Art der Erzeugung von Farben auf dem Monitor und im Druck sorgt bei der Bearbeitung von Grafiken und Bildern für Printproduktionen für gewisse Schwierigkeiten: Es ist so gut wie unmöglich zu erreichen, dass die Farben auf dem Monitor genauso aussehen wie später im Druck. Es ist ein großes Maß an Erfahrung notwendig, um dennoch die richtigen Farbtöne auszuwählen. Moderne Hard- und Software versucht zwar, das Aussehen der Farben über das sogenannte Color-Management geräteunabhängig identisch zu halten, aber letztlich hängt das Ganze von zu vielen verschiedenen Faktoren ab. Eine wichtige Rolle spielt beispielsweise die Beleuchtung im Arbeitsraum. Auch Bauart und Fabrikat des verwendeten Monitors sind entscheidend; darüber hinaus verändern sich die Farbeigenschaften von Monitoren mit zunehmendem Alter.

Die Art und Weise, wie Farben im Computer gespeichert werden, hängt vom jeweiligen Verwendungszweck ab. Konkret orientieren sich die sogenannten Farbmodelle, auf denen die Speicherung der Farben basiert, am Aufbau des Monitorbildes beziehungsweise an der traditionellen Zusammensetzung von Farben für den Druck.

Konkret bestehen Monitorbilder aus kleinen Punkten in drei verschiedenen Farben: Rot, Grün und Blau. Alle drei Einzelpunkte bilden zusammen ein *Pixel*, also einen einzelnen Bildpunkt, der mit einer beliebigen Farbe eingefärbt werden kann. Jede der drei Farbkomponenten kann mit unterschiedlicher Leuchtkraft strahlen. Wenn alle drei ihre volle Leuchtkraft erreichen, ergeben sie zusammen weißes Licht. Sind sie dagegen alle ausgeschaltet, sehen Sie an dieser Stelle die Farbe Schwarz. Dieses Verfahren der Mischung von Lichtfarben wird *Farbaddition* genannt, weil von Schwarz aus immer mehr Farbe hinzuaddiert wird, bis schließlich Weiß entsteht.

Beim Druck geschieht im Grunde genommen das Gegenteil. Die sogenannten Körper- oder Pigmentfarben, die zum Drucken verwendet werden, werden auf weißes Papier aufgetragen. Als Druckfarben werden die drei Komplementärfarben (gegenteilige Farben) der Lichtfarben verwendet: Cyan (die Komplementärfarbe zu Rot), Magenta (zu Grün) und Gelb (zu Blau).

Rein rechnerisch müsste das Übereinanderdrucken aller drei Druckgrundfarben mit voller Intensität Schwarz ergeben. Dass dies nicht der Fall ist, liegt an den physikalischen Eigenschaften der verwendeten Farbstoffe: Sie weichen stets ein wenig von mathematisch korrektem Cyan, Magenta und Gelb ab. Das Übereinanderdrucken ergibt daher kein Schwarz, sondern irgendein dunkles Grau-Braun. Den Effekt können Sie bei einigen besonders billigen älteren Tintenstrahldruckern beobachten, die nur mit den drei Farben Cyan, Magenta und Gelb drucken.

So gut wie alle höherwertigen Drucker und Druckmaschinen benutzen deshalb in der Praxis echtes Schwarz als vierte Druckfarbe, um Text, Konturen und Kontraste klarer darstellen zu können. Auf diese Weise entsteht der bekannte *Vierfarbdruck* (4C).

Die Mischung der Druckfarben auf Papier wird als *Farbsubtraktion* bezeichnet, weil die Helligkeit durch das Auftragen von Farbe immer weiter nachlässt.

Für die Speicherung von Farbbildern im Computer entstehen aus diesen Vorgaben zwei verschiedene Farbmodelle: RGB und CMYK.

Für die Bildschirmdarstellung werden Farben im *RGB*-Modus gespeichert, also als Mischung von Rot, Grün und Blau mit unterschiedlicher Intensität. Wird für alle drei Farben der Wert 0 gespeichert, ergibt sich Schwarz; wenn dagegen alle mit ihrem maximalen Wert gespeichert werden, entsteht Weiß. Wie hoch dieser maximale Wert konkret ist, hängt von der verwendeten Farbtiefe ab.

Allgemein gilt, dass derselbe Anteil aller drei Farben Grau ergibt. Je stärker die Werte für die drei Komponenten voneinander abweichen, desto höher ist die *Sättigung* der resultierenden Farbe – sie erscheint »bunter«.

Die Farbcodierung für den Vierfarbdruck ist der *CMYK*-Modus, die Mischung aus den vier Komponenten Cyan, Magenta, Gelb (Yellow) und Schwarz (Black; da B für Blau steht, wird hier der letzte Buchstabe verwendet). Das Farbmodell ist ein wenig komplexer als das *RGB*-Modell, da es durch die zusätzliche Verwendung von Schwarz häufig mehrere Möglichkeiten gibt, eine bestimmte Farbe darzustellen. Wenn bei diesem Farbmodell alle vier Farben den Wert 0 aufweisen, entsteht Weiß. Wenn Cyan, Magenta und Gelb mit ihrem Maximalwert gespeichert werden, ergibt dies rechnerisch Schwarz; der höchste Wert im Schwarzkanal bedeutet natürlich ebenfalls Schwarz.

Neben *RGB* und *CMYK*, die die Besonderheiten des Monitorbildes beziehungsweise des Ausdrucks berücksichtigen, sind noch zwei weitere Farbmodelle gebräuchlich:

- ▶ Beim *HSB*-Modell (Hue, Saturation and Brightness) wird eine Farbe durch die drei Informationen Farbton (Hue), Farbsättigung (Saturation) und Helligkeit (Brightness) dargestellt. Der Farbton wird als Winkel eines Kreises angegeben, in dem das gesamte Farbspektrum abgebildet ist. Die Sättigung entscheidet, wie blass oder kräftig eine Farbe erscheint; bei einem Sättigungsgrad von 0 ist sie grau. Die Helligkeit schließlich wird in Prozent angegeben: 0 % ist Schwarz und 100 % Weiß.
- ▶ Das *Lab*-Modell ist das umfangreichste Farbmodell, *RGB* und *CMYK* sind Teilmengen davon. Bei *Lab* werden zwei Achsen durch den Farbkreis gezogen.

Die Achse a verläuft von Rot nach Grün, während b von Blau nach Gelb reicht. Die Farbe wird auf dem Farbkreis durch den Schnittpunkt dieser Koordinaten dargestellt. Die Helligkeit (L), der dritte Wert, funktioniert wie gehabt. Der größte Vorteil von Lab ist die Geräteunabhängigkeit, jede Farbe wird exakt und eindeutig beschrieben.

Die Vorstufe des Lab-Modells wurde bereits 1931 von der Commission Internationale d'Eclairage (CIE), einem internationalen Gremium, standardisiert. Es entspricht dem Farbempfinden der meisten Menschen und wird daher als Referenzmodell für Farbkorrekturen eingesetzt.

Ein weiteres Merkmal, das die für ein digitales Bild gespeicherten Farben betrifft, ist die *Farbtiefe*. Sie besagt, welche Datenmenge für ein einzelnes Pixel gespeichert wird, bestimmt also die Anzahl unterschiedlicher Farben im Bild.

Für RGB-Bilder wird üblicherweise eine Farbtiefe von 24 Bit verwendet: Pro Grundfarbe werden 8 Bit gespeichert, es gibt also je 256 verschiedene Abstufungen von Rot, Grün und Blau. Damit lassen sich über 16,7 Millionen unterschiedliche Farben darstellen. Wenn 32 Bit zur Verfügung stehen, kann zusätzlich zu den drei Farbwerten auch noch ein *Alphakanal* verwendet werden, der den Transparenzgrad für jedes einzelne Pixel angibt. Bei CMYK-Bildern werden 32 Bit für das eigentliche Bild verwendet, weil jede der vier Grundfarben 8 Bit benötigt. Wirklich genutzt wird diese Datentiefe in der Praxis allerdings nicht, weil die CMYK-Anteile üblicherweise in Prozent und nicht als 8-Bit-Werte zwischen 0 und 255 angegeben werden.

Einige Bilddateiformate beschränken übrigens die Farbtiefe für das gesamte Bild auf deutlich geringere Bit-Breiten. Es gibt beispielsweise Bilder mit nur 16 Bit Farbtiefe (insgesamt 65.536 Farben), wodurch weniger verschiedene Mischungen der jeweiligen Grundfarben möglich sind.

Ein etwas anderer Weg wird bei Bildern beschritten, die insgesamt nur eine Farbtiefe von 8 Bit besitzen: Sie sind jeweils mit einer eigenen Farbpalette oder -tabelle ausgestattet, die die RGB-Werte der verwendeten Farben definiert. Ein solches Bild enthält also maximal 256 Farben, aber sie können beliebig gewählt werden. Farben aus einer durchnummerierten Palette werden als *indizierte Farben* bezeichnet.

Ob die verwendete Farbtiefe überhaupt auf dem Bildschirm angezeigt werden kann, hängt von der Ausstattung der Grafikkarte und vom entsprechenden Treiber ab. Ältere Grafikkarten können möglicherweise nur eine Farbtiefe von 16 Bit (65.536 Farben) oder gar nur 8 Bit (256 Farben) darstellen. Wenn nun ein Bild mit mehr Farben angezeigt werden soll, muss das Betriebssystem nicht vorhan-

dene Farben durch ihre nächsten verfügbaren Nachbarn ersetzen – gerade bei nur 8 Bit Farbtiefe sieht das Ergebnis wenig realistisch aus.

Erfreulicherweise ist praktisch jeder PC oder Mac, der in den letzten zehn Jahren gebaut wurde, in der Lage, 16,7 Millionen Farben (24 Bit RGB) anzuzeigen. Wenn eine Grafikkarte 24 oder gar 32 Bit Farbtiefe besitzt, wird dies als True Color bezeichnet, weil sie dadurch jede Farbe einer Bilddatei ohne Umrechnung darstellen kann.

Bildgröße und Auflösung

Zu den Merkmalen, die die Qualität eines Pixelbildes kennzeichnen, gehört neben der Farbtiefe auch die *Bildauflösung*. Sie bestimmt, wie viele Pixel pro Längeneinheit verwendet werden, also wie detailreich das Bild dargestellt wird. Die angegebene Längeneinheit ist meist Inch (Zoll), also 2,54 cm, manchmal aber auch Zentimeter.

Die Druckauflösung

Im Druck und bei der Bildschirmdarstellung wird die Einheit für die Auflösung ein wenig unterschiedlich bezeichnet: Auf dem Bildschirm beziehungsweise bei der Bildbearbeitung spricht man von Pixeln pro Inch (ppi) beziehungsweise Pixeln pro Zentimeter, bei der Ausgabe wird dagegen die Maßeinheit Punkte pro Inch (dpi – dots per inch) beziehungsweise Punkte pro Zentimeter eingesetzt. Der Grund dafür ist einfach: Während Pixel absolut quadratisch sind, sind Druckpunkte je nach Druckverfahren rund oder oval.

Darüber hinaus muss beim Drucken ein weiterer Wert berücksichtigt werden: die sogenannte *Rasterweite*. Druckmaschinen, Tintenstrahl- oder Laserdrucker sind in Wirklichkeit gar nicht in der Lage, Farbe mit unterschiedlicher Intensität zu Papier zu bringen. Stattdessen drucken sie unterschiedlich große Punkte, deren Verteilung für die unterschiedlichen Helligkeitswerte sorgt.

Damit aus den vier Druckfarben die gewünschten Mischungen entstehen können, werden ihre jeweiligen Punkte nicht genau übereinander gedruckt, sondern in unterschiedlichen Winkeln. Verbreitet ist die folgende Rasterwinkelverteilung: Cyan 15°, Magenta 75°, Gelb 0° und Schwarz 45°. Dies verhindert darüber hinaus die Entstehung eines Moiré-Effekts, bei dem die unpassende Überlagerung von Linien zu Darstellungsfehlern führt.

Die Rasterweite gibt den Abstand zwischen den Rasterlinien an. Sie wird in Linien pro Zentimeter (l/cm) oder Linien pro Inch (lpi) angegeben. Da neben der unterschiedlichen Größe von Rasterpunkten auch deren Anzahl innerhalb eines

Bereiches für unterschiedliche Helligkeitsabstufungen sorgen kann, werden zum Erreichen einer bestimmten Auflösung weniger Rasterlinien benötigt als erwartet. Beispielsweise verwendet der Standard-Offsetdruck 60 lpi zur Darstellung der Druckauflösung von 300 dpi. Im deutschen Sprachgebrauch bedeutet ein »60er Raster« allerdings in der Regel einen Wert von 60 l/cm – dies entspricht gut 150 lpi. Eine solche Druckqualität wird für höherwertige Printprodukte auf gestrichenem (versiegeltem, glänzendem) Papier verwendet.

Beachten Sie, dass sich die Auflösung für Printproduktionen stets auf die letztendliche Ausgabegröße auf dem Papier bezieht. Die Belichtungsfilm, die beim Offsetdruck einen wichtigen Zwischenschritt bilden, benötigen eine höhere Auflösung als der Ausdruck auf Papier, weil das Papier erheblich größer ist als der Film.

Je nach Bildart und Verwendungszweck werden unterschiedliche Druckauflösungen verwendet:

- ▶ Für den Standard-Vierfarbdruck gemischter Text-Bild-Vorlagen wird üblicherweise eine Druckauflösung von 300 dpi verwendet. Entsprechend werden im Bildbearbeitungsprogramm 300 ppi eingestellt, um solche Bilder vorzubereiten.
- ▶ Hochwertige Bilddrucke in Fotoqualität, beispielsweise für Hochglanzmagazine oder Bildbände, benötigen Druckauflösungen ab 600 dpi.
- ▶ Schwarzweiß- beziehungsweise Graustufendrucke, insbesondere von Strichzeichnungen, werden manchmal in noch höheren Auflösungen gedruckt, etwa mit 1.200 dpi.

Ein Problem mit der Bildschirmauflösung ergibt sich dadurch, dass Sie ein und denselben Bildschirm auf verschiedene Auflösungen einstellen können. Die meisten aktuellen Grafikkarten und Monitore unterstützen mindestens die folgenden Einstellungen:

- ▶ 640 × 480 Pixel
- ▶ 800 × 600 Pixel
- ▶ 1.024 × 768 Pixel
- ▶ 1.280 × 1.024 Pixel

Auch diese Werte sind nicht universell gültig, weil es beispielsweise einige Notebook-Monitore oder spezielle Flatscreens mit anderen Proportionen gibt, die statt der hier aufgeführten 4:3-Verhältnisse 16:10 oder gar 16:9 unterstützen. Abgesehen davon werden bei 19-Zoll-Monitoren oder größeren Geräten noch höhere Auflösungen eingesetzt.

Es ist unmöglich, für Monitore eine bestimmte Auflösung pro Längeneinheit anzugeben. Andererseits benötigen Bilddateien aus formalen Gründen einen solchen Wert. Deshalb wird für die Bildschirmdarstellung von Bildern eine rechnerische Auflösung von 72 ppi verwendet. Mit der Realität muss dieser Wert nichts zu tun haben. Die tatsächliche Auflösung ist völlig unterschiedlich, wenn beispielsweise auf einem 15- und auf einem 19-Zoll-Monitor jeweils 1.024×768 Pixel eingestellt werden.

Für die Wahl des Wertes von 72 ppi gibt es dennoch einen guten Grund: Da 72 pt (DTP-Punkt) genau 1 Zoll (Inch) betragen, sorgt diese Auflösung dafür, dass 1 Punkt einer Schriftgröße genau einem Pixel in einem Bild entspricht.

Abgesehen davon müssen Sie beim Bearbeiten von Bildern für den reinen Bildschirmeinsatz, also für Webseiten oder Multimedia-Produktionen, daran denken, dass die einzige verlässliche Maßeinheit das Pixel ist, während die Bilder für den DTP-Einsatz oft in Zentimetern oder Zoll gemessen werden, weil die absolute Ausgabegröße bekannt ist.

Eine Besonderheit müssen Sie übrigens noch beachten, wenn Sie Bilder mit einem Scanner digitalisieren: Es genügt nicht, ein Bild mit der gewünschten Bildschirm- oder Druckauflösung einzuscannen. Zum einen müssen Sie eventuell einen Skalierungsfaktor beachten: Ist das Bild beispielsweise 30×40 mm groß, soll aber mit einer Größe von 60×80 mm und einer Auflösung von 300 dpi gedruckt werden, dann müsste das Original schon rein rechnerisch mit 600 dpi gescannt werden, um die doppelte Größe bei gleichbleibender Qualität zu erreichen.

Aber auch dies genügt noch nicht: Da ein Bildpunkt mit einer bestimmten Farbe im ungünstigsten Fall genau zwischen zwei Scanlinien liegen könnte, wird in der Regel noch einmal die doppelte Auflösung verwendet, um diesen Fall auszuschließen und alle Farben und Details des ursprünglichen Bildes berücksichtigen zu können. Diese Verdoppelung der gewünschten Druckauflösung wird als *Qualitätsfaktor* bezeichnet. Ein Qualitätsfaktor von 1,5 ist gerade noch ausreichend; 2 ist optimal. Ein noch höherer Faktor bringt keine weitere Verbesserung mehr, sondern ermöglicht es Ihnen höchstens, das Bild später in einer größeren Ausgabegröße einzusetzen als ursprünglich geplant.

Der Qualitätsfaktor basiert auf dem sogenannten *Shannon-* oder *Nyquist-Theorem* (benannt nach seinen beiden voneinander unabhängigen Entdeckern). Dieser Satz besagt, dass für die Digitalisierung von Daten einer bestimmten Frequenz mindestens die doppelte Sampling-Rate erforderlich ist, weil in den »Austastlü-

cken« andernfalls Datenverlust und -verfälschungen drohen. Dies spielt übrigens auch in der Sound-Digitalisierung eine wichtige Rolle.

Letzten Endes ergibt sich also für die Scan-Auflösung die folgende Formel:

$$\text{Vorlagengröße} \times \text{Skalierungsfaktor} \times \text{Qualitätsfaktor} (2)$$

Angenommen, ein ursprünglich 15 × 20 mm großes Bild soll mit einer Breite von 40 mm und der Druckauflösung 300 dpi ausgegeben werden. Der Skalierungsfaktor beträgt in diesem Fall 40:15, also 2,67. Die benötigte Scan-Auflösung ist somit folgende:

$$300 \times 2,67 \times 2 \approx 1600$$

Datenkomprimierung

Eine bei Binärdateien häufig verwendete Option ist die *Datenkomprimierung*. Es handelt sich um mathematische Verfahren, die für eine Verringerung der Datenmenge sorgen sollen. Dies ist nützlich, weil komprimierte Daten weniger Platz auf Datenträgern belegen und schneller über das Internet übertragen werden können. Grundsätzlich lassen sich zwei Kompressionsverfahren voneinander unterscheiden:

- ▶ Die *verlustfreie Komprimierung* rechnet die vorhandenen Daten durch geeignete mathematische Verfahren um, so dass sie weniger Speicherplatz beanspruchen, aber wieder genau in ihren ursprünglichen Zustand zurückgerechnet werden können. Verlustfreie Verfahren werden beispielsweise für Archivformate wie ZIP oder Stuffit verwendet, aber auch für manche Bilddateien.

Das einfachste verlustfreie Kompressionsverfahren ist RLE (Run-Length Encoding oder zu Deutsch Lauflängencodierung). Bei dieser Methode werden aufeinanderfolgende identische Bytes durch ihren Wert und die Anzahl ihres Vorkommens abgekürzt.

- ▶ Eine Weiterentwicklung von RLE stellt das LZW-Verfahren dar, benannt nach seinen Entwicklern Lempel, Ziv und Welch. Es wird unter anderem zur Komprimierung der Bildformate GIF und TIFF verwendet und funktioniert folgendermaßen: In der Datei wird nach wiederkehrenden Datenmustern gesucht. Diese Muster werden durchnummeriert und nur einmal abgespeichert. An den entsprechenden Stellen, an denen sie eigentlich vorkommen sollten, steht nur noch ein Verweis auf die entsprechende Nummer und gegebenenfalls wiederum eine Anzahl.

Wie man sich leicht vorstellen kann, sind RLE und LZW besonders effizient, wenn ein Bild aus großen, einfarbigen Flächen besteht.

- ▶ Die grundsätzlich andere Methode ist die *verlustbehaftete Komprimierung*. Sie reduziert die Datenmenge, indem sie tatsächliche Daten aus der ursprünglichen Datei weglässt. Einfache verlustbehaftete Kompressionsmethoden wie die alte ADPCM-Komprimierung für Sound reduzieren die Datenmenge einfach ohne Unterschied durch Mittelwertbildung. Derartige Datenverluste fallen natürlich oft als extrem störend auf.

Moderne verlustbehaftete Kompressionsverfahren gehen dagegen von der Frage aus, auf welche Teile der Daten man am ehesten verzichten kann, ohne den Verlust allzu sehr zu bemerken. Dies führt zu Entwicklungen wie JPEG für die Komprimierung von Fotos, bei dem die meisten Farbtöne verworfen werden, weil Helligkeitsunterschiede stärker wahrgenommen werden. Ein anderes Beispiel ist MP3 für die Audio-komprimierung, das bevorzugt diejenigen Töne herausfiltert, die die meisten Menschen nicht oder nur unterschwellig hören.

Bilddateiformate

In diesem Abschnitt finden Sie kurze Beschreibungen der wichtigsten Bilddateiformate. Zu Beginn zeigt Tabelle 8.1 eine kurze Übersicht über die Fähigkeiten der Formate; weiter unten im Text werden sie dann näher erläutert.

Format	Farbtiefe	Farbmodi	Komprimierung	Weitere Daten
GIMP (.xcf)	beliebig	alle	—	alle, die GIMP bietet
Photoshop (.psd)	beliebig	alle	eigene, verlustfrei (optional)	alle, die Photoshop bietet (auch fast alle GIMP-Daten)
Photoshop 2.x (.psd)	beliebig	alle	—	keine Ebenen!
TIFF (.tif)	beliebig	alle	LZW, verlustfrei	Alphakanäle, Pfade
Encapsulated PostScript (.eps)	beliebig	alle	—	Vektordaten, Text, Schriften
JPEG (.jpg)	bis 24 Bit	RGB, CMYK, Graustufen	eigene, mit Verlust	Pfade
GIF (.gif)	8 Bit	indizierte Farben, Graustufen	LZW, verlustfrei	absolute Transparenz, Animation
PNG (.png)	8, 24 oder 32 Bit	indizierte Farben, RGB	eigene, verlustfrei	bei 32 Bit Alphakanal
BMP (.bmp)	beliebig	RGB, Graustufen, indizierte Farben	RLE (ähnlich LZW)	Alphakanäle
PICT (.pct)	beliebig	RGB, Graustufen, indizierte Farben	JPEG möglich	Vektordaten

Tabelle 8.1 Übersicht über wichtige Bilddateiformate

Über einige Bilddateiformate sollten Sie Näheres wissen, da sie häufig eingesetzt werden:

► *GIMP (XCF) und Photoshop (PSD)*

Der große Vorteil der Arbeitsdateiformate GIMP XCF und Photoshop PSD besteht darin, dass diese Formate sämtliche Daten von GIMP beziehungsweise der beliebten (kommerziellen) Bildbearbeitungssoftware Adobe Photoshop speichern können. Wenn Sie mit GIMP arbeiten, sollten Sie stets eine Arbeitskopie Ihrer bearbeiteten Bilder im XCF- oder Photoshop-Format behalten, weil kein anderes Dateiformat *alle* Bilddaten mitspeichert.

Der einzige Nachteil dieser Formate (neben ihrer recht stolzen Dateigröße) ist die kaum vorhandene Unterstützung durch andere Programme. Das Einzige, was manchmal möglich ist, ist ein – oft nur rudimentärer – Import in andere Bildbearbeitungsprogramme, speziell von PSD-Dateien.

XCF- und PSD-Dateien können ansonsten weder als Teil von Layouts an Druckereien weitergegeben noch auf Webseiten dargestellt werden. Sie benötigen stets Dateien in anderen Formaten für diese Einsatzzwecke.

Die spezielle Variante Photoshop 2.0, die von einigen älteren Fremdanwendungen als Importformat eingesetzt werden kann, unterstützt nicht einmal die heutzutage in jeder Bildbearbeitung fundamentalen Ebenen.

► *TIFF*

Das *Tagged Image File Format* (TIFF; Dateiendung *.tif*) ist das Standardformat für die Einbettung hochauflösender Pixelbilder in Layoutdokumente in der Druckvorstufe. Der Name bedeutet etwa »Bilddateiformat mit Marken«, wobei diese Marken (Tags) für spezielle Kennzeichnungen von Bildteilen und -inhalten stehen.

TIFF besitzt eine Reihe bedeutender Besonderheiten:

- Eine TIFF-Datei kann beliebig viele Farbkanäle enthalten. Diese können entweder für den Vierfarbdruck und zusätzliche Sonderfarben verwendet werden oder dienen in bestimmten Anwendungen als Alphakanäle, die die Transparenz bestimmen.
- In TIFF-Dateien können Sie neben den normalen Pixeldaten auch Vektorpfade speichern. Besonders wichtig sind in diesem Zusammenhang die Beschneidungspfade, die transparente Bereiche für Layoutprogramme definieren.
- Die LZW-Komprimierung ist optional. Sie muss nicht verwendet werden; einige ältere Programme sind dazu inkompatibel. Normalerweise ist LZW auf 8 Bit (256 Farben) beschränkt. Da TIFF die Farbkanäle einzeln speichert, gilt diese Beschränkung hier nur pro Kanal, so dass insgesamt eine

beliebige Farbtiefe möglich ist. In neueren TIFF-Varianten können Sie statt der LZW-Komprimierung auch eine ZIP- oder JPEG-Komprimierung wählen; ZIP komprimiert etwas stärker als LZW, während die verlustbehaftete JPEG-Komprimierung speziell für Fotos geeignet ist.

► *Die »Internet-Formate« GIF, JPEG und PNG*

Das *GIF*-Format (Graphics Interchange Format) wurde seit 1987 im Auftrag des damals verbreitetsten Online-Dienstes CompuServe entwickelt, war also von Anfang an als spezielles Online-Bildformat gedacht.

Ein GIF unterstützt maximal eine Farbtiefe von 8 Bit, das heißt, es zeigt höchstens 256 Farben aus einer angepassten Farbpalette an. Für Fotos und andere halbtonreiche Bilder mit Farbverläufen oder Ähnlichem ist es daher in der Regel nicht geeignet.

Andererseits wird ein GIF verlustfrei komprimiert; es verwendet auf jeden Fall die LZW-Komprimierung, die beim TIFF-Format optional zur Verfügung steht. Da bei dieser Kompressionsmethode nebeneinanderliegende, farbgleiche Pixel beziehungsweise wiederkehrende Muster zu einer Einheit zusammengefasst werden, wird ein solches Bild umso stärker komprimiert, je weniger Farben es enthält.

Ein weiteres unterstütztes Merkmal von GIF-Dateien ist die *Transparenz*. Die Farbtabelle eines Bildes kann den speziellen Eintrag »transparent« enthalten, um die Anzeige einzelner Pixel zu unterdrücken. Allerdings ist keine echte Alpha-Transparenz möglich; Pixel werden entweder mit voller Deckkraft angezeigt oder gar nicht.

Angenehm ist auch die Tatsache, dass ein GIF *interlaced* abgespeichert werden kann: In diesem Fall wird es beim Laden nicht zeilenweise von oben nach unten aufgebaut, sondern die Zeilen werden versetzt geladen. Gerade bei langsameren Internetverbindungen hat dies den Vorteil, dass Bilder schon während des Ladevorgangs im Ganzen erkennbar werden.

Es handelt sich bei GIF übrigens nicht einfach um ein Dateiformat für ein einzelnes Bild, sondern um ein Hüllformat für mehrere Bilder innerhalb derselben Datei. Der Header einer GIF-Datei kann Anweisungen enthalten, die die Anzeigedauer der einzelnen Bilder bestimmen. Dies ergibt die bekannten animierten GIFs, die beispielsweise häufig als Werbebanner auf Webseiten eingesetzt werden.

Das Dateiformat *JPEG* ist nach der *Joint Photographic Expert Group* benannt, einer Expertenkommission, die sich Anfang der 90er Jahre über die effiziente Komprimierung von Fotos Gedanken machte. Das Dateiformat selbst heißt eigentlich *JFIF* (JPEG File Interchange Format), benutzt aber üblicherweise die Dateiendung *.jpg*.

JPEG-Bilder unterstützen eine beliebige Farbtiefe von bis zu 8 Bits pro Kanal (bei den für das Web verwendeten RGB-Bildern also 24 Bits oder mehr als 16,7 Millionen Farben).

Die Komprimierung von Bildern dieses Dateiformats erfolgt stets verlustbehaftet. Sie basiert auf der farbphysiologischen Erkenntnis, dass Helligkeitsunterschiede erheblich stärker vom menschlichen Auge wahrgenommen werden als Farbtonschwankungen. Aus diesem Grund wird der Helligkeitswert jedes einzelnen Pixels gespeichert, während jeweils nur der Durchschnitt der Farbtöne aus einem quadratischen Bereich gespeichert wird. Die Größe dieses Quadrats ist der wichtigste Gesichtspunkt der einstellbaren JPEG-Qualität. Bei starker Vergrößerung sind diese Quadrate als »Schachbrettmuster« zu erkennen. Sie können bei JPEG-Bildern den Kompressionsfaktor frei wählen; je nach Anwendung werden 10 bis 100 verschiedene Stufen angeboten. Je stärker die Komprimierung, desto kleiner wird die resultierende Datei, aber es geht auch mehr Information verloren.

Besonders gut geeignet ist das JPEG-Format für Fotos und andere halbtone reiche Bilder. Dagegen eignet es sich überhaupt nicht für Illustrationen mit größeren gleichfarbigen Flächen, für die das GIF-Format oder eventuell PNG gewählt werden sollte.

Ähnlich wie GIF-Bilder interlaced abgespeichert werden können, besteht bei JPEG die Möglichkeit, das sogenannte Progressive-JPEG-Format auszuwählen. Es geht sogar noch einen Schritt weiter als Interlaced-GIF, indem es ein JPEG-Bild nicht nur zeilenweise, sondern in beide Richtungen versetzt lädt. Der einzige Nachteil dieser Variante besteht darin, dass ganz alte Browser (zum Beispiel die 3er-Versionen von Netscape und Internet Explorer) sie nicht unterstützen und stattdessen gar kein Bild anzeigen.

Bitte beachten Sie, dass Sie ein Bild, das bereits als JPEG abgespeichert wurde, nicht noch einmal in diesem Format speichern sollten – bei jedem JPEG-Kompressionsvorgang kommt es erneut zum Verlust von Bilddetails. Sie sollten also stets eine Arbeitskopie in einem verlustfreien Format wie Fireworks PNG, Photoshop PSD oder TIFF behalten.

Einer der größten Nachteile von JPEG-Dateien besteht darin, dass sie keine Transparenz unterstützen. Bilder in diesem Format sind auf jeden Fall rechteckig. Dies ist der einzige Grund, warum Sie manchmal gezwungen sind, Fotos im eigentlich völlig ungeeigneten GIF-Format zu speichern. Abgesehen davon ist JPEG ein Dateiformat für genau ein Bild und unterstützt daher keine Animation wie das GIF-Format.

Das PNG-Format (Portable Network Graphics; gesprochen »PING«) wurde als möglicher Nachfolger von GIF entwickelt – besonders, weil die freie Verwen-

derung von GIF bis Juli 2004 durch ein Patent auf das LZW-Verfahren bedroht wurde. PNG vereint in gewisser Weise die besten Eigenschaften von GIF und JPEG: Es komprimiert ohne Verlust, kann jedoch auch erheblich mehr als GIF. Während die 8-Bit-Variante genau dieselben Optionen bietet wie ein entsprechendes GIF und dabei auch noch etwas mehr Speicherplatz belegt, ist das 24-Bit-PNG aus einem wichtigen Grund sehr interessant: Als einziges Web-Dateiformat unterstützt es echte Alpha-Transparenz, das heißt, die Deckkraft eines Pixels kann in 256 Stufen von voll deckend bis absolut durchsichtig frei gewählt werden. Dies ermöglicht beispielsweise geglättete Ränder auf beliebigen Hintergründen und teiltransparente Bildteile wie Autofenster oder Brillengläser.

8.1.2 Bildbearbeitung mit GIMP

The GIMP (im Folgenden kurz GIMP), das GNU Image Manipulation Program, ist eine seit zehn Jahren intensiv entwickelte und daher weit fortgeschrittene Bildbearbeitungssoftware. Ihr Leistungsumfang lässt sich durchaus mit der kommerziellen Lösung Adobe Photoshop vergleichen (mit einigen Abstrichen im Bereich der CMYK-Farbseparation).

Um GIMP zu starten, wählen Sie einfach GRAFIK • BILDBEARBEITUNG • THE GIMP aus dem K-MENÜ beziehungsweise GNOME-Anwendungsmenü. Wenn Sie die Software noch nicht installiert haben: Sie befindet sich unter anderem in der YaST-Selektion GNOME-SYSTEM, da sie zum GNOME-Projekt gehört (historisch betrachtet ist es eigentlich umgekehrt: GNOME basiert auf dem für GIMP geschriebenen Toolkit GtK).

Nach dem Start präsentiert GIMP zunächst nur ein kleines Fenster mit einem kurzen Menü (drei Haupteinträge), einer Werkzeugpalette und Optionen für das jeweils ausgewählte Werkzeug – einen Screenshot sehen Sie weiter unten in »Die Werkzeugpalette«. Erst wenn Sie ein neues Bild erstellen oder eine vorhandene Bilddatei öffnen, werden in diesem Fenster zahlreiche weitere Menüpunkte angezeigt. Zusätzlich erscheint der GIMP Tipp des Tages, der manchmal nützliche Kurzbefehle und Arbeitshinweise offenbart, so dass Sie ihn bei den ersten paar GIMP-Sitzungen durchaus offen lassen sollten. Später können Sie per Kontrollkästchen bestimmen, dass er nicht mehr mitgestartet werden soll; in diesem Fall lässt er sich über HILFE • TIPP DES TAGES zurückholen.

Dateien erstellen, öffnen und speichern

Im Menü DATEI können Sie als Erstes vor allem eine neue Datei erstellen oder eine vorhandene öffnen. In Abbildung 8.1 wird der Dialog DATEI • NEU (oder

kurz (**Strg** + **N**) gezeigt. Unter VORLAGEN können Sie sich diverse voreingestellte Bildgrößen aussuchen. Alternativ lassen sich Breite und Höhe auch manuell einstellen – zur Auswahl stehen die Maßeinheiten PIXEL, ZOLL, MILLIMETER, PUNKT (DTP-Punkt, also Zoll) und PICA (12 Punkt); die Option MEHR bietet noch einige weitere Einheiten.

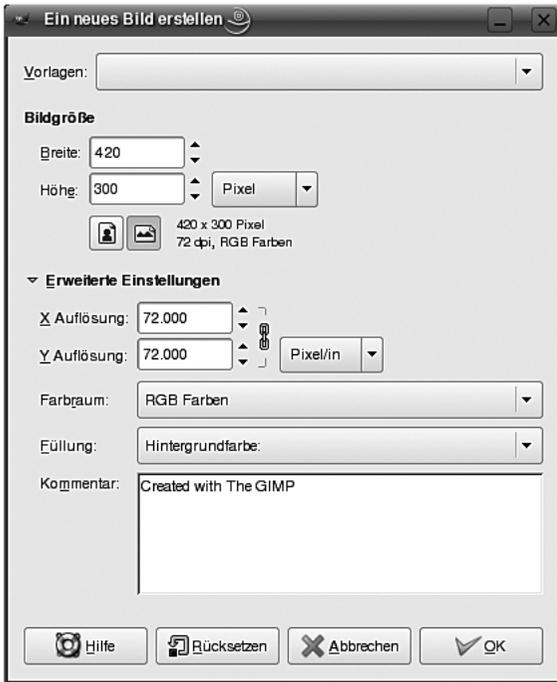


Abbildung 8.1 Der GIMP-Dialog zum Erstellen neuer Bilder

Alle Auswahlmöglichkeiten außer Pixel werden unmittelbar von der AUFLÖSUNG unter ERWEITERTE EINSTELLUNGEN beeinflusst. Standardmäßig werden X- und Y-Auflösung gleichzeitig geändert (Kettensymbol); nur in Ausnahmefällen ist es notwendig, sie unterschiedlich einzustellen. Für Bildschirmbilder (Screen- und Webdesign und so weiter) werden in aller Regel 72 PIXEL/IN ausgewählt; die Gründe wurden im obigen Theorie-Abschnitt erläutert. Für einen einfachen Tintenstrahl- oder Laserausdruck benötigen Sie meist 150 PIXEL/IN; qualitativ hochwertiger Fotodruck sollte dagegen mit mindestens 300 PIXEL/IN durchgeführt werden.

Schließlich können Sie noch den FARBRAUM des Bildes (RGB oder Graustufen) sowie unter FÜLLUNG eine der Einstellungen aktuelle VORDERGRUNDFARBE, HINTERGRUNDFARBE, WEISS oder TRANSPARENZ auswählen. Beachten Sie, dass Letzte-

res das Verhalten der untersten Ebene betrifft; alle zusätzlichen Bildebenen sind beim Erstellen zunächst ohnehin transparent.

DATEI • ÖFFNEN (**[Strg] + [O]**) ermöglicht das Laden eines vorhandenen Bildes von der Festplatte; abgesehen von einem Dateitypen-Filter rechts unter der Verzeichnisanzeige entspricht dieser Dialog anderen Datei-Öffnen-Dialogen.

Mit DATEI • HOLEN können Sie ein Bild aus verschiedenen Quellen importieren:

- ▶ ALS NEUES BILD EINFÜGEN erstellt ein neues Bild in der Größe des aktuellen Inhalts der Zwischenablage und fügt diesen ein.
- ▶ CAPTURE FROM CAMERA ermöglicht Ihnen den direkten Zugriff auf den Speicherchip einer Digitalkamera zum Aussuchen eines einzelnen Bildes.
- ▶ LOAD FROM CAMERA liest den gesamten Inhalt des Kamera-Speicherchips aus und ermöglicht Ihnen anschließend die Auswahl.
- ▶ SCREEN SHOT bietet einige Optionen zum Erstellen eines Screenshots (Bildschirmfoto); der betreffende Dialog wird in Abbildung 8.2 gezeigt: Wählen Sie je nach Bedarf EINZELNES FENSTER oder GANZER BILDSCHIRM; zusätzlich können Sie die Verzögerung einstellen, nach der das zurzeit ausgewählte Fenster beziehungsweise der aktuelle Zustand des Bildschirms festgehalten und als neues Bild in GIMP geöffnet wird. Wenn Sie sich für den Fenster-Modus entschieden haben, verwandelt sich der Cursor nach der angegebenen Sekundenzahl in ein Fadenkreuz, und Sie können das gewünschte Fenster anklicken (die Titelleiste genügt). Achten Sie darauf, dass es sich komplett im Vordergrund befindet, andernfalls bleiben die verdeckten Teile einfach schwarz.¹



Abbildung 8.2 Der »Screen Shot«-Dialog in GIMP

¹ Oder hatten Sie gedacht, X Window oder der Window-Manager machen sich die Mühe, Inhalte zu zeichnen, die sowieso nicht sichtbar sind? ;-)

- ▶ XSANE öffnet das gleichnamige Scan-Programm (siehe Kapitel 6, »Wichtige Hardware installieren«) und öffnet das eingescannte Bild automatisch in GIMP.

Über DATEI • ZULETZT GEÖFFNET können Sie die Bilder laden, die Sie als Letztes bearbeitet haben. Dafür stehen auch die Tastenkürzel **[Strg]+[1]**, **[Strg]+[2]** und so weiter bereit.

Das Speichern von Dateien erfolgt nicht mehr im Hauptfenster, sondern im Datei-Menü des jeweiligen Dokumentfensters. Wählen Sie dort DATEI • SPEICHERN UNTER (**[Strg]+[U]+[S]**), zum erneuten Speichern auch einfach DATEI • SPEICHERN (**[Strg]+[S]**). Im Speichern-unter-Dialog (siehe Abbildung 8.3) wird unter NAME der Dateiname eingegeben; IN ORDNER SPEICHERN ermöglicht die Auswahl des Verzeichnisses (voreingestellt ist praktischerweise Ihr Home-Verzeichnis). Wenn Sie sich den Speicherort interaktiv aussuchen möchten, können Sie auch den Ordner-Browser aufklappen. Der DATEITYP wird normalerweise NACH ENDUNG ausgewählt; bei Bedarf kann aber auch dieser Fensterteil aufgeklappt werden, um den Typ aus einer Liste zu wählen. Neben den oben diskutierten Standarddateiformaten stehen auch zahlreiche Exoten zur Verfügung; eher witzig als seriös ist zudem der Dateityp »ASCII Kunst«, der versucht, Ihr Bild möglichst originalgetreu in ASCII-Art umzusetzen.

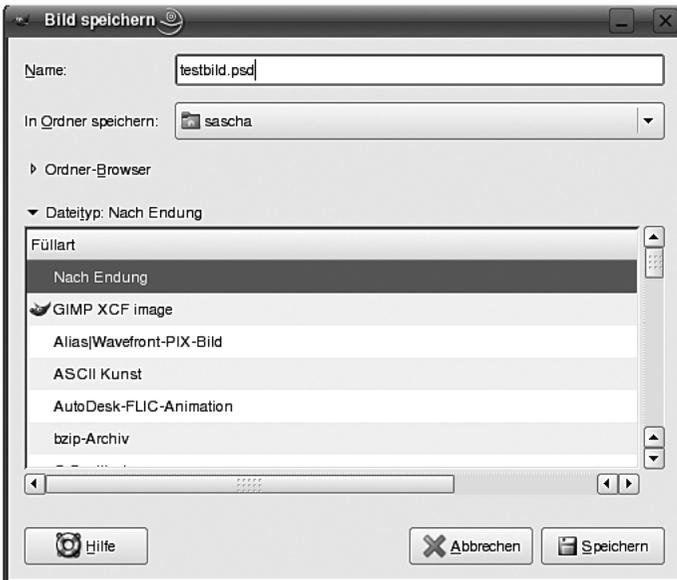


Abbildung 8.3 Der GIMP-Speicherdialg mit aufgeklappter Dateityp-Auswahl

Das Dokumentfenster

Die eigentliche Arbeitszentrale von GIMP (abgesehen von der Werkzeugpalette) ist das jeweilige Dokumentfenster. In Abbildung 8.4 sehen Sie ein Beispiel. Ganz oben befindet sich eine Menüleiste mit zahlreichen Befehlen, von denen Sie in diesem Kapitel natürlich nur die wichtigsten kennenlernen – es sind einfach zu viele, um sie alle zu behandeln; schließlich gibt es ganze Bücher über GIMP (siehe Anhang B).

Am linken und am oberen Rand des eigentlichen Dokumentbereichs finden Sie *Lineale*, die Ihnen die Positionierung und Bemaßung von Objekten ermöglichen. Die Maßeinheit sind standardmäßig Pixel; mit Hilfe des Menüs links unten (px) können Sie sie leicht in etwas Auflösungsabhängiges wie Millimeter oder Punkt ändern. Rechts daneben wird der Zoomfaktor ausgewählt.

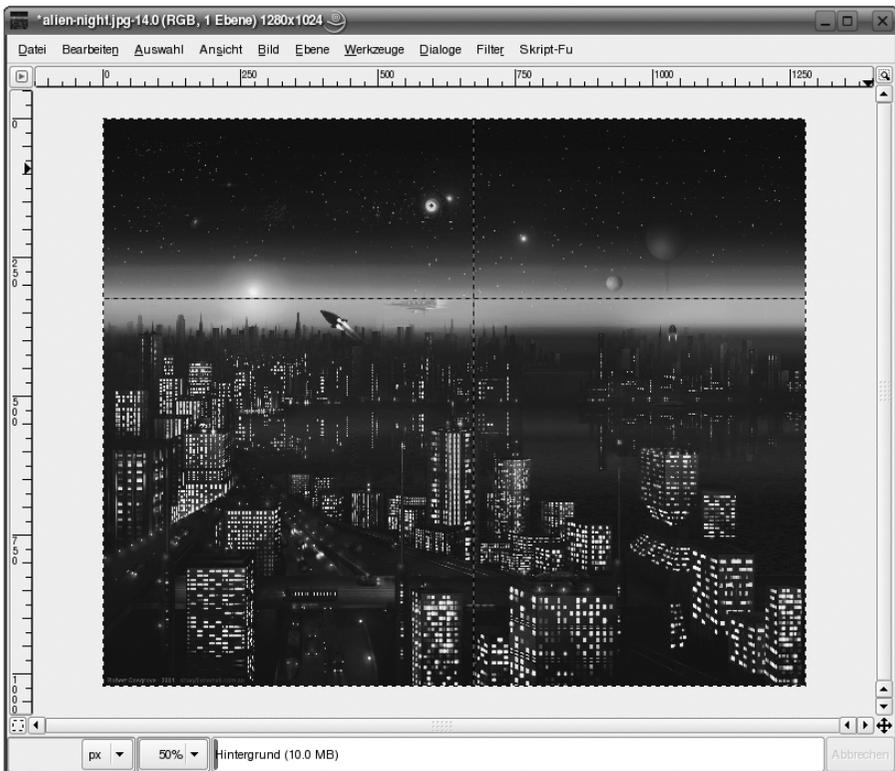


Abbildung 8.4 Ein GIMP-Dokumentfenster mit Linealen und Hilfslinien

Der Pfeil im Ursprung des Koordinatensystems (links oben) klappt ein Pull-down-Menü mit allen Befehlen auf, die sich auch in der Menüleiste befinden – dies

geschieht aus traditionellen Gründen, das heißt für ältere Window-Manager, die keine Menüleiste anzeigen können.

Aus den Linealen können Sie mit der Maus beliebig viele horizontale beziehungsweise vertikale *Hilfslinien* herausziehen. Wenn Sie später Elemente mit der Maus verschieben, verhalten sich diese Hilfslinien »magnetisch« und erleichtern es Ihnen auf diese Weise, Inhalte korrekt auszurichten.

Die Werkzeugpalette

Die GIMP-Werkzeugpalette enthält einen umfangreichen Satz von Arbeitsmaterial für die unterschiedlichsten Einsatzgebiete – grob gesagt Auswahl, Malen/Zeichnen und Modifizieren. Viele der Werkzeuge lassen durch ihre Form- und Namensgebung intuitiv errahnen, welche Aufgabe sie besitzen. In Abbildung 8.5 sehen Sie die Werkzeugpalette im Ganzen. In Abbildung 8.6 ist dann nur der eigentliche Werkzeugauswahlbereich zu sehen, wobei die einzelnen Werkzeuge nummeriert sind. Die Nummern werden in den Überschriften der nachfolgenden Einzelbeschreibungen aufgegriffen.



Abbildung 8.5 Die GIMP-Werkzeugpalette

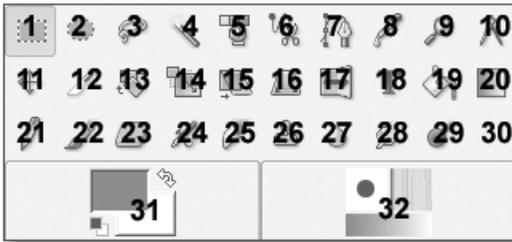


Abbildung 8.6 Die einzelnen Werkzeuge der GIMP-Palette, nummeriert für die nachfolgenden Beschreibungen

Die Rechteckauswahl (1)

Die *Rechteckauswahl*, die Sie auch einfach über die Taste **R** erreichen können (solange Sie sich nicht gerade im Texteingabemodus befinden), ermöglicht Ihnen die Auswahl eines rechteckigen Bereichs aus dem Bild oder der aktuellen Ebene. *Auswahlen* werden verwendet, um bestimmte Bearbeitungsschritte nur auf Teile des Bildes und nicht auf das gesamte Bild wirken zu lassen, aber natürlich auch für die üblichen Zwischenablage-Operationen.

Wenn Sie das Werkzeug einfach (ohne Modifikatortaste) verwenden, wird normal von Ecke zu Ecke ausgewählt. Halten Sie dagegen bestimmte Zusatztasten gedrückt, ändert sich das Verhalten:

- ▶ **⇧** fügt den ausgewählten Bereich zur bestehenden Auswahl hinzu.
- ▶ **Strg** zieht den neu ausgewählten Bereich von der bestehenden Auswahl ab und sorgt zusätzlich dafür, dass die Auswahl von der Mitte aus aufgezogen wird (um Letzteres für eine andere Art der Auswahl zu nutzen, müssen Sie diese in den Optionen ausdrücklich bestimmen; siehe unten).
- ▶ Mit **Strg**+**⇧** wird eine Schnittmenge gebildet, das heißt, nur der gemeinsame Bereich der bestehenden und der neuen Auswahl wird als neue Auswahl erstellt.

In den Optionen, das heißt unterhalb der eigentlichen Werkzeugwahl, kann der MODUS auch explizit festgelegt werden; von links nach rechts stehen die vier Schaltflächen für AUSWAHL ERSETZEN, ZUR AUSWAHL HINZUFÜGEN, VON AUSWAHL ABZIEHEN und AUSWAHLSCHNITTMENGE BILDEN.

Die Option KANTEN AUSBLENDEN erstellt einen allmählichen Intensitätsübergang zwischen Auswahl und Nichtauswahl über den gewählten Pixelbereich – ideal für Übergänge und Montagen. AUTOMATISCH VERKLEINERN beschränkt die Auswahl automatisch auf Bereiche, in denen die aktuelle Ebene Inhalte besitzt; die Zusatzoption VEREINIGUNG ÜBERPRÜFEN berücksichtigt dagegen alle Ebenen.

Darunter wird der Auswahlumfang festgelegt: FREIE AUSWAHL wählt genau den mit der Maus festgelegten Bereich aus; FESTE GRÖSSE ermöglicht die manuelle Eingabe von Breite und Höhe (in diversen Maßeinheiten) und die anschließende Auswahl per Einzelklick; FESTES SEITENVERHÄLTNIS beschränkt die Auswahl schließlich auf Quadrate statt Rechtecke.

Die Ellipsenauswahl (2)

Dieses Werkzeug (Kurtaste **E**) wählt elliptische Bereiche statt Rechtecke aus.

Die verfügbaren Optionen entsprechen sinngemäß genau der Rechteckauswahl; die zusätzliche Option KANTENGLÄTTUNG stellt Antialias ein, das heißt Übergangspixel mit verringerter Deckkraft.

Das Lasso (3)

Mit Hilfe des Lassowerkzeugs, das sich über die Kurtaste **L** aktivieren lässt, wählen Sie einen Bereich durch Freihandbewegung aus. Sobald Sie die Maustaste loslassen, wird der Anfangspunkt durch eine gerade Linie mit dem Endpunkt verbunden.

Die einzigen verfügbaren Optionen sind KANTENGLÄTTUNG und KANTEN AUSBLENDEN; sie entsprechen der Ellipsenauswahl.

Der Zauberstab (4)

Mit dem Zauberstab (Kurtaste **Z**) wählen Sie zusammenhängende Bereiche ähnlicher Farbe aus. Wie ähnlich die Farbe sein muss, bestimmt die Option SCHWELLE; die Werte reichen von 0 (exakt identisch) bis 255 (jede beliebige Farbe).

Der Farbbereichswähler (5)

Dieses Werkzeug (Tastenkürzel **⇧**+**0**) funktioniert im Wesentlichen genau wie der Zauberstab – mit dem einzigen Unterschied, dass farbähnliche Pixel aus dem gesamten Bild gewählt werden, nicht nur zusammenhängende.

Umrisse des Bildes wählen (6)

Jeder Klick mit diesem Werkzeug (Taste **I**) fügt einen weiteren Eckpunkt zur Auswahl hinzu. Die Auswahllinie zwischen den Punkten orientiert sich an Kanten beziehungsweise Farbkontrasten im Bild; dadurch ist das Werkzeug praktisch geeignet, um Objekte freizustellen.

Pfade erstellen und bearbeiten (7)

Mit dem Pfadwerkzeug () werden *Vektorzeichnungen* erstellt. Einfache Klicks erzeugen Eckpunkte, die jeweils durch gerade Linien miteinander verbunden werden. Durch Ziehen erstellen Sie dagegen Kurvenwendepunkte. Wenn Sie einen solchen Punkt anklicken, werden Hilfslinien angezeigt, an denen Sie ziehen können, um den Kurvenverlauf nachträglich zu modifizieren. Mit ein wenig Übung sind Pfade ein sehr nützliches Mittel, um geometrische Figuren zu zeichnen oder aus Pixelbildern auszuwählen.

In den Optionen des Werkzeugs können Sie zunächst den sogenannten BEARBEITUNGSMODUS einstellen: DESIGN dient dem Hinzufügen neuer Punkte. Wenn Sie BEARBEITEN wählen (oder im Designmodus  gedrückt halten), können Sie vorhandene Punkte verschieben oder über die Hilfslinien die Kurvenformen verändern. VERSCHIEBEN ermöglicht schließlich das Bewegen der gesamten Vektorzeichnung.

Die Option POLYGONAL zeichnet nur noch gerade Linien statt Kurven; Ziehen verschiebt in diesem Fall die Eckpunkte.

Sobald Sie mit dem Ergebnis zufrieden sind, können Sie zwei interessante Operationen mit Ihrem Pfad durchführen: AUSWAHL AUS PFAD ERZEUGEN erstellt eine Pixelauswahl in Form des Pfades; eine eventuelle Lücke wird dabei durch eine gerade Linie geschlossen. PFAD NACHZIEHEN ermöglicht es Ihnen dagegen, den Pfad mit einem auswählbaren Malwerkzeug nachzeichnen zu lassen.

Farben aus dem Bild wählen (8)

Die Pipette (Kurtaste ) ermöglicht es Ihnen, beliebige Pixel in einem Bild anzuklicken, um deren Farbe aufzunehmen.

Normalerweise wird genau die Farbe des Pixels an der Pipettenspitze gewählt. Die Option ABTASTGRÖSSE ermöglicht es Ihnen zusätzlich, einen Radius zwischen 1 und 30 Pixeln auszuwählen und so die durchschnittliche Farbe dieses Einzugsbereichs zu wählen.

Der AUSWAHLMODUS, der sich durch die -Taste auch dynamisch ändern lässt, bestimmt, was mit der gewählten Farbe geschehen soll: NUR AUSWÄHLEN öffnet lediglich ein Informationsfenster mit den Daten der gewählten Farbe (siehe Abbildung 8.7), während VORDERGRUNDFARBE beziehungsweise HINTERGRUNDFARBE ERSETZEN jeweils eine der beiden Arbeitsfarben einstellt. Die alternative Option ZUR FARBPALETTE HINZUFÜGEN – vorübergehend per -Taste aktivierbar – erweitert die aktuelle benutzerdefinierte Farbpalette um die gewählte Farbe.



Abbildung 8.7 Eine mit Hilfe der Pipette ermittelte Farbinformation

Vergößern/Verkleinern (9)

Das LUPENWERKZEUG besitzt zwar kein Tastenkürzel, aber Hinein- und Herauszoomen können Sie am schnellsten mit Hilfe der Tasten beziehungsweise . Beim Werkzeug selbst können Sie zusätzlich die Einstellung FENSTERGRÖSSE BEIM VERGRÖßERN UND VERKLEINERN ANPASSEN wählen – manchmal ist es praktisch, wenn das Fenster mitwächst. Wenn Sie statt zu klicken einen Rahmen aufziehen, wird der entsprechende Ausschnitt herangezoomt; die Option SCHWELLE bestimmt anhand der Rechteckgröße, in wie vielen Stufen dies geschieht.

Abstände und Winkel messen (10)

Mit diesem Werkzeug – das ebenfalls keine Kurztaste besitzt – können Sie Entfernungen in Bildern messen, indem Sie damit Linien aufziehen. Das Ergebnis – ein Abstand in Pixeln und ein Winkel in Grad – werden standardmäßig in der Statusleiste unter dem Bild angezeigt; alternativ können Sie die Option INFO-FENSTER VERWENDEN einstellen.

Ebenen und Auswahlen verschieben (11)

Dieses Werkzeug (eigene Kurztaste) erreichen Sie am schnellsten vorübergehend, indem Sie die Leertaste gedrückt halten. Die drei Schaltflächen am oberen Rand des Optionsfeldes wählen aus, was verschoben werden soll:

- ▶ EBENE TRANSFORMIEREN verschiebt die aktuelle Ebene; die beiden Radio-Buttons darunter bestimmen, wie diese ausgewählt wird: EBENE ODER HILFSLINIE AUSWÄHLEN verschiebt diejenige Ebene beziehungsweise Hilfslinie, die sich an der Cursorposition am weitesten im Vordergrund befindet. AKTIVE EBENE VERSCHIEBEN verlässt sich dagegen ausschließlich auf die Auswahl im EBENEN-Dialog. Ersteres ist oft schneller, während Letzteres eine präzise Ebenenauswahl ermöglicht.

- ▶ **AUSWAHL TRANSFORMIEREN** verschiebt die aktuelle Auswahl. Es ist wichtig, diese Option nicht misszuverstehen: Verschoben wird die *Auswahl selbst*; nach Anwendung dieser Funktion sind andere Pixel ausgewählt als vorher. Den *Inhalt* der aktuellen Auswahl verschieben Sie dagegen mit den weiter oben beschriebenen Auswahlwerkzeugen.
- ▶ **PFAD TRANSFORMIEREN** verschiebt den aktuellen Vektorpfad, genau wie die oben erwähnte Verschieben-Funktion des Pfadwerkzeugs.

Bildgröße ändern/Bild zuschneiden (12)

Mit dieser Funktion ($\square + \square$) können Sie Bilder oder Ebenen beziehungsweise ihre Inhalte per Auswahl verkleinern. Sobald Sie im Bild die Maustaste drücken, wird ein Dialog (siehe Abbildung 8.8) angezeigt – falls er Sie stört, können Sie ihn unterdrücken, indem Sie beim Klicken \square gedrückt halten. Hier können Sie folgende Parameter manuell einstellen: **URSPRUNG X** und **URSPRUNG Y** bilden die linke obere Ecke des gewünschten Bereichs; seine Größe wird durch **BREITE** und **HÖHE** festgelegt. Für beide Wertepaare können Sie jeweils getrennt die Maßeinheit einstellen. Wenn Sie das **SEITENVERHÄLTNIS** ändern, geschieht dies durch eine Anpassung der Höhe. Die Schaltfläche **AUS AUSWAHL** verwendet automatisch das Begrenzungsrechteck, das alle aktiven Auswahlbereiche umfasst, während **AUTOMATISCH SCHRUMPFEN** diejenigen Bereiche des Bildes beziehungsweise der Ebene weglässt, die keinen Inhalt besitzen.

Im Bild selbst können Sie den Ausschnitt auch mit der Maus wählen. Der Bereich außerhalb der Auswahl wird dabei leicht abgedunkelt, was eine bessere Vorschau ermöglicht.

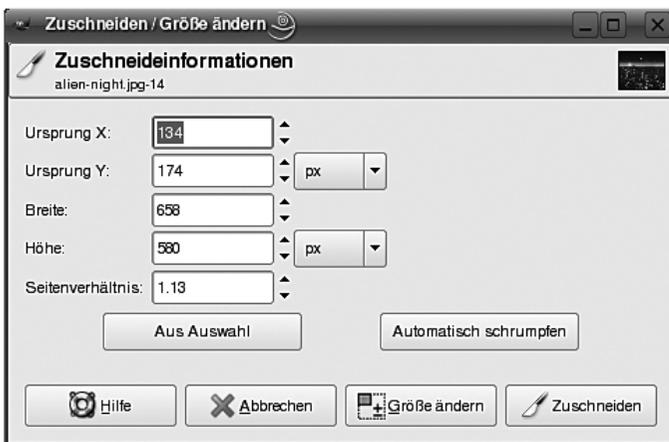


Abbildung 8.8 Der Einstellungsdialog zur Funktion »Zuschneiden/Größe ändern«

Die wichtigste Option ist die Auswahl zwischen den beiden – etwas irreführend betitelten – Funktionen ZUSCHNEIDEN und GRÖSSE ÄNDERN: ZUSCHNEIDEN ändert die *tatsächliche* Größe der Ebene beziehungsweise des Bildes, während GRÖSSE ÄNDERN nur die überflüssigen Inhalte entfernt, aber die Ausdehnung beibehält. Das Auswahlfeld NUR DIE AKTIVE EBENE ermöglicht es optional, die Funktion des Werkzeugs auf die aktuelle Ebene zu beschränken.

Um die gewählte Änderung tatsächlich durchzuführen, brauchen Sie nur den Auswahlbereich anzuklicken; alternativ können Sie auch im Dialog auf eine der Schaltflächen ZUSCHNEIDEN oder GRÖSSE ÄNDERN klicken. Falls Sie die Funktion dagegen ohne Änderung abrechnen möchten, müssen Sie **[ESC]** oder die Schaltfläche ABBRECHEN im Dialog betätigen.

Ebene oder Auswahl drehen (13)

Dieses Werkzeug (**[⇧]+[R]**) erlaubt das einfache Drehen der aktuellen Ebene, Auswahlumrandung oder Vektorzeichnung (wie beim Verschiebewerkzeug durch die WIRKT AUF-Schaltflächen auszuwählen). Sie können das gewünschte Element mit der Maus beliebig drehen; der ebenfalls angezeigte Mittelpunkt lässt sich zudem verschieben, was wiederum die Drehung beeinflusst. Im automatisch eingeblendeten Dialog DREHINFORMATIONEN können Sie diese Werte auch per Eingabe angeben.

Wichtig für die Drehung und andere Transformationen ist die INTERPOLATION, die Sie in den Optionen des Werkzeugs einstellen können; sie bestimmt, wie die neu zu zeichnenden Pixel des Bildes berechnet werden:

- ▶ KEINE setzt die vorhandenen Pixel einfach unverändert an die neue Position; bei Skalierungsoperationen bedeutet dies, dass bestimmte Pixel bei Bedarf einfach weggelassen oder verdoppelt werden.
- ▶ LINEAR zieht die Farben der vier links und rechts sowie oben und unten angrenzenden Nachbarn jedes Pixels zu Rate.
- ▶ KUBISCH bezieht sogar jeweils alle acht umgebenden Pixel in die Berechnung mit ein.

In der Praxis ist KEINE am besten für flächige, zeichnungsähnliche Grafiken geeignet, während Fotos und andere Halbtonvorlagen am sinnvollsten mit der Interpolationsmethode KUBISCH verarbeitet werden. LINEAR schließlich sollten Sie in Einzelfällen wählen, in denen KEINE zu grobpixelig beziehungsweise KUBISCH zu »matschig« wirkt.

Mit Hilfe der **[Strg]**-Taste oder des entsprechenden Optionsschalters können Sie die Winkel auf Vielfache von 15° beschränken.

Ebene oder Auswahl skalieren (14)

Das Skalierungswerkzeug ($\square + \square$) ermöglicht das Verkleinern oder Vergrößern des aktuellen Ebenen- oder Auswahlinhalts; genau wie bei den anderen Transformationswerkzeugen können alternativ auch Auswahlformen oder Vektorzeichnungen skaliert werden. Durch Ziehen mit der Maus oder per Eingabe im automatisch eingeblendeten Dialog können Sie die neue Größe des gewünschten Elements einstellen.

Die INTERPOLATION muss auch hier gemäß den obigen Ausführungen zum Drehwerkzeug gewählt werden. Unter EINSCHRÄNKUNGEN können Sie festlegen, dass Sie die Höhe (alternativ \square -Taste festhalten), die Breite (\square -Taste) oder das Seitenverhältnis ($\square + \square$) beibehalten möchten.

Ebene oder Auswahl scheren (15)

Auf dieselbe Weise wie Drehung und Skalierung verzerrt dieses Werkzeug (Tastenkürzel $\square + \square$) Ebenen, Auswahlformen oder Vektorzeichnungen parallelogrammförmig. Je nachdem, ob Sie zuerst nach rechts/links oder nach oben/unten ziehen, wird in X- beziehungsweise Y-Richtung geschert. Die Interpolationsmethode muss auch bei dieser Funktion fallweise gewählt werden.

Die Perspektive der Ebene oder Auswahl verändern (16)

Dieses Werkzeug ($\square + \square$) verzerrt das ausgewählte Element perspektivisch durch Ziehen an der jeweiligen Ecke. Funktionsweise und Optionen entsprechen den bereits beschriebenen Transformationswerkzeugen.

Die Ebene oder Auswahl spiegeln (17)

Das Spiegelungswerkzeug besitzt die Tastenkombination $\square + \square$. Mit jedem Mausklick wird das aktuell gewählte Element gespiegelt, so dass der zweite Mausklick die Ursprungsrichtung wiederherstellt. Durch Auswahlknöpfe im Optionsbereich können Sie zwischen horizontaler und vertikaler Spiegelung wählen; die \square -Taste wählt dynamisch die jeweils andere Option.

Text zum Bild hinzufügen (18)

Das Textwerkzeug (\square) dient dazu, Bilder mit beliebigem Text zu beschriften. Sobald Sie mit dem Werkzeug auf eine beliebige Stelle im Bild klicken, wird eine neue Textebene eingefügt, und es erscheint ein kleines Dialogfeld zur Texteingabe, der GIMP TEXT EDITOR (siehe Abbildung 8.9). Hier können Sie Text eingeben, der automatisch an der gewählten Stelle erscheint. Mit Hilfe der Schaltfläche ÖFFNEN können Sie auch eine externe Textdatei importieren, die im Bild angezeigt wird. LÖSCHEN entfernt den gesamten Text des aktuellen Bereichs; die bei-

den restlichen Schaltflächen dienen der Wahl der Schreibrichtung (LINKS NACH RECHTS für lateinische Schrift und andere oder RECHTS NACH LINKS für Schriften wie Hebräisch, Arabisch und so weiter). Sobald Sie auf SCHLIESSEN drücken, wird der Dialog geschlossen; ein Klick mit dem Textwerkzeug auf einen bestehenden Textbereich öffnet es wieder.

Sämtliche Aspekte der Zeichen- und -Absatzformatierung werden im Optionsbereich der Werkzeugpalette eingestellt: SCHRIFT und GRÖSSE, HINTING beziehungsweise AUTO-HINTING für bessere Lesbarkeit kleiner Schrift, KANTENGLÄTTUNG und FARBE sowie AUSRICHTUNG des Absatzes, EINZUG und ZEILENABSTAND. Bitte beachten Sie, dass ein einzelner Textbereich jeweils nur *einen* solchen Satz von Einstellungen besitzen darf.

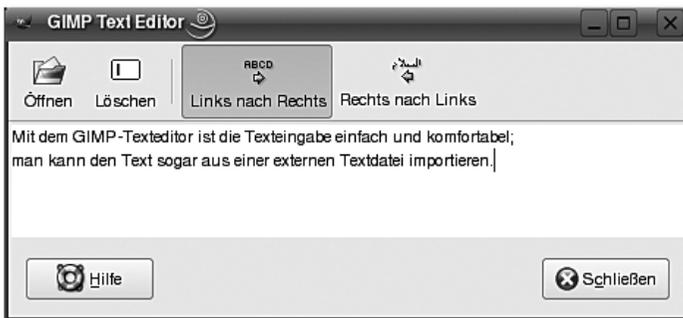


Abbildung 8.9 Text zum Bild hinzufügen mit dem GIMP Text Editor

Mit einer Farbe oder einem Muster füllen (19)

Das Füllwerkzeug ($\square + \text{B}$) füllt die aktuelle Ebene oder Auswahl mit einer Farbe oder einem Muster. Es ist das erste in einer Reihe von Malwerkzeugen, die allesamt den tatsächlichen Pixelinhalt einer Ebene modifizieren. Viele seiner Optionen gelten für alle diese Werkzeuge.

Als Erstes kann die DECKKRAFT eingestellt werden; die möglichen Werte liegen zwischen 0% (keine Wirkung) und 100% (voll deckend). Unabhängig von der Deckkraft bestimmt der MODUS, wie hinzugefügte und bisherige Farbe jedes Pixels einander beeinflussen. Die verfügbaren Modi sind:

- ▶ NORMAL
Die neue Farbe überdeckt die bisherige, je nach gewählter Deckkraft mit unterschiedlicher Intensität
- ▶ VERNICHTEND
Eine Deckkraft unter 100% wird nicht durch Mischung der einzelnen Pixel realisiert, sondern durch ein Zufallsmuster, indem eine dem Prozentsatz ent-

sprechende mittlere Anzahl von Pixeln eingefärbt wird. Das Ergebnis ist eine gesprenkelte Fläche.

- ▶ **HINTER**
Es werden nur transparente Bereiche der Ebene gefüllt.
- ▶ **FARBE ENTFERNEN**
Die Anteile der gewählten Füllfarbe werden aus jedem Pixel des Füllbereichs entfernt.
- ▶ **MULTIPLIKATION**
Multipliziert die Farbwerte miteinander. Im Großen und Ganzen wird das Ergebnis heller. Die Füllfarbe Schwarz (Farbwert 0) ergibt allerdings immer Schwarz, während Weiß (Wert 1) die ursprüngliche Farbe beibehält.
- ▶ **DIVISION**
Die bisherigen Farbwerte werden durch diejenigen der Füllfarbe dividiert; das Bild wird in der Regel dunkler.
- ▶ **BILDSCHIRM**
Die Pixel der Ebene werden mit dem Kehrwert der Füllfarbe multipliziert, das Ergebnis wird erneut invertiert. Insgesamt wird das Bild auf diese Weise heller; die Füllfarbe Weiß (Kehrwert 0) färbt alles komplett weiß ein, bei Schwarz (Kehrwert 1) bleibt die ursprüngliche Farbe dagegen erhalten.
- ▶ **ÜBERLAGERN**
Dieser Modus kombiniert die Modi **MULTIPLIKATION** und **BILDSCHIRM** miteinander. Daraus ergibt sich eine leichte Einfärbung des Füllbereichs mit der Füllfarbe.
- ▶ **ABWEDELN**
Der ursprüngliche Farbwert wird durch den Kehrwert des überlagernden dividiert. Auch diese Variante hellt das Bild auf, wobei Schwarz keine Änderung ergibt und Weiß alles überlagert.
- ▶ **NACHBELICHTEN**
Das Gegenteil von **ABWEDELN**; es wird mit dem Kehrwert der Füllfarbe multipliziert, so dass die Ebene insgesamt dunkler wird.
- ▶ **HARTE KANTEN**
Ist die gewählte Füllfarbe heller als 50 % Grau, dann wird multipliziert und das Bild wird heller, andernfalls wird der Bildschirm-Modus verwendet, der es dunkler macht. Die Füllfarben Schwarz und Weiß überdecken die bisherigen Pixel jeweils vollständig.
- ▶ **WEICHE KANTEN**
Auch bei diesem Modus hängt das Ergebnis von der Füllfarbe ab, allerdings wird es damit nur moderat aufgehellt beziehungsweise abgedunkelt.

- ▶ **FASER EXTRAHIEREN**
Die Ebene wird mit dem Kehrwert der Füllfarbe überblendet.
- ▶ **FASER MISCHEN**
Wie FASER EXTRAHIEREN, allerdings mit der Füllfarbe selbst.
- ▶ **UNTERSCHIED**
Es wird der Absolutwert der Farbdifferenz gebildet. Die Füllfarbe Weiß invertiert die ursprünglichen Pixel, Schwarz ergibt keine Änderung, alle anderen Farben führen zu eher surrealistischen Einfärbungen.
- ▶ **ADDITION**
Die Farbwerte werden einfach addiert; es ergibt sich eine lineare Aufhellung.
- ▶ **SUBTRAKTION**
Die Füllfarbe wird von der bisherigen abgezogen, so dass eine lineare Abdunklung entsteht.
- ▶ **NUR ABDUNKELN**
Die Farbwerte werden verglichen; der dunklere setzt sich durch.
- ▶ **NUR AUFHELLEN**
Auch hier werden die beiden Farbwerte einem Vergleich unterzogen. Allerdings wird in diesem Fall der hellere gewählt.
- ▶ **FARBTON**
Farbton, Sättigung und Helligkeit der beteiligten Farbwerte (die Einzelbestandteile des HSB-Schemas) werden getrennt voneinander betrachtet. Die Ebene wird mit dem Farbton der Füllfarbe eingefärbt.
- ▶ **SÄTTIGUNG**
In diesem Fall setzt sich nur die Sättigung der Füllfarbe durch.
- ▶ **FARBE**
Farbton und Sättigung der Füllfarbe werden übernommen, aber die Sättigung wird beibehalten.
- ▶ **WERT**
Nur der Helligkeitswert der neuen Farbe wird zum Füllen verwendet.

Beachten Sie, dass bei den beschriebenen Farbberechnungen die Rot-, Grün- und Blauwerte der Farben jeweils einzeln berechnet werden; dies macht ihre Wirkung verständlicher. Eine Ausnahme bilden die letzten vier Modi, die auf dem HSB-Modell basieren. Auf der Website zum Buch finden Sie farbige Abbildungen, die die Wirkung der Füllmodi an verschiedenen Beispielen demonstrieren.

Als Nächstes müssen Sie mit Hilfe der entsprechenden Optionsschalter oder schnell mit der `[Strg]`-Taste auswählen, mit welcher Farbe gefüllt werden soll. Die Möglichkeiten sind **VORDERGRUNDFARBE**, **HINTERGRUNDFARBE** oder ein vorde-

finiertes FÜLLMUSTER. Letzteres wird in der Palette MUSTER (DIALOGE • MUSTER oder **[Strg] + [F] + [P]**) ausgewählt.

Zu guter Letzt wird der zu füllende Bereich eingestellt: GANZE AUSWAHL FÜLLEN bedeckt die gesamte aktuelle Ebene beziehungsweise Auswahl mit der gewählten Füllfarbe oder dem aktuellen Muster. ÄHNLICHE FARBEN FÜLLEN erlaubt dagegen die Angabe einer Schwelle – sie bestimmt, wie ähnlich die Farben benachbarter Pixel sein müssen, um noch mitgefüllt zu werden. Die möglichen Werte reichen von 0 (nur identische Farbe) bis 255 (alle Pixel).

Mit einem Farbverlauf füllen (20)

Das Verlaufswerkzeug (Taste **[L]**) ähnelt dem eben besprochenen Füllwerkzeug stark, allerdings wird hier keine einzelne Farbe, sondern ein wählbarer Farbverlauf als Füllung verwendet. Neben DECKKRAFT und MODUS können Sie vor allem folgende Einstellungen vornehmen: Auswahl des FARBVERLAUFS selbst; das Auswahlfeld UMKEHREN dreht dessen Richtung um. Die FORM bestimmt, wie der Verlauf auf der Ebene verteilt wird – beispielsweise linear, also in paralleler Form, oder kreisförmig. Die WIEDERHOLUNG schließlich regelt, ob die Farben des Verlaufs nur einmal oder mehrmals durchlaufen werden.

Um den Verlauf anzuwenden, müssen Sie an der entsprechenden Stelle im Bild mit der Maus ziehen. Der erste Klick legt den Mittelpunkt fest (besonders wichtig etwa bei kreisförmigen Verläufen), durch das Ziehen wird anschließend die Verlaufsrichtung bestimmt.

Um vorhandene Verläufe zu modifizieren oder eigene zu definieren, wird die Palette FARBVERLÄUFE (DIALOGE • FARBVERLÄUFE oder **[Strg] + [G]**) verwendet. Hier finden Sie eine Liste aller vorhandenen Verläufe. Die untere Leiste enthält von links nach rechts folgende Schaltflächen: FARBVERLAUF BEARBEITEN, NEUER FARBVERLAUF, FARBVERLAUF DUPLIZIEREN, FARBVERLAUF LÖSCHEN (funktioniert nur bei selbstdefinierten Verläufen) sowie FARBVERLÄUFE NEU LADEN. Letzteres ist sinnvoll, wenn Sie versehentlich einen der vordefinierten Farbverläufe geändert haben.

Wenn Sie einen Verlauf zum Bearbeiten geöffnet oder einen neuen erstellt haben, befinden Sie sich in dem Dialog aus Abbildung 8.10. Um Farbmarkierungen (die kleinen Dreiecke) hinzuzufügen oder zu verändern, brauchen Sie die entsprechende Leiste nur mit der rechten Maustaste anzuklicken. Das ebenfalls in der Abbildung gezeigte Kontextmenü enthält zahlreiche Befehle dafür, die Sie in Ruhe ausprobieren können. Praktischerweise duplizieren Sie dazu vorhandene Farbverläufe und verändern sie je nach Wunsch.

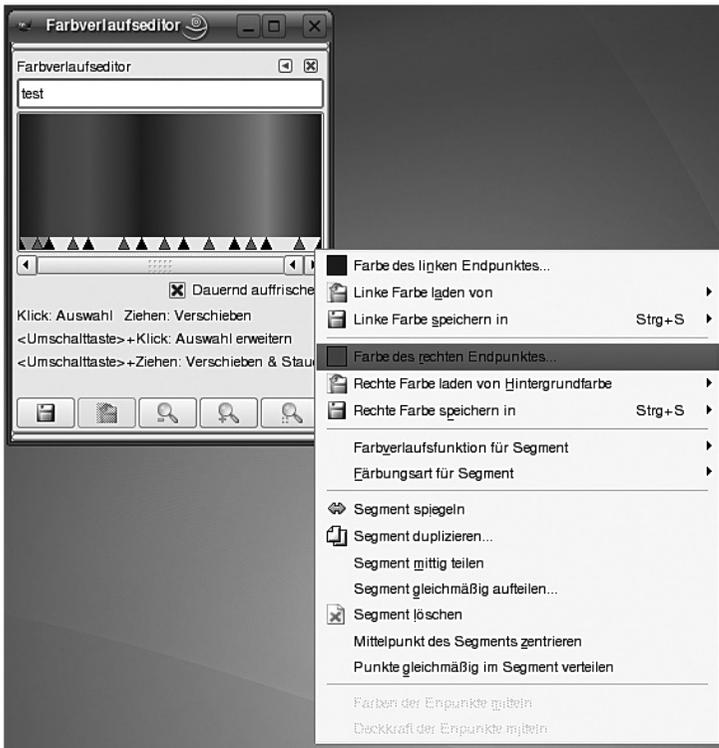


Abbildung 8.10 Einen Farbverlauf in GIMP bearbeiten

Der Stift (21)

Mit dem Stiftwerkzeug () können Sie Freihandzeichnungen in Vordergrundfarbe anfertigen. Die Besonderheit dieses Werkzeugs sind die harten Kanten – die Deckkraft nimmt zum Rand hin nicht ab. Unter der Option PINSEL können Sie die gewünschte Werkzeugspitze auswählen. Die Option DRUCKEMPFFINDLICHKEIT ist nur für die Arbeit mit einem Grafiktablett interessant; sie regelt, welche Aspekte des Buntstiftstrichs vom variablen Druck abhängen sollen.

VERBLASSEND bedeutet, dass der Strich über eine gewählte Anzahl von Pixeln hinweg schwächer wird. STEIGERND malt nicht sofort mit der vollen eingestellten Deckkraft, sondern erlaubt es, mehrere »Farbschichten« übereinanderzusetzen. FARBE AUS FARBVERLAUF schließlich malt nicht monochrom mit der Vordergrundfarbe, sondern verwendet den ausgewählten Farbverlauf mit diversen einstellbaren Optionen.

Der Pinsel (22)

Dieses Werkzeug (**P**) arbeitet genau wie der soeben beschriebene Stift, mit der Besonderheit, dass die Deckkraft mancher Pinselspitzen zum Rand hin kontinuierlich abnimmt.

Der Radiergummi (23)

Mit dem Radiergummi-Werkzeug (**⇧+E**) können Sie Ebeneninhalte manuell entfernen. Die Funktionsweise entspricht Stift und Pinsel, mit dem Unterschied, dass die behandelten Stellen einer Ebene transparent werden. Falls Sie eine DECKKRAFT unter 100% wählen, werden die Inhalte dagegen nicht vollständig entfernt, sondern nur prozentual durchsichtiger. Die Optionen des Radierers entsprechen den anderen Malwerkzeugen, mit zwei Ausnahmen: HARTE KANTEN stellt das Verhalten des Stifts ein, so dass die Strichränder nicht abgeschwächt werden; UN-RADIEREN macht Radiervorgänge selektiv rückgängig.

Die Sprühpistole (24)

Auch dieses Airbrush-Werkzeug (Taste **A**) funktioniert im Großen und Ganzen wie die anderen Malwerkzeuge. Allerdings wird die Farbe nach und nach aufgetragen, so dass ein Punkt umso stärker bedeckt wird, je länger Sie darauf verweilen. Dafür sind die beiden zusätzlichen Optionen RATE und DRUCK zuständig, die sich prozentual einstellen lassen.

Tinte (25)

Ein weiteres Malwerkzeug (Kurztaaste **K**) – es simuliert eine Schreibfeder, mit der sich kalligrafische Arbeiten anfertigen lassen. Dafür können Sie zahlreiche Aspekte der Werkzeugspitze einstellen: Unter JUSTIERUNG gibt die GRÖSSE die maximale Ausdehnung an, während der WINKEL die Grundposition regelt. Die Einstellungen unter EMPFINDLICHKEIT stellen dagegen die mögliche Abweichung vom Grundzustand ein: GRÖSSE und NEIGUNG können unterschiedlich stark variiert werden, während GESCHWINDIGKEIT die Verzögerung beim »Tintenfluss« bestimmt (der Maximalwert 1 ergibt dabei die stärkste Verzögerung, das heißt eigentlich die geringste Geschwindigkeit). Unter TYP können Sie schließlich eine von drei möglichen Federspitzenformen wählen.

Klonen (26)

Dieses Stempelwerkzeug (Taste **C**) ist eines der wichtigsten Mittel zur Bildretusche: Sie können damit Bildteile intuitiv an andere Stellen kopieren. Auf diese Weise lassen sich sowohl kleine Störungen überdecken (zum Beispiel Hautunreinheiten bei Porträts) als auch unerwünschte Objekte entfernen.

Als Erstes müssen Sie die Stelle im Bild auswählen, von der Sie Inhalte auswählen möchten. Klicken Sie die gewünschte Quelle dazu mit gedrückter **[Strg]**-Taste an. Anschließend können Sie wie gewohnt malen; ein Fadenkreuz zeigt dabei jeweils den Quellbereich an.

Optionen wie **DECKKRAFT**, **MODUS** oder **VERBLASSEN** entsprechen dabei den anderen Malwerkzeugen; neu sind dagegen die Einstellungen **QUELLE** und **AUSRICHTUNG**. Unter **QUELLE** können Sie sich alternativ zum Klonen eines Bildteils (Auswahl **BILDQUELLE**) auch eines der bereits angesprochenen Füllmuster aussuchen (**MUSTERQUELLE**).

Die Auswahlmöglichkeiten unter **AUSRICHTUNG** bedeuten Folgendes: **NICHT AUSGERICHTET** beginnt nach dem Loslassen und erneuten Drücken der Maustaste wieder am ursprünglichen Quellpunkt; **AUSGERICHTET** behält den relativen Abstand zwischen Quelle und Ziel über beliebig viele Malstriche hinweg bei. **REGISTRIERT** schließlich hält Quell- und Zielpunkt automatisch zusammen, eignet sich also im Grunde nur für Musterquellen.

Weichzeichnen/Schärfen (27)

Mit diesem Werkzeug (**[V]**) können Sie einzelne Bildbereiche nachträglich schärfen oder weichzeichnen; auf diese Weise können Sie beispielsweise fehlende Tiefenschärfe ausgleichen oder die Wichtigkeit oder Unwichtigkeit bestimmter Details verdeutlichen. Die Funktionsweise entspricht den anderen Malwerkzeugen; im Wesentlichen müssen Sie sich nur zwischen **WEICHZEICHNEN** und **SCHÄRFEN** entscheiden und können unter **RATE** einstellen, wie stark die entsprechende Funktion wirken soll.

Bild verschmieren (28)

Dieses Werkzeug (Taste **[S]**) nimmt die Farbe am jeweils angeklickten Punkt des Bildes auf und zieht sie weiter. Dies ermöglicht interessante Spezialeffekte, kann bei diffusen, eher abstrakten Bildteilen aber auch bisweilen den Stempel ersetzen beziehungsweise ergänzen.

Abwedeln/Nachbelichten (29)

Das letzte Werkzeug in der Palette dient dem gezielten Aufhellen (**ABWEDELN**) beziehungsweise Abdunkeln (**NACHBELICHTEN**) von Bildbereichen. Dabei können Sie sich entscheiden, ob Sie dunkle, mittlere oder helle Farbanteile bearbeiten möchten; diese werden als Schatten, Mitten beziehungsweise Glanzlichter bezeichnet.

Farben einstellen

Unter der eigentlichen Werkzeugpalette werden links die aktuelle Vorder- und Hintergrundfarbe (30) gezeigt; rechts sehen Sie die Einstellungen für Werkzeugspitze, Füllmuster und Farbverlauf (31). Sie können jedes dieser Felder anklicken, um die Auswahl zu ändern. Die Dialoge für Pinsel, Muster und Verläufe wurden bereits im Zusammenhang mit entsprechenden Werkzeugen erläutert.

Wenn Sie die aktuell gewählte Farbe anklicken, erscheint der in Abbildung 8.11 gezeigte Dialog VORDERGRUNDFARBE ÄNDERN beziehungsweise HINTERGRUNDFARBE ÄNDERN. Klicken Sie dagegen die jeweils andere Farbe an, wird sie zunächst als aktuelle Malfarbe ausgewählt, erst beim zweiten Klick erscheint der entsprechende Dialog.



Abbildung 8.11 Der GIMP-Farbwähler

Im Farbwähler können Sie sich links oben für vier verschiedene Registerkarten und damit Farbauswahlmethoden entscheiden:

- Die GIMP-Maus steht für den klassischen GIMP-Farbwahl-dialog: Das große Quadrat bietet jeweils ein gewisses Einstellungsspektrum für den auf der Skala rechts daneben gewählten Wert. Was auf der Skala eingestellt wird, bestimmen Sie mit Hilfe der Radio-Buttons rechts oben: Die Voreinstellung H (Hue) ist der Farbton; im Quadrat werden entsprechend Sättigung (von unten nach oben) und Helligkeit (von links nach rechts) eingestellt. S (Saturation) verlagert die Sättigung auf die Skala, so dass im Wählerquadrat Farbton und Helligkeit festgelegt werden. V (Value) schließlich bestimmt, dass die Hellig-

keit auf der Skala eingestellt wird und die beiden anderen Werte im großen Quadrat.

Alternativ können Sie auch eine der drei Komponenten Rot (R), Grün (G) oder Blau (B) auf der Skala auswählen.

- ▶ Die zweite Registerkarte bietet einen klassischen HSB- (oder HSV-)Farbwähler: Der äußere Kreis bestimmt den Farbton, während Sättigung und Helligkeit auf dem Dreieck in der Mitte eingestellt werden.
- ▶ Das Druckersymbol steht für die CMYK-Farbauswahl; in diesem Fall reichen die Werte nicht von 0 bis 255, sondern sind prozentual. Da es durch die zusätzliche Verwendung von Schwarz verschiedene Möglichkeiten gibt, dieselben Farben aufzubauen, können Sie über den SCHWARZAUSZUG regeln, wie stark dunkle, identische Farbanteile durch Schwarz dargestellt werden sollen.
- ▶ Die Registerkarte mit dem Pinsel schließlich dient der intuitiven, aber weniger präzisen Farbauswahl aus einem Spektrum.

Links unten werden zum Vergleich die aktuelle und die bisher ausgewählte Farbe gezeigt. Unten rechts befinden sich zwölf Rechtecke mit den zuletzt verwendeten Farben; mit Hilfe der kleinen Pfeiltaste können Sie die zurzeit eingestellte Farbe dorthin übernehmen. Darüber wird der *HTML*-Farbcode (hexadezimaler RGB) der aktuellen Farbe angezeigt.

Die kleine Doppelpfeil-Schaltfläche im Farbwähler der Werkzeugpalette vertauscht Vorder- und Hintergrundfarbe miteinander; die kleinen Quadrate stellen die Voreinstellung – Vordergrundfarbe Schwarz, Hintergrundfarbe Weiß – wieder her.

Einige weitere Arbeitsmittel

Wie bereits erwähnt, ist GIMP erheblich zu umfangreich, um hier erschöpfend behandelt zu werden – und das Programm lässt sich über eine praktische Schnittstelle durch Drittanbieter-Plug-ins oder gar eigene Skripte erweitern. Deshalb soll Ihre Aufmerksamkeit hier abschließend nur auf einige besondere Arbeitsmittel gelenkt werden; es lohnt sich, diese und weitere Hilfsmittel in der Praxis auszuprobieren.

Weiter oben wurden die *Ebenen* bereits grundsätzlich erwähnt. Öffnen Sie die EBENEN-Palette über DIALOGE • EBENEN oder schneller mit `[Strg]+[L]`. Eine Ebene ist eine Art transparente Folie, die Sie über ein vorhandenes Bild legen können, um darauf zu malen, ohne bestehende Inhalte zu vernichten. Transparenz wird in den Miniaturen der Ebenenpalette – und falls kein durchgehender Hintergrund vorhanden ist, auch im Bild selbst – durch ein Schachbrettmuster dargestellt.

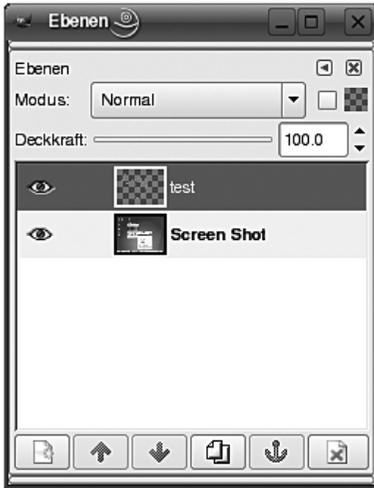


Abbildung 8.12 Die »Ebenen«-Palette in GIMP

Mit Hilfe des Augensymbols können Sie eine Ebene (oder mit  +Augensymbol alle anderen Ebenen) vorübergehend ein- oder ausblenden. Um auf einer bestimmten Ebene zu arbeiten, klicken Sie auf ihren Namen – neu gezeichnete oder eingefügte Elemente erscheinen dann exklusiv auf dieser Ebene.

Das besonders Interessante an Ebenen ist, dass sie nicht voll deckend übereinandergestapelt werden müssen: Genau wie die oben besprochenen Malwerkzeuge besitzen sie einen MODUS, der die Wirkung auf die darunterliegenden Pixel bestimmt, sowie eine DECKKRAFT.

Ganz unten in der EBENEN-Palette finden Sie eine Reihe von Buttons für folgende Aufgaben (von links nach rechts): NEUE EBENE ERSTELLEN, AKTUELLE EBENE NACH OBEN VERSCHIEBEN, AKTUELLE EBENE NACH UNTEN VERSCHIEBEN, EBENE DUPLIZIEREN, EBENE VERANKERN UND EBENE LÖSCHEN.

Wie bereits erwähnt, bestehen Mehrfarbenbilder aus einzelnen *Farbkanälen* – je nach Farbskala Rot, Grün und Blau oder Cyan, Magenta, Gelb und Schwarz. Mit Hilfe der Palette KANÄLE (DIALOGE • KANÄLE) können Sie jeden Farbkanal einzeln bearbeiten. Da die Farbkomponenten unterschiedlich verteilt sind, ermöglicht ein bestimmter Kanal mitunter eine besonders präzise Auswahl.

Auswahl ist in diesem Zusammenhang ohnehin ein gutes Stichwort: Intern sind Auswahlen nichts weiter als Kanäle, in denen Bereiche umso heller sind, je stärker ihre Auswahlintensität ist. Sie können sogar AUSWAHL • IN KANAL SPEICHERN wählen, um aus der aktuellen Auswahl einen neuen Kanal zu erstellen. Umge-

kehrt dient die vorletzte Schaltfläche unten in der KANÄLE-Palette dazu, eine Auswahl aus dem aktuellen Kanal zu erstellen.

Besonders wichtig für die Bearbeitung von Fotos sind die diversen Dialoge zur Farbanpassung unter EBENE • FARBEN. Die wichtigsten von ihnen sind:

- ▶ FARBTON SÄTTIGUNG ermöglicht es, einzelne Farbkomponenten zu modifizieren sowie ihre Sättigung und Helligkeit anzupassen.
- ▶ HELBIGKEIT KONTRAST ist der zentrale Anlaufpunkt zur Verbesserung über- oder unterbelichteter Bilder beziehungsweise Bildbereiche.
- ▶ WERTE ermöglicht unter anderem die Anpassung des Schwarz-, Mittelgrau- und Weißpunktes eines Bildes und somit einen Ausgleich bei fehlerhaftem Weißabgleich.

8.1.3 Das Vektorzeichenprogramm Inkscape

Die grundsätzliche andere Art der Computergrafik ist die Vektorgrafik, bei der mathematische Beschreibungen von Kurven und Linien gespeichert werden. openSUSE enthält verschiedene Programme, die mit Vektorzeichnungen umgehen können. Das leistungsfähigste und vielseitigste von ihnen ist *Inkscape*. Sie finden es im K-Menü oder GNOME-Menü unter ANWENDUNGEN • GRAFIK • VEKTORGRAFIK • SVG-VEKTORILLUSTRATOR/INKSCAPE. Abbildung 8.13 zeigt das Programm mit einigen Zeichnungen.

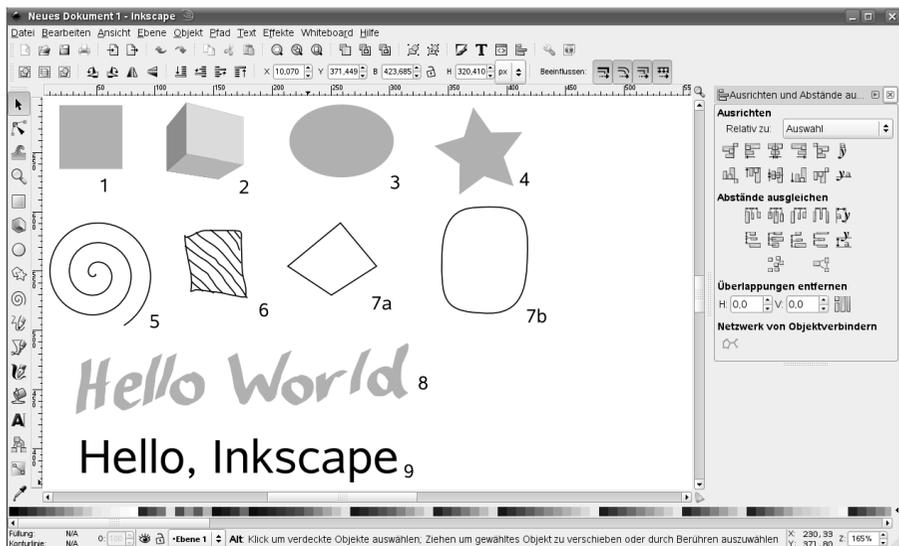


Abbildung 8.13 Der Bildschirm des Vektorzeichnungsprogramms Inkscape

Die Inkscape-Hauptwerkzeuge

Am linken Rand des Inkscape-Fensters finden Sie die Hauptwerkzeugleiste. Sobald Sie ein Werkzeug auswählen, können Sie in der unteren Symbolleiste, unmittelbar über dem eigentlichen Dokument, genauere Einstellungen dafür vornehmen. Einige der nachfolgend vorgestellten Werkzeuge sind nummeriert; die Nummern entsprechen den damit erstellten Zeichnungen aus Abbildung 8.13. Die Bezeichnungen in der Liste entsprechen den Tooltips, die beim Darüberfahren mit der Maus angezeigt werden. Von oben nach unten stehen folgende Werkzeuge zur Verfügung:

► *Objekte auswählen und verändern*

Klicken Sie ein Objekt an, um es auszuwählen. Wenn Sie die -Taste gedrückt halten, können Sie weitere Objekte zur Auswahl hinzufügen oder ausgewählte Objekte wieder abwählen. Klicken Sie ein Objekt mit der Maus an, und halten Sie die Maustaste gedrückt, um das Objekt an eine andere Position zu ziehen. Exakter, aber langsamer geht dies übrigens mit den Pfeiltasten.

Wenn Sie ein Objekt einmal anklicken, können Sie auch an den acht Anfassern um sein Begrenzungsrechteck ziehen, um es zu skalieren. Wenn Sie dabei  gedrückt halten, wird es aus der Mitte skaliert statt nur in die gewünschte Richtung. Mit  wird proportional skaliert, das Verhältnis zwischen Höhe und Breite wird also beibehalten. Beide Tasten sind auch kombinierbar. Sobald Sie ein Objekt zum zweiten Mal anklicken, ändert sich das Aussehen der Anfasser. Nun können Sie an den Eckanfassern ziehen, um es zu drehen; das Ziehen an den Anfassern in der jeweiligen Streckenmitte neigt das Objekt dagegen.

► *Bearbeiten der Knoten oder Anfasser eines Pfades*

Mit diesem Werkzeug wird kein gesamtes Objekt bearbeitet oder ausgewählt, sondern es geht um die einzelnen Eckpunkte beziehungsweise Kurvenankerpunkte, aus denen Zeichnungen bestehen. Das Ziehen an einem Eckpunkt verschiebt diesen Punkt und die mit ihm verbundenen Linien. Falls Sie dabei  festhalten, wird der Eckpunkt in einen Kurvenankerpunkt umgewandelt – die angrenzenden Linien werden entsprechend zu Kurven.

Wenn Sie an einem Kurvenankerpunkt ziehen, wird dieser ebenfalls verschoben. Das Ziehen an den Anfassern eines Kurvenankerpunkts ändert dagegen den Kurvenverlauf.

Bei einem mit dem weiter unten beschriebenen Werkzeug erstellten Rechteck oder Quadrat sorgt das Ziehen an dem runden Anfasser dafür, dass der Eckenradius geändert wird – Sie erhalten ein mehr oder weniger abgerundetes Rechteck. Bei Kreisen oder Ellipsen erzeugt das Ziehen am runden Anfasser

einen Kreis- beziehungsweise Ellipsenausschnitt; die Mausbewegung legt den Ausschnittswinkel fest.

- ▶ *Objekte verbessern durch Verformen oder Malen*
Dieses Werkzeug stellt in der unteren Symbolleiste verschiedene Manipulationsmöglichkeiten wie Verzerren oder Wölben zur Verfügung, die jeweils auf die im Einzugsbereich befindlichen Objekte wirken, insbesondere auf ihre Begrenzungslinien oder -kurven.
- ▶ *Zoomfaktor vergrößern oder verringern*
Das Zoomwerkzeug ändert die eigentliche Zeichnung nicht, sondern nur die Größe, in der sie angezeigt wird. Standardmäßig befinden Sie sich im Vergrößerungsmodus. Ein einfacher Klick vergrößert die Ansicht um einen Schritt. Alternativ können Sie einen Rahmen um einen bestimmten Bereich aufziehen, um den entsprechenden Ausschnitt zu vergrößern. Wenn Sie mit  klicken, wird die Ansicht dagegen um einen Schritt verkleinert.
- ▶ *Rechtecke und Quadrate erstellen (1)*
Standardmäßig werden durch Ziehen die äußeren Begrenzungen eines Rechtecks festgelegt. Mit gedrückter -Taste wird das Rechteck von der Mitte aus aufgezogen; mit STRG werden die Möglichkeiten auf ein Raster mit bestimmten Abständen beschränkt, was unter anderem das Zeichnen von Quadraten ermöglicht. Wie bereits erwähnt, steht die 1 für das entsprechende Element in Abbildung 8.13.
- ▶ *3-D-Box erzeugen (2)*
Mit diesem Werkzeug können Sie einen dreidimensionalen Quader oder Würfel aufziehen. Die genauen Eigenschaften werden in der unteren Symbolleiste eingestellt.
- ▶ *Kreise, Ellipsen und Bögen erstellen (3)*
Mit diesem Werkzeug werden Kreise und Ellipsen gezeichnet. Sein Verhalten entspricht dem Rechteckwerkzeug.
- ▶ *Sterne und Polygone erstellen (4)*
Mit diesem Werkzeug werden Sterne oder Polygone mit einer unterschiedlichen Anzahl von Zacken beziehungsweise Ecken erstellt. Die Anzahl wird in der unteren Symbolleiste eingestellt, genau wie die Frage, ob Sie ein Polygon oder einen Stern erstellen möchten. Das reine Zeichenverhalten entspricht dem Rechteck- und dem Ellipsenwerkzeug.
- ▶ *Spiralen erstellen (5)*
Zeichnet Spiralen, die Sie mit der Maus aufziehen können. Auch hier sind verschiedene Einstellungen verfügbar.

► *Freihandlinien zeichnen (6)*

Mit diesem Werkzeug können Sie beliebig zeichnen, indem Sie die Maustaste gedrückt halten und ziehen. Bessere Ergebnisse erzielen Sie natürlich mit einem Grafiktablett.

► *Bézier-Kurven (7b) und gerade Linien (7a) zeichnen*

Dies ist das wichtigste Werkzeug von Inkscape, da es die traditionelle Vektorzeichenweise darstellt. Ein Klick erzeugt einen Eckpunkt, der durch eine gerade Linie mit dem vorigen Punkt verbunden wird. Ziehen erzeugt dagegen einen Kurvenankerpunkt; Richtung und Stärke des Ziehens bestimmt den Verlauf der Kurve, die den vorigen mit dem neuen Punkt verbindet. Dies sind die besagten Bézier-Kurven (benannt nach Pierre Bézier, einem französischen Ingenieur, der diese mathematische Beschreibung von Kurven entwickelt hat). Im Prinzip ist die Arbeit mit dieser Funktion des Werkzeugs recht intuitiv, bedarf aber einer gewissen Übung.

Sie können das Zeichnen der aktuellen Kurve oder Linie jederzeit durch Esc abbrechen. Um die Zeichnung zu beenden (und zu behalten), drücken Sie dagegen ↵. Eine andere Möglichkeit ist erneutes Klicken beziehungsweise Ziehen auf dem Anfangspunkt; dies erzeugt einen geschlossenen Pfad, der beispielsweise mit Farbe gefüllt werden kann.

► *Kalligraphisch zeichnen (8)*

Dieses Werkzeug ahmt eine Schreibfeder mit wählbarer Stärke und Neigung nach; es entstehen Füllungen ohne sichtbare Konturlinien.

► *Abgegrenzte Gebiete füllen*

Wenn Sie dieses Werkzeug verwenden, können Sie geschlossene Pfade mit neuen Füllungen versehen; eine andere Möglichkeit besteht darin, die entsprechenden Pfade zu markieren und dann die gewünschte Farbe auszuwählen. Das Werkzeug besitzt allerdings noch die interessante Option *Lücken schließen*, um zu bestimmen, was geschehen soll, wenn nicht ganz geschlossene Pfade betroffen sind.

► *Textobjekte erstellen und bearbeiten (9)*

Ein einzelner Klick auf die Arbeitsfläche erzeugt einen Textbereich mit beliebiger Breite; Ziehen erstellt dagegen einen Textbereich mit fester Breite. Danach können Sie beliebigen Text eingeben und die Texteingenschaften (Schriftart, -größe, -farbe und so weiter) festlegen.

► *Objektverbinder erzeugen*

Wenn nacheinander zwei Objekte angeklickt werden, dann werden sie durch eine Linie miteinander verbunden. Diese Funktion ermöglicht das Zeichnen von Flussdiagrammen, Organigrammen und dergleichen.

- ▶ *Farbverläufe erstellen und bearbeiten*
Nicht immer sollen Zeichnungen mit einer einzelnen Farbe gefüllt werden. Farbverläufe können ästhetischer aussehen und ermöglichen unter Umständen das Zeichnen realistischerer Objekte. Hier können Sie Ihre Verläufe konfigurieren.
- ▶ *Farben aus dem Bild übernehmen*
Diese Pipette ermöglicht es Ihnen, Farbe, Kontureigenschaften oder einen Verlauf für künftige Objekte oder für das Füllwerkzeug aus dem Bild zu übernehmen.

Weitere Optionen

Weitere wichtige Bedienelemente befinden sich in der Hauptsymbolleiste (ganz oben). Nach den Standardhilfsmitteln zum Öffnen, Speichern oder für die Zwischenablage, die praktisch jede Anwendung besitzt, befinden sich die wichtigsten und Inkscape-spezifischsten von links aus gesehen in der drittletzten und der vorletzten Gruppe. Von links nach rechts bedeuten sie Folgendes:

- ▶ *Die gewählten Objekte gruppieren*
Wenn Sie diese Schaltfläche betätigen oder **[Strg]+[G]** drücken, verhalten sich die ausgewählten Objekte nach außen künftige wie ein einzelnes Objekt. Sie können also gemeinsam verschoben, skaliert oder gedreht werden, und es besteht keine Gefahr, dass sie versehentlich getrennt voneinander behandelt werden.
- ▶ *Gruppierung markierter Objekte aufheben*
Mit dieser Funktion (Kurzfassung: **[Strg]+[U]**) kann eine Gruppierung rückgängig gemacht werden. Dies ist manchmal wichtig, um die entsprechenden Objekte doch wieder einzeln zu bearbeiten.
- ▶ *Objektfarben, Farbverläufe, Strichbreiten, Pfeile, Strichmuster usw. ändern*
Öffnet am rechten Rand des Dokumentfensters einen umfangreichen Dialog, in dem Sie die Farb- und Kontureigenschaften Ihrer Zeichnungen festlegen können.
- ▶ *Schriftfamilie, Schriftgröße und andere Texteingenschaften ansehen und ändern*
Ermöglicht das praktische Einstellen aller wichtigen Texteingenschaften.
- ▶ *Zeige und ändere den XML-Baum des Dokuments*
Das interne Speicherformat von Inkscape ist SVG. Es handelt sich um ein spezielles XML-Format (Näheres zu XML siehe Kapitel 11, »Textbasierte Auszeichnungssprachen«). Mit dieser Schaltfläche können Sie den XML-Quelltext einsehen und gegebenenfalls ändern. Neben SVG kann Inkscape übrigens einige andere Dateiformate exportieren, was für die Zusammenarbeit mit

anderen Programmen wichtig ist. Beachten Sie aber, dass in einigen dieser Formate nicht mehr alle Aspekte Ihrer ursprünglichen Zeichnungen enthalten sind (natürlich erst recht nicht, wenn Sie sie als Pixelgrafiken exportieren). Aus diesem Grund sollten Sie stets eine SVG-Arbeitskopie behalten.

► *Objekte ausrichten und ihre Abstände ausgleichen*

Auch dieser Button öffnet einen Dialog am rechten Dokumentrand; weiter oben in Abbildung 8.13 ist er zu sehen. Hier erhalten Sie eine Reihe von Schaltflächen, um mehrere Objekte horizontal oder vertikal aneinander auszurichten und um mehrere von ihnen gleichmäßig zu verteilen.

Zuletzt ist es für ein Vektorgrafikprogramm besonders wichtig, dass Sie die Stapelreihenfolge überlappender Objekte ändern können. Die betreffenden Befehle finden Sie im Menü OBJEKT. Wählen Sie ANHEBEN (oder ) , um das ausgewählte Objekt eine Stufe weiter nach vorn zu holen. ABSSENKEN () stellt das Objekt dagegen um eine Stufe nach hinten. Weitere Optionen sind NACH GANZ OBEN ANHEBEN () und NACH GANZ UNTEN ABSSENKEN () . Um mehrere Objekte zu stapeln, zu sperren oder ein- und auszublenden, verfügt Inkscape zusätzlich über Ebenen.

8.2 Der Webbrowser Firefox

Eines der wichtigsten Anwendungsprogramme ist heute der Webbrowser; zum einen für den Besuch von Websites, zum anderen aber auch, weil immer mehr Anwendungen ins Internet oder Intranet verlegt und über den Browser bedient werden. openSUSE enthält mehrere Browser, aber der wichtigste und empfehlenswerteste ist zweifellos *Firefox*. In openSUSE 11.2 finden Sie die aktuelle Version 3.5. Firefox basiert auf den als Open Source freigegebenen Quellen des ursprünglichen Netscape Navigators (die heutige Version hat allerdings nicht mehr viel mit diesem gemeinsam) und wird von der Mozilla Foundation gepflegt.

8.2.1 Firefox im Überblick

In Abbildung 8.14 sehen Sie den Firefox im Einsatz. Beachten Sie, dass Sie natürlich erst einmal eine Internetverbindung herstellen müssen, um ihn sinnvoll einsetzen zu können. Wie dies geht, wird detailliert in Kapitel 13, »Netzwerkgrundlagen«, erklärt.



Abbildung 8.14 Der Webbrowser Mozilla Firefox

Das Webbrowsing beginnt üblicherweise mit der Eingabe einer URL in das lange Textfeld in der Adressleiste; diese befindet sich gleich unter dem Menü. Bereits während der Eingabe erhalten Sie Vorschläge. In Firefox 3 wurde die Vorschlagsliste überarbeitet: Sie enthält mehr Details und berücksichtigt die Häufigkeit Ihres Besuchs auf bestimmten Seiten. Frühere Versionen hielten sich nur an die Besuchsreihenfolge. Wenn Sie möchten, können Sie die Liste mit dem Pfeil am rechten Ende des Adressfeldes auch gleich aufklappen, ohne etwas einzugeben. Drücken Sie , um die eingetragene oder ausgewählte Adresse aufzurufen.

Links neben dem eigentlichen Adressfeld finden Sie folgende Schaltflächen (die Bezeichnungen entstammen ihren Tooltips):

- ▶ **EINE SEITE ZURÜCK** – der Browser merkt sich in der sogenannten History (zu Deutsch Verlauf), welche Seiten besucht wurden. Mit diesem Werkzeug oder mit  +  können Sie je eine Seite zurückblättern. Unter **EXTRAS • PRIVATE DATEN LÖSCHEN** können Sie die History und andere Spuren Ihrer Browsernutzung übrigens entfernen. Auf öffentlich genutzten Rechnern, beispielsweise in Internetcafés, sollten Sie dies nach der jeweiligen Sitzung tun. Auch für Entwickler ist diese Funktion sehr wichtig, da manche Änderungen nur zu sehen sind, wenn Sie den Cache (Zwischenspeicher zur Beschleunigung bereits geladener Seiten) leeren.

- ▶ EINE SEITE VOR – nachdem Sie in der History zurückgeblättert haben, können Sie sich mit dieser Schaltfläche oder mit **[Alt]+[→]** wieder vorwärts bewegen.
- ▶ KÜRZLICH BESUCHTE SEITEN – zeigt die letzten Einträge der History an und ermöglicht die gezielte Auswahl einer Seite daraus.
- ▶ AKTUELLE SEITE NEU LADEN – Seiten, die sich schnell ändern (beispielsweise aktuelle Nachrichten), können Sie mit dieser Schaltfläche oder mit **[Strg]+[R]** neu laden. Auch für Entwickler und Webdesigner ist diese Funktion interessant: Nachdem Sie die zurzeit angezeigte Seite im Editor geändert und gespeichert haben, können Sie auf diese Weise die neueste Version anzeigen lassen
- ▶ LADEN DIESER SEITE STOPPEN – falls das Laden zu lange dauert, können Sie es mit diesem Button oder mit **[Esc]** abbrechen.
- ▶ STARTSEITE – kehrt zur eingestellten Startseite zurück (Tastenkürzel: **[Alt]+[Pos1]**). Sie können die Startseite – und alle anderen Aspekte des Verhaltens von Firefox – unter BEARBEITEN • EINSTELLUNGEN ändern.

Eine sehr interessante Funktion von Webbrowsern sind die LESEZEICHEN. Mit ihrer Hilfe können Sie sich Webseiten merken, die Sie gern besuchen. Sie können ein Lesezeichen für die aktuelle Seite hinzufügen, indem Sie **[Strg]+[D]** drücken. Wenn Sie Unterordner bilden oder alte Lesezeichen löschen möchten, wählen Sie die Funktion LESEZEICHEN • LESEZEICHEN VERWALTEN.

Firefox gehörte zu den ersten Browsern, die »Tabbed Browsing« unterstützten: In ein und demselben Browserfenster werden mehrere Karteireiter (Tabs) mit unterschiedlichen Seiten angezeigt. Dies ist übersichtlicher als das Arbeiten mit mehreren Browserfenstern. Wählen Sie DATEI • NEUER TAB oder drücken Sie **[Strg]+[T]**, um einen Tab zu öffnen. Außerdem können Sie jeden Hyperlink auf einer Webseite mit der rechten Maustaste anklicken und LINK IN NEUEM TAB ÖFFNEN wählen.

Interessant ist auch die Funktion ANSICHT • VOLLBILD oder **[F11]**: Sie blendet die Menüleiste, die Adressleiste und alle Symbolleisten aus, die doch recht viel Platz belegen, und verwendet den gesamten Bildschirm für die eigentliche Webseite. Die Adressleiste wird übrigens eingeblendet, wenn Sie mit der Maus ganz nach oben fahren. Drücken Sie **[F11]** erneut, um zur Normalansicht zurückzukehren.

[Strg]+[F] oder **[/]** startet eine inkrementelle Suche im Inhalt der aktuellen Seite; bei jedem Tastendruck wird also bereits ein Suchergebnis angezeigt. Mit **[F3]** können Sie erneut nach dem aktuellen Begriff suchen.

Zuletzt sollten Sie sich die Funktion ANSICHT • SEITENQUELLTEXT ANZEIGEN oder **[Strg]+[U]** merken, mit der Sie den HTML- und CSS-Quellcode der aktuellen

Seite betrachten können. Nicht nur für Webdesigner und Entwickler kann ein Blick auf den Code interessant sein.

8.2.2 Firefox-Add-ons

Firefox ist bereits ab Werk ein sehr gut ausgestatteter Browser. Dennoch gibt es zahllose Erweiterungen von Drittanbietern, die ihn noch besser machen. Geprüfte und vertrauenswürdige Erweiterungen, die sich mit einem Mausklick installieren lassen, finden Sie unter <http://addons.mozilla.org> beziehungsweise auf der deutschen Seite <http://www.erweiterungen.de>. Gehen Sie ruhig selbst auf Entdeckungsreise; es lohnt sich. Drei Erweiterungen möchte ich Ihnen dennoch ans Herz legen:

- ▶ *Adblock Plus* ist ein mächtiger Werbe- und Popup-Filter, der die Daten zum Ausblenden unerwünschter Inhalte aus zahlreichen gut gepflegten Online-Listen bezieht.
- ▶ *Firebug* ist das praktischste Hilfsmittel für Webentwickler, das man sich vorstellen kann. Die Funktion *Inspect* ermöglicht beispielsweise das Markieren eines beliebigen Seitenteils mit der Maus; daraufhin werden seine HTML-, CSS- und JavaScript-Eigenschaften angezeigt. Umgekehrt können Sie auch im Quellcode ein Element markieren, das dann auf der Seite hervorgehoben wird. Ein anderer Modus zeigt zum Beispiel Dauer und Erfolg der diversen HTTP-Anfragen für die Elemente der aktuellen Seite an.
- ▶ *Web Developer* installiert eine zusätzliche Symbolleiste. In dieser können Sie beispielsweise JavaScript, CSS oder Java deaktivieren oder reaktivieren, verschiedene Elemente hervorheben oder die Seite online validieren (das heißt die formale Gültigkeit von HTML und CSS überprüfen).

8.3 Zusammenfassung

Einer der großen Vorteile einer Distribution wie openSUSE besteht darin, dass Sie nicht nur das Betriebssystem selbst erhalten, sondern auch Unmengen an kompatibler, optimal darauf abgestimmter Anwendungssoftware. In diesem Kapitel wurden einige besonders beliebte Beispiele vorgestellt:

- ▶ *Bildbearbeitung* ist ohne Zweifel eine der beliebtesten Aufgaben moderner PCs. Unter Linux steht dafür mit GIMP eine hervorragende Software zur Verfügung. Bevor Sie sich ins Vergnügen stürzen, sollten Sie die einführenden Bemerkungen zur Theorie in diesem Kapitel lesen – sie helfen Ihnen, die Bestandteile von GIMP und anderer Software besser zu verstehen. The GIMP

selbst ist ein überaus umfangreiches Programm, so dass im entsprechenden Abschnitt nur einige Grundlagen vermittelt werden konnten. Nachdem Sie sich so weit eingearbeitet haben, müssten Sie aber auch durch Ausprobieren weiterkommen. Neben den hier behandelten, recht intuitiv zu benutzenden Werkzeugen gibt es noch zahllose Dialoge, Filter und Erweiterungen.

- ▶ *Vektorgrafik* ermöglicht das Erstellen skalierbarer Zeichnungen mit Hilfe verschiedener Werkzeuge. Inkscape ist ein freies Programm, das diese Grafiken im XML-basierten SVG-Format speichert. Gerade das Arbeiten mit den typischen Bézier-Kurven-Werkzeugen bedarf einiger Übung, ermöglicht dann aber äußerst präzises Zeichnen.
- ▶ *Der Webbrowser Firefox* ist der empfehlenswerteste unter den mit openSUSE gelieferten Browsern. Es handelt sich um einen schnellen, stabilen und modernen Webbrowser mit zahlreichen interessanten Funktionen. Falls Ihnen diese noch nicht genügen, können Sie mit Hilfe zahlloser Add-ons weitere hinzufügen.

Was ist denn lang und dauerhaft unter den menschlichen Dingen?
– Aristoteles

9 Datenträger und Dateisysteme

Die Verwaltung verschiedener Laufwerke und Datenträger zur dauerhaften Speicherung von Informationen ist eine zentrale Aufgabe der Betriebssysteme. In diesem Kapitel erfahren Sie alles über den Umgang mit Festplatten, CD- und DVD-Laufwerken, USB-Sticks, Disketten sowie diversen virtuellen Dateisystemen.

9.1 Festplatten

Der bisher wichtigste Massenspeicher im Computer ist die *Festplatte*. Sie besteht aus einem Stapel runder Metallplatten, die auf einer gemeinsamen drehbaren Achse angeordnet sind. Die gabelförmig übereinander angeordneten Schreib-/Leseköpfe fahren zwischen diesen Platten nach innen und wieder nach außen. Alle diese Komponenten sind vakuumverschweißt; dies ist wichtig, weil die Köpfe in einer Entfernung über den Platten schweben, die einem Bruchteil der Dicke eines menschlichen Haares entspricht – im Vergleich dazu hätte ein Staubkorn die verheerende Wirkung eines Felsbrockens.

Die Speicherregionen auf der Festplatte wurden früher nach einem Schema nummeriert, das man als *CHS*-Verfahren (Cylinder, Head, Sector) bezeichnet: Die Zylinder (alternative Bezeichnung: Spuren, englisch »tracks«) sind konzentrische Kreise auf den einzelnen Scheiben, die von innen nach außen durchnummeriert werden. Mit den Kopfnummern werden nacheinander die einzelnen Schreib-/Leseköpfe, das heißt die Seiten der Platten, bezeichnet. Die Sektoren schließlich sind Unterteilungen der Zylinder – sie haben die Form von »Kuchenstücken«, sodass die Datendichte nach außen immer weiter abnimmt und auf diese Weise Speicherplatz vergeudet, da jeder Sektor eine feste Größe von 512 Byte besitzt.

Insbesondere aufgrund dieses letzten Nachteils wurde für neuere Festplatten ein Verfahren entwickelt, das den verfügbaren Speicherplatz ökonomischer nutzt: Die Zylinder werden individuell in unterschiedlich viele, aber gleich große Sektoren unterteilt. Die Sektoren der aufeinanderfolgenden Platten werden fortlau-

fend durchnummeriert. Dieses modernere Verfahren heißt *LBA* (Logical Block Addressing). Abbildung 9.1 zeigt den Unterschied zwischen CHS und LBA schematisch an je einer einzelnen Platte.

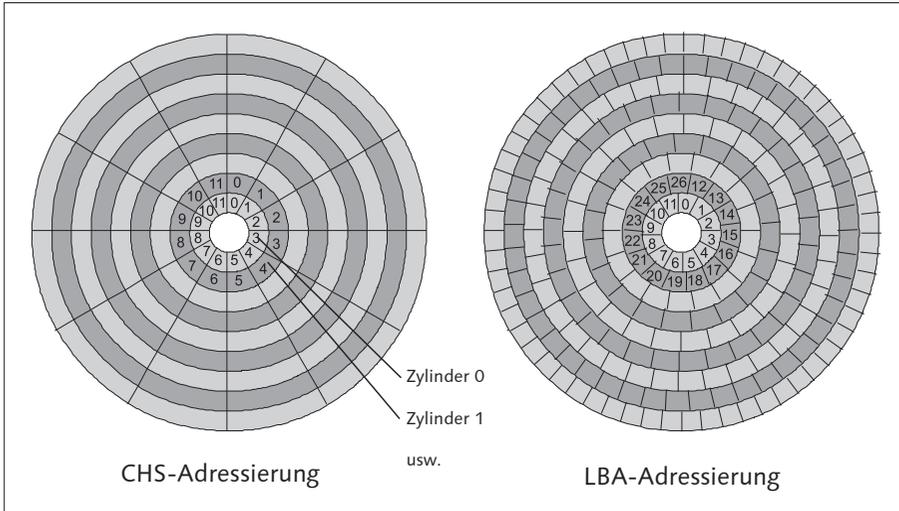


Abbildung 9.1 Die Festplatten-Adressierungsverfahren CHS und LBA im (schematischen) Vergleich

Die Größe aktueller Festplatten liegt üblicherweise zwischen 250 und 1.000 Gigabyte. Es werden aber auch wesentlich kleinere (ab etwa 60 GB) angeboten; insbesondere in der kleineren 2,5-Zoll-Bauform für Notebooks. Beachten Sie, dass die Angaben der Festplattenhersteller, die beispielsweise »250 GB« lauten, dezimal und daher irreführend sind: Es sind keine echten 250 Gigabyte (also $250 \times 1.024 \times 1.024 \times 1.024$ Byte) gemeint, sondern nur 250 Milliarden Byte – dies entspricht knapp 233 Gigabyte; zudem geht stets etwas Platz für die Formatierungslogik verloren.

Um die Geschwindigkeit von Festplatten zu messen, werden unterschiedliche Werte verwendet. Die erste wichtige Information ist die Umdrehungsgeschwindigkeit. Sie beträgt bei den meisten aktuellen Festplatten 7.200 U/Min.; einige ältere Modelle arbeiten nur mit 5.400 U/Min., während manche neuere Platten mit 10.000 U/Min. laufen. Es handelt sich hier um die Angabe einer konstanten Winkelgeschwindigkeit (Constant Angular Velocity oder kurz CAV). Bei der klassischen CHS-Adressierung werden Daten auf den Außenbezirken der Platte deutlich langsamer gelesen und geschrieben als innen, weil dieselbe Datenmenge auf einem erheblich längeren Ringabschnitt untergebracht ist.

Alle anderen Geschwindigkeitsangaben sind statistische Werte und werden durch Benchmarktests ermittelt. Die mittlere Zugriffszeit gibt an, wie lange es dauert, einen zufällig gewählten Sektor anzusteuern – man verwendet sehr viele Messungen, um die unterschiedlichsten aufeinanderfolgenden Sektorpositionen zu messen. Die Größenordnung der Werte liegt bei einem Bereich von 10 Millisekunden. Neben der mittleren Zugriffszeit wird noch die Geschwindigkeit des Schreib- und des Leseflusses gemessen. Diese Werte liegen bei aktuellen S-ATA-Festplatten zwischen 30 und 40 MByte/s.

Zu guter Letzt sind alle modernen Festplatten mit einem internen Cache ausgestattet, dessen Größe meist zwischen 2 und 8 MByte beträgt. Er beschleunigt Zugriffe auf die Festplatte durch Pufferung des Datenstroms. Sie können davon ausgehen, dass ein größerer Cache die Schreib- und Lesezugriffe noch ein wenig schneller macht.

Solid State Disks (SSD)

In letzter Zeit beginnt sich eine Alternative zur mechanischen Festplatte zu etablieren: Die Solid State Disk (SSD) besitzt nach außen einen gewöhnlichen Festplattenanschluss wie S-ATA, besteht jedoch intern aus einem Flash-ROM-Speicher, wie er auch für USB-Sticks oder Speicherkarten zum Einsatz kommt; daher übertrifft ihre Geschwindigkeit diejenige von Festplatten. Zurzeit sind SSDs noch teurer als herkömmliche Festplatten und besitzen eine geringere Speicherkapazität als diese, aber auf Dauer ist davon auszugehen, dass sie das Standardspeichermedium für PCs werden.

9.1.1 Partitionierung und Dateisysteme

Eine *Partition* ist eine Unterteilung der Festplatte, die vom Betriebssystem wie ein eigenständiges Laufwerk behandelt wird. Falls Sie mehrere Betriebssysteme auf demselben Rechner installieren möchten, sind Sie in den meisten Fällen sogar zur Partitionierung gezwungen; wie bereits im Rahmen der Installation erläutert wurde, kommt ein vernünftiges Linux-System auch allein nicht ohne Partitionierung aus.

Eine PC-Festplatte kann in bis zu vier Partitionen unterteilt werden. Eine von ihnen kann eine sogenannte erweiterte Partition sein, die wiederum bis zu 16 logische Laufwerke enthalten kann. Ein logisches Laufwerk hat aus der Sicht des Betriebssystems dieselben Eigenschaften wie eine Partition. Die normalen, nicht weiter unterteilbaren Partitionen werden als primäre Partitionen bezeichnet.

Das YaST-Modul zur Partitionierung wurde bereits in Kapitel 2, »openSUSE installieren«, recht ausführlich besprochen. Hier werden nur zwei Themen vertieft: der Aufbau der Partitionstabelle und die verschiedenen Dateisysteme. In

den nachfolgenden Unterabschnitten geht es zudem um das Kommando `mount`, um die Wiederherstellung versehentlich gelöschter Dateien und Verzeichnisse sowie um SoftRAID und LVM.

Beachten Sie an dieser Stelle noch, dass Sie Partitionen nicht modifizieren können, solange sie gemountet sind. Verwenden Sie dazu das Kommando `umount` (siehe nächster Unterabschnitt). Eventuell wird die jeweilige Partition zur Ausführung der grafischen Benutzeroberfläche oder anderer Systemdienste benötigt. In diesem Fall müssen Sie gegebenenfalls mittels

```
# init 2
```

in den Textmodus oder sogar mit

```
# init 1
```

in den Single-User-Modus wechseln, bevor Sie die Partition unmounten können. Beachten Sie, dass im letzteren Runlevel möglicherweise nicht alle Dateisystemtreiber zur Verfügung stehen – gegebenenfalls müssen Sie diese mittels `modprobe` nachladen.

Die unter `/` gemountete Root-Partition kann im laufenden Betrieb übrigens *gar nicht* abgekoppelt werden. Wenn Sie diese modifizieren möchten, müssen Sie von einem externen Datenträger booten – beispielsweise mit der weiter unten angesprochenen Rettungskonsole, die sich auf der openSUSE-Installations-DVD befindet.

Die Partitionstabelle

Damit Sie die in Kapitel 2 beschriebenen Partitionsvorgänge besser verstehen können, soll hier kurz auf die technischen Hintergründe der Festplattenpartitionierung eingegangen werden. Die Partitionsdaten werden auf der Festplatte selbst in einer sogenannten *Partitionstabelle* gespeichert; das YaST-Tool `SYSTEM • PARTITIONIERUNG` bearbeitet die Einträge in dieser Tabelle.

Der allererste Sektor einer jeden Festplatte – CHS 0,0,1 oder LBA 0 – ist der *Haupt-Bootsektor* (englisch »Master Boot Record« oder kurz »MBR«). In Tabelle 9.1 wird sein Aufbau gezeigt.

Startadresse	Maximale Länge (Byte)	Inhalt
0x0000 (0)	440	Bootloader
0x01B8 (440)	4	Disk-Signatur

Tabelle 9.1 Schematischer Aufbau eines Master Boot Records

Startadresse	Maximale Länge (Byte)	Inhalt
0x01BC (444)	2	Zwei Nullbytes (0x0000)
0x01BE (446)	64	Partitionstabelle
0x01FE (510)	2	MBR-Signatur (Magic Number)

Tabelle 9.1 Schematischer Aufbau eines Master Boot Records (Forts.)

Der Bootloader wird beim Start des PCs von dessen BIOS aufgerufen. Er dient dazu, das Betriebssystem zu starten oder – als Bootmanager – eines von mehreren zur Auswahl zu stellen. Unter Linux gibt es zwei verschiedene gängige Bootmanager, deren Starter an dieser Stelle residieren kann: GRUB und LILO. Beide werden weiter unten in diesem Kapitel erläutert.

Die eigentliche Partitionstabelle beginnt bei Byte 446 (hexadezimal 0x01BE) und ist maximal 64 Byte groß – für jeden der maximal vier Partitionseinträge stehen also 16 Byte zur Verfügung. Der Aufbau eines solchen Eintrags wird in Tabelle 9.2 gezeigt.

Relative Startadresse	Länge (Byte)	Inhalt
0x00 (0)	1	aktive (bootfähige) Partition: 0x80; nicht aktive: 0x00
0x01 (1)	3	Startsektor der Partition (CHS)
0x04 (4)	1	Partitionstyp (siehe unten)
0x05 (5)	3	Endsektor der Partition (CHS)
0x08 (8)	4	Startsektor (LBA oder relativ)
0x0C (12)	4	Anzahl der Sektoren

Tabelle 9.2 Aufbau eines Eintrags in einer Partitionstabelle

Das Byte 0x04 eines Partitionseintrags legt den Partitionstyp fest. Die wichtigsten bekannten Typen sind:

- ▶ 0x00: leer
- ▶ 0x01: FAT12 (Disketten)
- ▶ 0x04: FAT16 (bis 32 MB)
- ▶ 0x05: erweiterte Partition
- ▶ 0x06: FAT16 (ab 32 MiB)

- ▶ 0x07: HPFS (OS/2), NTFS (Windows NT)
- ▶ 0x0B: FAT32
- ▶ 0x0C: FAT32, BIOS-Erweiterung
- ▶ 0x0E: FAT16 (ab 32 MB, BIOS-Erweiterung)
- ▶ 0x0F: erweiterte Partition, BIOS-Erweiterung
- ▶ 0x12: EISA-Partition
- ▶ 0x42: dynamisches Volume
- ▶ 0x82: Linux Swap
- ▶ 0x83: Linux native
- ▶ 0x8E: Linux LVM
- ▶ 0xA0: Hibernation
- ▶ 0xA5: FreeBSD
- ▶ 0xA6: OpenBSD
- ▶ 0xA9: NetBSD
- ▶ 0xFD: RAID

Wie bereits in Kapitel 2 erwähnt, unterstützt das YaST-Partitionierungstool von Haus aus nur die Partitionstypen 0x0C (FAT32), 0x0F (erweiterte Partition), 0x82 (Swap), 0x83 (Linux native), 0x8E (LVM), 0xA0 (Hibernation) und 0xFD (RAID).

Wie Sie bereits wissen, können Partitionen sogenannte erweiterte Partitionen sein. Diese enthalten den Partitionstyp 0x0F. Am angegebenen Startsektor befindet sich eine Partitionstabelle mit zwei Einträgen: eine logische Partition (die ein beliebiges Dateisystem enthalten kann) sowie ein Verweis auf eine weitere Partitionstabelle nach diesem Schema. Auf diese Weise entsteht eine verkettete Liste logischer Partitionen.

Unter Linux werden die vier Grundpartitionen (egal ob primär oder erweitert) stets von 1 bis 4 durchnummeriert; die erste logische Partition besitzt stets die Nummer 5, auch wenn weniger primäre Partitionen vorhanden sind. EIDE- oder S-ATA-Festplatten heißen `hda`, `hdb` und so weiter, während SCSI- und SAS-Platten als `sda`, `sdb` und folgende bezeichnet werden. Somit heißt beispielsweise die erste Partition der zweiten EIDE-Platte `hdb1`, während die zweite logische Partition auf der ersten SCSI-Festplatte als `sda6` bezeichnet wird.

Da der Master Boot Record den Bootloader und die Partitionstabelle enthält, lohnt es sich, diesen zu sichern, beispielsweise auf eine Diskette. Dazu kann das Kommando `dd` verwendet werden, das Bytes direkt von einer (auch ungemoun-

teten) Gerätedatei lesen oder auf diese schreiben kann. Die folgende Anweisung kopiert den MBR der ersten EIDE-Festplatte in eine Datei:

```
# dd if=/dev/hda of=mbrcopy bs=512
```

Wie Sie eine Diskette mounten können, um den Bootsektor darauf zu kopieren, steht weiter unten im Unterabschnitt über das Kommando `mount`.

Wenn ein Unfall passiert ist (etwa ein Stromausfall zum falschen Zeitpunkt) und Sie die Partitionstabelle zurückschreiben möchten, wird das umgekehrte Kommando verwendet:

```
# dd if=mbrcopy of=/dev/hda bs=512
```

Dateisysteme

In Kapitel 2 wurden bereits einige von openSUSE unterstützte Dateisysteme angesprochen. Hier erhalten Sie einen etwas ausführlicheren Überblick über die konkreten Dateisysteme und ihre Fähigkeiten.

► Ext2

Das *Linux Extended File System 2* ist der Klassiker unter den Linux-Dateisystemen – bis zum Aufkommen der Journaling-Dateisysteme wurde es beinahe exklusiv verwendet. Das heutzutage nicht mehr unterstützte erste Linux Extended File System (Ext1) wurde geschrieben, um die wichtigsten Einschränkungen des ursprünglich verwendeten Minix-Dateisystems zu umgehen: 16-Bit-Adressierung, die zu einer maximalen Datei- und Partitionsgröße von 64 Megabyte (!) führte, sowie Dateinamen von maximal 16 und später 30 Zeichen.

Ext2 erweiterte die Fähigkeiten von Ext1 nochmals; vor allem unterstützte es maximale Partitionsgrößen von 4 Terabyte statt der 2 Gigabyte des Ext1-Dateisystems. Ansonsten wurden beide mit allen typischen Merkmalen eines UNIX-Dateisystems ausgestattet, insbesondere inodes zur internen Repräsentation der Datei-Metainformationen, Hard Links und Symlinks sowie die Unterstützung von Gerätedateien.

Wie bereits in Kapitel 2 erwähnt, brauchen Sie Ext2 heutzutage in der Regel nicht mehr als neues Dateisystem aufzusetzen; es wird eher aus Kompatibilitätsgründen unterstützt. Wenn Sie eine Partition auch von einem anderen Betriebssystem aus nutzen möchten, das Ext2 unterstützt, können Sie diese auch mit Ext3 formatieren – es ist voll abwärtskompatibel und kann von standardkonformen Ext2-Treibern gelesen und beschrieben werden. Das von Ext2 nicht unterstützte Journal wird dann beim nächsten Mounten mit dem Ext3-Treiber aktualisiert.

► *Ext3 und Ext4*

Das Linux Extended File System 3 ist der direkte Nachfolger von Ext2 und wie bereits erwähnt voll abwärtskompatibel zu diesem. Die wichtigste Neuerung ist die Implementierung eines *Journals*. Journaling-Dateisysteme sind sicherer als herkömmliche, weil sie alle Änderungen in Echtzeit protokollieren. Dabei kann zwischen einem klassischen Journal gewählt werden, das nur Metadaten enthält, oder einem modernen mit Metadaten und Nutzdaten. Nach einem Absturz, etwa durch Stromausfall, lässt sich der letzte stabile Zustand der Partition durch Zurückspielen des Journals wiederherstellen, meistens recht schnell und ohne Datenverluste. Bei klassischen Dateisystemen ist dagegen stets eine systematische Fehlersuche in der gesamten Partition (`fsck`, siehe unten) erforderlich.

Inzwischen ist der Nachfolger *Ext4* fertig. Neben diversen Performance- und Sicherheitsverbesserungen werden vor allem erheblich größere Partitionen unterstützt; die – zurzeit noch eher theoretische – Maximalgröße beträgt 1 Exabyte (1.024 Petabyte oder 1.048.576 Terabyte).

Ext4 ist das bevorzugte Dateisystem von openSUSE; es wird standardmäßig zur Formatierung der nativen Linux-Partitionen verwendet, wenn Sie sich nicht anderweitig entscheiden.

► *ReiserFS*

Das Dateisystem ReiserFS bietet seit Version 3 ebenfalls ein Journal. Anfangs wurden nur Metadaten protokolliert, in der aktuellen Version haben Sie genau wie bei Ext3 die Wahl. ReiserFS war das Standard-Dateisystem früherer SUSE-Linux-Versionen bis 10.1.

Ein gewisser Vorteil von ReiserFS gegenüber Ext3 besteht darin, dass dieses Dateisystem den verfügbaren Plattenplatz besser ausnutzt: Vor allem kleine Dateien können in den Verwaltungs-Clustern der inodes selbst gespeichert werden, zudem können sich Teile mehrerer Dateien eine Zuordnungseinheit teilen, während sie bei anderen Dateisystemen stets ganze Cluster belegen. Auch die Performance von ReiserFS ist in manchen Fällen besser als die von Ext3 – gerade bei der Arbeit mit vielen kleinen Dateien, wie sie etwa bei Web- oder Mail-Servern anfallen.

► *XFS*

Das dritte unter openSUSE verfügbare Journaling-Dateisystem wurde von Silicon Graphics für das hauseigene UNIX-Derivat IRIX entwickelt. Da es zur Zeit seiner Entwicklung das erste Journaling-Dateisystem für UNIX-Systeme war, erlangte es anfangs eine gewisse Verbreitung. Heute überwiegen eher die Nachteile: Das Journal speichert ausschließlich Metadaten und keine Nutzdaten, und es ist relativ schwierig, versehentlich gelöschte Daten auf XFS-Volumen wiederherzustellen.

► *FAT16*

Dies ist das ursprüngliche Dateisystem von MS-DOS und Windows 3.11 – für Disketten kommt sogar eine noch eingeschränktere Variante namens FAT12 zum Einsatz. »FAT« ist die Abkürzung für File Allocation Table (Dateizuordnungstabelle). In Form einer solchen Tabelle speichert dieses Dateisystem jeweils die Nummer des ersten Clusters, bei dem eine bestimmte Datei beginnt.

Da es sich bei FAT16 um ein 16-Bit-Dateisystem handelt, beträgt die maximale Anzahl von Zuordnungseinheiten auf einer Partition 65.536. Dies macht große Datenträger sehr ineffizient, da jede Datei immer ganze Zuordnungseinheiten belegt. Darüber hinaus ist die Größe einer FAT16-Partition auf 2 Gigabyte begrenzt.

► *FAT32*

Dieses Dateisystem besitzt exakt dieselbe Funktionsweise wie FAT16. Durch die 32-Bit-Adressierung wurde allerdings die Anzahl der Cluster pro Partition auf über 4 Milliarden erhöht, die Gesamtgröße einer Partition kann theoretisch 4 Terabyte betragen. Da alle gängigen Betriebssysteme mit FAT32 umgehen können, ist es das Standard-Dateisystem für USB-Sticks und die diversen Flash-Speicherkarten, die beispielsweise für Digitalkameras zum Einsatz kommen.

► *NTFS*

Das für Windows NT neu geschaffene New Technology File System ist eine Weiterentwicklung des von IBM und Microsoft entwickelten OS/2-Dateisystems HPFS. NTFS existiert in verschiedenen, leicht unterschiedlichen Versionen. Die Zuordnungseinheiten werden nicht mehr in einer einfachen Tabelle verwaltet, sondern in einer komplexen Baumstruktur, die erheblich schnellere Zugriffe ermöglicht und mehr Schutz vor Fehlern bietet. Des Weiteren bietet das Dateisystem eine UNIX-ähnliche Verwaltung von Benutzerrechten für den Eigentümer, die Gruppe und sonstige Benutzer sowie transparente Komprimierung und weitere Funktionen.

Lange Zeit konnten NTFS-Partitionen unter Linux nur gelesen, aber nicht geschrieben werden. Deshalb griffen die meisten User, auf deren Rechner Linux und Windows parallel installiert sind, in der Regel zu einer FAT32-Partition für den Austausch zwischen den beiden Systemen. Heute ist der Schreibzugriff auf zweierlei Art und Weise möglich: Die nativen Treiber des Kernels 2.6 ermöglichen eine Veränderung der vorhandenen Dateien im NTFS-Dateisystem, aber keine Neuanlage von Dateien. *Captive NTFS* erzeugt dagegen einen Windows-Emulations-Layer, in dem der Original-Microsoft-Treiber ausgeführt wird. Beachten Sie, dass Sie dazu die Datei *ntfs.sys* einer

vorhandenen Windows-Version (NT 4.0, 2000 oder XP) benötigen, da sie aus rechtlichen Gründen nicht mit openSUSE oder anderen Linux-Distributionen ausgeliefert werden darf.

► *ISO 9660*

Dies ist das ursprüngliche CD-ROM-Dateiformat für PCs. Für die bei seiner Einführung vorherrschenden MS-DOS- und Windows-3.1-Rechner waren seine vielfältigen Einschränkungen kein Problem: Genau wie ISO 9660 konnten auch deren Dateisysteme beispielsweise nur mit kurzen Dateinamen im »8.3«-Format umgehen: bis zu 8 Zeichen für den eigentlichen Dateinamen und 3 Zeichen für die durch einen Punkt getrennte Erweiterung, die unter Windows den Dateityp angibt. Erweiterungen wie Microsofts Joliet haben später dafür gesorgt, das ISO-Dateisystem auf abwärtskompatible Art und Weise mit langen Dateinamen auszustatten. Neuere Daten-CDs – und -DVDs erst recht – verwenden dagegen häufig ein Dateisystem, das UDF (Universal Disk Format) genannt wird und solche fortgeschrittenen Features von Hause aus beherrscht.

Daneben gibt es beispielsweise noch die Netzwerkdateisysteme NFS und SMB, die in Kapitel 14 besprochen werden, sowie einige virtuelle Dateisysteme, über die Sie weiter unten in diesem Kapitel etwas erfahren.

Über Floppy-Laufwerke und Disketten

Weitaus die meisten PCs und alle Macs werden heute ohne das klassische Diskettenlaufwerk ausgeliefert. Das 1980 von Sony entwickelte 3,5-Zoll-Laufwerk besitzt in seiner letzten, bis heute gültigen Spezifikation eine maximale Speicherkapazität von 1,44 Megabyte (2 Seiten × 80 Spuren × 18 Sektoren × 512 Byte pro Sektor) und ist nach wie vor recht langsam.

Eine Diskette oder Floppy Disk ist eine mit magnetisierbarem Metallstaub versetzte runde Kunststoffscheibe, die zum Schutz in einer quadratischen Kunststoffhülle steckt, die vollständig in das Laufwerk geschoben wird. Die 3,5-Zoll-Diskette ist mit einem kleinen Schiebeschalter ausgestattet; wird er geöffnet, dann ist die Diskette schreibgeschützt, in geschlossenem Zustand kann sie dagegen beschrieben werden.

Wenngleich Diskettenlaufwerke also kurz vor dem Aussterben sind, können sie als Nothelfer gute Dienste leisten – etwa wenn Sie, wie oben beschrieben, den Master Boot Record einer Festplatte darauf sichern.

Falls Ihr Rechner kein Diskettenlaufwerk mehr besitzt (und auch, falls er eines hat), können Sie sich heutzutage viel besser mit einem USB-Stick behelfen. Diese werden auch unter openSUSE meistens automatisch erkannt und auf dem Desktop bereitgestellt; andernfalls können Sie sie leicht mit `mount` (siehe unten) in Ihren Verzeichnisbaum einbinden.

9.1.2 mount, umount und /etc/fstab

Wie bereits erwähnt, wird das Kommando `mount` verwendet, um Partitionen oder Wechseldatenträger in das Dateisystem einzubinden. Die grundlegende Syntax lautet folgendermaßen:

```
mount [Optionen] Gerätedatei Verzeichnis
```

Das angegebene Verzeichnis muss existieren und leer sein – notfalls müssen Sie mittels `mkdir` ein neues erzeugen. Die wichtigsten Gerätedateien sind folgende:

- ▶ `/dev/hda` bis `/dev/hdd` sind die vier EIDE-Geräte in der Reihenfolge Primary Master, Primary Slave, Secondary Master und Secondary Slave; auch S-ATA-Geräte werden auf diese Weise angesprochen. Falls es sich um Festplatten handelt, muss zusätzlich die oben beschriebene Nummer der gewünschten Partition angegeben werden.
- ▶ `/dev/cdrom` ist eine alternative Methode, das CD-ROM-Laufwerk anzusprechen.
- ▶ `/dev/sda`, `/dev/sdb` und so weiter sind die entsprechenden SCSI- oder SAS-Festplatten in der Reihenfolge ihrer SCSI-IDs. Auch USB-Datenträger (USB-Sticks und externe Festplatten) besitzen solche Gerätenamen, da sie über modifizierte SCSI-Treiber angesprochen werden.
- ▶ `/dev/fd0` ist das Diskettenlaufwerk. Theoretisch steht `fd1` für das zweite, allerdings verfügt heute praktisch kein Rechner mehr über zwei Diskettenlaufwerke.

Die wichtigste Option des Befehls ist zweifellos die Angabe `-t Dateisystem`. Bei Standarddateisystemen kann sie mitunter weggelassen werden, aber in der Regel ist es sicherer, sie hinzuzufügen. Hier einige Beispiele:

- ▶ `ext3`
- ▶ `ext4`
- ▶ `reiserfs`
- ▶ `ntfs`
- ▶ `fat32`
- ▶ `nfs`
- ▶ `smbfs` beziehungsweise `cifs`

Die beiden letzten dienen dem Mounten externer Netzwerkfreigaben über das Network File System (UNIX-Klassiker) beziehungsweise eine Windows- oder Samba-Freigabe (SMB). Beide Dateisysteme werden in Kapitel 14, »Datei- und Verzeichnisdienstserver«, vertieft.

Das folgende Beispiel bindet die aktuell im Diskettenlaufwerk befindliche Diskette als Verzeichnis */disk* in das Dateisystem ein:

```
# mount /dev/fd0 /disk
```

Hier ein weiteres Beispiel; es bindet eine ReiserFS-Festplattenpartition ein:

```
# mount -t reiserfs /dev/hdb2 /test
```

Das letzte Beispiel schließlich mountet eine SMB-Freigabe (von einem Windows-Rechner oder aber einem Linux-PC, der den Windows-kompatiblen Dateiserver Samba ausführt):

```
# mount -t cifs //winbox/share /netzordner
```

Beachten Sie den doppelten Slash (//) vor dem Namen der Freigabe; er wird zur Kennzeichnung von Remote-Rechnern eingesetzt.

Wenn Sie `mount` ohne jegliche Parameter eingeben, wird eine Liste aller zurzeit eingebundenen Dateisysteme angezeigt.

Das Kommando `umount` (nicht etwa `UNmount`, da so lange Programmnamen zur Zeit seiner Entstehung noch nicht erlaubt waren) entfernt die Verknüpfung mit einem bestimmten Volume wieder. Vorher wird noch dafür gesorgt, dass alle Daten geschrieben werden, die eigentlich auf dieses Laufwerk gehören, sich aber derzeit im RAM-Cache befinden. Vor dem Entfernen von Wechseldatenträgern sollten Sie `umount` auf jeden Fall aufrufen.

Als Argument für den Befehl sollte bevorzugt das Verzeichnis angegeben werden, in dem der gewünschte Datenträger gemountet ist; notfalls können Sie aber auch die Gerätedatei selbst angeben. Das folgende Beispiel entkoppelt das Gerät, das unter */test* eingebunden wurde:

```
# umount /test
```

Die Systemkonfigurationsdatei */etc/fstab* enthält eine Liste derjenigen Laufwerke, die beim Systemstart automatisch gemountet werden. Zwei typische Einträge sehen beispielsweise so aus:

```
/dev/hda1 / reiserfs acl,usr_xattr 1 1
/dev/fd0 /media/floppy auto noauto,user,sync 0 0
```

Die Spalten enthalten von links nach rechts folgende Angaben:

► *Gerätepfad*

Die Kennung des einzubindenden Dateisystems. Übliche Formate sind */dev/** für lokale Laufwerke oder Partitionen, *//Host/Pfad* für Remote-Netzwerkfrei-

gaben. Zusätzlich existieren einige besondere Angaben für Spezialdateisysteme, etwa `proc` für das weiter unten beschriebene virtuelle Prozess-Dateisystem oder `usbfs` für die Überwachung der USB-Anschlüsse auf Hot-Plugging-Geräten.

► *Mountpoint*

Der Verzeichnispfad, unter dem das jeweilige Device eingebunden werden soll.

► *Dateisystem*

Eine Liste der wichtigsten Angaben, die hier stehen können, finden Sie oben unter der Beschreibung von `mount`. Die Spezialangabe `auto` bei Wechseldatenträgern wie dem oben gezeigten Diskettenlaufwerk `/dev/fd0` ist notwendig, weil diese unterschiedliche Dateisysteme enthalten können. Falls die automatische Erkennung scheitert, müssen Sie die betreffenden Volumes in Einzelfällen manuell mounten.

► *Optionen*

Hier können Sie eine Reihe genauerer Angaben darüber machen, wie das jeweilige Volume eingebunden werden soll. Die wichtigsten im Überblick (die Standardangabe steht jeweils vorn):

- `rw|ro`: Soll das Volume im Schreib- und Lesemodus (`rw`) oder schreibgeschützt (`ro`) gemountet werden?
- `atime|noatime`: Sollen Datum und Uhrzeit des letzten Zugriffs aktualisiert werden (`atime`) oder nicht (`noatime`)?
- `nouser|user`: Darf dieses Volume nur durch `root` gemountet werden (`nouser`) oder auch durch normale Benutzer (`user`)?
- `auto|noauto`: Soll das Volume automatisch beim Systemstart gemountet werden (`auto`) oder nicht (`noauto`)?
- `defaults`: Kurzfassung für die Standardwerte `rw,atime,nouser,auto`.
- `async|sync`: Sollen Schreibzugriffe gecachet werden, bis genug Zeit zum Schreiben ist (`async`), oder soll alles sofort geschrieben werden (`sync`)? Der asynchrone Zugriff verbessert verständlicherweise die Performance, während der synchrone sicherer ist und in Einzelfällen für journalfreie Dateisysteme (oder, wie oben gezeigt, für das Diskettenlaufwerk) eingesetzt wird.
- `acl`: Access Control Lists (Zugriffskontrolllisten) können zusätzlich zur klassischen UNIX-Rechteverwaltung aufgesetzt werden. Zusätzlich zu den Rechten »sonstiger Benutzer« regeln sie im Detail, welcher Drittbenedutzer auf welche Weise auf eine Ressource zugreifen darf.

- ▶ *Dump*

In dieser Spalte steht der Wert »1«, falls der Backup-Klassiker `dump` diese Partition in die Datensicherung einbeziehen soll, oder »0«, wenn dem nicht so ist. Auch modernere Backup-Programme lesen diesen Wert aus.

- ▶ *Fsck-Reihenfolge*

Beim Systemstart werden die einzubindenden Dateisysteme routinemäßig mittels `fsck` (siehe unten) überprüft. Die Einträge in dieser Spalte regeln, in welcher Reihenfolge dies geschehen soll; der spezielle Wert »0« bedeutet, dass das betreffende Volume gar nicht geprüft wird.

9.1.3 Weitere Dateisystem-Kommandos

Hier noch einige weitere Kommandos zur Behandlung von Dateisystemen und Verzeichnissen im Überblick:

- ▶ `du` (»disk usage«) gibt in Kilobyte an, wie viel Speicher in den angegebenen Verzeichnissen belegt ist. Dabei werden alle Unterverzeichnisse rekursiv durchwandert und ihre Größen addiert, sodass am Schluss je eine Summe für die übergeordneten Knoten steht.

Die Option `-c` gibt zusätzlich eine Gesamtsumme für die verschiedenen Verzeichnisse aus, während `-h` (»human-readable«) das Ganze in angenehmerer Form, das heißt je nach Verzeichnisgröße auch in Megabyte oder Gigabyte, angibt.

- ▶ `fsck` (»file system check«) überprüft den angegebenen Datenträger auf Fehler. Als Argument muss eine Partition im Geräteformat wie etwa `/dev/hda1` angegeben werden. Beachten Sie, dass gemountete Dateisysteme zwar überprüft, aber nicht repariert werden können. Sie müssen sie zunächst unmounten; im Falle der Root-Partition müssen Sie sogar von einem externen Volume starten. Vor jedem eventuellen Reparaturversuch wird nachgefragt.

`fsck` wird bei jedem Systemstart aufgerufen, um die Volumes aus `/etc/fstab` zu überprüfen. In regelmäßigen Abständen (zum Beispiel jedes zwanzigste Mal) werden die Volumes dabei automatisch einer kompletten Prüfung unterzogen. Außerdem wird `fsck` auf jeden Fall nach einem kompletten Systemabsturz (meist durch Stromausfall) gestartet.

- ▶ `mkfs` (»make file system«) dient dem Formatieren eines Datenträgers mit einem bestimmten Dateisystem. Der gewünschte Dateisystemtyp kann mithilfe der Option `-t` angegeben werden. Beispiele sind etwa `ext2`, `ext3` oder `reiserfs`. Der Datenträger wird als Gerätedatei oder über sein Mount-Verzeichnis angegeben.

Im Grunde handelt es sich um einen Starter für die konkreten Formatierprogramme der einzelnen Dateisysteme wie `mke2fs`, `mke3fs` oder `mkreiserfs`. Das besondere Programm `mkisofs` erzeugt ISO-9660-Images. Sie können diese Images über das Loop-Device (siehe unten) mounten und Daten hineinkopieren, bevor Sie die fertige Image-Datei auf CD oder DVD brennen (auch dies wird weiter unten beschrieben).

9.1.4 SoftRAID

Die Verwendung sogenannter *RAID*-Systeme ist häufig im Serverbereich anzutreffen. Das *Redundant Array of Independent Disks* (manchmal auch »Inexpensive Disks«) fasst mehrere physikalische Festplatten oder Partitionen zu einer Einheit zusammen, entweder aus Performancegründen oder für eine Verbesserung der Datensicherheit. RAID wird entweder durch eine spezielle Hardwarekomponente namens RAID-Controller bereitgestellt, oder es wird als SoftRAID über Treiber des Betriebssystems zur Verfügung gestellt – Linux enthält die entsprechende Unterstützung als Kernelmodul, das sich über YaST konfigurieren lässt.

Man unterscheidet acht verschiedene RAID-Levels, die von 0 bis 7 durchnummeriert werden und sich bezüglich der Art und Weise, wie sie die einzelnen Festplatten nutzen, voneinander unterscheiden; daneben gibt es diverse Mischungen aus diesen Haupt-Levels. Die wichtigsten Levels sind 0, 1 und 5:

- ▶ *RAID Level 0: Stripe Set.* Die Speicherkapazitäten mehrerer Festplatten werden in einzelne »Streifen« zerschnitten. Daten werden abwechselnd auf den verschiedenen Platten gespeichert, wobei die Streifen jeder Platte fortlaufend beschrieben werden. Diese Methode steigert nur die Performance des Systems, aber nicht die Sicherheit. Sie benötigen mindestens zwei gleich große Festplatten, deren gesamte Kapazität dann zur Verfügung steht.
- ▶ *RAID Level 1: Mirroring.* Die Daten einer ganzen Festplatte werden jeweils komplett auf einer zweiten Platte gespeichert. Diese Variante bietet die höchstmögliche Sicherheit, aber keinerlei Performancesteigerung. Hier werden ebenfalls zwei gleich große Festplatten benötigt, wobei die verfügbare Kapazität natürlich nur derjenigen einer der beiden Platten entspricht.
- ▶ *RAID Level 5: Stripe Set mit Parity.* Diese RAID-Version bietet einen Kompromiss zwischen Geschwindigkeits- und Sicherheitsverbesserung: Es wird ein Stripe Set erzeugt wie bei Level 0; allerdings wird zusätzlich ein Bereich auf jeder Platte für Prüfsummen (Parity) verwendet, um nachträglich Fehler beseitigen zu können. Für Level 5 brauchen Sie drei gleich große Festplatten; die verfügbare Kapazität entspricht derjenigen von zweien.

Um SoftRAID zu nutzen, benötigen Sie mehrere unformatierte Partitionen mit dem Typ 0xFD (RAID) oder 0x83 (Linux native), die möglichst gleich groß sein sollten (andernfalls bestimmt die kleinste von ihnen den nutzbaren Platz). Damit das RAID-System Ihnen wirklichen Nutzen bezüglich Performance und/oder Sicherheit bringt, sollten diese Partitionen zudem auf unterschiedlichen physikalischen Festplatten liegen.

Wählen Sie im YaST-Partitionierungsprogramm ANLEGEN, um die betreffenden Partitionen zu erstellen. Achten Sie darauf, dass die Option FORMATIEREN deaktiviert ist. Anschließend können Sie RAID • RAID ANLEGEN wählen und den detaillierten Anweisungen folgen, um Ihr SoftRAID-Array zu erstellen.

9.1.5 LVM

Der *Logical Volume Manager* ermöglicht die Verwaltung virtueller Volumes, die über die Grenzen eines einzelnen physischen Datenträgers hinausgehen. Praktischerweise können nachträglich physische Volumes hinzugefügt werden, um die logischen Volumes zu vergrößern. Wenn Sie eine LVM-Konfiguration vornehmen möchten, können Sie die gleichnamige Schaltfläche in der YaST-Partitionsverwaltung betätigen.

Vor der ersten Verwendung von LVM müssen Sie möglicherweise das Paket `lvm2` nachinstallieren. YaST fragt automatisch nach und kann dies für Sie übernehmen, wenn Sie die LVM-Funktion starten.

9.2 CDs, DVDs und Blu-ray Discs

Zu den wichtigsten Wechseldatenträgern gehören heutzutage die optischen Volumes CD, DVD und zunehmend auch die Blu-ray Disc. In diesem Abschnitt erfahren Sie nach einer Einführung in die technische Spezifikation vor allem, wie Sie beschreibbare optische Medien unter openSUSE behandeln können.

9.2.1 Technische Grundlagen

Die eigentliche Informationsfläche einer CD, DVD oder Blu-ray Disc besteht jeweils aus einer hauchdünnen Metallschicht, die von einem Laserstrahl mit einer bestimmten Brennweite abgetastet werden kann. Die verschiedenen Farben (Gold, Silber, Blau oder Grün) kommen durch unterschiedliche Metallegierungen zustande. Vertiefungen in dieser Oberfläche (Pits) wechseln sich mit der normalen, unversehrten Fläche (Land) ab und bilden so das Bitmuster. Die unterschiedlichen Reflexionseigenschaften der Pit- und Land-Bereiche werden vom

Lesekopf des CD- oder DVD-Players als Daten, Musik oder Video interpretiert. Zum Schutz vor Beschädigung wird die empfindliche Metallschicht durch eine verhältnismäßig dicke Schutzschicht aus durchsichtigem Kunststoff abgedeckt.

Über CDs

Die *Compact Disc* (CD) wurde 1982 von Philips und Sony erfunden, zunächst als digitaler Tonträger (Audio-CD). Nach gut 10 Jahren war es der Musikindustrie gelungen, fast den gesamten Tonträgermarkt auf CDs umzustellen, denn mit ihrer kompakten Größe, ihrer Unempfindlichkeit, den guten Digitalaudio-Eigenschaften (44,1 kHz Sampling-Rate, 16 Bit Sampling-Tiefe, Stereo) und Platz für 74 Minuten Musik ist sie für den normalen Konsumentenmarkt besser geeignet als die klassische Vinyl-LP – auch wenn HiFi-Freaks behaupten, dass sie die bessere Qualität der Analogschallplatte hören können.

Es lag nah, dieses Speichermedium für die Speicherung von Daten zu verwenden. Zuerst gab es nur die CD-ROM, eine ab Werk mit vorgegebenem Inhalt versehene Daten-CD, die für die Lieferung von Software oder für Multimedia-Präsentationen verwendet wird. Seit etwa 1995 wird so gut wie jeder PC mit einem CD-ROM-Laufwerk ausgestattet, das diese Datenträger lesen kann – inzwischen handelt es sich mehrheitlich um DVD-Laufwerke, die auch CDs unterstützen.

Eine klassische CD-ROM besitzt eine Speicherkapazität von maximal 650 Megabyte, bei neueren Versionen sind es 700 oder gar 800. Da Audio-CDs mit weniger Verwaltungsdaten auskommen, passt etwas mehr Musik darauf, als die direkte Umrechnung dieser Werte in etwa 10 MByte pro Minute Musik vermuten lassen würde, nämlich 74, 80 oder 90 Minuten. Beachten Sie, dass ältere Audio-CD-Player keine CDs mit einer längeren Spieldauer als 74 Minuten abspielen können.

Die Spezifikationen für die verschiedenen CDs werden in den sogenannten *Rainbow Books* oder »bunten Büchern« dokumentiert:

- ▶ Das *Red Book* spezifiziert die Audio-CD, die aus einem Inhaltsverzeichnis (Table of Contents) und bis zu 99 Audiotracks besteht. Beachten Sie, dass eine Audio-CD, auf der die ärgerlichen Kopierschutzmaßnahmen der Musikindustrie verwendet werden, nicht dem Red Book entspricht und deshalb eigentlich gar keine richtige Audio-CD ist – erfreulicherweise werden inzwischen fast keine kopiergeschützten CDs mehr verkauft, da sie bei den Kunden zu Recht sehr schlecht ankamen.

Eine aktuelle Ergänzung der Audio-CD ist dagegen ein Format, das als CD-Text bezeichnet wird: Neben den reinen Audiodaten können kurze Textinformationen wie Interpreten und Titel in die Tracks geschrieben werden. Neue CD-

Player und CD-Abspielprogramme für Computer zeigen diese Informationen an; aktuelle Brennsoftware kann sie auch auf beschreibbare CDs schreiben.

- ▶ Das *Yellow Book* ist der Standard für die CD-ROM, die ursprüngliche Daten-CD. Sie war von Anfang an als Mixed-Mode-CD ausgelegt, kann also neben den Computerdaten auch Audiotracks enthalten.
- ▶ Im *Green Book* wurde das Format der Philips CDi (interactive) festgelegt; der CDi-Player war ein an den Fernseher anschließbares Gerät zum Abspielen von Multimedia-Präsentationen. Es kam kurz vor dem Siegeszug der Multimedia-CD-ROM und des Webs auf den Markt und verkaufte sich schlecht.
- ▶ Das *Orange Book* standardisiert die beschreibbare CD (CD-R, für »recordable«) und die wiederbeschreibbare CD (CD-RW, »ReWritable«).
- ▶ Das *White Book* definiert das Format der Video-CD (nicht etwa der DVD).
- ▶ Das *Blue Book* ist eine Erweiterung des Yellow-Book-Standards im Hinblick auf Mixed-Mode-CDs: Es wird genauer festgelegt, wie Audio- und Daten-Tracks aufeinanderfolgen sollen, damit die CD sowohl von einem alten Audio-CD-Player als auch von einem modernen CD-ROM-Laufwerk in einem Computer abgespielt werden kann.

Die Geschwindigkeit eines CD-ROM-Laufwerks wird als Vielfaches der Datenübertragungsrate eines Audio-CD-Players angegeben: Audio-CDs werden mit 150 KByte/s abgespielt; ein 48x-CD-ROM-Laufwerk schafft entsprechend $48 \times 150 = 7.200$ KByte/s oder etwa 7,03 MByte/s. Anders als bei den Festplatten handelt sich bei diesen Geschwindigkeitsangaben um eine konstante lineare Geschwindigkeit (Constant Linear Velocity, abgekürzt CLV) – trotz der unterschiedlichen Radien der verschiedenen Spuren geschieht das Lesen und Schreiben immer gleich schnell.

Bei den beschreibbaren CD-Rs werden die Pits durch einen besonders starken Laserstrahl in die Metalloberfläche gebohrt. Die wiederbeschreibbare CD-RW verwendet dagegen Pits mit einer viel geringeren Tiefe, die beim Überschreibvorgang wieder »abgeschliffen« werden können. Auf diese Weise lässt sich eine CD-RW bis zu 1.000 Mal beschreiben. Beachten Sie, dass CD-ROM-Laufwerke und Audio-CD-Player, die ungefähr vor Ende 1998 gebaut wurden, keine CD-RWs lesen können.

Da das Wiederbeschreiben einer CD-RW aufwendiger ist, geschieht es beim gleichen Brenner langsamer als das Beschreiben einer normalen CD-R. Aus diesem Grund enthalten CD-RW-Brenner in der Regel drei Geschwindigkeitsangaben in ihrer Spezifikation oder gar aufgedruckt auf das Gerät selbst: 40x/12x/48x bedeutet zum Beispiel, dass der CD-Brenner CD-Rs mit 40-facher Geschwindigkeit

beschreibt, CD-RWs 12-fach wiederbeschreibt und sämtliche CDs mit 48-facher Geschwindigkeit liest.

Als die beschreibbaren CDs eingeführt wurden, musste übrigens sämtlicher Inhalt in einem einzigen Durchgang darauf gebrannt werden. Bei Audio-CDs heißt dieses Verfahren Disc-at-once. Erst etwas später wurde ein Verfahren entwickelt, das man als Multisession-Format bezeichnet: Eine CD kann in mehreren Durchgängen (Sessions) gebrannt werden. Nach jeder neuen Session wird ein neues Inhaltsverzeichnis geschrieben. Sie können sich aussuchen, ob die Dateien der alten Sessions darin enthalten sein sollen, sodass sie weiterhin lesbar sind, oder ob sie weggelassen werden sollen – es ist, als seien sie nie auf der CD enthalten gewesen. Das spezielle Multisession-Verfahren für Audio-CDs wird als Track-at-once bezeichnet.

Jede Session besteht aus einem Anfangsbereich namens Lead-in-Area, den eigentlichen Daten und einem Abschlussbereich, der Lead-out-Area. Da diese Grenzbereiche eine gewisse Größe haben, steht auf einer CD mit vielen Sessions etwas weniger Speicherplatz für Daten zur Verfügung als auf einer CD, die in einem einzigen Durchgang gebrannt wurde.

Über DVDs

Die *Digital Versatile Disc* (DVD) sieht rein äußerlich genauso aus wie eine CD. Auch die DVD ist ein optischer Datenträger, der jedoch eine viel höhere Speicherdichte aufweist. Zudem gibt es Double-Layer-DVDs, die statt einer Metallschicht zwei übereinander angeordnete besitzen; sie werden von einem Laser mit unterschiedlicher Brennweite abgetastet. Darüber hinaus existieren doppelseitige DVDs mit ein bis zwei Schichten pro Seite, aber mit dem Nachteil, dass keine Seite ein Label enthalten kann; nur unmittelbar um das Loch herum befindet sich eine unscheinbare Beschriftung.

Die Speicherkapazität einer DVD beträgt 4,7 Gigabyte pro Schicht. Um einen kompletten Spielfilm in hoher Qualität mitsamt Surround-Tonspuren und Untertiteln in mehreren Sprachen sowie mit einer interaktiven Menüsteuerung zu speichern, werden in der Regel Double-Layer-DVDs verwendet.

DVDs werden zum einen für die Verbreitung von Verleih- und Verkaufsvideos eingesetzt; der Wohnzimmer-DVD-Player hat den VHS-Videorekorder inzwischen flächendeckend verdrängt. Zum anderen ist die DVD-ROM das Standard-Speichermedium für Software-Installationspakete: Betriebssystem-Distributionen, Multimedia-Enzyklopädien und aufwendige Computerspiele werden in der Regel auf DVD ausgeliefert, weil die Arbeit mit einem Stapel CDs inzwischen genau so lästig wäre wie der Umgang mit mehreren Installationsdisketten in den 90er-Jahren.

Um DVD-ROMs zu verwenden, benötigen Sie ein DVD-ROM-Laufwerk. Grundsätzlich können diese Laufwerke auch CDs lesen. Die Geschwindigkeit wird als Vielfaches der Datenrate eines Video-DVD-Players (etwa 1,3 MByte/s) angegeben. Ein 16x-DVD-Laufwerk erreicht also eine Datenübertragungsleistung von gut 20,8 MByte/s. Es werden grundsätzlich zwei Geschwindigkeitswerte wie 16x/48x angegeben, wobei der zweite Wert für die Geschwindigkeit steht, mit der das Laufwerk CD-ROMs liest (natürlich als Vielfaches der CD-Player-Rate).

Das Chaos der Datei- und Datenformate, das bei älteren CDs immer wieder für Verwirrung sorgt, herrscht bei DVDs nicht. Alle DVDs, ob sie nun Video- oder Programmdateien enthalten, verwenden dasselbe formale Datenformat, das als UDF (Universal Disk Format) bezeichnet wird. Es kann von allen aktuellen Betriebssystemen ohne Schwierigkeiten bei den Dateinamen gelesen werden.

Für beschreibbare beziehungsweise wiederbeschreibbare DVDs gibt es insgesamt drei zueinander inkompatible Formate:

1. Die DVD-R und die zugehörige DVD-RW bilden das älteste Format. Diese Datenträger können von einem normalen DVD-ROM-Laufwerk oder DVD-Player gelesen werden. Wenn Sie Video-DVDs brennen möchten, ist dieses Format vorzuziehen, weil die Videoqualität etwas besser ist als bei den anderen Formaten.
2. Etwas später wurde der DVD+R- beziehungsweise DVD+RW-Standard entwickelt. Auch diese Scheiben sind für normale Player geeignet. In puncto Geschwindigkeit und Fehlervermeidung sind diese »Plus«-Formate dem »Minus«-Standard überlegen.
3. Völlig inkompatibel zu den beiden anderen Formaten, aber auch zu normalen DVD-ROM-Laufwerken und Video-DVD-Playern, ist die DVD-RAM. Andererseits stellt sie von allen drei Formaten die schnellste und zuverlässigste Datenspeicherung zur Verfügung. Wenn Sie also eine verlässliche, moderne Datensicherungslösung benötigen, aber keinen Wert auf Kompatibilität legen, ist die DVD-RAM das Richtige für Sie.

Damit Sie mit allen Formaten zurechtkommen, sollten Sie sich einfach einen Kombibrenner kaufen, wie sie beispielsweise von Sony, NEC oder LG angeboten werden. Die meisten können Plus- und Minus-Datenträger beschreiben; nur wenige von ihnen können auch mit DVD-RAMs umgehen.

DVD-Nachfolger

Für die Nachfolge der DVD wurden zwei verschiedene Formate vorgeschlagen. Die Blu-ray Disc wurde von Elektronikunternehmen wie Philips, Sony und LG entwickelt. Ihre Kapazität beträgt 25 GByte beziehungsweise 50 GByte im Dual-

Layer-Format. Die von Firmen wie Microsoft, IBM und Toshiba entwickelte HD DVD verwendet eine andere Laser-Wellenlänge; sie besitzt eine Kapazität von 15 GByte im Single- und 30 GByte im Dual-Layer-Format. Die für beide Formate verfügbaren Standalone-Player, Computerlaufwerke und Brenner können auch mit normalen DVDs und CDs umgehen, aber in der Regel nicht mit dem jeweils anderen Format (die wenigen verfügbaren Hybrid-Geräte sind um ein Vielfaches teurer als diejenigen für eines der beiden Formate).

Anfang 2008 hat Toshiba angekündigt, die HD DVD und die zugehörigen Player nicht mehr weiterzuentwickeln. Damit kann man diesen Systemstreit als erledigt ansehen, und die Blu-ray Disc steht als Sieger fest. Seitdem erscheinen immer mehr Spielfilme auf Blu-ray, aber fast keine mehr als HD DVD.

9.2.2 CDs und DVDs brennen

openSUSE ist mit diversen Tools zum Brennen optischer Datenträger ausgestattet. Im Folgenden wird zuerst das Konsolen-Tool `wodim` beschrieben, anschließend das grafische KDE-Brennprogramm `K3b`.

Das Konsolen-Tool `wodim`

Ursprünglich kam unter Linux und anderen UNIX-Systemen ein CD-Brenn-Paket namens `cdrtools` zum Einsatz; das eigentliche Konsolen-Brennprogramm, das darin enthalten war, hieß `cdrecord`. Die `cdrtools` sind allerdings erst seit 2006 Open-Source-Software, zudem ist ihre Lizenz inkompatibel zur GPL. Aus diesem Grund kommt in openSUSE und den meisten anderen aktuellen Distributionen eine Abspaltung namens `cdrkit` zum Einsatz. Das Brennprogramm selbst heißt hier `wodim` (Write Optical Disk Media). Es reagiert aber auch auf den Namen `cdrecord`, und die meisten Parameter sind kompatibel. `wodim` verfügt allerdings über einige Erweiterungen, zumal das ursprüngliche `cdrecord` früher nur mithilfe einer kostenpflichtigen Erweiterung in der Lage war, DVDs zu brennen.

Die grundlegende Syntax von `wodim` lautet:

```
wodim [Optionen] [Track1] [...] [TrackN]
```

Die wichtigsten Optionen sind:

- ▶ `dev=Gerätedatei` – die Gerätedatei des zu verwendenden Brenners gemäß den Konventionen, die weiter oben beim `mount`-Befehl beschrieben wurden. Falls kein Gerät angegeben wird, wertet `wodim` die Umgebungsvariable `CDR_DEVICE` aus.
- ▶ `blank=all|fast|...` – falls das Medium eine CD-RW, DVD-RW oder DVD+RW ist, wird ihr bisheriger Inhalt mit diesem Kommando gelöscht. Der

Modus `all` löscht dabei alle Tracks, und `fast` löscht sie so schnell wie möglich. Es gibt noch andere Optionen; konsultieren Sie dafür die Manpage `wodim(1)`.

- ▶ `-v` – der Verbose-Modus, in dem genauere Informationen ausgegeben werden.
- ▶ `-dummy` – der Brennvorgang wird nur simuliert. Empfehlenswert, um Fehler zu vermeiden. Beachten Sie, dass `-dummy` und andere Optionen mit langen Namen bei `wodim` mit nur einem Minuszeichen geschrieben werden.
- ▶ `-data` – alle nachfolgenden Tracks sollen als Datentracks gelten. Normalerweise macht `wodim` aus `.wav`- und `.au`-Dateien automatisch einzelne Audio-tracks und aus allen anderen Dateitypen Datentracks. Die Option `-data` sowie das nachfolgend beschriebene `-audio` können nicht nur im Optionenteil vor der Trackliste stehen, sondern auch zwischen den Tracks, und gelten dann jeweils für alle nachfolgenden Tracks.
- ▶ `-audio` – interpretiert alle nachfolgenden Tracks als Audiotracks.

Das folgende Beispiel brennt das x86-32-Bit-Image der Installations-DVD von openSUSE 11.2 auf den Brenner unter `/dev/hdc` (wie der `#`-Prompt zeigt, sollten Sie `wodim` vorzugsweise als `root` ausführen):

```
# wodim -dev=/dev/hdc -v openSUSE-11.2-DVD-i586.iso
```

Hier noch ein weiteres Beispiel – es simuliert das Brennen aller `.wav`-Dateien im aktuellen Verzeichnis als Audiotracks:

```
# wodim -dev=/dev/hdc -v -dummy -audio *.wav
```

Wenn Sie umgekehrt den Inhalt einer Audio-CD in Form von Audio-Dateien speichern möchten (CD-Ripping), können Sie dazu das Kommando `cdparanoia` verwenden. Das folgende Beispiel speichert die Tracks der aktuell eingelegten CD als `.wav`-Dateien:

```
# cdparanoia -wB
```

Die Option `-w` steht dabei für das WAV-Format. `-B` (Batch-Modus) sorgt dafür, dass jeder Track als eigene Datei gespeichert wird; standardmäßig erhalten Sie sonst eine einzelne Sounddatei mit allen Tracks. Weitere Optionen können Sie in der Manpage `cdparanoia(1)` nachlesen.

Das grafische Brennprogramm K3b

Das praktischste Werkzeug zum Brennen optischer Datenträger unter Linux ist K3b aus dem KDE-Projekt. Es handelt sich im Prinzip um ein grafisches Frontend für diverse Kommandozeilentools, auf die hier allerdings nicht näher eingegangen wird. Wenn K3b bei Ihnen installiert ist, können Sie es über `K-MENÜ • MUL-`

TIMEDIA • CD/DVD BRENNEN • K3b starten; in GNOME finden Sie die entsprechenden Menüs unter ANWENDUNGEN.



Abbildung 9.2 Die Arbeitsoberfläche des Brennprogramms »K3b«

Nach dem Start präsentiert sich K3b wie in Abbildung 9.2. Zunächst stehen vier Grundaufgaben zur Verfügung:

- ▶ NEUE AUDIO-CD
- ▶ NEUE DATEN-CD
- ▶ NEUE DATEN-DVD
- ▶ CD KOPIEREN

Eine Daten-CD brennen

An dieser Stelle soll exemplarisch das Brennen einer Daten-CD beschrieben werden. Klicken Sie dazu als Erstes auf NEUE DATEN-CD. Nun können Sie sich im oberen Bereich links durch die Ordnerhierarchie bewegen. Jedes Dateisymbol, das Sie im rechten Fenster anklicken, wird zur Liste der zu brennenden Dateien im unteren Bereich hinzugefügt. In der Statusleiste am unteren Rand wird angezeigt, wie viel des verfügbaren Speicherplatzes bisher belegt wurde und wie viel noch frei ist.

Weitere Optionen stehen zur Verfügung, wenn Sie mit der rechten Maustaste in den unteren Bereich klicken. Wählen Sie beispielsweise NEUER ORDNER, um Ihre künftige CD in Verzeichnissen zu organisieren; außerdem können Sie Dateien umbenennen oder wieder entfernen. Zum Schluss können Sie unter DATENTRÄGER-NAME eingeben, wie die CD heißen soll.

Wenn Sie mit Ihrem CD-Layout zufrieden sind, klicken Sie auf BRENNEN. Stellen Sie auf der Registerkarte BRENNEN die Brenngeschwindigkeit und weitere Optionen ein. Sie können beispielsweise auch SIMULIEREN wählen, um das Brennen mit der gewählten Geschwindigkeit nur zu testen. Die Registerkarte ABBILD ermöglicht das Erstellen eines ISO-Images – nützlich zur Verbreitung brennfertiger CDs im Internet, zur Archivierung oder auch, um mehrere identische Datenträger zu erstellen. Unter DATEISYSTEM und ERWEITERT können Sie schließlich die oben beschriebenen Dateisystem-Details wie ISO, Joliet oder UDF mit zahlreichen Zusatzoptionen einstellen.

Wenn Sie alles eingestellt haben, können Sie auf SPEICHERN klicken, um die Projekteinstellungen für einen späteren Brennvorgang zu speichern. Wenn Sie dagegen die Schaltfläche BRENNEN anklicken, wird der Brennvorgang mit den gewählten Einstellungen sofort gestartet.

Bearbeiten und Brennen der openSUSE-Installations-Images

Besonders wichtig ist K3b oder ein anderes Brennprogramm natürlich zum Erstellen der openSUSE-Installations-DVDs oder -LiveCDs aus den Images, die Sie von *opensuse.org* oder einer der zahlreichen Mirror-Sites heruntergeladen haben. Wählen Sie dazu EXTRAS • CD-ABBILDDATEI BRENNEN beziehungsweise EXTRAS • ISO-ABBILDDATEI AUF DVD BRENNEN. Klicken Sie das Ordnersymbol unter ABBILD ZUM BRENNEN an, und wählen Sie die passende ISO-Image-Datei aus. Ein sehr praktisches Feature von K3b ist, dass die bereits in Kapitel 2 empfohlene MD5-Prüfsumme automatisch ermittelt wird.

9.3 Archivdateien und virtuelle Dateisysteme

Neben den bisher beschriebenen datenträger- beziehungsweise partitionsbasierten Dateisystemen stehen einige spezielle virtuelle Dateisysteme zur Verfügung. Im weiteren Sinne gehören auch die diversen Arten von Archivdateien dazu, die als Erste in diesem Abschnitt besprochen werden.

9.3.1 Archivdateien verwenden

Archivdateien begegnen Ihnen täglich beim Download von Software und anderen Inhalten. Es geht um eine Methode, mehrere Dateien in eine gemeinsame Container-Datei zu verpacken und nach Möglichkeit auch zu komprimieren. Während unter Windows das PKZIP-Format (unter anderem in WinZip) dominiert, das Dateisammlungen in einem Arbeitsschritt archiviert und komprimiert, wird auf UNIX-artigen Systemen in der Regel eine Kombination verwendet: Das Archivprogramm `tar` (das nicht selbst komprimieren kann) und eines von mehreren externen Kompressionsprogrammen (die zumindest früher nicht selbst archivieren konnten).

`tar` ist die Abkürzung für *Tape ARchive*. Da klassische Magnetbänder nur sequenziell beschrieben und gelesen werden können, ist es nützlich, alle zu sichernden Dateien zunächst in eine einzige große Hülldatei zu schreiben.

Wenn Sie `tar` auf zwei kurze Textdateien anwenden und sich das Ergebnis anzeigen lassen, können Sie sehen, wie es intern funktioniert. Erstellen Sie zunächst die beiden Textdateien:

```
$ cat >text1
Ich bin Text 1.
[Strg]+[D]
$ cat >text2
Ich bin Text 2.
[Strg]+[D]
```

Als Nächstes sollen die beiden Dateien in einem Tar-Archiv gesammelt werden. Die Option zum Speichern heißt `-c` (Langform `--create`). Wenn Sie Dateien an ein bestehendes Archiv anhängen möchten, wird stattdessen `-r` (`--append`) verwendet. Zusätzlich wird die Option `-f` Archivdatei benötigt, weil die Ausgabe in einer Datei und nicht auf einem Bandlaufwerk landen soll. Meistens wird auch noch `-v` (`verbose`) benutzt; dieser Parameter listet die verarbeiteten Dateien auf. Das folgende Beispiel speichert die beiden Textdateien in einem Archiv namens *texte.tar*:

```
$ tar cvf texte.tar text1 text2
text1
text2
```

Nun können Sie sich das erzeugte Archiv anzeigen lassen. Da die einzelnen Felder innerhalb von Tar-Archiven durch Nullzeichen (ASCII 0 oder Escape-Sequenz `\0`) getrennt werden, sollten Sie geeignete Mittel verwenden, um diese durch Leerzeichen, Tabs, Zeilenumbrüche oder andere Zeichen Ihrer Wahl zu ersetzen. Das folgende Beispiel sendet die Ausgabe durch eine Pipe an ein direkt auf der

Kommandozeile geschriebenes Perl-Skript (die Option `-e` sorgt dafür, dass Perl den übergebenen String nicht als Dateinamen, sondern als Skript auswertet):

```
$ cat texte.tar |perl -e 'while(<>) {s/\0+/\n/g; print; }'
text1
0000644
0000000
0000000
00000000020
10434566641
010545
0
ustar
sascha
users
Ich bin Text 1.

text2
0000644
0000000
0000000
00000000020
10434566655
010553
0
ustar
sascha
users
Ich bin Text 2.
```

Die Perl-Substitutionsanweisung `s/\0+/\n/g` ersetzt eine Folge von einem oder mehreren Null-Zeichen (`\0+`) durch je einen Zeilenumbruch (`\n`), und dies beliebig oft in einer Eingabezeile (Option `/g` für »global«). Mehr über die Arbeit mit regulären Ausdrücken in Perl erfahren Sie in Kapitel 17, »System-Automatisierung«.

Wie Sie sehen, enthält jeder Dateieintrag im Archiv vor dem eigentlichen Inhalt den Dateinamen, verschiedene Informationsfelder (unter anderem die Dateirechte, hier `0644`), Benutzer und Gruppe.

Wenn Sie ein solches Archiv wieder entpacken möchten, wird statt `-c` die Option `-x` (`--extract`) eingesetzt. Hier die notwendige Eingabe, um `texte.tar` wieder zu entpacken:

```
$ tar xvf texte.tar
textel
texte2
```

Wenn Sie ein fertiges Tar-Archiv komprimieren möchten, haben Sie unter anderem die Wahl zwischen den Kompressionsformaten GNUZip, das weiter verbreitet und damit kompatibler ist, und BZip2, das effizienter komprimiert. Wenn Sie *texte.tar* gzip-komprimieren möchten, geben Sie Folgendes ein:

```
$ gzip texte.tar
```

Das Ergebnis ist eine Datei namens *texte.tar.gz*; die ursprüngliche *.tar*-Datei wird gelöscht. Zum Entpacken dient folgende Eingabe:

```
$ gunzip texte.tar.gz
```

Dies ersetzt umgekehrt die komprimierte Datei durch die entpackte Version *texte.tar*.

Für die BZip2-Komprimierung wird dagegen dieses Kommando verwendet:

```
$ bzip2 texte.tar
```

Der entsprechende Befehl zum Entpacken lautet:

```
$ bunzip2 texte.tar.bz2
```

Die moderne GNU-Version von *tar* kann die gzip- oder BZip2-Komprimierung beziehungsweise -Dekomprimierung als zusätzlichen Arbeitsschritt gleich mit übernehmen. Dazu wird zusätzlich die Option *-z* (GNUZip) beziehungsweise *-j* (BZip2) verwendet. Den Archivdateinamen sollten Sie in diesem Fall direkt mit der zusätzlichen Endung für das gewählte Kompressionsformat versehen. Das folgende Beispiel erstellt aus *texte1* und *texte2* das komprimierte GNUZip-Archiv *texte.tar.gz*:

```
$ tar czvf texte.tar.gz text1 text2
text1
text2
```

Dekomprimiert und entpackt wird die neue Datei wie folgt:

```
$ tar xzvf texte.tar.gz
```

Wenn Sie stattdessen *.bz2*-Komprimierung wünschen, lauten die beiden Eingaben so:

```
$ tar cjvf texte.tar.bz2 text1 text2
$ tar xjvf texte.tar.bz2
```

Wenn Sie sich die Inhalte eines Archivs auflisten lassen möchten, ohne es zu entpacken, können Sie die Option `l` statt `x` verwenden; die Option `v` erledigt in diesem Zusammenhang die Aufgabe von `-l` beim Kommando `ls`. Das folgende Beispiel listet das Inhaltsverzeichnis der Datei *texte.tar.gz* ausführlich auf:

```
$ tar tzvf texte.tar.gz
-rw-r--r-- sascha/users 14 2008-04-05 08:16 text1
-rw-r--r-- sascha/users 14 2008-04-05 08:16 text2
```

Linux kann übrigens auch mit den in der Windows-Welt verbreiteten PKZIP-Dateien umgehen. Verwenden Sie das Kommando `zip` zum Verpacken und `unzip` zum Entpacken. Die allgemeine Syntax für `zip` lautet:

```
zip Zipdatei Datei[en]
```

Das folgende Beispiel verpackt alle Dateien im aktuellen Verzeichnis, deren Name mit »text« beginnt, in die ZIP-Datei *texte.zip*:

```
$ zip texte.zip text*
```

Um diese Datei wieder auszupacken, wird folgendes Kommando verwendet:

```
$ unzip texte.zip
```

Wenn Sie nur die Inhalte sehen möchten, können Sie `unzip` mit der Option `-l` einsetzen:

```
$ unzip -l texte.zip
```

9.3.2 Loop-Devices

Weiter oben war bereits von ISO-Images die Rede. Es handelt sich um Dateisysteme, die in gewöhnlichen Dateien gespeichert werden. Linux besitzt eine Möglichkeit, um Images und andere als Dateien gespeicherte Dateisysteme zu mounten: die sogenannten Loop-Devices. Es handelt sich um die speziellen Gerätedateien */dev/loop0* bis */dev/loop7*. Jedes dieser Devices lässt sich über das Kommando `losetup`, das folgende Grundsyntax besitzt, mit einer Image-Datei verknüpfen:

```
losetup /dev/loopN Dateiname
```

Das folgende Beispiel verknüpft das Image *test.iso* im aktuellen Arbeitsverzeichnis mit dem Loop-Device `loop0`:

```
# losetup /dev/loop0 test.iso
```

Anschließend kann das Loop-Device wie ein gewöhnliches Volume gemountet werden, wobei Sie wieder optional den Dateisystemtyp angeben können. Beispiel:

```
# mount -t iso9660 /dev/loop0 /tmp/cd_image
```

9.3.3 Das /proc-Dateisystem

Eine Besonderheit stellt das virtuelle Dateisystem dar, das standardmäßig unter */proc* gemountet wird. Es wird zur Laufzeit dynamisch erstellt und enthält Informationen über den Status von Geräten und Prozessen. Die meisten virtuellen Dateien darin können nur gelesen werden. Das folgende Beispiel zeigt den Inhalt von */proc/meminfo* mit Informationen über die Nutzung des Arbeitsspeichers an:

```
# cat /proc/meminfo
MemTotal:      256724 kB
MemFree:       5116 kB
Buffers:       96480 kB
Cached:        51680 kB
SwapCached:    29116 kB
Active:        155692 kB
Inactive:      49736 kB
HighTotal:     0 kB
HighFree:     0 kB
LowTotal:     256724 kB
LowFree:      5116 kB
SwapTotal:    514040 kB
SwapFree:     416896 kB
Dirty:         0 kB
Writeback:     0 kB
Mapped:       90004 kB
Slab:         36852 kB
CommitLimit:  642400 kB
Committed_AS: 395900 kB
PageTables:    2096 kB
VmallocTotal: 770040 kB
VmallocUsed:   9732 kB
VmallocChunk: 760036 kB
HugePages_Total: 0
HugePages_Free: 0
Hugepagesize: 4096 kB
```

Wenn Sie sich mittels `ls` im Verzeichnis */proc* umschaun, werden Sie feststellen, dass es dort noch weitere interessante Informationen gibt. Allerdings haben Sie

für viele davon auch bereits Kommandos kennengelernt, die die betreffenden Daten in benutzerfreundlicheren Formaten anzeigen.

Manche Konfigurationseinstellungen lassen sich übrigens durch Ausgabeumleitung in die betreffenden */proc*-Dateien vornehmen (was allerdings nur in Einzelfällen wirklich notwendig ist; fast immer ist es einfacher, Konfigurationsdateien zu bearbeiten). Die folgende Eingabe ändert beispielsweise den Hostnamen des Rechners:

```
# echo "geeko" >/proc/sys/kernel/hostname
```

9.4 Die Boot-Konfiguration

Weiter oben in diesem Kapitel wurde bereits der Aufbau des Master Boot Records beschrieben, in dem sich unter anderem der Bootloader befindet, der das eigentliche Betriebssystem startet. An dieser Stelle geht es um die Konfiguration des Bootloaders selbst. Unter Linux sind zwei verschiedene Bootmanager verbreitet: der Klassiker LILO und die modernere Variante GRUB. Dieser Abschnitt konzentriert sich vor allem auf GRUB, der heutzutage Standard ist, und erwähnt LILO eher am Rande.

9.4.1 Den Bootloader konfigurieren

Wählen Sie in YaST **SYSTEM • KONFIGURATION DES BOOTLOADERS**, um den Bootmanager einzurichten. Dies ist notwendig, falls es noch nicht bei der Systeminstallation geschehen ist oder wenn Sie gemäß der Anleitung in Kapitel 6 einen neuen Kernel kompiliert haben. Das YaST-Applet enthält zwei Registerkarten:

► *Abschnittsverwaltung*

In diesem Bereich wird eingestellt, welche Auswahlmöglichkeiten im Bootmenü zur Verfügung stehen sollen – und zwar zunächst einmal unabhängig vom konkreten Bootmanager. Wählen Sie **BEARBEITEN**, um die Optionen eines Eintrags zu ändern, oder **HINZUFÜGEN**, um einen neuen zu erstellen.

Für existierende Einträge stehen folgende Optionen zur Verfügung:

- **OPTIONALER PARAMETER FÜR KERNEL-BEFEHLSZEILE:** Hier können diverse Optionen wie etwa spezielle Kernelmodule (Näheres in Kapitel 6) angegeben werden.
- **KERNEL-ABBILD:** Bestimmt, welches Kernel-Image der jeweilige Abschnitt starten soll. Der Linux-Standardeintrag bootet den Kernel */boot/vmlinuz*. Falls Sie selbst einen neuen Kernel kompiliert haben (siehe Kapitel 6,

»Wichtige Hardware konfigurieren«), tragen Sie hier den Dateinamen ein, unter dem Sie ihn nach */boot* kopiert haben.

- ▶ **INITIAL RAM DISK:** Wenn hier ein Eintrag besteht, wird beim Start eine virtuelle Diskette (RAM-Disk) geladen – das Konzept ist eine Art ältere Variante der oben beschriebenen Loop-Devices. Beim Standardeintrag für den Linux-Kernel wird */boot/initrd* verwendet.
- ▶ **ROOT-DEVICE:** Die Festplatte beziehungsweise Partition, die als Wurzel des Verzeichnisbaums (*/*) eingebunden werden soll.
- ▶ **VGA-MODUS:** Auflösung und Farbtiefe der Grafikkarte, die nach dem Booten eingestellt werden sollen.
- ▶ Wenn Sie sich für das Erstellen eines neuen Abschnitts entscheiden, haben Sie die Wahl zwischen den folgenden Möglichkeiten:
 - ▶ **AUSGEWÄHLTEN ABSCHNITT KLONEN:** Erstellt eine Kopie des aktuellen Abschnitts, die Sie dann nach Ihren Wünschen modifizieren können. Dies ist beispielsweise ideal für einen neu kompilierten Kernel, bei dem sich lediglich der Pfad der Kernel-Datei ändert.
 - ▶ **KERNEL (LINUX):** Standardabschnitt zum Booten eines Linux-Kernels
 - ▶ **KERNEL ÜBER XEN:** Die Virtualisierungssoftware XEN erlaubt den Betrieb mehrerer virtueller Maschinen auf demselben Rechner. Die aktuelle Version 3 gehört zum Lieferumfang von openSUSE und kann leicht über YaST konfiguriert werden (**SYSTEM • VIRTUAL MACHINE MANAGEMENT**). Voraussetzung für den Einsatz von XEN ist, dass Sie einen entsprechend modifizierten Kernel booten, der als erste der virtuellen Maschinen (Domain 0) dient. openSUSE enthält bereits ab Werk einen solchen Kernel und den betreffenden Bootmenü-Eintrag. Mit diesem Menüpunkt können Sie einen weiteren erstellen.
 - ▶ **ANDERES SYSTEM (CHAINLOADER):** Wenn auf Ihrem System weitere Betriebssysteme installiert sind, können Sie den Linux-Bootmanager verwenden, um diese alternativ zu Linux zu starten. Gängige Systeme wie Windows werden bei der Installation automatisch erkannt und hier eingetragen. Der Bootloader des betreffenden Systems muss sich dafür im Bootsektor der jeweiligen Partition befinden.

Als Nächstes müssen Sie dem neuen Abschnitt einen Namen geben und bestimmen, für welche Partition er gelten soll. Danach geht es je nach Art des Abschnitts weiter: Bei Linux-Kerneln (klassisch oder XEN) erscheint der oben beschriebene Dialog, während Sie bei der Boot-Konfiguration für andere Betriebssysteme bereits mit der Auswahl des Geräts fertig sind.

► *Bootloader-Installation*

Diese Registerkarte bestimmt, welcher Bootloader tatsächlich installiert werden soll und wohin. Wählen Sie unter `Typ` den gewünschten Bootloader (GRUB oder LILO). Anschließend können Sie sich für den `Speicherort` entscheiden: Master Boot Record der jeweiligen Festplatte (zum Beispiel `/dev/hda`), Bootsektor der Partition (etwa `/dev/hdb2`) oder Diskette. Letzteres ist als Sicherheitskopie recht nützlich.

In aller Regel genügt es, den Bootloader in den Bootsektor einer bestimmten Partition zu installieren. Es kann gefährlich sein, ihn im Master Boot Record der Festplatte zu platzieren, falls andere Betriebssysteme installiert sind. Diese können dann gegebenenfalls nicht mehr starten (dies betrifft vor allem ältere Windows-Versionen).

Unter dem Menüpunkt `ANDERE` können Sie tiefer ins Detail gehen. Wenn Sie dort `KONFIGURATIONSDATEIEN BEARBEITEN` wählen, werden die hinter den Kulissen erstellten Dateien des jeweiligen Bootloaders angezeigt. GRUB besitzt drei solche Dateien:

- `/boot/grub/device.map`: GRUB verwendet ein eigenwilliges Schema zur Bezeichnung von Datenträgern und Partitionen: Festplatten heißen (hd0), (hd1) und so weiter – unabhängig von ihrer Architektur. Die Partitionen werden entsprechend als (hd0,0), (hd0,1) und folgende bezeichnet. Das Diskettenlaufwerk heißt (fd0). Diese Konfigurationsdatei wird benutzt, um die GRUB-Gerätenamen den eigentlichen Gerätedateien zuzuordnen. Dazu enthält sie zweispaltige Einträge wie diesen:

```
(hd0) /dev/hda
```

- `/boot/grub/menu.lst`: Dies ist das eigentliche Bootmenü. Als Erstes gibt es einige globale Darstellungsoptionen – etwa `default` für die Standardauswahl (die Einträge werden dazu ab 0 durchnummeriert) oder `timeout` für die Wartezeit in Sekunden, bis dieser Standardeintrag automatisch gebootet wird. Darunter stehen die einzelnen Abschnitte; derjenige für den Standard-Linux-Kernel sieht beispielsweise so aus (wenn Sie sie genau betrachten, erkennen Sie die einzelnen Optionen von der Registerkarte `ABSCHNITTSVERWALTUNG` wieder):

```
title openSUSE 11.2
  root (hd0,1)
  kernel /boot/vmlinuz root=/dev/sda2 vga=0x314 \
    resume=/dev/sda1 splash=silent showopts
  initrd /boot/initrd
```

- ▶ */etc/grub.conf*: Einstellungen für den Aufruf des Kommandozeilen-Tools `grub`, das hier dank der komfortablen YaST-Konfiguration nicht besprochen zu werden braucht.

Für LILO gibt es dagegen nur eine Konfigurationsdatei, nämlich */etc/lilo.conf*. Sie enthält zunächst ebenfalls einige globale Optionen wie `timeout=Wartezeit` oder `default="Name des Standardeintrags"`. Danach folgen die eigentlichen Boot-Einträge; der Standardeintrag für den normalen Linux-Kernel sieht folgendermaßen aus:

```
image = /boot/vmlinuz
    label = SUSE_Linux
    append = " resume=/dev/sda1 splash=silent showopts"
    vga = 0x314
    initrd = /boot/initrd
    root = /dev/sda2
```

9.4.2 Die Rettungskonsole

Wenn das Booten aus irgendeinem Grund nicht mehr funktioniert, ist noch nicht alles verloren. Legen Sie bei einem solchen Problem einfach einen openSUSE-Installationsdatenträger ein, und wählen Sie `RETTUNGSSYSTEM` aus dessen Bootmenü.

Nach dem recht schnellen Booten finden Sie sich in einer Textkonsole wieder und sind `root` (ohne ein Passwort einzugeben). Viele der in diesem Kapitel und im Rest des Buches vorgestellten Konsolenkommandos stehen auch hier zur Verfügung. Sie können also nach Belieben (manuell) Datenträger mounten und unmounten, Konfigurationsdateien betrachten und mit `vi` editieren, Verzeichnisinhalte lesen oder die oben beschriebenen Dateisystem-Verwaltungsbefehle verwenden.

Beachten Sie, dass kein deutscher Tastaturtreiber geladen wird. Dadurch sind insbesondere folgende Tasten anders belegt:

- ▶ `[Z]` und `[Y]` sind vertauscht
- ▶ Der `/` liegt auf der Taste `[_]`.
- ▶ Das `-` ist dagegen über die Taste `[B]` zu erreichen.
- ▶ `<` und `>` erreichen Sie mithilfe der Tastenkombinationen `[⇧]+[_]` beziehungsweise `[⇧]+[B]`.

Tabelle 9.3 zeigt eine ausführlichere Übersicht über wichtige Sonderzeichen und ihre Erzeugung auf einem US-englischen Tastatur-Layout mit deutscher Beschriftung.

lung. Beachten Sie in diesem Zusammenhang bitte, dass das britische Tastatur-Layout wieder ein wenig anders ist als das amerikanische.

Gewünschtes Zeichen	Zeichen auf deutscher Tastatur	Tastenkombination
!	!	⇧ + 1
"	Ä	⇧ + Ä
\$	\$	⇧ + 3
&	/	⇧ + 7
@	"	⇧ + 2
/	-	-
()	⇧ + 9
)	=	⇧ + 0
[ü	Ü
]	+	+
{	ü	⇧ + Ü
}	*	⇧ + +
\	#	#
<	;	⇧ + .
>	:	⇧ + .
	'	⇧ + #
:	ö	Ö
:	Ö	⇧ + Ö
?	–	⇧ + -
+	˘	⇧ + ~
-	ß	ß
–	?	⇧ + ß
=	˘	˘
z	y	Y
y	z	Z

Tabelle 9.3 Erzeugen der wichtigsten Sonderzeichen auf einer deutsch beschrifteten Tastatur mit US-englischem Tastatur-Layout

Sie können die Rettungskonsole auch verwenden, um ein vergessenes `root`-Passwort zurückzusetzen:

1. Erstellen Sie ein neues Verzeichnis, und mounten Sie die Partition, auf der sich das Verzeichnis `/etc` befindet (in der Regel die Root-Partition), dorthin. Beispiel:

```
# mkdir /disk
# mount -t ext3 /dev/hda1
```

2. Öffnen Sie die Datei `/etc/passwd` (nun unter dem Verzeichnis `/disk`) zum Bearbeiten mit `vi`:

```
# vi /disk/etc/passwd
```

Der `root`-Eintrag sieht wahrscheinlich etwa so aus:

```
root:x:0:0:root:/root:/bin/bash
```

3. Entfernen Sie das auf die Shadow-Datei verweisende `x`:

```
root::0:0:root:/root:/bin/bash
```

4. Speichern Sie die Datei, und starten Sie das System ohne CD beziehungsweise DVD neu:

```
# shutdown -r now
```

5. Beim nächsten Start können Sie sich auf einer Textkonsole als `root` ohne Passwort einloggen und das Kommando `passwd` verwenden, um ein neues Passwort zu erstellen.

9.5 Zusammenfassung

Die Verwaltung von Datenträgern und Dateisystemen ist eine der wichtigsten Aufgaben eines Betriebssystems. In diesem Kapitel haben Sie die entsprechenden Administrationswerkzeuge kennengelernt.

Die wichtigste Art von Datenträgern sind Festplatten. Nach einer Einführung in die technischen Grundlagen wurde im ersten Abschnitt dieses Kapitels ausführlich erläutert, welche Partitionstypen und Dateisysteme zur Verfügung stehen und wie sie sich nutzen lassen. Die eigentlichen Partitionierungswerkzeuge wurden übrigens bereits in Kapitel 2 vorgestellt.

Im zweiten Abschnitt haben Sie das Wichtigste über CDs und DVDs erfahren. Als Brennsoftware wurden das Konsolen-Tool `wodim` und die KDE-Lösung `K3b` vorge-

stellt, aber die grundlegenden Erläuterungen lassen sich leicht auf andere Brennprogramme übertragen.

Als Nächstes ging es um den Umgang mit virtuellen Dateisystemen, insbesondere um Loop-Devices, über die sich Image-Dateien als normale Verzeichnisse mounten lassen, sowie um das Verwaltungdateisystem */proc*.

Zum Schluss wurde noch die Boot-Konfiguration behandelt – der Unterschied zwischen den Bootmanagern GRUB und LILO und wie sich diese mittels YaST konfigurieren lassen. Falls das System einmal nicht mehr booten sollte, können Sie die Rettungskonsole von einem Installationsdatenträger starten.

»Why are we hiding from the police, daddy?« –
»Because we use vi son, they use Emacs.«
– Anonymous¹

10 Klassische Texteditoren

Einige der per Maus bedienbaren, unter KDE oder GNOME verfügbaren Texteditoren haben Sie bereits in Kapitel 3, »Systemrundgang und Grundkonfiguration«, kennengelernt. Die Benutzung dieser Programme ist leicht zu erlernen, aber dennoch stellen sie nicht für jeden Anwendungszweck die optimale Methode der Texteingabe und -bearbeitung auf einem UNIX-System dar. Wenn Sie sich auf reinen Text ohne visuelles Layout konzentrieren möchten, ist der Griff zur Maus stets umständlich. Deshalb werden in diesem Kapitel zwei weitverbreitete Editoren vorgestellt, die zu jedem vollständigen Linux-System gehören: Vi(m) und Emacs. Jeder von ihnen besitzt spezifische Stärken und Schwächen, die hier einigermaßen ausführlich dargestellt werden.

Anfangs mag es anstrengend erscheinen, sich unzählige Tastenkombinationen zu merken, aber wenn Sie sich erst an die Arbeit mit einem der beiden Editoren gewöhnt haben, werden Sie Text schneller und effizienter bearbeiten können als je zuvor: Nicht umsonst habe ich beinahe den gesamten Text dieses Buches in diesen Editoren verfasst und erst zur endgültigen Formatierung in OpenOffice.org übernommen.

10.1 vi und Vim

vi (sprich: »wie Ei«) ist der klassische Texteditor für UNIX-Systeme. Er wurde bereits in den 1970er-Jahren von *Bill Joy* programmiert; im Grunde handelte es sich um eine Vollbilderweiterung für den Zeileneditor `ex`. Das erkennt man bei vi und allen seinen Abkömmlingen noch heute daran, dass sich hinter einem Doppelpunkt `ex`-Kommandos eingeben lassen.

1 [»Warum verstecken wir uns vor der Polizei, Papa?« – »Weil wir vi benutzen, mein Sohn, und sie Emacs.«] Diese Anspielung auf den ewigen Flamewar zwischen den Anhängern der beiden hier vorgestellten Editoren gibt es natürlich auch umgekehrt!

`vi` oder ein dazu kompatibler Editor steht in der Regel auf jedem beliebigen UNIX- oder Linux-System zur Verfügung. openSUSE und die meisten anderen Linux-Distributionen enthalten die wesentlich leistungsfähigere Variante *Vim* (Vi Improved) von *Bram Molenaar*; eine andere verbreitete Version ist beispielsweise `elvis`. In jeder openSUSE-Standardinstallation existiert ein Link namens `vi`, der auf `vim` verweist, sodass Sie den Editor Vim starten können, indem Sie eines der beiden folgenden Kommandos eingeben:

```
$ vi
$ vim
```

Wenn Sie einen Dateinamen dahintersetzen, wird die angegebene Datei zum Bearbeiten geöffnet – falls sie noch nicht existierte, wird sie neu angelegt. Sie können – durch Leerzeichen getrennt – auch mehrere Dateien angeben. Der Wechsel zur jeweils nächsten Datei erfolgt durch Eingabe von `:` `next` `↵`.

Die Bedienung von `vi` und seinen Nachfolgern ist etwas gewöhnungsbedürftig, weil es zwei grundlegende Arbeitsmodi gibt: den *Befehlsmodus* und den *Eingabemodus*. Nach dem Start befindet sich `vi` zunächst im Befehlsmodus – durch spezielle Tastenkürzel können Sie sich im Text bewegen oder Verwaltungsaufgaben wie Löschen oder Speichern durchführen. Durch Drücken von `I` gelangen Sie in den Eingabemodus, in dem Sie Text eintippen können. Auch die Taste `A` aktiviert den Eingabemodus, beginnt aber ein Zeichen hinter der bisherigen Cursorposition – zum Erweitern eines bestehenden Textes ist dies oft praktischer. `↵+I` setzt den Cursor an den Anfang der aktuellen Zeile und wechselt dann in den Eingabemodus; `↵+A` wählt entsprechend das Zeilenende. `O` (der Buchstabe o, nicht die Zahl Null) fügt unter der aktuellen Zeile eine Leerzeile ein und setzt den Eingabecursor hinein; mit `↵+O` geschieht dasselbe mit einer Leerzeile *über* der aktuellen Zeile.

`ESC` wechselt zurück in den Befehlsmodus. Anders als der klassische `vi` kennzeichnet Vim den Einfügemodus durch `-- EINFÜGEN --` am linken unteren Bildschirmrand. Am einfachsten ist es aber, vor der Eingabe von Befehlen vorsichtshalber `ESC` zu drücken: Falls Sie sich bereits im Befehlsmodus befinden, ändert sich nichts daran.

Bitte beachten: Sämtliche Tastenkombinationen in diesem Abschnitt, die ein Sonderzeichen ergeben – etwa `↵+7` für `/` – beziehen sich auf die deutsche Tastaturbelegung. Deshalb wird das Ergebniszeichen jeweils mit angegeben; wenn Ihre Tastatur anders eingestellt ist, müssen Sie jeweils die Taste oder Tastenkombination verwenden, die zu diesem Zeichen führt. Wie man einige häufige Zeichen auf einer deutschen Tastatur eingibt, wenn englisches Tastaturlay-

out eingestellt ist, erfahren Sie in Kapitel 9 (da dies manchmal in Notfall- und Rettungssystemen vorkommt).

10.1.1 Navigation im Text

Die zeichen- beziehungsweise zeilenweise Navigation im Text erfolgt im Befehlsmodus durch folgende Tasten: Mit **H** bewegen Sie den Cursor nach links, mit **J** nach unten, mit **K** nach oben und mit **L** nach rechts. In modernen Linux-Distributionen funktionieren auch die Pfeiltasten, aber die vier nebeneinander liegenden Buchstabentasten sind nicht die schlechteste Wahl.

Mithilfe der Taste **O** können Sie sich zum Zeilenanfang bewegen; **\$** – das heißt **↩+4** – springt dagegen zum Zeilenende. Die Tasten **Pos1** beziehungsweise **Ende** erfüllen dieselbe Aufgabe. Der Begriff »Zeile« steht dabei für einen beliebig langen Zeichenblock, der durch einen Zeilenumbruch abgeschlossen wird und durchaus mehrere Bildschirmzeilen lang sein kann.

Runde Klammern dienen dem Bewegen zwischen Sätzen: **(** beziehungsweise **↩+8** bewegt sich zum Anfang des vorigen Satzes, **)** oder **↩+9** zum nächsten. Geschweifte Klammern – **[** und **]** – springen zum Anfang des nächsten beziehungsweise des vorigen Absatzes (gekennzeichnet durch eine Leerzeile). Sie werden mithilfe der Tastenkombinationen **Alt Gr+7** beziehungsweise **Alt Gr+9** erzeugt.

Eine Zahl mit einem Pipe-Zeichen | (**Alt Gr+<**) dahinter springt zum angegebenen Zeichen in der aktuellen Zeile: **7|** **|** bewegt sich beispielsweise zur Position 7. Hat eine Zeile weniger Zeichen als der Wert, den Sie angeben, dann bewegt sich der Cursor zum letzten.

- bewegt den Cursor zum Anfang der vorigen Zeile, während **+** oder **↵** zum ersten Nicht-Leerzeichen in der nächsten Zeile springen.

Auch relativ zum sichtbaren Ausschnitt des Textes lässt es sich navigieren: **↩+H** bewegt den Cursor an den Anfang der ersten sichtbaren Zeile, **↩+M** auf die erste Position der mittleren und **↩+L** an den Beginn der letzten.

W bewegt den Cursor um ein Wort weiter. Dabei gelten auch Satzzeichen wie Punkt oder Komma als eigenständige Wörter; **↩+W** beschränkt sich dagegen auf die eigentlichen Wörter, das heißt auf alle Kombinationen aus alphanumerischen Zeichen. Mit **B** beziehungsweise **↩+B** können Sie auf dieselbe Weise je ein Wort zurückblättern.

Mit der Taste **E** springen Sie zum nächsten Wortende, mit **G E** zum vorherigen. Auch bei diesen beiden Befehlen gibt es **↩**-Optionen, um sie auf richtige Wörter zu beschränken: **↩+E** beziehungsweise **G ↩+E**.

Zähler verwenden

Sämtlichen Befehlstasten können Sie eine Zahl voranstellen, um deren Wirkung zu vervielfachen. Beispielsweise bewegt **3 L** den Cursor um drei Zeichen nach links, während **4 B** um vier Wörter nach links wandert. Bei mehrbuchstabigen Befehlen steht die Anzahl entweder davor oder in der Mitte – **3 D D** beziehungsweise **D 3 D** löscht etwa drei Zeilen ab der aktuellen.² Auch dem Befehl **\$** können Sie einen Zähler voranstellen; er springt dann nicht zum Ende der aktuellen Zeile, sondern entsprechend viele Zeilen weiter. Die Taste **0** für den Zeilenanfang besitzt diese Funktion dagegen nicht, weil sie selbst eine Zahl ist; leider geht es auch mit **Pos1** nicht.

Sprünge zu absoluten Zeilennummern sind mit **↕+G** möglich; dies nützt zum Beispiel Programmierern, um zu einer Zeile zu springen, die in einer Fehlermeldung angegeben wurde. Ohne Zähler springt die Option zum Textende, während **4 7 ↕+G** den Cursor beispielsweise an den Anfang von Zeile 47 bewegt. **G G** springt übrigens an den Textanfang und **↕+G** an den Anfang der letzten Zeile des Textes.

Scrollen

Zum Scrollen werden Tastenkombinationen mit **Strg** verwendet, die trotzdem nur im Befehlsmodus funktionieren. **Strg+U** scrollt einen halben Bildschirm nach oben, **Strg+D** entsprechend weit nach unten. Um eine einzelne Zeile hochzuscrollen, können Sie **Strg+Y** drücken; in die entgegengesetzte Richtung geht es mit **Strg+E**. Einen ganzen Bildschirm nach oben oder unten geht es mit **Strg+B** beziehungsweise **Strg+F**. **Z Z** positioniert die Zeile mit dem Cursor genau in der Bildschirmmitte; entsprechend wird sie mit **Z T** an den oberen und mit **Z B** an den unteren Fensterrand gescrollt.

Zeichen suchen

Die Taste **F**, gefolgt von einem beliebigen Zeichen, springt zum nächsten Vorkommen des entsprechenden Zeichens nach der aktuellen Cursorposition, allerdings nur in der aktuellen Zeile. Der Zähler wählt entsprechende weitere Vorkommen des Zeichens. Wenn der Cursor sich beispielsweise am Anfang des Wortes »Linux-Distribution« befindet und Sie **F I** eingeben, dann befinden Sie sich anschließend auf dem i von »Linux«. **3 F I** würde dagegen zum zweiten der drei i in »Distribution« springen. **↕+F** sucht entsprechend rückwärts. **T** und **↕+T** sind Variationen dieser Befehle, die den Cursor je ein Zei-

2 Formal gibt es einen Unterschied zwischen den beiden Varianten: **3 D D** führt dreimal den Befehl "Eine Zeile löschen" aus, während **D 3 D** den Löschbefehl auf eine Bewegung von drei Zeilen anwendet. Praktisch bedeuten sie allerdings dasselbe.

chen vor beziehungsweise hinter dem Ergebnis positionieren. Wenn Sie die gewünschte Suchtaste bereits gedrückt haben, aber gar nicht suchen möchten, können Sie statt des zu suchenden Zeichens einfach `[Esc]` betätigen. Mit `;` können Sie den letzten Suchbefehl wiederholen; `,` kehrt seine Richtung um. Beachten Sie, dass diese Art der Suche zwischen Groß- und Kleinschreibung unterscheidet.

Der Befehl `%` (`[↔]+[5]`) springt zwischen korrespondierenden Klammern hin und her, dies gilt für `()` runde, `[]` eckige und `{}` geschweifte Klammern. Besonders beim Programmieren ist diese Funktion sehr nützlich. Wie Ihnen vielleicht bereits aufgefallen ist, zeigt Vim beim Eingeben einer schließenden Klammer stets kurz die zugehörige öffnende an. Mit Zähler hat das Prozentzeichen übrigens eine andere Aufgabe: Es dient tatsächlich der prozentualen Navigation. Mit `[5][0][%]` gelangen Sie zum Beispiel in die Mitte des Dokuments.

10.1.2 Textsuchbefehle

Vim verfügt über leistungsfähige, Regexp-basierte Suchfunktionen. Um nach dem nächsten Vorkommen eines regulären Ausdrucks zu suchen, geben Sie im Befehlsmodus `/` ein, das heißt `[↔]+[7]`. Tippen Sie anschließend einen beliebigen String oder regulären Ausdruck ein, nach dem Sie suchen möchten, und drücken Sie zum Schluss `[↵]`. Innerhalb der Eingabezeile können Sie mit den Pfeiltasten `[←]` und `[→]` sowie mit `[Entf]` und `[Backspace]` Korrekturen vornehmen. Mit den Pfeiltasten `[↑]` und `[↓]` können Sie die bisherigen Suchbegriffe durchblättern; dies funktioniert genau wie die Shell-History. Sollten Sie während der Eingabe zu dem Schluss kommen, dass Sie doch nicht suchen möchten, drücken Sie `[Esc]`.

Die folgenden Zeichen haben in der von Vim unterstützten Regexp-Teilmenge eine besondere Bedeutung: `.` `*` `[]` `^` `%` `/` `\` `?` `~` `$`. Soll eines dieser Zeichen in Ihrem Suchbegriff vorkommen, müssen Sie ihm wie üblich einen Backslash voranstellen.

Falls Sie nach einem weiteren Vorkommen desselben Ausdrucks suchen möchten, genügt `/` `[↵]`. Um von der aktuellen Cursorposition aus rückwärts zu suchen, müssen Sie `[?]` (`[↔]+[B]`) statt `/` verwenden. Ein weiteres Vorkommen des Ausdrucks in derselben Richtung finden Sie dann mit `[?]` `[↵]`.

Eine noch schnellere Möglichkeit, die Suche in der jeweils aktuellen Richtung zu wiederholen, bietet die Taste `[N]`. `[↔]+[N]` sucht ebenfalls weiter, kehrt allerdings die bisherige Suchrichtung um. Wenn Sie dem `[N]` einen Zähler voranstellen, werden entsprechend einige Suchergebnisse übersprungen; `[3][N]` sucht etwa nach dem dritten Vorkommen des Suchmusters ab der aktuellen Cursorposition.

Wird das Dokumentende erreicht, so sucht Vim an dessen Anfang weiter; in der Statuszeile erscheint eine entsprechende Mitteilung. Entsprechend wird am Ende weitergesucht, falls eine Rückwärtssuche den Anfang überschreitet. Wenn Sie dieses Verhalten ausschalten möchten, ist die Eingabe eines Befehls erforderlich. Befehle beginnen mit `:` und enden mit `↵`. Der Befehl zum Abschalten des Suchumbruchs lautet `: nowrapscan ↵`; um ihn wieder einzuschalten, wird `: wrapscan ↵` verwendet.

Neben den bekannten, in Kapitel 4 besprochenen Regexp-Konstrukten verwendet Vim eine besondere Syntax für Wortanfänge beziehungsweise -enden: `\<` bezeichnet einen Wortanfang, `\>` ein Wortende. `\<öl` findet also das Wort »ölig«, aber nicht »Köln«. Entsprechend trifft `öl\>` auf »Erdöl«, aber nicht auf »einölen« zu. `\<öl\>` schließlich trifft nur auf »öl« (klein geschrieben) als ganzes Wort zu.

Im Standardmodus unterscheidet die Suche zwischen Groß- und Kleinschreibung. Um dies zu ändern, müssen Sie den Befehl `: set ignorecase ↵` eingeben. Entsprechend schaltet `: set noignorecase ↵` sie wieder ein.

Eine weitere interessante Funktion ist die Suche nach einem weiteren Vorkommen des Wortes, auf dem sich der Cursor gerade befindet. Drücken Sie `*` – `+↵` – für eine Vorwärts- und `#↵` für eine Rückwärtssuche nach diesem Wort. Wie üblich überspringt ein vorangestellter Zähler einige Ergebnisse. Diese beiden Befehle suchen ausschließlich nach ganzen Wörtern; wenn Sie das aktuelle Wort als Teilstring suchen möchten, lauten die Kommandos `G *` beziehungsweise `G #`.

Suchoptionen

Als nützliche Zusatzoption steht die Markierung (Highlighting) sämtlicher Vorkommen eines Suchbegriffs zu Verfügung. Diese wird mittels `: set hlsearch ↵` aktiviert. Künftige Suchbefehle invertieren jede Fundstelle im gesamten Dokument. Wenn Sie die Funktion wieder abschalten möchten, müssen Sie den Befehl `: set nohlsearch ↵` eingeben. Wollen Sie dagegen nur die aktuelle Markierung ausblenden, das Highlighting als solches aber beibehalten, dann lautet der Befehl `: nohlsearch ↵`.

Hier noch einige weitere Suchoptionen: `: set incsearch ↵` schaltet die inkrementelle Suche ein – bei der Eingabe der einzelnen Zeichen wird die nächstgelegene, auf den bisherigen Teilausdruck passende Fundstelle markiert. Ausschalten können Sie diese Funktion erwartungsgemäß mit `: set noincsearch ↵`.

Vim kann nicht nur suchen, sondern auch automatisch ersetzen. Um das nächste Vorkommen von »Text1« durch »Text2« zu ersetzen, können Sie Folgendes ein-

geben: `:S / Text1 / Text2 ↵`. Um alle Vorkommen von »Text1« in der aktuellen Zeile zu ersetzen, müssen Sie den Optionsschalter `G` hinzufügen: `:S / Text1 / Text2 / G ↵`. Ein Prozentzeichen vor dem `S` sorgt zudem dafür, dass alle Zeilen durchsucht werden – `:%S / Text1 / Text2 / ↵ ↵` ersetzt jeweils das erste Vorkommen in jeder Zeile, während `:%S / Text1 / Text2 / G ↵` die Ersetzung im gesamten Text durchführt. Wenn Sie nicht jedes Vorkommen ersetzen möchten, können Sie zusätzlich die Option `C` verwenden, um vor jeder Ersetzung einzeln gefragt zu werden: `:%S / Text1 / Text2 / G C ↵`. Beachten Sie aber, dass es mühselig sein kann, einige hundert Bestätigungen durchzuführen – in manchen Fällen ist es einfacher, die Suchkriterien mithilfe regulärer Ausdrücke genauer einzugrenzen.

10.1.3 Sprungmarken

Bei jedem Sprung mit `G` und jedem der gerade besprochenen Suchbefehle merkt Vim sich die vorherige Cursorposition als automatische Marke. Um zu dieser Marke zurückzukehren, können Sie `^ ^` drücken, das heißt zweimal `↵+↵`. Wenden Sie den Befehl ein weiteres Mal an, so springt er wieder zum vorherigen Sprungziel und dann immer zwischen den beiden Positionen hin und her. Mit `Strg+O` können Sie schrittweise zu früheren Marken zurückblättern, mit `Strg+I` oder `↵` wieder vorwärts.

Mit `M`, gefolgt von einem beliebigen Buchstaben, können Sie an der aktuellen Cursorposition eine benannte Marke setzen, zum Beispiel `M A` oder `M X`. Somit stehen 26 solcher Marken zur Verfügung. Um später zu einer solchen Marke zu springen, geben Sie `^` und den entsprechenden Buchstaben ein, also etwa `^ A` beziehungsweise `^ X`. Wenn Sie `'` – `↵+#` – statt des Backticks verwenden, beispielsweise `' A`, springt der Cursor nicht zur Marke selbst, sondern zum Anfang der Zeile, die diese enthält.

Der Befehl `:marks ↵` zeigt eine Liste aller Marken an, die beispielsweise so aussieht (die Zeilen wurden jeweils gekürzt, damit sie in den Satzspiegel passen):

```
:marks
Mark Zeile Sp Datei/Text
'      23   0 Vim verfügt über leistungsfähige, Regexp-
a      22   0 =Suchbefehle=
b      33   0 =Sprungmarken=
c      35  157 Mit (M), gefolgt von einem beliebigen
m      18   0 Die Taste (F), gefolgt von einem
0     466   9 /usr/share/vim/vim63/doc/usr_03.txt
1      21   0
```

```

2      17      0 ~/susebuch/draft1.txt
3     286      1 ~/susebuch/commandref.txt
4       1      0 ~/susebuch/testcommands2.txt
5      21      0 ~/susebuch/testcommands.txt
6      43      0 ~/susebuch/testcommands.txt
7     590      0 ~/susebuch/commandref.txt
8     563      0 ~/susebuch/commandref.txt
9     545      0 ~/susebuch/commandref.txt
"      31    117 Hier noch einige weitere Suchoptionen:
[      36      0 Der Befehl (:):marks(ENTER) zeigt eine
]      37      0
^      37      0
.      36     92 Der Befehl (:):marks(ENTER) zeigt eine

```

Die Beispielliste zeigt die benannten Marken a, b, c und m. Die nummerierten Marken 0 bis 9 verweisen auf die zuletzt bearbeiteten Dateien – bevor Sie zu einer von ihnen wechseln, sollten Sie die Arbeit an der aktuellen Datei mit `:` `W` `←` speichern. Daneben gibt es noch einige Spezialmarken; hier die wichtigsten: ' ist die Cursorposition vor dem letzten Sprung. " bezeichnet die Position beim letzten Speichern der Datei. [und] ist die Anfangs- beziehungsweise Endposition der letzten Änderung.

10.1.4 Löschen, Kopieren und Einfügen

Im Eingabemodus können Sie das Zeichen unter dem Cursor mit der `Entf`-Taste löschen. Backspace löscht das zuletzt eingegebene Zeichen vor dem Cursor; dies funktioniert allerdings nur mit Zeichen, die seit dem letzten Wechsel in den Eingabemodus hinzugefügt wurden. Im Befehlsmodus kann auch die Taste `X` benutzt werden, um das Zeichen unter dem Cursor zu löschen; `⇧+X` dagegen entfernt das Zeichen davor. Wie üblich können Sie einen Zähler verwenden, um mehrere Zeichen zu löschen. Beispielsweise löscht `5 X` das Zeichen unter dem Cursor und die vier nachfolgenden Zeichen.

Um größere Blöcke zu löschen, wird die Taste `D` (für delete), gefolgt von einem der weiter oben beschriebenen Bewegungskommandos, verwendet. Beispielsweise löscht `D $` die aktuelle Zeile ab der aktuellen Position, während `D 0` umgekehrt alle Zeichen vom Zeilenanfang bis zur Cursorposition löscht. Auch Kombinationen wie `D 4 W` zum Löschen der nächsten vier Wörter oder `D 1 0 H` zum Entfernen der zehn Zeichen vor dem Cursor sind möglich. Sie können sogar den Text von der aktuellen Cursorposition bis zu einer beliebigen Marke löschen, indem Sie beispielsweise `D ~ A` eingeben. Zum Löschen einer ganzen Zeile müssen Sie `D D` eingeben; `D 2 D` löscht entsprechend

zwei Zeilen. Ein praktisches Kürzel ist $\text{D} + \text{D}$ für D $\text{\$}$, das heißt für das Löschen bis zum Zeilenende.

Einfügen

Der zuletzt gelöschte Bereich wird automatisch in einem Puffer zwischengespeichert, das heißt, er wird eigentlich ausgeschnitten statt gelöscht. Mit P können Sie den zuletzt ausgeschnittenen Text an der aktuellen Cursorposition einfügen, mit $\text{D} + \text{P}$ dahinter. Bei Bedarf können Sie dies beliebig oft wiederholen oder auch einen Zähler verwenden, um die Anzahl der einzufügenden Kopien festzulegen. Um bei einem Dreher schnell zwei Zeichen zu vertauschen, platzieren Sie den Cursor auf dem vorderen und drücken Sie X P . Beispiel: Sie haben versehentlich »dre« statt »der« eingegeben. Wenn Sie auf dem r die Taste X drücken, verschwindet es; der Befehl P fügt es anschließend hinter dem e wieder ein.

Wenn Sie den Befehl C (change) statt D verwenden, wird ebenfalls Text in der gewünschten Richtung und Menge gelöscht, aber anschließend befinden Sie sich automatisch im Eingabemodus. Zum Ersetzen einer ganzen Zeile wird in diesem Fall C C verwendet. Auch bei dieser Option befindet sich der gelöschte Text im Puffer; um ihn wieder einzufügen, müssen Sie den Einfügemodus aber natürlich mit ESC verlassen. Wenn Sie nur das Zeichen unter dem Cursor ersetzen möchten, können Sie statt C L auch S drücken; $\text{D} + \text{S}$ ersetzt dagegen den Rest der Zeile ab der aktuellen Position, genau wie C $\text{\$}$. Noch interessanter ist unter Umständen R – dieser Befehl ersetzt das Zeichen unter dem Cursor durch das als Nächstes Eingetippte, ohne überhaupt in den Eingabemodus zu wechseln.

Manchmal kann es vorkommen, dass Sie bestehenden Text zeichenweise überschreiben möchten. Dies erledigt der Ersetzungsmodus, den Sie mit $\text{D} + \text{R}$ einschalten können. Wenn Sie das letzte Zeichen einer Zeile überschrieben haben, wird diese übrigens automatisch verlängert; Zeilenumbrüche werden nicht eigenmächtig eingefügt. Außerdem können Sie die Taste Einf verwenden, um zwischen dem Einfüge- und dem Ersetzungsmodus zu wechseln.

Schließlich existiert auch noch eine Option, um Text nicht zu löschen, sondern nur in den Puffer zu kopieren – die entsprechende Taste ist Y für yank. Y Y kopiert in diesem Fall eine komplette Zeile.

10.1.5 Textobjekte

Ein weiteres Hilfsmittel zum Löschen und Kopieren ist die Arbeit mit Textobjekten. Anders als die Sprungbefehle beziehen sich diese nämlich jeweils auf das aktuelle Objekt. Angenommen, der Cursor befindet sich gerade mitten in einem

Wort, das Sie löschen möchten. Dies geht am schnellsten, wenn Sie **D** **A** **W** eintippen – das Textobjekt `aw` steht für »a word«, also für das aktuelle Wort.

Tabelle 10.1 zeigt einige wichtige Textobjekte im Überblick.

Textobjekt	Name	Funktion
<code>aw</code>	a word	Das aktuelle Wort; mit Zähler auch folgende.
<code>iw</code>	inner word	Das aktuelle Wort; Leerzeichen gelten als eigene Wörter.
<code>as</code>	a sentence	Der aktuelle Satz mit nachfolgendem Leerzeichen.
<code>is</code>	inner sentence	Der aktuelle Satz ohne Leerzeichen.
<code>ap</code>	a paragraph	Der aktuelle Absatz mit nachfolgender Leerzeile.
<code>ip</code>	inner paragraph	Der aktuelle Absatz ohne Leerzeile.

Tabelle 10.1 Die wichtigsten Vim-Textobjekte

Als Satz im Sinne der Textobjekte gilt eine Zeichenfolge, die durch `.`, `!` oder `?` und ein nachfolgendes Leerzeichen abgeschlossen wird. Ein Absatz wird durch eine Leerzeile begrenzt.

10.1.6 Der visuelle Modus

Im Gegensatz zum ursprünglichen `vi` kennt Vim eine Alternative zur Bestimmung der Zeichen, die gelöscht oder kopiert werden: den visuellen Modus (Visual Mode). Sobald Sie im Befehlsmodus **V** drücken, markieren alle Bewegungsbefehle und sogar Textobjekte den Text ab der aktuellen Cursorposition; dies wird durch Invertierung gekennzeichnet. Sobald der gewünschte Textbereich markiert ist, können Sie eine der folgenden Tasten drücken: **D** zum Löschen, **C** zum Ändern (Löschen mit Wechsel in den Eingabemodus) oder **Y** zum Kopieren. Um den visuellen Modus unverrichteter Dinge zu verlassen, können Sie **Esc** drücken.

Eine spezielle Variante ist der visuelle Zeilenmodus (Visual Line Mode). Er wird mit **↵**+**V** aktiviert und markiert grundsätzlich ganze Zeilen. Noch ein wenig komplexer ist der visuelle Blockmodus (Visual Block Mode), der mit **Strg**+**V** gestartet wird. Hier können Sie rechteckige Blöcke markieren, was in der Regel nur einen Sinn ergibt, wenn Ihre Zeilen kürzer sind als die Fensterbreite. Ein gutes Beispiel ist die Arbeit mit Tabellen. Stellen Sie sich vor, Sie haben folgende Tabelle eingegeben:

Name	Vorname	Beruf
----	-----	-----
Becker	Adelbert	Bäcker
Müller	Friedbert	Müller
Dankwart	Diethelm	Tankwart
Schmidt	Gerhard	Schmied

Jetzt fällt Ihnen ein, dass Sie die Spalten mit den Vor- und Nachnamen vertauschen möchten. Platzieren Sie dazu den Cursor auf dem V von »Vornamen«. Drücken Sie anschließend `[Strg]+[V]` und bewegen Sie sich dann beispielsweise mit `[J]` und `[L]` nach rechts unten, bis der gesamte Block bis links neben dem S von »Schmied« markiert ist. Die Taste `[D]` löscht diesen Block und speichert ihn im Puffer. Bewegen Sie den Cursor auf das N von »Name« und drücken Sie `[>]+[P]`, um den Block links daneben einzufügen. Sie erhalten das folgende Ergebnis:

Vorname	Name	Beruf
-----	----	-----
Adelbert	Becker	Bäcker
Friedbert	Müller	Müller
Diethelm	Dankwart	Tankwart
Gerhard	Schmidt	Schmied

In allen Varianten des visuellen Modus' können Sie die Taste `[O]` benutzen, um zum jeweils anderen Ende der Markierung zu wechseln und diese so auf beiden Seiten zu erweitern beziehungsweise zu verringern.

10.1.7 Befehle wiederholen, rückgängig machen und wiederherstellen

Wenn Sie einen Fehler gemacht haben, können Sie die Taste `[U]` betätigen, um die letzte Änderung zurückzunehmen, mit vorangestelltem Zähler auch mehrere Änderungen auf einmal. Das Gegenteil erreichen Sie mit `[Strg]+[R]`: Wenn Sie eine Änderung zu Unrecht rückgängig gemacht haben, können Sie damit den vorherigen Zustand wiederherstellen. Hier ein Komplettbeispiel:

1. Sie wechseln in den Eingabemodus, geben den Satz »nur ein Tset« ein, und verlassen den Eingabemodus wieder:

```
[I] nur ein Tset [Esc]
```

2. Sie stellen fest, dass *s* und *e* in »Tset« vertauscht werden müssen. Sie möchten `[X]` `[P]` verwenden, um die beiden Buchstaben zu vertauschen. Irrtümlich platzieren Sie den Cursor aber auf dem *e* statt auf dem *s* und erhalten:

```
nur ein Tste
```

3. Drücken Sie nun dreimal hintereinander `U` oder direkt `3 U` (jedenfalls einmal zu viel) – die gesamte Eingabe verschwindet.
4. Jetzt können Sie `Strg+R` drücken, um die irrtümlich entfernte Eingabe zurückzuholen.
5. Zu guter Letzt können Sie den Cursor nun auf dem `s` von »Tset« platzieren und `X P` drücken, um die korrekte Änderung durchzuführen.

Eine etwas andere Aufgabe als `Strg+R` hat die Taste `.`: Sie wiederholt den letzten Befehl. Beachten Sie, dass Bewegungs- und Suchbefehle auf diese Weise genauso wenig wiederholt werden können wie das soeben besprochene Rückgängigmachen. Es geht ausschließlich um echte Änderungen.

Angenommen, Sie möchten »vi« in einem Text an fast allen Stellen durch »Vim« ersetzen (aber nicht an allen, weil darin beispielsweise die Entwicklung von Vim aus dem ursprünglichen `vi` beschrieben wird). Suchen Sie zunächst mit `/ vi` nach der Zeichenkette »vi«. Wenn Sie eine Stelle gefunden haben, die geändert werden soll, drücken Sie `C W`, tippen Sie `Vim` ein und drücken Sie `Esc`. Nun können Sie mit `N` weitersuchen. Jedes »vi«, das ausgetauscht werden soll, können Sie nun einfach mit `.` ändern.

10.1.8 Datei- und Fensterverwaltung

Für den Umgang mit Dateien und Editorfenstern stehen diverse Befehle zur Verfügung. Um die aktuelle Datei zu speichern, müssen Sie `: W` drücken. Alternativ können Sie `: W Leertaste Neuer-Name` eingeben, um die aktuelle Datei unter einem neuen Namen zu speichern. Dies speichert eine Kopie der Datei; das nächste `: W` speichert wieder unter dem ursprünglichen Namen.

Um das Bearbeiten der aktuellen Datei zu beenden und zu einer anderen zu wechseln, wird der Befehl `: edit Dateiname` verwendet. Falls Sie die bisherige Datei nicht gespeichert hatten, erhalten Sie zunächst nur eine Fehlermeldung wie diese:

```
E37: No write since last change (add ! to override)
```

In diesem Fall können Sie entweder mit `: W` speichern oder aber Sie wiederholen den `edit`-Befehl mit einem zusätzlichen Ausrufezeichen, um die Datei ungespeichert zu schließen: `: edit ! Dateiname`. Wenn Sie eine völlig neue Datei bearbeiten möchten, geben Sie einfach einen neuen Namen ein, der noch nicht verwendet wird.

Wie bereits beschrieben, können Sie Vim bei Bedarf zur Bearbeitung mehrerer Dateien öffnen. Der Wechsel zur jeweils nächsten Datei erfolgt dann mit `: next`

`↵`). Hier kann die Variante `next` `!` verwendet werden, wenn Sie die vorherige Datei nicht speichern möchten; alternativ existiert die Schnellschreibweise `:` `wnext` `↵` zum Speichern und anschließenden Wechseln.

Um eine Datei vom Datenträger einzulesen, können Sie auch `:` `R` `Dateiname` `↵` eingeben. Dies öffnet keinen neuen Editor, sondern fügt die gesamte Datei an der aktuellen Cursorposition ein – ideal, um mehrere Dateien zusammenzukopieren.

Alternativ besteht die Möglichkeit, die aktuelle Datei zu verbergen, um eine andere zu bearbeiten. Geben Sie dazu `:` `hide edit` `Dateiname` `↵` ein. Das Ergebnis sind mehrere sogenannte Puffer (englisch buffers) zur Bearbeitung von Dateien. Wenn Sie `:` `buffers` `↵` oder das weniger eingängige `:` `ls` `↵` eingeben, wird eine nummerierte Liste aller geöffneten Puffer angezeigt. Wenn Sie zu einem anderen Puffer wechseln möchten, müssen Sie `:` `buffer` `Nummer` `↵` eingeben, also zum Beispiel `:` `buffer 2` `↵`. Statt der Nummer können Sie auch einen Namensbestandteil eingeben; Vim wechselt zur ersten Datei, auf die dieser passt.

Mehrfenstermodus

Wenn Sie bei der Arbeit mit einem bestimmten Text einen anderen lesen möchten, besteht die Möglichkeit, den Editorbereich in zwei Fenster zu unterteilen. Der Befehl dafür lautet `:` `split` `↵`. Abbildung 10.1 zeigt den Zustand unmittelbar nach dem Teilen: Im oberen und im unteren Fenster wird jeweils dieselbe Datei bearbeitet. Um zwischen den beiden Fenstern hin- und herzuwechseln, wird die Tastenfolge `Strg+W` `W` verwendet. Solange Sie in beiden Fenstern dieselbe Datei bearbeiten, betreffen Änderungen auch beide Textdarstellungen – solange dieselben Zeilen dargestellt werden, können Sie dies sogar live beobachten.

Voneinander getrennt werden die Fenster erst, wenn Sie in einem von ihnen eine andere Datei öffnen, zum Beispiel mit dem soeben beschriebenen `:` `edit` `Dateiname` `↵`. Da die bisherige Datei zu diesem Zeitpunkt noch im anderen Fenster offen ist, brauchen Sie in diesem Fall auch kein `!`, um den Wechsel zu erzwingen. Schneller geht es, den `split`-Befehl gleich mit dem Namen einer anderen Datei einzugeben. Beispiel: `:` `split` `datei2.txt` `↵`. Ein weiterer Befehl öffnet ein neues, leeres Fenster: `:` `new` `↵`.

Um das aktuelle Fenster wieder zu schließen, können Sie `:` `close` `↵` eingeben, aber auch andere Befehle zum Beenden wie `:` `Q` `!` oder `␣+Z` `␣+Z` funktionieren.

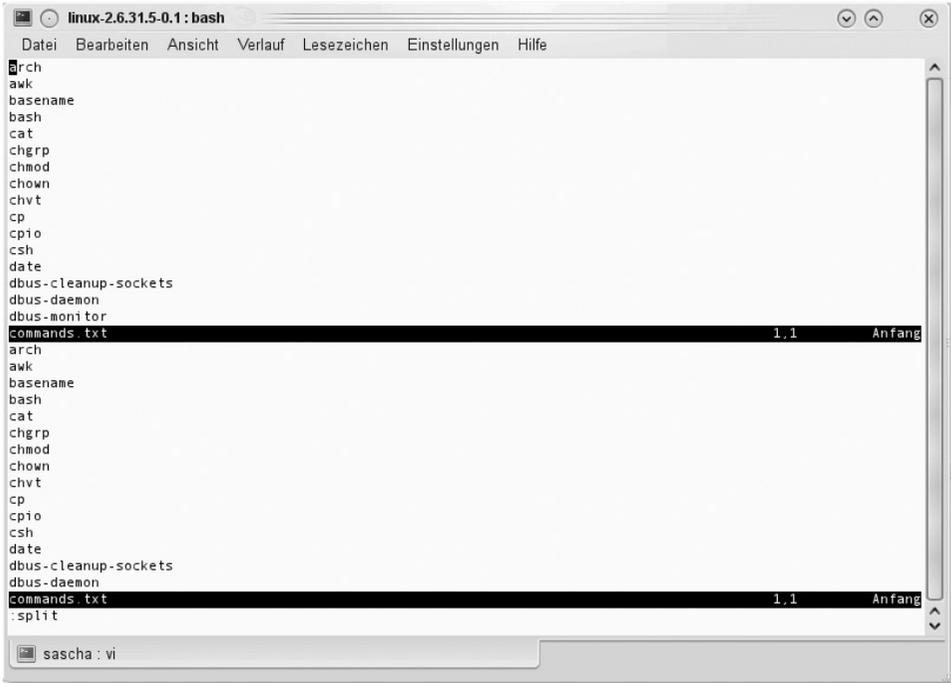


Abbildung 10.1 Vim nach dem Teilen der Ansicht in zwei Fenster

Wenn Sie mehrere `split`- oder `new`-Befehle verwenden, wird das Fenster entsprechend mehrfach unterteilt. Nicht nur in diesem Zusammenhang kann es sinnvoll sein, die Größe der neuen Fensterbereiche festzulegen. Dies geschieht, indem Sie dem Befehl eine Zeilenanzahl voranstellen. Beispielsweise erzeugt `:6split` ein neues Fenster für die aktuelle Datei mit sechs Zeichen Höhe. Mit `[Strg]+W +` können Sie das aktuelle Fenster nachträglich um eine Zeile vergrößern, mit `[Strg]+W -` verkleinern; auch diese Befehle erlauben Zähler. Für eine neue absolute Zeilenanzahl wird `Anzahl [Strg]+W Unterstrich` verwendet, zum Beispiel `7 [Strg]+W Unterstrich` für sieben Zeilen.

Alternativ zu `split` können Sie übrigens `:vsplit` verwenden, um das Fenster vertikal statt horizontal zu unterteilen. Die Trennung zwischen den Fenstern wird durch invertierte `|`-Zeichen angezeigt. Wenn Sie das Hauptfenster mehrfach in beide Richtungen unterteilt haben, ermöglicht `[Strg]+W`, gefolgt von einer der vier Grundrichtungstasten `H`, `J`, `K` und `L`, den Wechsel in das Fenster, das in der jeweiligen Richtung liegt.

Im Mehrfenstermodus können Sie `:Q all` eingeben, um Vim insgesamt zu schließen; genauso dient `:W all` dem Speichern aller geöffneten Dateien.

10.1.9 Weitere Features

Nachdem die grundlegende Bedienung von Vim in den vorigen Abschnitten beschrieben wurde, lernen Sie hier noch eine kleine Auswahl fortgeschrittener Funktionen des Editors kennen. Wenn Sie noch mehr wissen möchten, können Sie die umfangreiche Hilfe-Funktion des Programms konsultieren. Geben Sie dazu im Befehlsmodus `:` `help` `<Enter>` ein. Innerhalb der Hilfe finden Sie zahlreiche Verknüpfungen, zum Beispiel im Inhaltsverzeichnis oder auch als Hinweise auf verwandte Themen. Um einen solchen Link zu aktivieren, müssen Sie sich mit dem Cursor zwischen die beiden `|`-Zeichen bewegen und dann `[Strg]+[J]`, auf einer deutschen PC-Tastatur also `[Strg]+[Alt Gr]+[9]`, drücken. Geschlossen wird die Hilfe mit `:` `Q` `<Enter>`.

Daneben sind sogar einige Bücher zum vi(m)-Editor erschienen; entsprechende Hinweise finden Sie in Anhang B.

Ein weiteres nützliches Hilfe-Feature: Wenn Sie auf dem aktuellen Wort die Tastenkombination `<F5>+[K]` betätigen, wird nach einer Manpage zum entsprechenden Thema gesucht. Sobald Sie mit dem Lesen fertig sind, drücken Sie `Q`, um das Programm `man` zu verlassen, und anschließend `<Enter>`, um vom Shell-Bildschirm zu Vim zurückzukehren.

Da moderne Terminalemulationen farbigen Text darstellen können, ist Syntax-Highlighting für diverse Programmier- und Auszeichnungssprachen weitverbreitet. Auch Vim beherrscht diese Funktion; geben Sie `:` `syntax on` `<Enter>` ein, um sie zu aktivieren. In aller Regel erkennt der Editor die Sprache aus der Dateierdung und hebt die Syntax korrekt hervor. Erfolgreich überprüft habe ich es unter anderem mit PHP (Endung `.php`), C (`.c` und `.h`), C++ (`.cpp`), Java (`.java`), Perl (`.pl`), Ruby (`.rb`) HTML (`.htm` oder `.html`) und Shell-Skripten (`.sh`).

Ebenfalls ist für Programmierer die Fähigkeit der automatischen Einrückung interessant. Geben Sie `:` `set autoindent` `<Enter>` oder kurz `:` `set ai` `<Enter>` ein, um sie zu aktivieren. Sobald Sie nun eine Zeile mit Leerzeichen einrücken, bleibt diese Einrückung in der nächsten Zeile erhalten. Benutzen Sie die Taste `<Left>`, um den Text später wieder auszurücken. `:` `set noautoindent` `<Enter>` deaktiviert das Feature wieder.

Sehr nützlich ist auch die Fähigkeit von Vim, zwischendurch einen einzelnen Shell-Befehl auszuführen. Geben Sie dazu `:` `!` `Befehl` `<Enter>` ein. Damit Sie die Ausgabe des Befehls lesen können, bleibt der Shell-Bildschirm stehen, bis Sie `<Enter>` drücken. Alternativ können Sie natürlich auch wie in jedem Konsolenprogramm `[Strg]+[Z]` drücken, beliebig viele Shell-Kommandos ausführen und mit `fg` zu Vim zurückkehren.

Mithilfe der Tastenfolge `G` `Strg`+`G` können Sie die aktuelle Anzahl der Wörter im Text zählen – das Ergebnis ist eine Statistik wie die folgende in der Statuszeile:

```
Spal 1 von 0; Zeile 145 von 145; Wort 4564 von 4564;
Byte 32230 von 32230
```

Einige praktische Helfer erleichtern das Öffnen anderer Dateien. `:` `edit` `.` `↵` öffnet einen kleinen Dateibrowser für das aktuelle Verzeichnis. Sie können sich mit den üblichen Vim-Funktionen darin bewegen. Sobald sich der Cursor auf der gewünschten Datei befindet, können Sie `O` drücken, um sie in einem separaten Fensterbereich zu öffnen. Wenn Sie den Cursor innerhalb der aktuellen Datei auf einen Dateinamen oder eine URL bewegen und `F` `G` drücken, wird Vim versuchen, die entsprechende Ressource zu laden und zum Bearbeiten zu öffnen.

Besonders häufig genutzte `:`-Befehle können Sie übrigens in eine Vim-Startdatei schreiben, damit sie jedes Mal automatisch geladen werden. Diese Datei muss `.vimrc` heißen und in Ihrem Home-Verzeichnis liegen. Innerhalb dieser Datei werden die `ex`-Befehle ohne Doppelpunkt angegeben. Eine kurze Startdatei für Programmierer sieht beispielsweise so aus:

```
syntax on
set autoindent
set number
```

Dies aktiviert Syntaxhervorhebung, automatische Einrückung und die Anzeige von Zeilennummern. Letzteres wurde bisher noch nicht besprochen; wenn Sie es wieder aufheben möchten, müssen Sie wie üblich `:` `set nonumber` `↵` eingeben.

10.1.10 Vim-Befehle im Überblick

Nach den vorangegangenen Tutorial-Abschnitten fasst Tabelle 10.2 noch einmal die wichtigsten der bisher behandelten Tastenkombinationen und Kommandos kompakt zusammen. Speziellere Kommandos wie Textobjekte oder visueller Modus bleiben hier außen vor. Falls nicht anders angemerkt, gelten sie alle für den Befehlsmodus.

Tastenkürzel/Befehl	Bedeutung
Moduswechsel	
<code>I</code>	Wechsel in den Eingabemodus an akt. Position
<code>A</code>	Wechsel in den Eingabemodus nach akt. Position

Tabelle 10.2 Die wichtigsten Vim-Tastenkürzel und -kommandos im Überblick

Tastenkürzel/Befehl	Bedeutung
<code>↩+I</code>	Wechsel in den Eingabemodus am Zeilenanfang
<code>↩+A</code>	Wechsel in den Eingabemodus am Zeilenende
<code>O</code>	Wechsel in den Eingabemodus unter aktueller Zeile
<code>↩+O</code>	Wechsel in den Eingabemodus über aktueller Zeile
<code>Esc</code>	Wechsel aus Eingabemodus in den Befehlsmodus
Navigation im Text	
<code>H</code>	Cursor um ein Zeichen nach links bewegen
<code>J</code>	Cursor um ein Zeichen nach unten bewegen
<code>K</code>	Cursor um ein Zeichen nach oben bewegen
<code>L</code>	Cursor um ein Zeichen nach rechts bewegen
<code>O</code> (Null, nicht O)	Cursor an den Zeilenanfang
<code>↩+4</code> , d. h. <code>\$</code>	Cursor ans Zeilenende
<code>↩+8</code> , d. h. <code>(</code>	Cursor zum vorigen Satzanfang
<code>↩+9</code> , d. h. <code>)</code>	Cursor zum nächsten Satzanfang
<code>Alt Gr+7</code> , d. h. <code>[</code>	Cursor zum vorigen Absatzanfang
<code>Alt Gr+9</code> , d. h. <code>]</code>	Cursor zum nächsten Absatzanfang
n <code>Alt Gr+<</code> , d. h. n <code>[</code>	Cursor zu Zeichen n in der aktuellen Zeile
<code>-</code>	Cursor zum Anfang der vorigen Zeile
<code>+</code> oder <code>↵</code>	Cursor zum ersten Nicht-Leerzeichen der nächsten Zeile
<code>↩+H</code>	Cursor zum Anfang der ersten sichtbaren Zeile
<code>↩+M</code>	Cursor zum Anfang der Zeile in der Bildschirmmitte
<code>↩+L</code>	Cursor zum Anfang der letzten sichtbaren Zeile
n <code>↩+5</code> , d. h. n <code>%</code>	Cursor in die Zeile, die dem angegebenen prozentualen Anteil des Textes entspricht
<code>W</code>	Cursor ein Wort weiter (Satzzeichen als eigene Wörter)
<code>↩+W</code>	Cursor ein Wort weiter (inkl. Satzzeichen)
<code>B</code>	Cursor ein Wort zurück (Satzzeichen als eigene Wörter)
<code>↩+B</code>	Cursor ein Wort zurück (inkl. Satzzeichen)

Tabelle 10.2 Die wichtigsten Vim-Tastenkürzel und -kommandos im Überblick (Forts.)

Tastenkürzel/Befehl	Bedeutung
<code>E</code>	Cursor zum nächsten Wortende
<code>G E</code>	Cursor zum vorigen Wortende
<code>n G</code>	Cursor in Zeile <i>n</i>
<code>G G</code>	Cursor an den Textanfang
<code>↶+G</code>	Cursor an den Anfang der letzten Zeile
<code>n</code> Tastenkürzel	Die angegebene Funktion <i>n</i> -mal ausführen (z. B. <code>3 H</code> für drei Zeichen nach links; nicht nur für Navigation verfügbar)
Scrollen (Bildschirm bewegen, ohne den Cursor zu verschieben)	
<code>Strg+Y</code>	Eine Zeile nach oben scrollen
<code>Strg+E</code>	Eine Zeile nach unten scrollen
<code>Strg+U</code>	Einen halben Bildschirm nach oben scrollen
<code>Strg+D</code>	Einen halben Bildschirm nach unten scrollen
<code>Strg+B</code>	Einen ganzen Bildschirm nach oben scrollen
<code>Strg+F</code>	Einen ganzen Bildschirm nach unten scrollen
<code>Z Z</code>	Die Zeile mit dem Cursor in die Bildschirmmitte scrollen
<code>Z T</code>	Die Zeile mit dem Cursor an den oberen Bildschirmrand scrollen
<code>Z B</code>	Die Zeile mit dem Cursor an den unteren Bildschirmrand scrollen
Zeichen suchen	
<code>F Zeichen</code>	Erstes Vorkommen des angegebenen Zeichens in der aktuellen Zeile suchen
<code>n F Zeichen</code>	Das <i>n</i> . Vorkommen des angegebenen Zeichens in der aktuellen Zeile suchen
<code>↶+F Zeichen</code>	Rückwärts in der aktuellen Zeile nach dem angegebenen Zeichen
<code>:</code>	Den vorherigen Suchbefehl wiederholen
<code>.</code>	Den vorherigen Suchbefehl in die andere Richtung wiederholen

Tabelle 10.2 Die wichtigsten Vim-Tastenkürzel und -kommandos im Überblick (Forts.)

Tastenkürzel/Befehl	Bedeutung
<code>%</code>	Zwischen korrespondierenden Klammern (auch eckigen oder geschweiften) hin- und herspringen. Mit Zähler völlig andere Bedeutung (siehe unter Navigation)
Volltextsuche und Ersetzen	
<code>/ Regexp ↵</code>	Vorwärts im Text nach dem angegebenen regulären Ausdruck suchen
<code>? Regexp ↵</code>	Rückwärts im Text nach dem angegebenen regulären Ausdruck suchen
<code>N</code>	In der aktuellen Richtung erneut nach dem vorherigen regulären Ausdruck suchen
<code>↵+N</code>	In der entgegengesetzten Richtung erneut nach dem vorherigen regulären Ausdruck suchen
<code>: wrapscan ↵</code>	Suche darf Dokumentende/-anfang überschreiten (Standard)
<code>: nowrapscan ↵</code>	Suche darf Dokumentende/-anfang nicht mehr überschreiten
<code>: set hlsearch ↵</code>	Künftig alle gefundenen Ergebnisse hervorheben
<code>: nohlsearch ↵</code>	Die aktuelle Hervorhebung ausblenden
<code>: set nohlsearch ↵</code>	Die gefundenen Ergebnisse nicht mehr hervorheben
<code>: set incsearch ↵</code>	Inkrementelle Suche – bereits während der Eingabe zu suchen beginnen
<code>: set noincsearch ↵</code>	Die inkrementelle Suche abschalten
<code>: S / Text1 / Text2 ↵</code>	Das nächste Vorkommen von Text1 in der aktuellen Zeile durch Text2 ersetzen
<code>: S / Text1 / Text2 ↵ G ↵</code>	Alle Vorkommen von Text1 in der aktuellen Zeile durch Text2 ersetzen
<code>: % S / Text1 / Text2 ↵ ↵</code>	Das erste Vorkommen von Text1 in jeder Zeile durch Text2 ersetzen
<code>: % S / Text1 / Text2 ↵ G ↵</code>	Alle Vorkommen von Text1 in jeder Zeile (d.h. alle Vorkommen im gesamten Text) durch Text2 ersetzen
<code>: % S / Text1 / Text2 ↵ G C ↵</code>	Wie der vorherige Befehl, aber vor jedem einzelnen Ersetzungsvorgang nachfragen (die Option <code>C</code> steht auch für alle anderen Optionen zur Verfügung)

Tabelle 10.2 Die wichtigsten Vim-Tastenkürzel und -kommandos im Überblick (Forts.)

Tastenkürzel/Befehl	Bedeutung
Sprungmarken	
<code>^ ^</code>	Sprung zur letzten Sprungmarke (Ergebnis einer Suche oder eines <code>G</code> -Sprungbefehls)
<code>Strg + O</code>	In der Liste der Sprungmarken schrittweise rückwärts blättern
<code>Strg + I</code>	In der Liste der Sprungmarken schrittweise vorwärts blättern
<code>M A bis M Z</code>	An der aktuellen Cursorposition eine der benannten Marken A bis Z einrichten
<code>^ A bis ^ Z</code>	Zu einer der benannten Marken A bis Z springen
<code>' A bis ' Z</code>	Zum Anfang der jeweiligen Zeile springen, in der sich eine der benannten Marken A bis Z befindet
<code>: marks ↵</code>	Liste aller Marken ausgeben
Löschen, Kopieren und Einfügen	
<code>X</code>	Das Zeichen unter dem Cursor ausschneiden (löschen und in einem Puffer speichern)
<code>↵ + X</code>	Das Zeichen links neben dem Cursor ausschneiden
<code>D Navigationsbefehl</code>	Die dem Navigationsbefehl entsprechende Menge Text ausschneiden: <code>D 3 H</code> löscht beispielsweise drei Zeichen nach links
<code>D D</code>	Ganze Zeile ausschneiden
<code>C Navigationsbefehl</code>	Wie <code>D</code> , aber anschließend in den Eingabemodus wechseln
<code>C C</code>	Wie <code>D D</code> , aber anschließend in den Eingabemodus wechseln
<code>Y Navigationsbefehl</code>	Die dem Navigationsbefehl entsprechende Menge Text in den Puffer kopieren (yank)
<code>Y Y</code>	Ganze Zeile kopieren
<code>P</code>	Den Text aus dem Puffer an der aktuellen Cursorposition einfügen
<code>↵ + P</code>	Den Text aus dem Puffer hinter der aktuellen Cursorposition einfügen
<code>X P</code>	Das aktuelle und das nachfolgende Zeichen vertauschen

Tabelle 10.2 Die wichtigsten Vim-Tastenkürzel und -kommandos im Überblick (Forts.)

Tastenkürzel/Befehl	Bedeutung
Wiederholen, rückgängig machen, wiederherstellen	
<code>U</code>	Den letzten Befehl rückgängig machen (in vi nur einmal, in Vim mehrmals möglich – auch mit Zähler)
<code>Strg+R</code>	Den zuletzt rückgängig gemachten Befehl wiederherstellen (nur Vim)
<code>.</code>	Den vorherigen Befehl wiederholen
Datei- und Fensterverwaltung	
<code>:W ↵</code>	Die aktuelle Datei speichern
<code>:W NeuerName ↵</code>	Eine Kopie der aktuellen Datei unter <code>NeuerName</code> speichern; mit der aktuellen weiterarbeiten
<code>:edit Dateiname ↵</code>	Das Editieren der aktuellen Datei beenden (muss gespeichert sein) und die angegebene laden
<code>:edit !Dateiname ↵</code>	Das Editieren der aktuellen Datei abbrechen (wird nicht gespeichert!) und die angegebene laden
<code>:wedit Dateiname ↵</code>	Die aktuelle Datei automatisch speichern und schließen und die angegebene laden
<code>:hide edit Dateiname ↵</code>	Die aktuelle Datei verbergen und die angegebene laden
<code>:buffers ↵</code>	Nummerierte Liste aller Puffer (geöffneten Dateien)
<code>:buffer Nummer ↵</code>	Zum Puffer mit der angegebenen Nummer wechseln
<code>:Q</code>	vi beenden (alle Dateien müssen gespeichert sein)
<code>:Q !</code>	Beenden von vi erzwingen (nicht gespeicherte Dateien werden verworfen)

Tabelle 10.2 Die wichtigsten Vim-Tastenkürzel und -kommandos im Überblick (Forts.)

10.2 Emacs

Ein weiterer, noch leistungsfähigerer Editor heißt *Emacs*. Der Name »Emacs« steht für »Editor Macros«, weil er ursprünglich als Satz von Makrobefehlen für einen älteren Editor entworfen wurde. Einer der wichtigsten Autoren der ersten Versionen ist der GNU-Begründer Richard Stallman.

Inzwischen ist Emacs ein eigenständiger Editor, der in vielen verschiedenen Varianten für etliche Plattformen verfügbar ist – nicht nur für alle erdenklichen

UNIX-Arten, sondern beispielsweise auch für Windows. Die Hauptentwicklungslinie ist der GNU Emacs, den Sie von der Homepage des GNU-Projekts (www.gnu.org) oder den zahlreichen weltweiten Mirror-Sites herunterladen können. Die aktuelle Version ist 23.x, was Ihnen einen Eindruck davon vermitteln mag, wie lange diese Software bereits existiert – das Programm ist ausgereift und sehr stabil.

Eine weitere Variante trägt den Namen XEmacs, was ein wenig verwirrend ist, weil beide Versionen heute standardmäßig im GUI-Modus ausgeführt werden. Ein modernes SUSE-System stellt beide Typen zur Verfügung; `emacs` startet den GNU Emacs, während `xemacs` den gleichnamigen Editor lädt. Die Unterschiede zwischen den beiden sind relativ geringfügig; in diesem Abschnitt wird speziell der GNU Emacs beschrieben.

Emacs verfügt über einen erheblich größeren Befehlsumfang als `vi`; es handelt sich letzten Endes nicht nur um einen Editor, sondern um einen vollwertigen Shell-Ersatz mit zusätzlichen Funktionen wie E-Mail, unzähligen Betriebsmodi für diverse Textsorten und sogar Spielen.³

Die meisten Emacs-Befehle werden durch Tastenkombinationen mit `Strg` (auf englischen PC-Tastaturen `Ctrl`) oder `Meta` gebildet. Wenn Sie auf Ihre Tastatur schauen, werden Sie feststellen, dass Sie gar keine Meta-Taste haben; die gibt es nur an einigen alten Terminals und verschiedenen UNIX-Workstations. In den meisten Terminalprogrammen oder in der GUI-Version von Emacs kann die `Alt`-Taste dafür verwendet werden. Einige Terminal-Emulationen bestehen dagegen darauf, dass Sie Meta-Tastenkombinationen durch Drücken der Taste `Esc` und anschließendes Betätigen der entsprechenden Kombinationstaste eingeben.

Die Emacs-Dokumentation verwendet eine etwas eigenwillige Schreibweise für die Tastenkombinationen. Diese Schreibweise wird hier kurz erläutert und anschließend verwendet, damit Sie sich nicht umzugewöhnen brauchen, wenn Sie die zahlreichen Hilfeseiten im Emacs selbst lesen: `C-x` für `Strg+X`. Meta-Tastenkürzel werden dagegen mit vorangestelltem `M-` dargestellt. `M-x` bedeutet also je nach Umgebung, in der Sie arbeiten, entweder `Alt+X` oder `Esc X`. Letzteres funktioniert übrigens immer.

Emacs wird mit einer umfangreichen Online-Hilfe geliefert. `C-h t` startet ein interaktives Schritt-für-Schritt-Tutorial, in dem Sie die gängigsten Befehle auspro-

³ Wegen des immensen Leistungsumfangs, gepaart mit der gewöhnungsbedürftigen Steuerung, kursiert folgender Witz: »Emacs ist ein cooles Betriebssystem, dem leider ein vernünftiger Editor fehlt.«

bieren können. C-h k Tastenkombination beschreibt die Funktionalität der entsprechenden Eingabe, während C-h f Funktion Informationen über eine mittels M-x erreichbare Funktion liefert. C-h ? zeigt schließlich eine Liste weiterer Hilfefunktionen an. Noch mehr Informationen bieten die in Anhang B aufgelisteten Emacs-Bücher.

10.2.1 Dateien, Buffer und Fenster

Der GNU Emacs wird durch Eingabe von emacs gestartet. Standardmäßig öffnet sich das in Abbildung 10.2 gezeigte grafische Fenster; die Option -nw beziehungsweise --no-window öffnet dagegen die klassische Ansicht im Terminalfenster. Der Hauptvorteil der grafischen Oberfläche ist das Menü; es zeigt neben jedem Menübefehl die zugehörige Tastenkombination an und ist somit ideal zum Lernen geeignet.

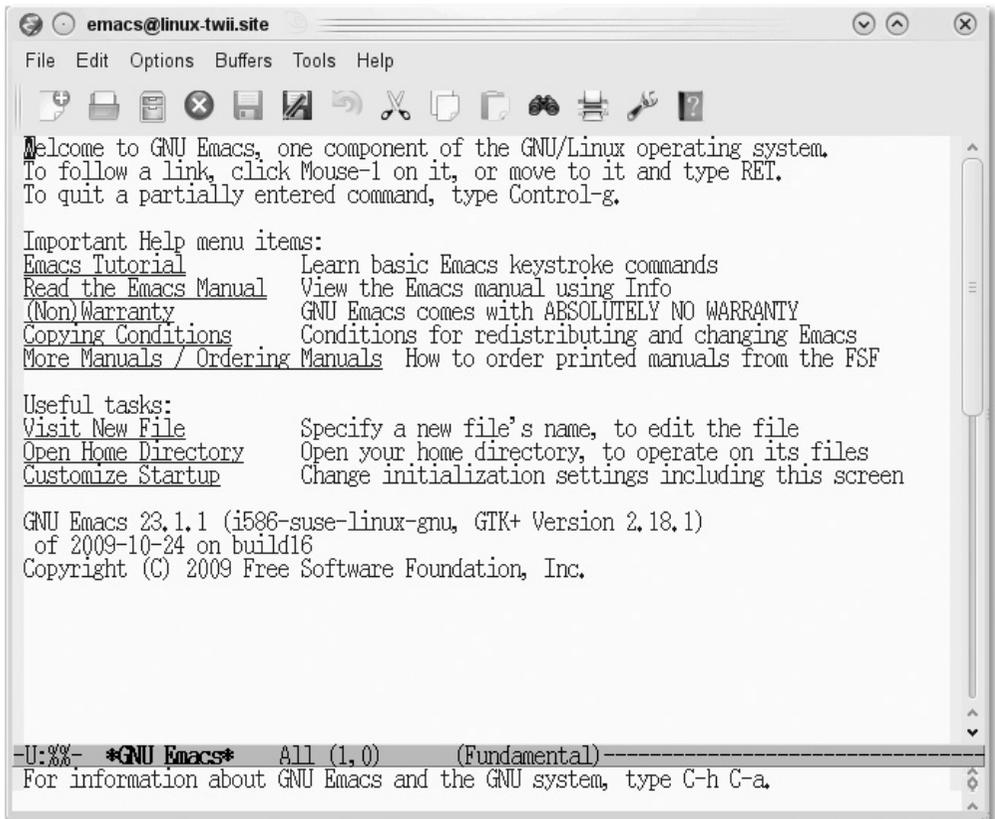


Abbildung 10.2 Das Startfenster des GNU Emacs

Sie können beim Start des Editors eine oder mehrere Dateien zum Bearbeiten angeben. Die einzelnen Arbeitsbereiche für Dateien werden als *Buffer* bezeichnet. Wenn Sie keinen Dateinamen angegeben haben, befinden Sie sich zunächst im *Scratch-Buffer* (Notizblock), der für ungespeicherte Notizen vorgesehen ist. Wenn Sie eine speicherbare Datei editieren möchten, müssen Sie diese zunächst mit `C-x C-f` *Dateiname* »besuchen« (die Emacs-Dokumentation verwendet den Ausdruck »visit«). Falls diese Datei bereits existiert, wird sie geöffnet, ansonsten neu angelegt. Sollten Sie versehentlich die falsche Datei geöffnet haben, drücken Sie `C-x C-v` – der aktuelle Dateiname wird ganz unten angezeigt, sodass Sie ihn verbessern können. Sobald Sie `↵` drücken, wird die angegebene Datei statt der bisherigen geladen. Sollten Sie zu diesem Zeitpunkt bereits Änderungen durchgeführt haben, gehen diese (nach einer Sicherheitsbestätigung) verloren. Eine interessante Variante bietet der Befehl `C-x i`: Er fügt die geladene Datei an der aktuellen Cursorposition in die vorhandene ein.

Wenn mehrere Buffer geöffnet sind, können Sie mithilfe der Tastenkombination `C-x b` *Buffername* `↵` zum Buffer mit dem angegebenen Namen wechseln (ohne Angabe eines Buffers wird reihum zum jeweils nächsten gewechselt). `C-x k` schließt den angegebenen Buffer ganz; wenn Sie einfach `↵` drücken, wird der aktuelle Buffer geschlossen. Eine Liste aller verfügbaren Buffer erhalten Sie mit `C-x C-b` in einem abgetrennten Bereich, der als Fenster bezeichnet wird. Mit `C-x 2` können Sie ein solches Fenster selbst erzeugen; `C-x 4 f` *Dateiname* `↵` öffnet eine neue Datei in einem separaten Fenster. Sie können mit `C-x o` zwischen den beiden Fenstern hin- und herwechseln. Mit `M-C-v` können Sie den Inhalt des Fensters scrollen, in dem Sie zurzeit nicht arbeiten. `C-x 1` schließt das nicht aktive Fenster und behält nur das aktuelle.

Wenn Sie mit `C-x C-b` die Bufferliste geöffnet haben, können Sie mit `C-x o` in diese wechseln. Verwenden Sie `C-n` und `C-p` oder die Pfeiltasten, um sich auf und ab zu bewegen. Drücken Sie in einer beliebigen Zeile `e` oder `f`, um zum angegebenen Buffer zu wechseln. Dieser erscheint im unteren Fenster; `C-x 1` vergrößert dies auf das gesamte GUI-Fenster oder Terminal. In der Bufferliste können Sie die Taste `d` einsetzen, um einen Buffer zum Löschen zu markieren, während `s` ihn zum Speichern vormerkt. `u` entfernt die Markierungen vorzeitig wieder. Ausgeführt werden diese Befehle für beliebig viele Buffer, sobald Sie `x` drücken.

10.2.2 Die Emacs-Modi

Je nach Dateityp (den Emacs an der Dateiendung erkennt) befinden Sie sich in einem der verschiedenen Emacs-Arbeitsmodi, die vor allem Programmierern das Leben erleichtern. Einige der wichtigsten Modi sind `Fundamental` (der grundle-

gende Modus ohne besondere Eigenschaften), `Text` (Modus für einfachen Fließtext wie Artikel oder Ähnliches), `C` (für die gleichnamige Programmiersprache) oder `HTML` (zum Editieren von Webseiten). Jeder dieser Modi verfügt über besondere Funktionen wie automatische Einrückung, Anzeigen der korrespondierenden Klammer beim Schließen oder Ähnliches.

Sie können den Modus auch manuell über `M-x Modusname` wechseln: `M-x fundamental` wechselt beispielsweise in den Fundamental Mode, `M-x c-mode` in den C-Modus oder `M-x html-mode` in den HTML-Modus. `M-x` Schlüsselwort dient allgemein der Eingabe eines Befehls in der Langform und muss jeweils mit `↵` abgeschlossen werden; bei besonders langen Befehlen können Sie nach einigen Zeichen `↵` drücken, weil Emacs – wie die Shell – Eingabevervollständigung durchführt. Wenn Sie dagegen nach einem oder mehreren Anfangsbuchstaben eines Befehls `↵` drücken, erhalten Sie eine Liste der passenden Befehle.

Der einzelige Bereich am unteren Bildschirm- beziehungsweise Fensterrand, in den Sie mit `M-x` gelangen, wird *Minibuffer* genannt. Sie können ihn durch `C-g` wieder verlassen, wenn Sie doch keinen Befehl eingeben möchten. Andere unerwünschte Funktionen können Sie dagegen oft durch dreimaliges Drücken von `Esc` (und nicht etwa `Meta`) abbrechen.

In vielen Modi wird kein automatischer Wortumbruch durchgeführt, das heißt, Emacs fügt keine festen Zeilenumbrüche in den Text ein. In solchen Fällen wird das Ende einer Bildschirmzeile markiert, falls die eigentliche Textzeile länger ist: auf der Konsole durch `\`, in der GUI-Version dagegen durch einen kleinen, umgebogenen Pfeil. Wenn Sie `M-x auto-fill` eingeben, wechselt Emacs in den Auto-fill-Untermodus und sorgt künftig für Wortumbrüche nach höchstens 80 Zeichen. In der Statuszeile macht sich dies durch den Zusatz `Fill` in der Modusangabe bemerkbar; statt `(Text)` steht dort dann beispielsweise `(Text Fill)`. Natürlich gerät ein Absatz wieder aus den Fugen, wenn Sie ihn nachträglich editieren – in diesem Fall können Sie `M-q` drücken, um ihn neu umzubrechen. Beachten Sie aber, dass Sie Absätze dafür durch Leerzeilen trennen müssen – andernfalls wird der gesamte Text als ein einziger langer Absatz betrachtet, was speziell Fachtexten mit Aufzählungen oder Listings überaus schlecht bekommt.

Ein weiterer Untermodus ist der *Ersetzungsmodus* – hier wird das Zeichen unter dem Cursor jeweils durch ein neu eingetipptes ersetzt. Sie schalten diesen Modus durch die Taste `Einfg` ein und wieder aus. Der Ersetzungsmodus wird durch `Ovrt` in der Statuszeile gekennzeichnet, zum Beispiel `(Text Ovrt Fill)`. Genau wie bei Vim verlässt der Modus auch hier nicht die aktuelle Zeile, sondern erweitert sie bei Bedarf.

10.2.3 Navigation und einfache Textbearbeitung

Zeichenweise durch den Text navigieren können Sie mit den Pfeiltasten oder alternativ mit C-f (nach rechts, forward), C-b (nach links, backward), C-p (nach oben, previous line) beziehungsweise C-n (nach unten, next line). C-a bewegt den Cursor zum Zeilenbeginn und C-e zum Zeilenende. M-a und M-e springen dagegen zum vorigen beziehungsweise nächsten Satzbeginn (ein Satzzeichen, gefolgt von einem Leerzeichen). C-v scrollt um einen ganzen Bildschirm nach unten, M-v nach oben. M-f wandert um ein Wort nach rechts, M-b nach links. M-< springt zum Anfang der gesamten Datei, M-> (Alt+⏪+⏩) zu ihrem Ende.

C-d löscht das Zeichen unter dem Cursor, während Backspace das vorherige Zeichen löscht. M-d löscht das folgende Wort, M-⏪ das vorige. C-k löscht den Bereich vom Cursor bis zum Ende der Zeile, während M-k den Text bis zum Satzende entfernt. Sehr nützlich ist auch die Funktion C-t: Sie tauscht das aktuelle mit dem vorherigen Zeichen. Angenommen, Sie haben »Txe« statt »Text« eingegeben: Bewegen Sie den Cursor auf das e und drücken Sie C-t, um die Position der beiden Zeichen zu wechseln.

Ähnlich wie der Vim verfügt auch Emacs über eine Zählerfunktion, um einen Befehl mehrfach auszuführen. Es gibt zwei Möglichkeiten, den Zähler einzugeben: C-u Anzahl Befehl oder M-Anzahl Befehl. Wollen Sie den Cursor beispielsweise um sechs Wörter weiterbewegen, so müssen Sie C-u 6 M-f oder M-6 M-f eingeben. Ebenso bewegt C-u 5 C-b den Cursor um fünf Zeichen nach links, genau wie M-5 C-b. Eine Besonderheit ergibt sich bei den Scrollbefehlen C-v und M-v: Der Zähler bewegt den Bildschirmausschnitt nicht etwa um die angegebene Anzahl von Fensterhöhen vor oder zurück, sondern stattdessen um die entsprechende Anzahl von Zeilen. Beispielsweise blättert C-u 7 C-v um sieben Zeilen nach unten, während M-4 M-v um vier Zeilen nach oben springt. In Emacs funktioniert die Wiederholung sogar mit der Eingabe einzelner Zeichen. Wenn Sie zum Beispiel /* C-u 40 * eingeben, erhalten Sie folgende Zeile:

```
/* *****
```

Dies könnten Sie praktischerweise in einem C-Programm einsetzen, um Ihre Kommentare besser lesbar zu machen.

Markieren, kopieren und einfügen

C-Leertaste setzt eine Markierung. Der markierte Bereich reicht automatisch von der gesetzten Markierung bis zur aktuellen Cursorposition. Um den gesamten Bereich zu löschen, genauer gesagt auszuschneiden, wird die Tastenkombination C-w verwendet. M-w kopiert den Bereich dagegen zum späteren Einfügen. In bei-

den Fällen können Sie ihn mithilfe von `C-y` wieder einfügen. Danach können Sie bei Bedarf mehrfach `M-y` betätigen, um nicht den zuletzt gelöschten beziehungsweise kopierten Text einzufügen, sondern einen früheren.

In der GUI-Variante von Emacs können Sie Text übrigens auch mit der Maus markieren; anschließend lassen sich wiederum die soeben beschriebenen Funktionen ausführen.

Befehle rückgängig machen

Mit `C-x u` oder `C-_` können Sie die letzte Änderung rückgängig machen. Auch hier funktioniert der Zähler: `C-u 7 C-x u` macht zum Beispiel die letzten sieben Änderungen rückgängig. Beachten Sie, dass bei Texteingaben jeweils bis zu 20 Zeichen zu einer Einheit zusammengefasst wird, die nur »im Paket« rückgängig gemacht werden kann.

Einen offiziellen Befehl zum Wiederherstellen gibt es nicht. Wenn Sie eine Änderung zu viel rückgängig gemacht haben, genügt es aber, den Cursor zu bewegen und anschließend erneut den Undo-Befehl zu verwenden. Da Cursorbewegungen keine Änderung durchführen und somit nicht zu den Befehlen gehören, die sich rückgängig machen lassen, zählt das vorherige Undo nun als letzte Änderung.

Suchen und Ersetzen

`C-s` leitet eine inkrementelle Suche ein – während der Eingabe des Suchbegriffs springt der Cursor zur jeweils nächstgelegenen Stelle, die diesem Begriff entspricht. Entsprechend sucht `C-r` rückwärts. `C-s C-s` wiederholt die letzte Suche.  verlässt den Suchbefehl vollständig. `M-C-s` sucht nach einem regulären Ausdruck statt einer einfachen Zeichenfolge. Die Syntax der regulären Ausdrücke entspricht `grep`.

`M-%` ist der Befehl zum Suchen und Ersetzen. Zunächst wird nach der zu ersetzenden Zeichenfolge gefragt, anschließend nach dem Ersetzungstext. Die Eingabe beider Zeichenfolgen muss mit  abgeschlossen werden. An jeder einzelnen Fundstelle werden Sie gefragt, ob Sie die Ersetzung durchführen möchten. `Y` oder Leertaste bedeutet ja, `N` oder  steht für nein, und `!` heißt, dass alle künftigen Fundstellen ohne weitere Rückfrage ersetzt werden sollen.

Wenn Sie Ihre aktuelle Arbeit speichern möchten, wird dazu der Befehl `C-x C-s` verwendet. `C-x s` fragt für jeden geänderten Buffer, ob er gespeichert werden soll. `C-x C-w Dateiname`  speichert die Datei dagegen unter dem angegebenen neuen Namen. Mit `C-x C-c` können Sie Emacs beenden. Vorher werden Sie gefragt, ob Sie die modifizierten Buffer speichern möchten.

10.2.4 Spezielle Funktionen

Was bereits über den Vim gesagt wurde, gilt für Emacs erst recht: Es gibt Unmengen weiterer Befehle, für die in diesem Buch leider kein Platz ist. Mit dem hier gebotenen Ausschnitt aus der Funktionsvielfalt von Emacs können Sie beliebige Textdateien bearbeiten. Mit etwas Übung werden Sie damit irgendwann schneller arbeiten können als mit einem grafisch orientierten Editor, in dem Sie oft zur Maus greifen müssen, um spezielle Befehle zu verwenden.

Abgesehen davon können Sie diesen Editor völlig frei konfigurieren. Zum einen können Sie die Tastenkürzel selbst frei belegen, zum anderen ist der Emacs beliebig erweiterbar: Er ist mit einer eigenen Variante der Programmiersprache Lisp ausgestattet. Diese Sprache könnte Programmierer durch ihren eigenwilligen Ansatz abschrecken, sie ist aber speziell an die Funktionen von Emacs angepasst und erleichtert so das Programmieren von Zusatzfunktionen. Diese Themen führen hier allerdings zu weit.

Shell-Kommandos ausführen

Emacs besitzt zwei sehr praktische Methoden, Shell-Befehle auszuführen. Zum einen führt `C-u M-! Befehl`  den angegebenen Befehl aus und fügt dessen Ausgabe an der aktuellen Cursorposition ein. Hier ein Beispiel mit dem aktuellen Verzeichnisinhalt – die Eingabe lautet `C-u M-! ls -l` .

```

insgesamt 1992
-rw-r--r--  1 sascha users 961142 2007-05-17 20:45 abb_04_01.tif
-rw-r--r--  1 sascha users   187 2007-10-13 20:35 bash.txt
-rw-r--r--  1 sascha users  10328 2007-01-19 23:38 commandref.txt
-rw-r--r--  1 sascha users   3467 2008-01-19 16:38 draft1_erg.txt
-rw-r--r--  1 sascha root    3416 2007-09-15 14:05 draft1.txt
-rw-r--r--  1 sascha users  15374 2007-09-14 22:23 draft1_work.odt
-rw-r--r--  1 sascha users   3372 2008-01-19 01:17 draft1_work.txt
-rw-----  1 sascha users  16522 2007-09-15 00:24 editoren.txt
-rw-r--r--  1 sascha users  16763 2008-01-23 12:41 editoren_utf8.txt
-rw-r--r--  1 sascha users  14274 2008-01-24 23:32 emacs_notes.txt
-rw-r--r--  1 sascha users   1893 2008-01-20 22:07 gimpnotes.txt
-rw-r--r--  1 sascha users  14309 2007-09-21 00:31 mysql.txt
-rw-r--r--  1 sascha users   1073 2008-01-22 17:36 regex.txt
-rw-r--r--  1 sascha users    97 2008-01-23 11:48 TestClass.java
-rw-r--r--  1 sascha users  34370 2008-01-23 20:22 vi.txt

```

Die zweite Shell-Variante ist mitunter noch praktischer: `M-x shell` öffnet einen Buffer mit einer vollständigen Shell. Hier können Sie genauso arbeiten wie in jedem anderen Terminal; zusätzlich stehen Ihnen sämtliche Emacs-Befehle und

-Optionen zur Verfügung. Aus diesem Grund benutzen viele Emacs-Fans den Editor als universelle Arbeitsumgebung.

Mit Verzeichnissen arbeiten

Ein weiteres interessantes Hilfsmittel ist der `dired`-Modus (directory editor) zum Betrachten von Verzeichnisinhalten. Starten Sie Emacs dazu mit einem Verzeichnis- statt einem Dateinamen. In der laufenden Sitzung lautet der Befehl zum Start eines `dired`-Buffers `C-x d`. Im letzteren Fall werden Sie im Minibuffer nach dem gewünschten Verzeichnis gefragt – als Standard wird das aktuelle Arbeitsverzeichnis angezeigt.

Innerhalb des `dired`-Fensters (siehe Abbildung 10.3) können Sie sich mit den üblichen Emacs-Befehlstasten bewegen. Erreichen Sie dabei ein Verzeichnis, so können Sie dessen Inhalt mittels `↵` in einem neuen `dired`-Buffer öffnen. Es ist nicht möglich, in diesen Buffer Text einzugeben; stattdessen besitzen einige Tasten eine spezielle Funktion: Wenn Sie `S` drücken, wird abwechselnd nach Änderungsdatum (neueste zuerst) und alphabetisch nach Dateinamen sortiert. Die Taste `V` öffnet die Datei, auf der sich der Cursor gerade befindet, zum Lesen – mit `Q` kehren Sie zu `dired` zurück. Mit `E` oder `F` lässt sich die aktuelle Datei in einem neuen Buffer zum Bearbeiten öffnen.

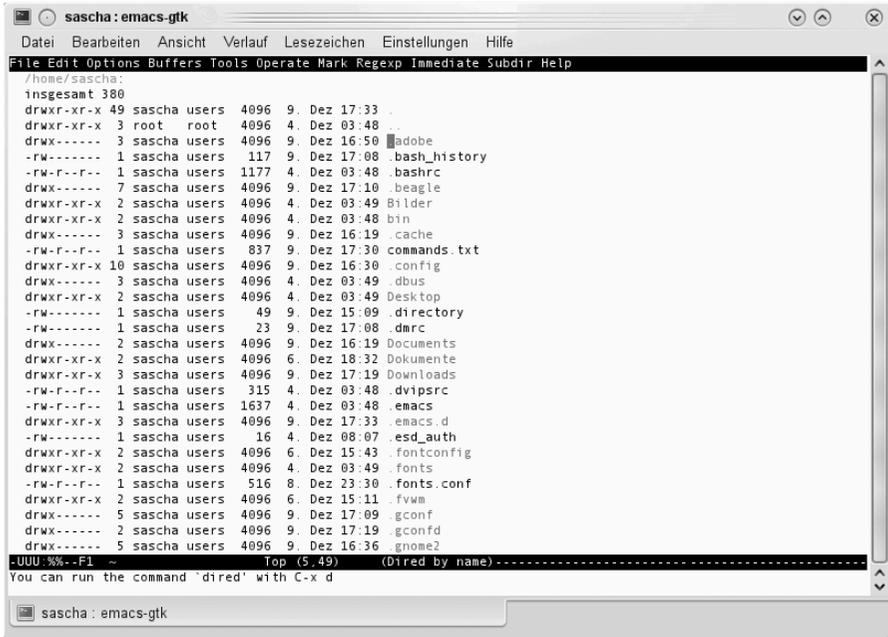


Abbildung 10.3 GNU Emacs (Konsolenbetrieb) im `dired`-Modus

Genau wie in der Bufferliste können Sie auch im `direc`-Modus Dateien zum Löschen markieren, indem Sie `d` drücken; `u` entfernt die Markierung wieder, während `x` sie endgültig ausführt, nachdem Sie `yes` eingegeben haben. Selbstverständlich können Sie Dateien nur löschen, wenn Sie Schreibrechte daran besitzen. Das praktische Kürzel `#` markiert alle Autosave-Dateien (`#Dateiname#`) zum Löschen; entsprechend werden alle Backup-Dateien (deren Namen mit `~` enden) über das Zeichen `~` zum Löschen vorgemerkt.

Emacs enthält eine Funktion zum Lesen von Manpages: Geben Sie `M-x man` `[↵]` ein, gefolgt vom Namen der Manpage, die Sie öffnen möchten, und abschließend erneut `[↵]`. Die Manpage wird im Hintergrund geöffnet und anschließend in einem Fenster gezeigt. Sie können sie mittels `M-C-v` seitenweise durchblättern, ohne das aktuelle Fenster zu verlassen; alternativ können Sie auch mit `C-x o` in dieses Fenster wechseln, um beispielsweise Manual-Text zu kopieren.

Einige Spezialfunktionen

Es gibt, wie bereits erwähnt, unzählige weitere Funktionen, deren genauere Betrachtung in diesem Abschnitt zu weit führen würde. Hier nur eine kurze Liste interessanter Features und wie Sie sie erreichen:

- ▶ `M-x display-time` zeigt die aktuelle Uhrzeit in der Statusleiste an
- ▶ `M-x calendar` öffnet ein Fenster mit dem Emacs-Kalender; die weiter oben beschriebenen Navigationsbefehle beziehen sich hier in aufsteigender Reihenfolge auf Tage, Wochen, Monate und so weiter anstatt auf Zeichen, Zeilen oder Absätze. Beispielsweise blättern Sie mit `C-f` einen Tag weiter und mit `C-p` eine Woche zurück.
- ▶ `C-x m` startet einen Buffer, in dem Sie eine E-Mail schreiben können; mit `C-c C-c` wird sie versandt.
- ▶ `M-x rmail` `[↵]` öffnet den eingebauten Mail-Reader, in dem Sie Ihre ankommenden E-Mails lesen können.
- ▶ `M-x gnus` `[↵]` ist der Befehl für den Start des eingebauten Newsgroup-Readers Gnus.
- ▶ `M-x doctor` `[↵]` startet die eingebaute Variante von *Joseph Weizenbaums* interaktivem Psychotherapeuten *Eliza* – nach all diesen Tastaturkürzeln vielleicht genau das, was Sie brauchen: Geben Sie Text ein und drücken Sie zweimal `[↵]`, daraufhin antwortet Eliza durch Analyse und Umformulierung Ihrer Eingabe.
- ▶ `M-x tetris` ist die Emacs-Umsetzung des beliebten Spieles; verwenden Sie die Pfeiltasten, um die Steine zu bewegen und zu drehen.

10.2.5 Emacs-Befehle im Überblick

In Tabelle 10.3 finden Sie eine Übersicht über die wichtigsten Emacs-Tastenkürzel und -befehle; sie alle wurden bereits in den vorangehenden Tutorial-Unterabschnitten bearbeitet. Wie bereits erwähnt steht C-Taste gemäß der Emacs-Dokumentation für `[Strg]+[Taste]`, während M-Taste in der Regel für `[Alt]+[Taste]` und notfalls für `[Esc]` steht.

Tastenkürzel/Befehl	Bedeutung
Dateien, Buffer und Fenster	
C-x C-f Dateiname	Die angegebene Datei laden (Emacs-Begriff »visit«). Falls sie noch nicht existiert, wird sie neu angelegt
C-x C-v	Den Namen der aktuellen Datei im Minibuffer anzeigen, um ihn zu editieren und die resultierende Datei zu laden/anzulegen
C-x i Dateiname	Die angegebene Datei an der aktuellen Cursorposition in die derzeitige Datei einfügen
C-x b Buffername	Zum Buffer mit dem angegebenen Namen wechseln
C-x k [Buffername]	Den angegebenen bzw. den aktuellen Buffer schließen
C-x C-b	Liste aller derzeit geöffneten Buffer in separatem Fenster (Abschnitt)
e oder f	In der Bufferliste den aktiven Buffer im anderen Fenster öffnen
d	In der Bufferliste einen Buffer zum Löschen markieren
s	In der Bufferliste einen Buffer zum Speichern markieren
u	In der Bufferliste eine Markierung aufheben
x	In der Bufferliste alle markierten Lösch- und Speichervorgänge ausführen
C-x 2	Ein zweites Fenster erzeugen
C-x 4 f Dateiname	Die angegebene Datei in einem neuen Fenster öffnen/erstellen
C-x o	Wechsel zum jeweils anderen Fenster (auch in die Bufferliste)
C-x v	Im jeweils anderen Fenster scrollen
C-x 1	Das nicht aktive Fenster schließen

Tabelle 10.3 Die wichtigsten Emacs-Befehle im Überblick

Tastenkürzel/Befehl	Bedeutung
Zähler	
C-u Anzahl Befehl	Den angegebenen Befehl so oft durchführen, wie durch Anzahl angegeben
M-Anzahl Befehl	Den angegebenen Befehl so oft durchführen, wie durch Anzahl angegeben
Navigation im Text	
C-f	Cursor um ein Zeichen nach rechts bewegen (forward)
C-b	Cursor um ein Zeichen nach links bewegen (backward)
C-p	Cursor um eine Zeile nach oben bewegen (previous line)
C-n	Cursor um eine Zeile nach unten bewegen (next line)
M-b	Cursor um ein Wort nach rechts bewegen
M-f	Cursor um ein Wort nach links bewegen
C-a	Cursor zum Zeilenanfang bewegen
C-e	Cursor zum Zeilenende bewegen
M-a	Cursor zum vorigen Satzbeginn bewegen
M-e	Cursor zum nächsten Satzbeginn bewegen
C-v	Um einen ganzen Bildschirm nach unten scrollen
M-v	Um einen ganzen Bildschirm nach oben scrollen
M-<	Cursor zum Textanfang bewegen
M->	Cursor zum Textende bewegen
Löschen	
C-d	Das Zeichen unter dem Cursor löschen
Backspace	Das Zeichen links vom Cursor löschen
M-d	Das folgende Wort löschen
M-Backspace	Das vorige Wort löschen
C-k	Vom Cursor bis zum Ende der Zeile löschen
M-k	Vom Cursor bis zum Satzende löschen
C-t	Das aktuelle und das vorherige Zeichen vertauschen

Tabelle 10.3 Die wichtigsten Emacs-Befehle im Überblick (Forts.)

Tastenkürzel/Befehl	Bedeutung
Markieren, kopieren und einfügen	
C-Leertaste	Markierung setzen
C-w	Den Text von der Markierung bis zur aktuellen Cursorposition ausschneiden
M-w	Den Text von der Markierung bis zur aktuellen Cursorposition kopieren
C-y	Den zuletzt ausgeschnittenen bzw. kopierten Text an der akt. Cursorposition einfügen
M-y	Durch die zuvor ausgeschnittenen bzw. kopierten Texte »blättern«, um den richtigen auszusuchen
Suchen und ersetzen	
C-s	Inkrementelle Suche vorwärts einleiten
C-r	Inkrementelle Suche rückwärts einleiten
C-s C-s	Erneut nach dem vorherigen Suchbegriff suchen
	Den Suchmodus verlassen
M-C-s	Nach einem regulären Ausdruck statt nach einem einfachen String suchen
M-%	Interaktives Suchen und Ersetzen: Suchbegriff eingeben, Ersetzungsstring eingeben, für jede Stelle mit y oder Leertaste bestätigen, mit n oder Entf ablehnen, oder mit ! für alle Fundstellen bestätigen
Speichern und beenden	
C-x C-s	Den aktuellen Buffer speichern
C-x s	Alle Buffer (mit Nachfrage) speichern
C-x C-w Dateiname	Die aktuelle Datei unter dem angegebenen neuen Namen speichern
C-x C-c	Emacs beenden (mit Nachfrage, ob die geänderten Buffer gespeichert werden sollen)

Tabelle 10.3 Die wichtigsten Emacs-Befehle im Überblick (Forts.)

10.3 Zusammenfassung

Es ist zugegebenermaßen recht viel Übung erforderlich, um mit einem oder gar beiden in diesem Kapitel vorgestellten Texteditoren sicher umgehen zu können. Für »Vieltipper« – sowohl Programmierer als auch Autoren – lohnt sich der Aufwand allerdings. Auch für die vernünftige Arbeit mit den im nächsten Kapitel behandelten Auszeichnungssprachen XML, (X)HTML und LaTeX ist es sinnvoll, sich Vim- oder Emacs-Kenntnisse anzueignen.

An Vim fällt zunächst auf, dass dieselben einfachen Tastendrücke je nach Modus zur Texteingabe wie zur Bearbeitung dienen. Was zunächst als verwirrender Nachteil erscheint, kann auch sehr praktisch sein: Verrenkungen durch Tastenkombinationen sind für die meisten Funktionen nicht nötig. Andererseits ist der Moduswechsel unter Umständen etwas langsamer als der Einsatz von Tastenkürzeln während des Editierens.

Vim ist eine moderne Erweiterung des klassischen vi-Editors und verfügt daher trotz der scheinbar archaischen Arbeitsweise über Funktionen wie Syntaxhervorhebung oder automatisches Einrücken.

Emacs besitzt noch erheblich mehr Funktionen als Vim. Unter anderem können Sie ihn als praktischen Shell-Ersatz, Mail- und Newsreader und sogar zum Spielen verwenden. Im Gegensatz zum Vim verwendet Emacs Tastenkombinationen mit `[Strg]` und `[Alt]`. Die etwas eigenwillige Originaldokumentation (und in Folge dessen auch dieses Kapitel) bezeichnet diese Tastenkürzel als C-"Zeichen" beziehungsweise M-"Zeichen".

Um gut schreiben zu können, muss man etwas Kühleres in den Adern haben als Blut.
– Truman Capote

11 Textbasierte Auszeichnungssprachen

Die UNIX-Welt, zu der Linux gehört, ist die Welt der Klartextformate. Besonders die sogenannten Auszeichnungssprachen (englisch »Markup Languages«) sind allgegenwärtig. Grundsätzlich bietet die Verwendung klartextbasierter Dokumentformate eine Reihe von Vorteilen gegenüber Binärdateien:

- ▶ Sämtliche Konfigurations- und Strukturinformationen sind für Menschen lesbar und können notfalls auch manuell geändert werden.
- ▶ Die Dokumente lassen sich auf jedem beliebigen Computersystem in einem einfachen Texteditor öffnen und eventuell bearbeiten. Der reine Textinhalt erschließt sich auch Anwendern, die die verwendeten Auszeichnungsbefehle nicht verstehen.
- ▶ Der Austausch von Dokumenten mit anderen Anwendungen, neuen Versionen einer Anwendung oder Programmiersprachen ist erheblich einfacher als bei Binärformaten.

In diesem Kapitel lernen Sie die wichtigsten von ihnen kennen; gleichzeitig werden einige mit openSUSE gelieferte Werkzeuge zu ihrer Bearbeitung vorgestellt. Die betreffenden Sprachen sind:

- ▶ XML
- ▶ (X)HTML und CSS
- ▶ LaTeX

11.1 XML

Die *Extensible Markup Language* (erweiterbare Auszeichnungssprache) XML ist mehr als ein einzelnes Dokumentformat für einen spezifischen Verwendungszweck. Es handelt sich vielmehr um eine Metasprache zur Definition beliebig vieler Auszeichnungssprachen unter Berücksichtigung vergleichsweise weniger for-

maler Kriterien (diese müssen allerdings streng befolgt werden). Diese mithilfe von XML definierten Sprachen können zum Beispiel Textdokumente, Vektorgrafiken, multimediale Präsentationen, Datenbanken oder andere Arten von strukturierten Daten beschreiben. Das von OpenOffice.org verwendete Open Document Format (siehe Kapitel 7, »OpenOffice.org«) besteht beispielsweise aus ZIP-Archiven, in denen XML-Dateien in einem speziellen Format enthalten sind.

XML ist eine Empfehlung (Recommendation) des World Wide Web-Consortium (W3C) und damit ein Quasistandard. Es handelt sich um eine schlanke, moderne und an Internet-Bedürfnisse angepasste Weiterentwicklung der klassischen Metasprache SGML (Standard Generalized Markup Language), die Ende der 60er-Jahre des vorigen Jahrhunderts erfunden und in den 80ern entscheidend weiterentwickelt wurde.

Die XML-Spezifikation selbst enthält, wie erwähnt, nur wenige formale Regeln für den Aufbau von Dokumenten. Wenn Sie sich an diese – weiter unten erläuterten – Regeln halten, erzeugen Sie ein *wohlgeformtes* XML-Dokument. Darüber hinaus besteht aber auch die Möglichkeit, Standards für XML-Dokumentformate zu definieren und Dokumente von diesen Standards abhängig zu machen. Dokumente, die auf solchen Formatdefinitionen basieren, sind nicht nur wohlgeformt, sondern auch *gültig*. Die klassische, bereits für SGML verwendete Sprache für solche Dokumentklassen ist die *Document Type Definition* oder *DTD*. Moderne Alternativen, die zum einen leistungsfähiger sind und zum anderen selbst in reinem XML verfasst werden, sind *XML Schema* und *RELAX NG* (siehe <http://www.relaxng.org>).

XML ist eine *universelle* Sprache; es spielt keine Rolle, ob Sie Ihre Musik-CD-Sammlung, Ihre Doktorarbeit oder Ihre Geschäftsdaten in einem XML-Dokument speichern. Wichtig ist allerdings, dass XML immer nur die Struktur und den Aufbau der Daten beschreibt. Das Layout der Druck- oder Webversion von Textinhalten muss durch eine externe Stil- oder Formatierungssprache beschrieben werden, beispielsweise durch das weiter unten angesprochene XSLT.

Übrigens werden in der Praxis gar nicht so oft neue Dokumentformate für eigene Anwendungen entwickelt. Der überwiegende Anteil der Anwendungen von XML beruht auf dem Einsatz vorhandener XML-basierter Sprachen. Beispiele sind hier etwa die Webseiten-Auszeichnungssprache *XHTML*, das Vektorgrafikformat *SVG* oder die beliebte Handbuch- und Dokumentationssprache *DocBook*. XHTML wird im nächsten Abschnitt ausführlich erläutert.

Sie können XML-Dateien mit Ihrem bevorzugten Texteditor eingeben; Vim besitzt ein Syntax-Highlighting-Schema dafür, Emacs sogar einen eigenen Modus. Wichtig ist, dass der gewählte Editor den Zeichensatz unterstützt, den

das XML-Dokument verwendet. Im HTML-Abschnitt lernen Sie Quanta Plus als Beispiel für einen komfortablen grafischen Editor kennen, der neben HTML auch XML beherrscht.

XML-Dokumente werden als Textdateien gespeichert, entweder mit der Endung `.xml`, wenn es sich um allgemeine XML-Dokumente handelt, oder mit einer speziellen Dateierweiterung, falls ein besonderes XML-Format verwendet wird. Zum Beispiel werden XHTML-Dokumente üblicherweise mit der Endung `.htm` oder `.html` gespeichert; SVG-Dateien besitzen dagegen die Erweiterung `.svg`.

Der für Webserver und E-Mail-Anwendungen wichtige MIME-Type allgemeiner XML-Dokumente ist `text/xml`, während spezielle Formate entweder einen ganz eigenen Typ wie `text/html` oder eine Kombination wie `image/svg+xml` bilden. Beachten Sie, dass ein SVG-Dokument zwar formal noch immer ein Textdokument ist, aber den Verwendungszweck eines Bildes erfüllt – deshalb der Haupttyp `image`.

11.1.1 Der Aufbau von XML-Dokumenten

Jedes XML-Dokument besteht aus einer Hierarchie ineinander verschachtelter Steueranweisungen, die als *Elemente* oder *Tags* bezeichnet werden, und kann zusätzlich einfachen Text enthalten. Die XML-Tags werden in spitze Klammern gesetzt, also zwischen ein `<`-Zeichen und ein `>`-Zeichen. Sie können ein oder mehrere Attribute in der Form `attribut="wert"` enthalten. Jedes Tag wird unter Angabe seiner Bezeichnung geöffnet (zum Beispiel `<test>`) und weiter unten im Dokument durch eine Wiederholung mit vorangestelltem Slash (`/`) wieder geschlossen (etwa `</test>`). Wahrscheinlich haben Sie eine solche Syntax schon einmal gesehen, wenn Sie sich den Quellcode von HTML-Dokumenten angesehen haben, die im Prinzip genauso aufgebaut sind.

Die grundlegenden Bestandteile von XML-Dokumenten

Hier sehen Sie ein einfaches XML-Dokument, in dem zwei Musik-CDs aus einer umfangreichen Sammlung dargestellt werden:

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<cds>
  <cd num="1">
    <artist type="band">Metallica</artist>
    <title>Master of Puppets</title>
    <genre>Metal</genre>
    <year>1986</year>
    <country>USA</country>
    <tracks>
```

```

    <track num="1">
      <title>Battery</title>
      <duration>312</duration>
    </track>
    <track num="2">
      <title>Master of Puppets</title>
      <duration>514</duration>
    </track>
    <!-- 5 weitere Tracks -->
    <track num="8">
      <title>Damage Inc.</title>
      <duration>331</duration>
    </track>
  </tracks>
</cd>
<cd num="2">
  <artist type="band">Iron Maiden</artist>
  <title>Piece of Mind</title>
  <genre>Metal</genre>
  <year>1983</year>
  <country>UK</country>
  <tracks>
    <track num="1">
      <title>Where Eagles Dare</title>
      <duration>369</duration>
    </track>
    <!-- 7 weitere Tracks -->
    <track num="9">
      <title>To Tame A Land</title>
      <duration>446</duration>
    </track>
  </tracks>
</cd>
<!-- und viele weitere CDs -->
</cdfs>

```

Jedes XML-Dokument beginnt mit einer `xml`-Steueranweisung, die die verwendete XML-Version (bisher immer 1.0) und den Zeichensatz des Dokuments angibt:

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
```

Die hier verwendete Zeichencodierung `utf-8` ist eine spezielle Schreibweise des Unicode-Zeichensatzes: Die ASCII-Zeichen 0 bis 127 benötigen nur 1 Byte, alle anderen Unicode-Zeichen dagegen 2 bis 4. In allen gängigen Editoren, Program-

men und in den Shells unter openSUSE ist UTF-8 Standard, sodass dies die beste Wahl ist. Selbstverständlich können Sie auch andere Zeichensätze angeben, zum Beispiel `iso-8859-1` für den USA/Westeuropa-Standardzeichensatz oder `gb2312` für Chinesisch (VR China). Dazu muss der verwendete Zeichensatz allerdings von Ihrem Text- oder XML-Editor unterstützt werden.

Das Attribut `standalone` gibt an, ob sich das Dokument auf ein externes Formatdokument wie eine DTD oder ein XML Schema bezieht oder nicht – `yes` besagt, dass das Dokument selbstständig ist und nicht von einer solchen Standardisierung abhängt.

Steueranweisungen (auch *PI*, Processing Instructions) können in XML-Dokumenten an beliebiger Stelle vorkommen. Es handelt sich um Anweisungen für interpretierende Geräte oder Programme, die mit dem Dokument selbst nichts zu tun haben.

Es ist wichtig, dass *genau ein* Tag das gesamte Dokument umschließt. Das Wurzelement des vorliegenden Dokuments sieht folgendermaßen aus:

```
<cds>
  <!-- umschlossener Inhalt -->
</cds>
```

Innerhalb des Wurzelements sind weitere Elemente mit ihren Unterelementen und Textinhalten verschachtelt. Das Wichtigste ist dabei, dass Sie auf die korrekte Reihenfolge bei der Verschachtelung achten. Eine Schreibweise wie die folgende ist nicht gestattet:

```
<track num="1"><title>Where Eagles Dare</track></title>
```

Richtig muss es folgendermaßen lauten:

```
<track num="1"><title>Where Eagles Dare</title></track>
```

Die hierarchische Gliederung von XML-Dokumenten ergibt eine Art Baumdiagramm. Das Wurzelement bildet logischerweise die Wurzel, die verschachtelten Elemente sind die Äste und Zweige und die Textinhalte die Blätter.

Die *Namen* von Tags und den weiter unten besprochenen Attributen dürfen aus Buchstaben und Ziffern sowie aus den folgenden Sonderzeichen bestehen: `_` (Unterstrich), `-` (Bindestrich) und `.` (Punkt). Das erste Zeichen darf keine Ziffer sein. Es wird zwischen Groß- und Kleinschreibung unterschieden.

Wie bereits indirekt erwähnt, muss jedes XML-Tag, das geöffnet wurde, auch wieder geschlossen werden. Erst das öffnende und das schließende Tag zusammen ergeben das eigentliche Element. Geschlossen wird ein Tag durch die Wie-

derholung seines Namens mit vorangestelltem Slash (/). In der Regel steht zwischen den beiden Tags Inhalt, der aus tiefer verschachtelten Tags oder aus einfachem Text bestehen kann. Mitunter kommt es jedoch vor, dass ein Element *leer* ist, also keinen weiteren Inhalt enthält.

Zum Beispiel könnte die Liste für jede CD ein zusätzliches Element enthalten, das auf den Dateinamen und den MIME-Type einer Abbildung des CD-Covers verweist. Für diese Angaben würden sich Attribute (siehe nächster Unterabschnitt) besonders gut eignen:

```
...
<title>Master of Puppets</title>
<cover file="metallica_master.png" type="image/png">
</cover>
...
```

Da alle erforderlichen Informationen über das Cover-Bild bereits in den Attributen stehen, benötigt das Element `cover` keinen verschachtelten Inhalt. Falls Sie bereits Kenntnisse in HTML haben, könnten Ihnen die dort verwendeten »Einfach-Tags« wie der Zeilenumbruch `
` vertraut sein. In XML ist eine solche Ausnahme nicht zulässig; diese laxe Schreibweise stammt noch aus SGML. Als »Entschädigung« bietet XML allerdings eine kurz gefasste Schreibweise für leere Elemente an. Das `cover`-Element könnten Sie entsprechend dieser Syntax auch folgendermaßen schreiben:

```
<cover file="metallica_master.png" type="image/png" />
```

Der Slash am Ende des Tags ersetzt also das vollständige, schließende Tag. Das Leerzeichen vor dem End-Slash ist nach der eigentlichen XML-Syntax nicht erforderlich. Allerdings ist es nützlich, um kompatiblen Code für Browser abzuliefern, die nur HTML, aber nicht das neue XHTML verstehen. Der oben erwähnte Zeilenumbruch wird in der voll kompatiblen Fassung also so geschrieben:

```
<br />
```

Alte Browser ignorieren den Slash einfach, während XML-Parser kein Problem mit dem Leerzeichen haben. Falls Sie dagegen einfach `
` schreiben, versteht ein älterer Browser Sie möglicherweise nicht, weil er dies für ein unbekanntes Tag hält.

Das Element `cd` enthält jeweils ein Attribut namens `num`. Attribute stehen nur beim öffnenden Tag und werden beim schließenden niemals wiederholt. Die Form ist stets `attributname="attributwert"` oder `attributname='attributwert'`. Die doppelten oder einfachen Anführungszeichen sind zwingend erforderlich. Zwischen dem Attributnamen, dem Gleichheitszeichen und dem öffnen-

den Anführungszeichen ist kein Abstand erlaubt, während mehrere Attribute durch Whitespace voneinander getrennt werden, genau wie Tag-Name und Attribut.

Allgemein werden Attribute häufig verwendet, um Ordnungskriterien oder Meta-Informationen für Elemente anzugeben. Allerdings ergäbe sich kein Unterschied im Informationsgehalt des Dokuments, wenn Sie statt

```
<cd num="1">
  ...
</cd>
```

die folgende Schreibweise wählen würden:

```
<cd>
  <num>1</num>
  ...
</cd>
```

Beachten Sie jedoch, dass Attribute immer nur für Informationen geeignet sind, die nur einmal vorkommen. Es ist nämlich nicht zulässig, zwei Attribute gleichen Namens in ein und demselben Tag zu verwenden. Außerdem ist ein Attribut naturgemäß nicht für mehrgliedrige Angaben geeignet, die eigentlich verschachtelt werden müssen.

Das folgende Beispiel zeigt einen extrem schlechten Stil bei der Verwendung von Attributen, da es mehrere gleichartige Attribute verwendet:

```
<cd num="1"
  artist1="Hetfield, James (vocals, guitar)"
  artist2="Hammet, Kirk (guitar)"
  artist3="Burton, Cliff (bass)"
  artist4="Ulrich, Lars (drums)">
  ...
</cd>
```

Wie sähe dies erst aus, wenn dreißig Musiker an der CD mitgewirkt hätten (wenn ein klassisches Orchester statt einer Rockband beteiligt wäre, könnte dies leicht vorkommen)?

Die folgende Schreibweise ist noch schlimmer, da sie sowohl mehrere gleichartige Attribute verwendet als auch Informationen mit Verschachtelungsbedarf in Attribute packt:

```
<cd num="1"
  artist1-lastname="Hetfield" artist1-firstname="James"
  artist1-instruments="vocals, guitar">
```

```

artist2-lastname="Hammet" artist2-firstname="Kirk"
artist2-instruments="guitar"
artist3-lastname="Burton" artist3-firstname="Cliff"
artist3-instruments="bass"
artist4-lastname="Ulrich" artist4-firstname="Lars"
artist4-instruments="drums">
...
</cd>

```

Diese beiden schlechten Beispiele sollten Sie so schnell wie möglich wieder vergessen; sie sind zwar formal zulässig, stilistisch aber vollkommen indiskutabel.

Einige Zeichen sind in XML-Dokumenten nicht zulässig, sondern müssen durch spezielle Escape-Sequenzen ersetzt werden, die als *Entity-Referenzen* bezeichnet werden. Eine Entity-Referenz beginnt mit einem &-Zeichen, darauf folgt ein spezieller Code und am Ende ein Semikolon. Tabelle 11.1 zeigt die fünf Zeichen, die nicht gestattet sind, und die passenden Entity-Referenzen.

Zeichen	Entity-Referenz	Bedeutung
<	<	less than
>	>	greater than
&	&	Ampersand (»and per se and«)
'	'	Apostroph
"	"	quotation mark

Tabelle 11.1 Die fünf Standard-Entity-Referenzen in XML

Alle diese Zeichen haben im XML-Code eine spezielle Bedeutung: < und > umschließen die Tags. Die Werte von Attributen stehen in Anführungszeichen oder wahlweise in einfachen Anführungszeichen (Apostrophen). Das &-Zeichen schließlich leitet eben die Entity-Referenzen ein.

Neben diesen vorgefertigten Entity-Referenzen können Sie in XML auch beliebige Unicode-Zeichen numerisch angeben. Die Syntax ist entweder `&#Dezimalcode;` oder `&#xHexadezimalcode;`. Beispielsweise können Sie ein »Registered Trademark« (®) durch die Zeichenfolge `®` (dezimal) oder `®` (hexadezimal) erzeugen.

Schließlich können Sie eigene Entities definieren und in Ihren Dokumenten Referenzen darauf verwenden. Dies ermöglicht Ihnen den schnellen Zugriff auf häufig benötigte Sonderzeichen oder sogar XML-Codeblöcke. Die Definition von

Entities wird innerhalb von DTDs durchgeführt, die weiter unten behandelt werden.

Der normale Text in XML-Dokumenten besteht aus sogenannten *PCDATA*-Abschnitten; »PCDATA« steht für »parsed character data«. Dies bedeutet, dass einige Sonderzeichen innerhalb des Textes gemäß ihrer speziellen XML-Bedeutung behandelt werden – es sei denn, Sie verwenden die im vorigen Unterabschnitt vorgestellten Entity-Referenzen.

In einigen Fällen können Entity-Referenzen sehr störend sein. Stellen Sie sich zum Beispiel eine XML-Version des vorliegenden Abschnitts vor – andauernd werden XML-Codebeispiele gezeigt. Diese müssen in normalen Textabschnitten die Entity-Referenzen verwenden.

Der folgende Code ist die XML-Entsprechung eines Ausschnitts aus dem vorigen Unterabschnitt:

```
<para>
  Das <code>cover</code>-Element könnten Sie nach dieser
  Syntax auch folgendermaßen schreiben:
</para>
<codeblock>
  &lt;cover file=&quot;metallica_master.png&quot;
    type=&quot;image/png&quot; /&gt;
</codeblock>
<para>
  Der Slash am Ende des Tags ersetzt also das komplette
  schließende Tag.
</para>
```

Das XML-Codebeispiel zwischen den Tags `<codeblock>` und `</codeblock>` ist durch die Häufung von Entity-Referenzen absolut unleserlich. Um solche Probleme zu vermeiden, haben sich die XML-Entwickler ein spezielles Format für solche Textblöcke ausgedacht, die sogenannten *CDATA*-Abschnitte. Diese Abkürzung bedeutet »character data«. Innerhalb dieser speziellen Bereiche sind alle Sonderzeichen erlaubt und werden nicht als XML interpretiert.

Ein *CDATA*-Abschnitt wird durch die Sequenz `<![CDATA[` eingeleitet und durch `]]>` abgeschlossen. Mithilfe eines *CDATA*-Abschnitts wird das obige Beispiel sofort viel lesbarer:

```
<para>
  Das <code>cover</code>-Element könnten Sie nach dieser
  Syntax auch folgendermaßen schreiben:
</para>
```

```

<codeblock>
<![CDATA[
  <cover file="metallica_master.png"
    type="image/png" />
]]>
</codeblock>
<para>
  Der Slash am Ende des Tags ersetzt also das komplette
  schließende Tag.
</para>

```

Es versteht sich von selbst, dass die Zeichenfolge `]]>` innerhalb eines CDATA-Blocks unzulässig ist – schließlich beendet sie eben diesen Block. Genau deshalb wurde eine so unwahrscheinliche Zeichensequenz gewählt.

Wohlgeformtheit

Jedes XML-Dokument muss eine Reihe formaler Regeln erfüllen, um *wohlgeformt* zu sein. XML-Dateien, die diesen Regeln nicht genügen, werden von XML-Parsern in Anwendungen und Programmiersprachen nicht verarbeitet, sondern erzeugen Fehlermeldungen. In diesem Unterabschnitt werden die Regeln für die Wohlgeformtheit noch einmal explizit erläutert. Zwar wurden sie bereits am Rande erwähnt, sind aber wichtig genug, um genauer erklärt zu werden.

Hier sehen Sie zunächst eine kurze Liste aller Regeln für die Wohlgeformtheit:

- ▶ Ein XML-Dokument benötigt genau ein *Wurzelement*: Ein bestimmtes Element muss alle anderen Elemente und Textinhalte umschließen.
- ▶ Alle Elemente müssen korrekt ineinander verschachtelt werden; das zuletzt geöffnete Element wird als Erstes wieder geschlossen.
- ▶ Jedes Element besteht aus einem öffnenden und einem schließenden Tag; »Einfach-Tags« wie in klassischem HTML gibt es nicht. Für leere Tags existiert die spezielle Kurzfassung mit dem End-Slash.
- ▶ Attribute haben die Form `name="wert"`. Der Wert muss stets in Anführungszeichen stehen.
- ▶ Die Namen von Elementen und Attributen dürfen nur Buchstaben, Ziffern, Unterstriche, Bindestriche und Punkte enthalten. Es wird zwischen Groß- und Kleinschreibung unterschieden. Das erste Zeichen darf keine Ziffer sein.
- ▶ Bestimmte Zeichen sind in XML-Dokumenten nicht zulässig: `<`, `>`, `&`, `"` und `'` müssen durch die *Entity-Referenzen* `<`, `>`, `&`, `"`; beziehungsweise `'` ersetzt werden. Die Definition weiterer Entity-Referenzen ist zulässig.

- ▶ CDATA-Blöcke ermöglichen die beliebige Verwendung der Sonderzeichen, die normalerweise durch Entity-Referenzen ersetzt werden müssen. Ein CDATA-Abschnitt steht zwischen `<![CDATA[` und `]]>`.

Die meisten dieser Regeln wurden weiter oben bereits ausführlich genug erläutert. Im Folgenden wird allerdings noch einmal die Bedeutung des Wurzelements und der korrekten Verschachtelung von Tags hervorgehoben.

Die Forderung nach einem Wurzelement bedeutet, dass Code wie der folgende kein vollständiges XML-Dokument bildet:

```
<cd>
  <artist type="band">Extreme</artist>
  <title>Saudades de Rock</title>
</cd>
<cd>
  <artist type="band">Guns N&apos; Roses</artist>
  <title>User Your Illusion I</title>
</cd>
```

Dies ist bestenfalls ein Dokument-Fragment. Manche XML-fähigen Anwendungen sind in der Lage, mit solchen Fragmenten umzugehen. Darauf dürfen Sie sich allerdings niemals verlassen. Korrekt wäre dagegen folgende Fassung:

```
<cds>
  <cd>
    <artist type="band">Extreme</artist>
    <title>Saudades de Rock</title>
  </cd>
  <cd>
    <artist type="band">Guns N&apos; Roses</artist>
    <title>User Your Illusion I</title>
  </cd>
</cds>
```

Das Element `cds` ist das Wurzelement des gesamten Dokuments; die Tags `<cds>` und `</cds>` umschließen alle anderen Inhalte.

Bei vielen vordefinierten XML-basierten Dokumentformaten ist das Wurzelement ein Hinweis auf das Format selbst. Beispielsweise lautet das Wurzelement eines XHTML-Dokuments `html`. Der gesamte Inhalt von HTML-Dokumenten wird also von den Tags `<html>` und `</html>` umschlossen.

Wie bereits erwähnt, muss die korrekte Verschachtelungsreihenfolge von XML-Elementen beachtet werden. Tags werden von außen nach innen geöffnet und in

umgekehrter Reihenfolge wieder geschlossen. Das zuletzt geöffnete Tag wird demnach zuerst geschlossen.

Diese Regel ist im Grunde leicht zu merken und einzuhalten. Vielleicht verwirrt es Sie aber in dem Fall, dass zwei Tags unmittelbar hintereinander geöffnet werden, die Sie später auch wieder gleichzeitig schließen möchten.

Stellen Sie sich beispielsweise vor, Sie hätten für die Formatierung eines Textdokuments zwei Elemente namens `fett` und `kursiv` definiert und wollten nun einige Wörter `fett` *und* `kursiv` darstellen. In diesem Fall könnte es leicht passieren, dass Sie denken: Diese Wörter sollen `fett` und `kursiv` sein. Wenig später denken Sie sich: Dieses Wort soll schon nicht mehr `fett` und nicht mehr `kursiv` sein. Dieser Sprachgebrauch würde die folgende Formulierung in XML nahelegen:

```
Dieser Text ist <fett><kursiv>fett und kursiv</fett></kursiv> und
dieser nicht mehr.
```

Allerdings ist dieses Konstrukt absolut verboten. Die richtige Syntax lautet natürlich folgendermaßen:

```
Dieser Text ist <fett><kursiv>fett und kursiv</kursiv></fett> und
dieser nicht mehr.
```

Gute XML- oder HTML-Editoren weisen im Übrigen schon während der Eingabe darauf hin, dass Sie eine falsche Verschachtelung verwendet haben.

11.1.2 DTDs und XML Schema

Bereits in der Einleitung wurde erwähnt, dass XML-Dokumente neben der Wohlgeformtheit zusätzlich von einem Standard abhängen können. Ein solcher Standard enthält Regeln, die bestimmen, welche Elemente und Attribute erforderlich oder zulässig sind und in welcher Reihenfolge sie stehen müssen beziehungsweise dürfen. Die traditionelle Methode, um einem XML-Dokument derartige Beschränkungen aufzuerlegen, ist die Document Type Definition (DTD). Dieses Format wurde bereits für SGML entworfen und mit einigen notwendigen Änderungen und Ergänzungen für XML übernommen. Eine neuere, rein XML-basierte Alternative, die obendrein mehr Möglichkeiten bietet, ist XML Schema. Beide Sprachen werden in diesem Abschnitt behandelt.

Document Type Definitions (DTDs)

Eine Document Type Definition, die einen bestimmten XML-Datentyp definiert, steht in der Regel in einer externen Datei mit der Endung `.dtd`. Um ein XML-Dokument an die Regeln dieser DTD zu binden, müssen Sie noch vor dem Wur-

zelement eine `<!DOCTYPE>`-Deklaration hineinschreiben. Hier wird die URL oder allgemeiner die ID der verwendeten DTD angegeben. Es kann sich dabei um eine SYSTEM-ID handeln, die stets eine URL benötigt, oder um eine PUBLIC-ID, die sich auf eine öffentlich standardisierte DTD bezieht.

Angenommen, es existiert eine DTD für das XML-Dokument aus dem vorigen Abschnitt, die sich im gleichen Verzeichnis befindet wie das Dokument. Die passende `<!DOCTYPE>`-Angabe verwendet in diesem Fall eine SYSTEM-ID und lautet folgendermaßen:

```
<!DOCTYPE cds SYSTEM "cds.dtd">
```

Genauso gut kann diese DTD in einem anderen Verzeichnis im Dateisystem oder sogar auf einem anderen Server im Internet liegen. Die DTD-Datei *xml-buecher.dtd* liegt unter anderem unter <http://buecher.lingoworld.de/opensuse/listings/13/xml-buecher.dtd>. Wenn Sie sich auf diese URL beziehen möchten (immer mit der Gefahr, dass sie geändert oder entfernt werden könnte), funktioniert dies folgendermaßen:

```
<!DOCTYPE cds SYSTEM "http://buecher.lingoworld.de/opensuse/
listings/11/cds.dtd">
```

Eine PUBLIC-ID verwendet dagegen ein standardisiertes Format, um die DTD unabhängig von ihrer konkreten URL zu kennzeichnen. Eine Anwendung, die ein Dokument auf der Grundlage dieser DTD validiert (ihre Gültigkeit überprüft), muss allerdings eine eingebaute Version der DTD enthalten oder wissen, wie sie diese online finden kann. Deshalb werden in der Praxis nur wenige öffentliche DTDs häufig verwendet. Das folgende Beispiel zeigt einen Verweis auf die XHTML-DTD des W3C:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN">
```

Um sicherzustellen, dass die DTD auf jeden Fall gefunden wird, wenn der Validator sie benötigt, wird in der Praxis meist zusätzlich eine URL angegeben:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

DTDs beschreiben, welche Elemente und Attribute in welcher Reihenfolge in einem Dokument zulässig sind und welche Daten sie jeweils enthalten dürfen.

Ein Element wird mithilfe einer `<!ELEMENT>`-Deklaration angegeben. Formal sieht diese Deklaration folgendermaßen aus:

```
<!ELEMENT elementname (elementinhalt)>
```

Der Elementinhalt kann aus einer Liste verschachtelter Elemente bestehen oder auf einfachen Textinhalt hinweisen. Enthält ein Element verschachtelte Tags, dann werden für diese wiederum `<!ELEMENT>`-Definitionen angegeben.

Die zulässigen Attribute für ein Element werden dagegen in eine `<!ATTLIST>`-Angabe geschrieben. Die formale Schreibweise ist folgende:

```
<!ATTLIST elementname attr1 TYP #REQUIRED
                    attr2 TYP #IMPLIED
                    ...>
```

Der häufigste TYP für Attribute ist `CDATA`, also die Angabe beliebiger Zeichen. `#REQUIRED` oder `#IMPLIED` geben an, ob das Attribut erforderlich ist oder nicht. Ein Attribut mit der Angabe `#REQUIRED` muss angegeben werden, `#IMPLIED` definiert dagegen ein optionales Attribut. Eine dritte zulässige Angabe ist `#FIXED`, was für ein vorgegebenes Attribut steht: Wird es nicht angegeben, ist es dennoch automatisch mit seinem Standardwert vertreten; wenn Sie es explizit angeben, muss es dagegen den vorgegebenen Wert besitzen.

Übrigens können Sie auch für jedes Attribut eine eigene `<!ATTLIST>`-Definition schreiben. In keinem der beiden Fälle ist die Reihenfolge der Attribute innerhalb des Elements verbindlich.

Hier sehen Sie eine vollständige DTD für das weiter oben vorgestellte Dokument *cds.dtd*. Die zusätzlich erläuterte Erweiterung um ein `cover`-Element für eine Abbildung des CD-Covers ist bereits enthalten:

```
<!ELEMENT cds (cd+)>
  <!ELEMENT cd (artist, title, cover, genre, year,
    country, tracks)>
  <!ATTLIST cd num CDATA #REQUIRED>
  <!ELEMENT artist (#PCDATA)>
    <!ATTLIST artist type CDATA #REQUIRED>
  <!ELEMENT title (#PCDATA)>
  <!ELEMENT cover EMPTY>
  <!ATTLIST cover file CDATA #REQUIRED
    type CDATA #REQUIRED>
  <!ELEMENT genre (#PCDATA)>
  <!ELEMENT year (#PCDATA)>
  <!ELEMENT country (#PCDATA)>
  <!ELEMENT tracks (track+)>
    <!ELEMENT track (title, duration)>
    <!ATTLIST track num CDATA #REQUIRED>
    <!ELEMENT title (#PCDATA)>
    <!ELEMENT duration (#PCDATA)>
```

Übrigens ist die Reihenfolge, in der Sie die Deklarationen in der DTD vornehmen, vollkommen gleichgültig. Allerdings können Sie leicht durcheinanderkommen, wenn Sie sich nicht dauerhaft an eine selbstgewählte Reihenfolge halten. Eine empfehlenswerte Reihenfolge, die auch im vorliegenden Beispiel verwendet wird und in vielen gut geschriebenen DTDs anzutreffen ist, funktioniert nach den folgenden Regeln:

- ▶ Als Erstes wird das Wurzelement deklariert.
- ▶ Falls das Wurzelement Attribute besitzt, folgt als Nächstes seine `<!ATTLIST>`-Deklaration.
- ▶ Anschließend werden nacheinander alle unterhalb des Wurzelements zulässigen Elemente in der angegebenen Reihenfolge deklariert.
- ▶ Nach jedem Element folgt – falls vorhanden – seine `<!ATTLIST>`-Angabe; anschließend werden wiederum alle in diesem Element zulässigen Elemente durch ihre `<!ELEMENT>`-Angaben deklariert.

Auf diese Weise enthält die DTD ein genaues Abbild der Verschachtelung der durch sie beschriebenen XML-Dokumente. Dies können Sie zusätzlich durch Einrückungen verdeutlichen.

Die meisten Elemente in der gezeigten DTD enthalten im Übrigen keine weiter verschachtelten Elemente mehr, sondern das Inhaltsmodell (`#PCDATA`), also beliebigen Text. Das Element `cover` besitzt dagegen die spezielle Angabe `EMPTY` – es handelt sich um ein leeres Tag, das keine weiteren Inhalte aufweisen darf. Wie bereits weiter oben erläutert, muss es später im Dokument nicht umständlich als `<cover ...></cover>` geschrieben, sondern kann durch `<cover .../>` abgekürzt werden.

Im vorigen Unterabschnitt haben Sie die `<!ELEMENT>`-Deklaration schon grundsätzlich kennengelernt. Die häufigsten Formen dieser Deklaration sind die Aufzählung der zulässigen Elemente oder `#PCDATA` für einfachen Text.

Die umfangreichste Liste von Elementen in der Beispiel-DTD enthält die Angabe für das Element `cd`:

```
<!ELEMENT cd (artist, title, cover, genre, year,
  country, tracks)>
```

In dieser Schreibweise bedeutet die Definition, dass innerhalb des Elements `cd` alle angegebenen Elemente in der vorgegebenen Reihenfolge vorkommen müssen.

Statt einer festgelegten Reihenfolge können Sie auch eine durch | getrennte Liste von Alternativen angeben. Das folgende Beispiel zeigt ein Element namens `anschrift`, das entweder das Element `postfach` oder `strasse` enthalten kann:

```
<!ELEMENT anschrift (postfach | strasse)>
```

Die beiden folgenden Alternativen sind gültige Verwendungen des Elements `anschrift`:

```
<anschrift>
  <postfach>1234567</postfach>
</anschrift>

<anschrift>
  <strasse>Alte Straße 12</strasse>
</anschrift>
```

Solche Angaben lassen sich durch Klammern auch verschachteln. Diese verbesserte Version von `anschrift` verlangt entweder ein Postfach oder eine Straße und eine Hausnummer:

```
<!ELEMENT anschrift (postfach | (strasse, hausnr))>
```

Sie können sogar Alternativen und Pflichtangaben beliebig mischen, wie die folgende vollständige Fassung einer Anschrift zeigt:

```
<!ELEMENT anschrift (name, (postfach | (strasse, hausnr)), plz,
ort)>
```

Eine Anschrift besteht also aus einem Namen, gefolgt von einem Postfach oder einer Straße und Hausnummer, anschließend kommt die Postleitzahl und zum Schluss der Ort. Die beiden folgenden Anschriften erfüllen dieses Format:

```
<anschrift>
  <name>Galileo Press</name>
  <strasse>Rheinwerkallee</strasse>
  <hausnr>4</hausnr>
  <plz>53227</plz>
  <ort>Bonn</ort>
</anschrift>

<anschrift>
  <name>MICROGRAFX (Deutschland) GmbH</name>
  <postfach>14 18</postfach>
  <plz>85704</plz>
  <ort>Unterschleißheim</ort>
</anschrift>
```

Alternativen können auch helfen, wenn Ihnen die Reihenfolge bestimmter Elemente egal ist. Die folgende Variante der `cd`-Deklaration stellt es frei, die Reihenfolge von `year` und `country` beliebig zu vertauschen:

```
<!ELEMENT cd (artist, title, cover, genre,
  ((year, country) | (country, year)), tracks)>
```

Zu viele verschiedene Reihenfolgen können Sie also nicht zulassen, weil die Liste sonst erheblich zu lang würde.

Sie können einer Liste von Alternativen auch `#PCDATA` voranstellen, um statt der möglichen Tags auch beliebigen Text zuzulassen. Beispielsweise könnte eine Anschrift neben einem Postfach oder einer Straße/Hausnummer-Folge auch einen anderen Zusatz enthalten, der als einfacher Text ausgedrückt wird:

```
<!ELEMENT anschrift (name, (#PCDATA | postfach |
  (strasse, hausnr)), plz, ort)>
```

Auf diese Weise könnten Sie auch anders formulierte Anschriften, wie man sie manchmal in ländlichen Gegenden findet (»Gutshof Erlenbach« oder Ähnliches), ohne Probleme angeben.

Jedes Element und jede Gruppe von Inhalten kann in einem XML-Dokument auch mehrmals vorkommen. Zu diesem Zweck bietet die DTD-Sprache verschiedene Modifikatoren an, die Sie hinter ein Element oder einen geklammerten Ausdruck setzen können, um seine Häufigkeit anzugeben:

- ▶ ? – Der Inhalt darf einmal oder keinmal vorkommen.
- ▶ + – Der Inhalt muss mindestens einmal vorkommen.
- ▶ * – Der Inhalt darf beliebig oft vorkommen.

Beispielsweise können Personen nicht nur einen Vornamen haben, sondern auch mehrere. Um diese Vornamen voneinander zu trennen, könnten Sie ein Element namens `person` wie folgt schreiben:

```
<!ELEMENT person (name, vorname+)>
```

Dadurch könnten Sie einen Personennamen folgendermaßen angeben:

```
<person>
  <name>Goethe</name>
  <vorname>Johann</vorname>
  <vorname>Wolfgang</vorname>
</person>
```

Auch die Angabe ? für ein- oder keinmal kann sehr nützlich sein. Bei dem cd-Beispiel ist es etwa angebracht, das Cover-Bild optional zu setzen, weil für manche CDs vielleicht keines verfügbar ist:

```
<!ELEMENT cd (artist, title, cover?, genre,
  ((year, country) | (country, year)), tracks)>
```

Das Gleiche gilt für die Angabe von Postleitzahl oder Straße und Hausnummer bei Anschriften: Große Postempfänger besitzen manchmal ihre eigene Postleitzahl, die eine weitere Angabe überflüssig macht. Die ultimative Fassung des Elements `anschrift` sieht demnach so aus:

```
<!ELEMENT anschrift (name, (#PCDATA | postfach |
  (strasse, hausnr))?, plz, ort)>
```

Der Modifikator * für beliebig oft (auch keinmal) könnte beispielsweise nützlich sein, um die Adels- oder akademischen Titel einer Person anzugeben. Die folgende Definition berücksichtigt so gut wie alle Eventualitäten bei der Angabe von Personennamen:

```
<!ELEMENT person (anrede, titel*, vorname+, name,
  geburtsname?)>
```

Eine Person kann nach diesem Schema beliebig viele Titel tragen, einen oder mehrere Vornamen führen und einen vom aktuellen Namen abweichenden Geburtsnamen haben oder auch nicht. Die folgenden Beispiele genügen diesem Modell:

```
<person>
  <anrede>Herr</anrede>
  <titel>Dr.</titel>
  <vorname>Klaus</vorname>
  <vorname>Peter</vorname>
  <name>Schmitz</name>
</person>
```

```
<person>
  <anrede>Frau</anrede>
  <titel>Prof.</titel>
  <titel>Dr.</titel>
  <vorname>Annette</vorname>
  <name>Schmitz</name>
  <geburtsname>Müller</geburtsname>
</person>
```

Die folgende kleine DTD definiert ein Format für einfache Textdokumente mit wenigen Auszeichnungsmöglichkeiten:

```
<!ELEMENT dokument ((ueberschrift?, absatz+)+)>
  <!ELEMENT ueberschrift (#PCDATA)>
  <!ELEMENT absatz ((#PCDATA | fett | kursiv)+)>
    <!ELEMENT fett (#PCDATA)>
    <!ELEMENT kursiv (#PCDATA)>
```

Ein Dokument kann laut dieser DTD einen oder mehrere Blöcke enthalten, die aus einer oder keiner Überschrift und einem oder mehreren Absätzen bestehen. Eine Überschrift enthält nur einfachen Text. Ein Absatz dagegen kann einen oder mehrere Teile enthalten, die aus einfachem Text, einem fetten oder einem kursiven Bereich bestehen. Die fetten oder kursiven Bereiche enthalten wiederum einfachen Text.

Das folgende kurze Beispiel zeigt ein Dokument, das sich an diese DTD hält:

```
<dokument>
  <ueberschrift>XML</ueberschrift>
  <absatz>
    <fett>XML</fett> ist ein vom <kursiv>W3C</kursiv>
    definiertes Metaformat für die Definition von
    <kursiv>Auszeichnungssprachen</kursiv>.
  </absatz>
  <absatz>
    Inzwischen gibt es Unmengen von Formaten, die
    auf XML basieren, beispielsweise <fett>XHTML</fett>,
    <fett>SVG</fett> oder <fett>MathML</fett>.
  </absatz>
</dokument>
```

Wenn Sie das Dokument mit geeigneten Mitteln (zum Beispiel mit einem XSLT-Template, siehe unten) verarbeiten, könnte es in einem fertigen Layout zum Beispiel folgendermaßen aussehen:

XML

XML ist ein vom W3C definiertes Metaformat für die Definition von *Auszeichnungssprachen*.

Inzwischen gibt es Unmengen von Formaten, die auf XML basieren, beispielsweise *XHTML*, *SVG* oder *MathML*.

Formal ein wenig einfacher, aber inhaltlich dafür komplexer als bei den Elementen sind die DTD-Regeln für die Definition von Attributen. Wie oben beschrie-

ben, werden Attribute mithilfe einer `<!ATTLIST>`-Deklaration angegeben. Beispielsweise befindet sich in der `cds-DTD` die folgende Attributliste für das Element `cover`:

```
<!ATTLIST cover file CDATA #REQUIRED
                type CDATA #REQUIRED>
```

Jede Attributangabe besteht aus dem Attributnamen (hier `file` und `type`), dem Attributtyp (im Beispiel `CDATA` für einen beliebigen Textinhalt) und der Angabe, ob das Attribut erforderlich ist (in diesem Fall `#REQUIRED` für erforderlich).

Alternativ könnten Sie die beiden Attribute des Elements `cover` auch folgendermaßen angeben:

```
<!ATTLIST cover file CDATA #REQUIRED>
<!ATTLIST cover type CDATA #REQUIRED>
```

Es ist mit keiner der beiden Definitionsmethoden möglich, eine bestimmte Reihenfolge der Attribute eines Elements festzulegen. Wegen der besonderen Bedeutung der Attribute als nähere Bestimmungen eines Elements wäre dies auch gar nicht wünschenswert. Es spielt überhaupt keine Rolle, ob zuerst der Dateiname oder zuerst der MIME-Type einer CD-Cover-Abbildung angegeben wird; beide sind erforderlich, um das Bild in einer eventuellen Anwendung korrekt darstellen zu können.

Was Attributangaben besonders komplex macht, ist die Tatsache, dass es zehn verschiedene Attributtypen gibt. Der häufigste von allen ist `CDATA` für einen beliebigen Textstring. Fast genauso häufig wird der spezielle Inhaltstyp »Aufzählung« verwendet. Für diesen Typ wird kein spezielles Schlüsselwort angegeben, sondern lediglich eine in Klammern stehende, durch `|`-Zeichen getrennte Liste von Alternativen.

Beispielsweise definiert die folgende `<!ATTLIST>` die Attribute eines Elements namens `farbe`, das die Intensität einer der drei RGB-Grundfarben `rot`, `gruen` oder `blau` angibt:

```
<!ATTLIST farbe ton          (rot|gruen|blau) #REQUIRED
                intensitaet CDATA          #REQUIRED>
```

Trotz der vielen verschiedenen Attributtypen gibt es keine vernünftige Möglichkeit, die Intensität auf eine ganze Zahl zwischen 0 und 255 einzuschränken – es sei denn, Sie haben Lust, statt `CDATA` die vollständige Liste (`0 | 1 | 2 | ... | 254 | 255`) einzugeben (oder von einem kleinen Skript generieren zu lassen). Absolut unmöglich ist im Übrigen die Beschränkung auf einen bestimmten

Datentyp wie »ganze Zahl«, »Datum/Uhrzeit« oder Ähnliches. Für solche Feinheiten ist das weiter unten vorgestellte XML Schema geeigneter.

Was die verschiedenen Attributtypen dagegen zu bieten haben, sehen Sie übersichtlich in Tabelle 11.2.

Typ	Erläuterung	Beispiele
CDATA	beliebiger Text	"Hallo", "25"
(Aufzählung)	Liste möglicher Alternativen wie (rot gruen blau)	"kreuz" aus ("kreuz" "pik" "herz" "karo")
NMTOKEN	Darf nur die Zeichen enthalten, die in XML-Tag-Namen erlaubt sind. Ermöglicht eine stärkere Inhaltskontrolle als CDATA.	"13.05.2003", "hallo", "funny-names"
NMTOKENS	Liste mehrerer NMTOKEN-Werte. XML-Parser behandeln Leerzeichen als Trennzeichen für mehrere Einzelwerte.	"kueche diele bad" (Parser bildet die Einzelwerte kueche, diele und bad)
ID	Werte wie bei NMTOKEN; jeder Wert muss jedoch im gesamten XML-Dokument einmalig sein. Dies ist nützlich für eindeutige Schlüssel.	"isbn3898421376", "B12345" (leider ist eine reine Zahl verboten, weil sie kein gültiger XML-Name ist)
IDREF	Bezug auf ein Attribut vom Typ ID eines anderen Elements. Dient der Definition von Bezügen wie in relationalen Datenbanken (siehe Kapitel 12).	(siehe ID)
IDREFS	Eine durch Leerzeichen getrennte Liste mehrerer IDs, auf die Bezug genommen wird.	"isbn3898421376 isbn3898423557"
ENTITY	Verweist auf ein innerhalb der DTD definiertes Entity (Standard-Entities wie lt oder quot sind nicht zulässig).	(Entities werden weiter unten genau erläutert)

Tabelle 11.2 Die zulässigen Attributtypen in DTDs

Typ	Erläuterung	Beispiele
ENTITIES	Eine durch Leerzeichen getrennte Liste mehrerer ENTITY-Werte.	
NOTATION	Der Wert eines speziellen DTD-Konstrukts vom Typ <code><!NOTATION></code> , das die Abkürzung von SYSTEM- oder PUBLIC-IDs ermöglicht.	"gif", bezogen auf: <pre><!NOTATION gif SYSTEM "image/gif"> <!NOTATION jpg SYSTEM "image/jpeg"> <!NOTATION png SYSTEM "image/png"> <!ATTLIST bild typ NOTATION(gif jpg png)></pre>

Tabelle 11.2 Die zulässigen Attributtypen in DTDs (Forts.)

Die meisten dieser Attributtypen werden Sie in eigenen DTDs wahrscheinlich niemals verwenden. Die allermeisten Aufgaben können Sie mit `CDATA` und Aufzählungen erledigen; wenn Sie XML für datenbankähnliche Aufgaben einsetzen, werden Sie wahrscheinlich auch `ID` und `IDREF` beziehungsweise `IDREFS` nützlich finden.

Die letzte Angabe innerhalb einer Attribut-Definition gibt an, ob das Attribut erforderlich ist oder nicht, und besagt, ob es einen Standardwert für dieses Attribut gibt. Die vier möglichen Werte sind folgende:

- ▶ `#REQUIRED`
Das Attribut muss auf jeden Fall angegeben werden; es gibt keinen Standardwert.
- ▶ `#IMPLIED`
Das Attribut kann weggelassen werden; einen Standardwert gibt es auch hier nicht.
- ▶ `#FIXED`
Das Attribut hat stets den hinter `#FIXED` angegebenen Standardwert. Wird es nicht angegeben, dann wird es vom Parser trotzdem mit dem Standardwert ausgewertet. Falls es explizit angegeben wird, muss es dagegen genau den Standardwert aufweisen.
- ▶ `Literal`
Wenn Sie statt eines der drei Schlüsselwörter nur einen Standardwert in Anführungszeichen angeben, hat das Attribut diesen Standardwert, wenn Sie es weglassen. Geben Sie es dagegen explizit an, dann erhält es den entsprechenden Wert.

Die folgende kleine DTD definiert einen einfachen Farbverlauf aus zwei RGB-Farben. Das leere Element `rgb` besitzt verschiedene Attribute, um Rot-/Grün-/Blauwert und Deckkraft (`alpha`) zu definieren:

```
<!ELEMENT verlauf (rgb, rgb)>
  <!ELEMENT rgb EMPTY>
  <!ATTLIST rgb rot CDATA "255"
              gruen CDATA "255"
              blau CDATA "255"
              alpha CDATA "100">
```

Die Standard-RGB-Farbe ist demnach Weiß mit einer Deckkraft von 100 %. Einen Verlauf von Schwarz nach Weiß, beide mit 100% Deckkraft, können Sie also mit minimalem Aufwand folgendermaßen definieren:

```
<verlauf>
  <rgb rot="0" gruen="0" blau="0" />
  <rgb />
</verlauf>
```

Ein XML-Parser, der die DTD verarbeitet und diese Elemente liest, ergänzt sie automatisch zu folgender Langform:

```
<verlauf>
  <rgb rot="0" gruen="0" blau="0" alpha="100" />
  <rgb rot="255" gruen="255" blau="255" alpha="100" />
</verlauf>
```

Entities bieten vor allem die Möglichkeit, häufig vorkommende XML-Blöcke abzukürzen sowie Zeichen darzustellen, die im aktuellen Zeichensatz des XML-Dokuments nicht vorkommen. Im XML-Dokument werden die in der DTD definierten Entities durch *Entity-Referenzen* angegeben. Die fünf fest in XML eingebauten Entity-Referenzen wurden bereits weiter oben erwähnt.

Weitere Entities können Sie auf einfache Art und Weise innerhalb einer DTD definieren. Das folgende Entity definiert eine Copyright-Zeile:

```
<!ENTITY copymsg "Copyright 2008 by Galileo Computing">
```

Wenn Sie an irgendeiner Stelle eines XML-Dokuments, das von dieser DTD abhängt, die Entity-Referenz `©msg;` verwenden, wird sie automatisch durch den String "Copyright 2008 by Galileo Computing" ersetzt. Entities können aber nicht nur reinen Text enthalten, sondern auch XML-Auszeichnungen. Das folgende Beispiel kürzt eine vollständige CD gemäß der oben vorgestellten DTD auf das Entity `&cd;` herunter:

```
<!ENTITY cd '
  <cd num="3">
    <artist type="band">Guns N&apos; Roses</artist>
    <title>Appetite for Destruction</title>
```

```

<genre>Hard Rock</genre>
<year>1987</year>
<country>USA</country>
<tracks>
  <track num="1">
    <title>Welcome to the Jungle</title>
    <duration>273</duration>
  </track>
  <!-- 10 weitere Tracks -->
  <track num="12">
    <title>Rocket Queen</title>
    <duration>372</duration>
  </track>
</tracks>
</cd>
'>

```

Beachten Sie im obigen Beispiel die einfachen Anführungszeichen, in denen der Code für das Album steht – sie ermöglichen, dass Sie die doppelten Anführungszeichen der enthaltenen Attributwerte einfach stehen lassen können.

Längere Entities müssen Sie nicht innerhalb der DTD selbst definieren, sondern können sie in eine externe XML-Datei schreiben. Auf diese Datei wird dann in der Entity-Deklaration mithilfe einer SYSTEM-ID verwiesen:

```
<!ENTITY cd2 SYSTEM "cd2.xml">
```

Damit die Entity-Referenz `&cd2;` aufgelöst werden kann, muss im Verzeichnis, in dem sich die DTD befindet, eine formal korrekte XML-Datei namens `cd2.xml` vorliegen.

Namensräume

Eine besondere Eigenschaft von XML-Dokumenten besteht darin, dass Sie mehrere Dokumenttypen miteinander vermischen können. Zu diesem Zweck wurden die *Namensräume* (englisch »name spaces«) eingeführt, die die Unterscheidung von Elementen aus verschiedenen DTDs zulassen.

Der *Standard-Namensraum* eines Dokuments zeichnet sich dadurch aus, dass Sie seine Elemente ohne spezielles Präfix verwenden können. Werden weitere Namensräume eingebunden, dann wird deren Elementen ein Namensraum-Präfix vorangestellt, das durch einen Doppelpunkt vom eigentlichen Elementnamen getrennt wird.

Angenommen, Sie fügen in die Datei mit den CDs ein neues Element namens `description` ein, das eine kurze Beschreibung der CD im HTML-Format enthält. Zu diesem Zweck können Sie den verwendeten HTML-Tags das Namensraum-Präfix `html` voranstellen, um sie von den Elementen des Standard-Namensraums zu unterscheiden. Hier ein kurzes Beispiel:

```
<cd num="4">
  <artist type="band">Extreme</artist>
  <title>Extreme II: Pornograffiti</title>
  <genre>Metal</genre>
  <year>1990</year>
  <country>USA</country>
  <description>
  <html:p>Das zweite, kommerziell bisher erfolgreichste Album
  der Bostoner Rockband <html:b>Extreme</html:b>
  besticht durch funky angehauchten Metal, der
  vor allem von der Genialität des Ausnahmegitarristen
  <html:i>Nuno Bettencourt</html:i> lebt.</html:p>
  </description>
  <tracks>
    <track num="1">
      <title>Decadence Dance</title>
      <duration>409</duration>
    </track>
    <!-- 11 weitere Tracks -->
    <track num="13">
      <title>Hole Hearted</title>
      <duration>219</duration>
    </track>
  </tracks>
</cd>
```

Die in der obigen Beschreibung verwendeten HTML-Auszeichnungen definieren einen Absatz, Fett- und Kursivschrift. Sie werden in Abschnitt 11.2, »HTML, XHTML und CSS«, genauer erläutert.

Namensräume werden mithilfe von `xmlns`-Angaben innerhalb eines Elements in das Dokument eingebunden. In der Regel stehen sie im Wurzelement. Der Standard-Namensraum wird einfach als `xmlns` bezeichnet, während zusätzliche Namensräume mit `xmlns:namensraum` angegeben werden. Das folgende Beispiel bindet das `xml-buecher`-Format als Standard-Namensraum und das HTML-Format als Erweiterung ein:

```
<xml-buecher xmlns="http://buecher.lingoworld.de/opensuse/listings/
11/common/cds" xmlns:html="http://www.w3.org/1999/xhtml">
```

Unter den angegebenen URLs muss sich kein spezielles Dokument befinden, das den Namensraum definiert – allerdings ist es üblich, dort eine kurze Beschreibung des Namensraums im HTML-Format abzulegen. Wichtig ist nur, dass verschiedene Namensraum-Angaben auch unterschiedliche URLs verwenden.

XML Schema

XML Schema bietet eine alternative Möglichkeit, Standards für XML-Dokumente einzurichten. Gegenüber DTDs besitzt dieses Format vor allem zwei Vorteile:

- ▶ Die zulässigen Inhalte für Elemente und Attribute können wesentlich genauer angegeben werden.
- ▶ Das Format ist selbst XML-basiert, verwendet also keine separate Syntax wie DTDs, sondern besteht aus wohlgeformten XML-Dokumenten.

Eine Schema-Definition steht in einer Datei mit der Endung *.xsd*. Die meisten aktuellen XML-Parser unterstützen ein Schema als Alternative zu einer DTD für die Validation von Dokumenten.

Hier ein einfaches Schema für Adresslisten-Dokumente auf der Basis des weiter oben definierten Adressbeispiels:

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="adressliste">
    <xs:complexType>
      <xs:element name="person" minOccurs="1"
maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="anrede"
type="xs:string" />
            <xs:element name="titel" type="xs:string"
minOccurs="0" />
            <xs:element name="vorname" type="xs:string"
maxOccurs="unbounded" />
            <xs:element name="name" type="xs:string" />
            <xs:element name="anschrift">
              <xs:complexType>
                <xs:choice>
                  <xs:element name="strasse"
type="xs:string" />
                  <xs:element name="plz"
type="xs:string" />
                </xs:choice>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```

        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:complexType>
</xs:element>
</xs:schema>

```

Grundsätzlich wird jedes Element mithilfe einer `<xs:element>`-Deklaration angegeben. Elemente, die nur einfachen Textinhalt und keine verschachtelten Elemente oder Attribute besitzen, benötigen im Schema das Attribut `type`, das den zulässigen Typ des Inhalts angibt. Einige zulässige Typen sind `"xs:string"` für beliebigen Text, `"xs:integer"` für ganze Zahlen oder `"xs:language"` für eine ISO-Sprachangabe wie `en` (Englisch) oder `de` (Deutsch). Diese und andere Inhaltstypen können sowohl für Elemente als auch für Attribute verwendet werden.

Elemente, die verschachtelte Elemente, gemischten Inhalt oder Attribute enthalten dürfen, benötigen für die Angabe dieser Komponenten einen `<xs:complexType>`-Block. Für verschachtelte Elemente steht innerhalb dieses Blocks entweder ein einzelnes Element, eine durch einen `<xs:sequence>`-Abschnitt umschlossene Liste aufeinanderfolgender Elemente oder eine durch `<xs:choice>` umhüllte Aufzählung von Alternativen.

Attribute – die im obigen Beispiel nicht vorkommen – werden übrigens folgendermaßen deklariert (hier am Beispiel der im Einführungsbeispiel verwendeten CD-Nummer):

```
<xs:attribute name="num" type="xs:integer" />
```

Angenommen, Sie möchten ein Element `buchtitel` deklarieren, dessen Inhalt einfacher Text ist und das ein Attribut namens `isbn` enthält. Da Elemente mit Attributen immer einen `<xs:complexType>`-Block benötigen, in dem ihr Inhalt definiert wird, können Sie nicht mehr einfach `type="xs:string"` schreiben. Stattdessen sieht die Definition nun so aus:

```

<xs:element name="buchtitel">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="isbn" type="xs:string" />
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>

```

Der `<xs:simpleContent>`-Block gibt an, dass das Element trotz des `<xs:complexType>` nur einfachen Inhalt und keine verschachtelten Tags enthält. `<xs:extension>` gibt den Inhaltstyp des Elements selbst an (hier "xs:string"), während die hineinverschachtelten `<xs:attribute>`-Elemente die Attribute und ihre Datentypen definieren.

Interessant ist schließlich noch die Möglichkeit, über `minOccurs` und `maxOccurs` die minimale und die maximale Anzahl von Elementen eines Typs anzugeben. Beide haben den Standardwert 1; ein Element ohne weitere Angaben muss genau einmal vorkommen. Ein spezieller Wert für `maxOccurs` ist "unbounded". Das entsprechende Element darf unbegrenzt oft vorkommen.

Das folgende kurze Beispiel genügt dem oben definierten Schema und zeigt außerdem, wie Sie die Verwendung des Schemas im XML-Dokument angeben.¹

```
<?xml version="1.0"?>
<adressliste xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="adressen.xsd">
  <person>
    <anrede>Herr</anrede>
    <titel>Doktor</titel>
    <vorname>Dieter</vorname>
    <name>Heinze</name>
    <anschrift>
      <strasse>Alte Straße 34</strasse>
    </anschrift>
  </person>
  <person>
    <anrede>Frau</anrede>
    <vorname>Maria</vorname>
    <vorname>Theresia</vorname>
    <name>Huber</name>
    <anschrift>
      <postfach>1234567</postfach>
    </anschrift>
  </person>
</adressliste>
```

Über dieses kurze Beispiel hinaus ist XML Schema eine sehr umfangreiche Sprache, die sehr detaillierte Definitionen für Dokumentformate ermöglicht. Der

1 Die hier sehr verkürzt behandelte Schema-Spezifikation wird noch erheblich komplexer, wenn Namensräume ins Spiel kommen. »xsi:noNamespaceSchemaLocation« bindet nur Schema-Definitionen ein, die keine speziellen Namensräume deklarieren.

kleine Einblick in diesem Unterabschnitt hat hoffentlich gezeigt, dass die Möglichkeiten von XML Schema weit über DTDs hinausgehen. Wenn Sie alle Möglichkeiten ausschöpfen möchten, sollten Sie das (derzeit vergriffene) Buch »XML Schema« von Marco Skulschus und Marcus Wiederstein (Bonn 2004, Galileo Computing) lesen.

11.1.3 XSLT

Wenn Sie dem Kapitel bis hierhin gefolgt sind, wird Ihnen aufgefallen sein, dass Sie zwar eine Reihe verschiedener XML-Dokumente und -Fragmente gesehen haben, aber noch keinen Screenshot oder eine andere Darstellung eines fertig verarbeiteten Dokuments. Das liegt daran, dass XML einzig und allein die Struktur eines Dokuments beschreibt. Für das Layout sind separate Formatierungsangaben zuständig, die meist in der *Extensible Stylesheet Language (XSL)* verfasst werden.

XSL besteht aus zwei verschiedenen Komponenten: Die in diesem Abschnitt behandelten *XSL-Transformations (XSLT)* beschreiben die Umwandlung (Transformation) beliebiger XML-Dokumente in andere Formate wie HTML oder PDF, während *XSL Formatting Objects (XSL-FO)* eine eigene Sprache zur Definition von Formatierungen und Stilen bilden. Eine dritte Alternative für die Formatierung bieten die am häufigsten für HTML-Dokumente eingesetzten *Cascading Style-sheets (CSS)*, die im nächsten Abschnitt erläutert werden.

Ein XSLT-Dokument beschreibt, welche Elemente und Attribute eines XML-Eingabedokuments in welche Bestandteile eines Ausgabedokuments umgesetzt werden. Um die einzelnen Komponenten des Eingabedokuments zu identifizieren, verwendet XSLT eine Sprache namens *XPath*, die den Zugriff auf die Baumstruktur jedes XML-Dokuments ermöglicht.

Um XSLT einsetzen zu können, benötigen Sie eine spezielle Anwendung namens XSLT-Prozessor. Dieses Programm nimmt ein XSLT-Stylesheet und ein XML-Dokument entgegen und erzeugt gemäß den Regeln des Stylesheets das gewünschte Ausgabedokument.

Einer der bekanntesten XSLT-Prozessoren ist *Apache Xalan*, den Sie unter xml.apache.org/xalan herunterladen können. Es handelt sich um eine Java-Anwendung, die Sie von der Kommandozeile aus folgendermaßen aufrufen können, falls bei Ihnen mindestens das Java Runtime Environment (JRE) installiert ist:

```
$ java org.apache.xalan.xslt.Process -IN cds.xml
  -XSL cds2htm.xsl -OUT cds.html
```

Der Kommandozeilenparameter `-IN` gibt, wie Sie wahrscheinlich richtig vermutet haben, das Eingabedokument an, `-XSL` bezeichnet die XSLT-Stylesheet-Datei mit der Endung `.xsl` und `-OUT` die Ausgabedatei.

Die häufigste Umwandlung – und die einzige, die in diesem kurzen Unterabschnitt demonstriert wird – ist diejenige von allgemeinem XML nach HTML.

Hier ein kurzes XSLT-Stylesheet, das die zu Beginn dieses Kapitels vorgestellte Datei `cds.xml` in ein einfaches HTML-Dokument umwandelt:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="cds">
    <html>
      <head>
        <title>Rock-CDs</title>
      </head>
      <body>
        <h1>Rock-CDs</h1>
        <xsl:for-each select="cd">
          <h2><xsl:value-of select="title" /></h2>
          <p>
            von
            <xsl:value-of select="artist" />
          </p>
          <p>
            <xsl:value-of select="country" />,
            <xsl:value-of select="year" />
          </p>
          <p>
            <xsl:for-each select="tracks/track">
              <xsl:value-of select="@num" />
              <xsl:text> </xsl:text>
              <xsl:value-of select="title" />
              <xsl:text> </xsl:text>
              <xsl:value-of select="floor(duration div 60)" />:
              <xsl:if test="(duration mod 60)
                &lt; 10">0</xsl:if><xsl:value-of select="duration mod 60" />
              <br />
            </xsl:for-each>
          </p>
        </xsl:for-each>
      </body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```

Wie Sie sehen, wird die gesamte Struktur des HTML-Dokuments einfach aufgeschrieben, umschlossen und unterbrochen von XSLT-Angaben, die die Inhalte bestimmter Elemente aus *cds.xml* an den passenden Stellen einfügen.

Im Einzelnen werden im Beispiel die folgenden XSLT-Elemente verwendet:

▶ `xsl:stylesheet`

Dieses Element ist das Wurzelement des gesamten XSLT-Stylesheets.

▶ `xsl:template`

Erwartet als Inhalt des Attributs `match` den Namen eines XML-Elements aus dem Dokument, dessen Inhalte verarbeitet werden sollen. Bevor Sie die erste Zeile des Ausgabedokuments schreiben, müssen Sie mittels `xsl:template` einen gültigen Pfad zu einem XML-Element angeben. Meistens wird das Wurzelement des Eingabedokuments ausgewählt. Zu diesem Zweck bietet XSLT die Kurzfassung `match="/"` an. Ein Template kann auch mittels `<xsl:template name="Bezeichnung">...</template>` deklariert und dann mithilfe von `<xsl:call-template name="Bezeichnung" />` explizit aufgerufen werden.

In die `call-template`-Aufrufe können Sie beliebig viele Parameterübergaben im Format `<xsl:with-param name="Parametername" select="Wert/XPath-Ausdruck" />` verschachteln; diese werden dann zu Beginn einer benannten Template-Definition mittels `<xsl:param name="Parametername"/>` entgegengenommen. Innerhalb von XPath-Ausdrücken im aufgerufenen Template werden diese Parameter als `$Parametername` angesprochen. Auf diese Weise erledigen Templates auch eine ähnliche Aufgabe wie Funktionen in Programmiersprachen.

▶ `xsl:for-each`

Dieses Element funktioniert ähnlich wie `xsl:template`. Allerdings durchläuft es sämtliche Elemente des unter `select` angegebenen Typs in einer Schleife.

▶ `xsl:text`

Kann verwendet werden, um explizit Whitespace einzufügen, der vom XSLT-Prozessor andernfalls womöglich ignoriert wird.

▶ `xsl:value-of`

Liest den Textinhalt eines Elements, das PCDATA oder CDATA enthält. Wenn Sie dem Attributwert von `select` ein `@` voranstellen, wird nicht der Wert eines Elements ausgelesen, sondern der Wert eines Attributs des aktuellen Elements, das zuvor mittels `xsl:template` oder `xsl:for-each` ausgewählt wurde (Näheres zum Zugriff auf den XML-Baum mittels XPath erfahren Sie am Ende dieses Abschnitts). Wie das Beispiel der `duration` zeigt, kann `xsl:value-of` auch Ausdrücke auswerten (in diesem Fall wird aus den Sekunden die ganze Minutenzahl sowie die Restsekundenzahl errechnet).

Eine Kurzfassung für `xsl:value-of`, die speziell für die Ausgabe von Attributwerten gedacht ist, besteht darin, den auszugebenden XPath-Ausdruck direkt in geschweiften Klammern in die Anführungszeichen des jeweiligen Attributs einzusetzen. Das folgende Beispiel übernimmt ein Attribut namens `font-size` aus dem aktuell gelesenen Element in ein gleichnamiges HTML-`style`-Attribut eines Absatzes:

```
<p style="font-size: {@font-size}">...</p>
```

Wenn Sie statt einfachem Text eine verschachtelte Struktur aus dem Original-XML-Dokument übernehmen möchten, können Sie stattdessen `xsl:copy-of` verwenden.

► `xsl:if`

Mit `<xsl:if test="Ausdruck">...</xsl:if>` kann ein bestimmter Teil des Stylesheets optional gesetzt werden – es wird nur ausgeführt, wenn der Ausdruck zutrifft. Im vorliegenden Beispiel wird eine Null vor die Restsekundenzahl gesetzt, falls diese kleiner als 10 ist.

Beachten Sie, dass die Vergleichsoperatoren `<`, `>`, `<=` und `>=` mit den entsprechenden Entity-Referenzen als `<`, `>`, `<=` beziehungsweise `>=` geschrieben werden müssen.

Abbildung 11.1 zeigt das Ergebnisdokument, wie es von Firefox dargestellt wird.

Neben den hier verwendeten XSLT-Konstrukten existiert noch eine Reihe weiterer, die den Rahmen dieses Kapitels sprengen würden. Im Grunde ist XSLT eine Turing-vollständige Programmiersprache, wenn auch mit etwas eigenwilliger Syntax.

Das Wichtigste, was Sie beachten müssen, ist die korrekte Reihenfolge bei der Konstruktion von XSLT-Stylesheets: Sie müssen sowohl die Hierarchie des Eingabedokuments mittels `xsl:template`- und `xsl:foreach`-Tags korrekt verarbeiten als auch die Wohlgeformtheit des Ausgabedokuments beachten.

Neben den eigentlichen XSLT-Elementen sollten Sie auch noch einige XPath-Ausdrücke kennen, um erfolgreich Templates schreiben zu können. Hier die wichtigsten im Überblick:

- Das jeweils aktive Element (Kontextelement), das beispielsweise mittels `xsl:template` oder `xsl:for-each` ausgewählt wurde, hat einen einfachen Punkt (`.`) als Namen. Beachten Sie, dass das Attribut `test` einer `xsl:if`-Anweisung den Kontext nicht ändert.

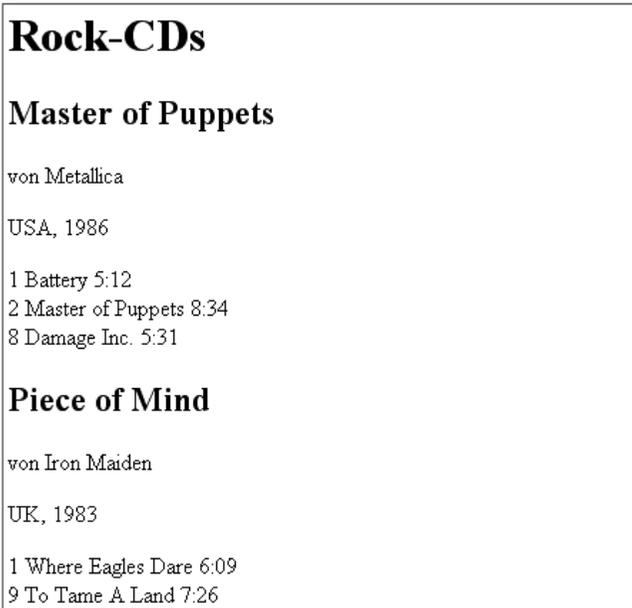


Abbildung 11.1 Ausgabe des Ergebnisses der XSLT-Transformation im Browser Firefox

- ▶ Direkte Kindelemente des Kontextelements können einfach mit ihrem Namen angesprochen werden. Angenommen, Sie haben die Struktur `<interpret><name/></interpret>` und das Kontextelement ist `<interpret>`, dann genügt `name` für den Zugriff auf den Kindknoten `name`. `./name` ist allerdings auch zulässig.
- ▶ In verschachtelten Strukturen werden die Elemente durch `/` voneinander getrennt. Wird beispielsweise die Struktur `<cd><interpret><name/></interpret></cd>` verarbeitet, und das Kontextelement ist `<cd>`, dann können Sie mit `interpret/name` auf den Knoten `<name>` zugreifen.
- ▶ Ein `*` greift auf alle direkten Kinder des gewählten Knotens zu. Nützlich ist dies vor allem für `xsl:for-each`-Konstrukte. Mithilfe der XPath-Funktion `local-name(.)` können Sie dann innerhalb der Schleife herausfinden, um welche Art von Element es sich jeweils handelt. Auf diese Weise können Sie etwa mit `xsl:if` überprüfen, welches Element gerade verarbeitet wird.
- ▶ Attribute werden, wie bereits erwähnt, durch ein vorangestelltes `@` von Elementen unterschieden. In der Struktur `<track num="..." />` greift `@num` (beziehungsweise `track/@num`, falls `<track>` nicht das Kontextelement ist) beispielsweise auf den Wert des Attributs `num` zu.

11.2 HTML, XHTML und CSS

In diesem Abschnitt wird (X)HTML vorgestellt, die grundlegende Sprache zum Schreiben von Webseiten. Hinzu kommen die Grundlagen der Formatierung mit Cascading Stylesheets (CSS). Zum Schluss wird mit Quanta Plus ein praktisches Werkzeug zum Editieren von HTML-Dokumenten gezeigt.

11.2.1 HTML und XHTML

HTML steht für *Hypertext Markup Language* (Auszeichnungssprache für Hypertext, also Text mit integrierten Querverweisen). HTML-Code sieht im Wesentlichen genauso aus wie das im vorigen Abschnitt vorgestellte XML – die gemeinsame Wurzel von XML und HTML ist die Auszeichnungssprache SGML. Das klassische HTML ist eine bestimmte SGML-DTD; es sind also nur ganz bestimmte Tags in einer vorgegebenen Anordnung erlaubt. Die aktuelle (und vorläufig letzte) Version dieser Sprache ist HTML 4.01.

Da in herkömmlichem HTML eine Reihe von Freiheiten gestattet sind, die in XML nicht mehr gelten, wurde vom W3C inzwischen eine neue HTML-Variante namens XHTML eingeführt, die von XML abgeleitet ist und für die dieselben Regeln gelten wie für alle anderen XML-Dokumente. Allerdings sind alle bisher verfügbaren Browser tolerant genug, die gelockerte Syntax der alten HTML-Versionen weiterhin zuzulassen.

Inzwischen wurde vom W3C das neue Format HTML 5.0 freigegeben (sowohl in einer XML-basierten als auch in einer klassischen Fassung); es gibt allerdings bisher nur wenige Browser, die die darin enthaltenen Erweiterungen unterstützen.

Die Grundstruktur von XHTML-Dokumenten

Wie alle XML-Varianten besteht auch XHTML aus wohlgeformten Dokumenten; das Wurzelement ist `<html>`. Innerhalb dieses äußersten Containers besteht eine HTML-Seite aus einem *Head* (Kopf), der den *Titel* der Seite und andere Meta-Informationen enthält, und dem *Body* (Körper), in dem der eigentliche Inhalt des Dokuments steht.

Daraus ergibt sich der folgende grundlegende Aufbau einer HTML-Seite:

```
<html>
  <head>
    <title> Titel des Dokuments </title>
  </head>
  <body>
    Der sichtbare Inhalt
  </body>
</html>
```

Im Einzelnen bedeuten die diversen Bestandteile des Dokuments Folgendes:

- ▶ `<html>` teilt dem Browser mit, dass HTML-Code folgt.
- ▶ `<head>` leitet den Dokumentkopf ein.
- ▶ `<title>` leitet den Titel des Dokuments ein.
- ▶ `</title>` schließt den Titel; ein Schrägstrich (Slash) vor dem Tag-Namen bedeutet, dass die entsprechende Auszeichnung an dieser Stelle endet.
An dieser Stelle folgen eventuell weitere technische Informationen, die nicht zum sichtbaren Inhalte der Seite gehören.
- ▶ `</head>` schließt den Kopf.
- ▶ `<body>` leitet den eigentlichen, sichtbaren Inhalt des Dokuments ein.
Der Inhalt selbst besteht aus Text und weiteren ineinander verschachtelten HTML-Tags.
- ▶ `</body>` schließt den Inhalt.
- ▶ `</html>` schließt das gesamte Dokument.

Im Head *muss* das Element `<title>` vorkommen, alle anderen Head-Angaben sind freiwillig. Der Titel des Dokuments ist ein wichtiges Strukturelement, das an folgenden Stellen angezeigt wird:

- ▶ In der Titelleiste eines Browserfensters erscheint er zusammen mit dem Namen des Browsers selbst.
- ▶ Er steht in der Liste der Bookmarks oder Lesezeichen Ihres Browsers, wenn Sie die Seite dort ablegen.
- ▶ In den Ergebnislisten von Suchmaschinen wird der Titel als anklickbarer Link für eine Fundstelle verwendet.

Ein geeigneter Titel sollte den Namen der Website beziehungsweise des Anbieters sowie den Namen der einzelnen Seite enthalten. Hier einige gute Beispiele:

- ▶ XHTML-Kurs – Die erste Webseite
- ▶ Galileo Computing, Buch – openSUSE 11.2
- ▶ Müller GmbH: Produkte

Über dem Dokument-Start-Tag `<html>` steht häufig eine DOCTYPE-Verarbeitungsanweisung, wie sie auch für andere Arten von XML-Dokumenten verwendet wird:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

Textstrukturierung und -formatierung

Bei der Textformatierung in HTML ist zu beachten, dass der Browser sich grundsätzlich nicht für Textformatierungen im Quelltext interessiert. Ob Sie ein Leerzeichen oder zehn setzen, ob Sie einen Zeilenumbruch erzeugen oder nicht: Der Browser setzt den gesamten Fließtext einfach hintereinander. Er erzeugt nur dann automatische Zeilenumbrüche, wenn der rechte Fensterrand erreicht wird.

Wollen Sie dagegen willkürlich einen Zeilenumbruch erzeugen, dann müssen Sie das folgende HTML-Tag verwenden:

```
<br />
```

Beachten Sie jedoch, dass Sie auf keinen Fall die Zeilenlänge von Fließtext bestimmen dürfen, indem Sie hinter jede Zeile ein `
` setzen. Sie riskieren sonst, dass sich bei einer anderen Browserfenster-Größe erzwungene und automatische Zeilenumbrüche abwechseln und auf diese Weise den Zeilenfall völlig zerstören. Sie dürfen feste Zeilenumbrüche also nur an Stellen verwenden, an denen tatsächlich eine neue Zeile beginnen soll, beispielsweise für einfache Aufzählungen, Sinnabschnitte oder Gedichtzeilen.

Ein weiteres Problem ist die Darstellung der *Umlaute* und anderer *Sonderzeichen*: Um weltweit kompatibel zu sein, verwendet HTML zunächst nur reinen ASCII-Code, also die 128 international identischen Zeichen. Es gibt zwei Möglichkeiten, dies zu ändern.

Sie können zum einen im Head den Zeichensatz angeben, der im Dokument verwendet wird. Dazu dient ein sogenanntes `<meta>`-Tag, das Konfigurationsinformationen in Form von Name/Wert-Paaren enthält (Näheres siehe unten). Um beispielsweise `iso-latin-1` – den Standardzeichensatz für die USA und Westeuropa – einzustellen, lautet das entsprechende Tag folgendermaßen:

```
<meta http-equiv="content-type"
content="text/html;charset=iso-8859-1" />
```

Alternativ können Sie die deutschen Umlaute und viele andere Sonderzeichen durch Entity-Referenzen erzeugen. Wie im XML-Abschnitt erläutert, beginnen sie mit einem `&`-Zeichen und enden mit einem Semikolon. Dazwischen befindet sich ein Kürzel, das jeweils das Sonderzeichen beschreibt. Diese Sonderzeichen müssen ohne Abstand in den Fließtext eingebunden werden, damit sie normal im Wort erscheinen.

Tabelle 11.3 zeigt einen Überblick über die gängigsten Sonderzeichencodierungen.

Zeichen	Codierung	Erläuterungen
Umlaute und diakritische Zeichen:		
ä	ä	a-Umlaut
ö	ö	o-Umlaut
ü	ü	u-Umlaut
Ä	Ä	A-Umlaut
Ö	Ö	O-Umlaut
Ü	Ü	U-Umlaut
ß	ß	S-Z-Ligatur
é	é	e mit Akut
è	è	e mit Gravis
ê	ê	e mit Zirkumflex
à	á	a mit Akut
ç	ç	c mit Cedille
ø	ø	o mit / (Slash)
œ	œ	o-e-Ligatur
æ	æ	a-e-Ligatur
ñ	ñ	n mit Tilde
Wegen Gefahr von Missverständnissen codierte Zeichen:		
<	<	less than
>	>	greater than
"	"	quotation mark
&	&	Ampersand (»and per se and«)
Weitere Sonderzeichen:		
€	€	Euro
£	£	Pfund Sterling
¥	¥	Yen
¡	&ixcl;	inverse exclamation mark

Tabelle 11.3 Die wichtigsten HTML-Umlaute und -Sonderzeichen

Zeichen	Codierung	Erläuterungen
¿	¿	inverse question mark
©	©	Copyright
®	®	Registered Trademark
§	§	section
„	„	Anführungszeichen 99 unten (Deutsch Anfang)
“	“	Anführungszeichen 66 oben (Deutsch Ende, Englisch Anfang)
”	”	Anführungszeichen 99 oben (Englisch Ende)
«	«	left angular quote (Deutsch Ende; Französisch Anfang)
»	»	right angular quote (Deutsch Anfang; Französisch Ende)

Tabelle 11.3 Die wichtigsten HTML-Umlaute und -Sonderzeichen (Forts.)

Ein weiteres wichtiges Sonderzeichen konnte leider nicht in der Tabelle untergebracht werden, da es für sich allein vollkommen unsichtbar ist: Das Zeichen ` `; es steht für Non-Breaking Space (nicht umbrechendes Leerzeichen) und erfüllt zwei wichtige Aufgaben:

- ▶ Da der Browser Wörter und andere Elemente, die im HTML-Code durch beliebig viel Whitespace (Leerzeichen, Zeilenumbrüche, Tabulatoren) voneinander getrennt sind, stets durch genau ein Leerzeichen trennt, kann das ` ` zur Erzeugung mehrerer aufeinanderfolgender Leerzeichen verwendet werden.
- ▶ Wenn Sie Wörter oder andere Inhalte in derselben Zeile zusammenhalten möchten, können Sie sie durch dieses Zeichen voneinander trennen, da es – seinem Namen entsprechend – den Zeilenumbruch verhindert. Um etwa »openSUSE 11.2« zusammenzuhalten, müssten Sie im HTML-Code `openSUSE 11.2` schreiben.

Zur Kennzeichnung von *Absätzen* wird das folgende HTML-Tag eingesetzt:

```
<p> ... </p>
```

Das `p` steht für »paragraph« (Absatz). Falls Sie sich entschließen, Ihren Text durch Absätze zu gliedern, sollten Sie dies im gesamten Dokument tun – oder es ganz lassen; wie immer kann Inkonsequenz zu unvorhergesehenen Problemen führen.

Neben der Einteilung von Texten in Sinnabschnitte besitzen Absätze eine weitere Fähigkeit: Über das Attribut `align` können Sie die Textausrichtung bestimmen.

```
<p align="left"> ... </p>
```

erzeugt einen linksbündigen Absatz; das ist allerdings Standard und wird deshalb in der Regel nicht angegeben. Anders sieht es aus, wenn Sie einen Zeichensatz mit der Laufrichtung von rechts nach links verwenden, beispielsweise für Arabisch. In diesem Fall wäre der Standard die rechtsbündige Ausrichtung.

```
<p align="center"> ... </p>
```

erzeugt einen zentrierten Absatz. Mit

```
<p align="right"> ... </p>
```

wird ein rechtsbündiger Absatz erzeugt. Zu guter Letzt existiert noch die folgende Einstellung:

```
<p align="justify"> ... </p>
```

Dies erzeugt einen Absatz mit Blocksatz. Für Bildschirmlayouts ist dies in der Regel nicht ratsam, zumal Blocksatz ohne eine leistungsfähige automatische Silbentrennung nicht besonders ästhetisch wirkt.

Einen Sonderfall der Absätze bilden die *Überschriften*. Sie können in HTML in sechs verschiedenen Hierarchiestufen verwendet werden:

```
<h1>...</h1>
<h2>...</h2>
...
<h6>...</h6>
```

Das `h` steht für »headline« (Überschrift). Browser stellen die verschiedenen Stufen in der Regel durch unterschiedliche Schriftgrößen dar. In einem Dokument wird normalerweise genau eine `<h1>`-Überschrift als Hauptüberschrift verwendet, die weitere Gliederung des Inhalts erfolgt meist durch `<h2>` bis `<h4>`. Noch weitergehende Unterteilungen sind eher selten und können ein Zeichen dafür sein, dass Sie die Inhalte besser auf mehrere Einzeldokumente verteilen sollten.

Beachten Sie, dass die `<h...>`-Tags auf keinen Fall für die allgemeine Einstellung der Schriftgröße verwendet werden dürfen: Zum einen sind Überschriften eigenständige Absätze, zum anderen werden sie standardmäßig fett dargestellt.

Auch Überschriften kennen das Attribut `align` mit denselben Werten wie Absätze.

Als Letztes gibt es noch die spezielle Formatierung `<pre>`. Sie erzeugt im engeren Sinne keine Absätze, sondern vorformatierten Text (`pre` steht für »preformatted«). Text zwischen diesen beiden Tags wird mitsamt allen Leerzeichen und Zeilenumbrüchen aus dem HTML-Code dargestellt – interessant ist dies zum Beispiel, um Programmquelltexte auf Webseiten zu veröffentlichen, da diese auf eine sauber eingerückte Darstellung angewiesen sind. Hier ein kleines Beispiel:

```
<pre>
public class HelloWorld {
    public static void main(String args[])
    {
        system.out.println ("Hello, World!");
    }
}
</pre>
```

Auch zur Darstellung von ASCII-Art – kleinen Zeichnungen, die nur aus ASCII-Zeichen bestehen – ist vorformatierter Text geeignet. Wunderschöne künstlerische Beispiele finden Sie etwa unter www.ascii-art.de; hier folgt dagegen mein eigenes bescheidenes Werk:

```
<pre>
      o 0
      o  _
      ||_||o| _____
      |   | -\_THE\_ -\_WEB\_ -\_TRAIN\_
      /00-00\  0  0    0  0    0  0
</pre>
```

Listen und Aufzählungen

Listen und Aufzählungen sind ein gutes Mittel zum Ordnen von Informationen. Da HTML ursprünglich für die wissenschaftliche Dokumentation erfunden wurde, ist es nicht weiter verwunderlich, dass einige Möglichkeiten zur Erstellung solcher Listen angeboten werden.

Eine *nicht nummerierte Aufzählung* oder »bullet list« wird von den folgenden Tags umschlossen:

```
<ul>...</ul>
```

`ul` steht dabei für »unordered list« (englisch für unsortierte Liste).

Innerhalb dieses Bereiches werden die einzelnen Listenpunkte durch ``-Tags umschlossen:

```
<li>Einzelnformation</li>
```

li bedeutet »list item« (Listenpunkt). In HTML 4.0 ist das ein Einfach-Tag, das nur *vor* dem Text des Listeneintrags stehen muss; XHTML kennt diese Schreibweise natürlich nicht.

Das Attribut `type` kann sowohl beim - als auch beim einzelnen -Tag stehen und gibt den gewünschten Stil der Aufzählungspunkte an. Die folgenden Werte sind möglich:

- ▶ `disc` – ein gefüllter runder Punkt
- ▶ `circle` – ein Kreis, also ein innen hohler runder Punkt
- ▶ `square` – ein quadratischer Punkt

Wenn Sie das Attribut weglassen, werden verschachtelte Listen je nach Ebene nacheinander mit den drei Punktarten versehen.

Der folgende Codeabschnitt erzeugt eine Aufzählung mit vier Unterpunkten:

```
<h3>Bei XHTML zu beachten:</h3>
<ul>
  <li>XML-Header verwenden</li>
  <li>Jedes Tag schlie&szlig;en</li>
  <li>Alle Tags klein schreiben</li>
  <li>Attributwerte immer in Anf&uuml;hrungszeichen</li>
</ul>
```

Die Struktur

```
<ol>...</ol>
```

bezeichnet eine *nummerierte Liste* (»ordered list«). Auch in dieser Liste wird ein Listenpunkt durch angegeben.

Das -Tag kennt den Parameter `type`, der die Art der verwendeten Aufzählung angibt. Dabei sind folgende Werte möglich:

- ▶ "1" – arabische Ziffern (Standard)
- ▶ "A" – Großbuchstaben
- ▶ "a" – Kleinbuchstaben
- ▶ "I" – römische Zahlen
- ▶ "i" – kleingeschriebene römische Zahlen, also i, ii, iii, iv und so weiter

Das Attribut `start` bezeichnet dagegen den Wert, mit dem die Nummerierung beginnt. Unabhängig vom jeweiligen `type` muss der Wert numerisch angegeben werden.

Soll etwa eine alphabetische Liste mit D. beginnen, dann schreiben Sie den folgenden HTML-Code:

```
<ol type="A" start=4> ... </ol>
```

Nummerierte und nicht nummerierte Listen können beliebig ineinander verschachtelt werden. Dabei gehört das jeweilige `` immer zu derjenigen Liste, die zuletzt geöffnet wurde.

Der folgende Codeabschnitt erzeugt eine nummerierte Hauptliste, in die jeweils nicht nummerierte Aufzählungen verschachtelt sind:

```
<h3>Was Sie für das Surfen im Web benötigen</h3>
<ol>
  <li>Eine DFÜ-Verbindung</li>
  <ul>
    <li>DSL</li>
    <li>ISDN</li>
    <li>Modem</li>
  </ul>
  <li>Ein Account bei einem Provider, z.B.</li>
  <ul>
    <li>NetCologne</li>
    <li>Arcor</li>
  </ul>
  <li>Ein Browser</li>
  <ul>
    <li>Firefox</li>
    <li>Konqueror</li>
    <li>sonstige ...</li>
  </ul>
</ol>
```

Im Browser sieht dies schematisch betrachtet so aus (wenn auch mit anderen Aufzählungszeichen):

Was Sie für das Surfen im Web benötigen:

- ▶ Eine DFÜ-Verbindung
 - ▶ DSL
 - ▶ ISDN
 - ▶ Modem

- ▶ Einen Account bei einem Provider, z.B.
 - ▶ NetCologne
 - ▶ Arcor
- ▶ Einen Browser
 - ▶ Firefox
 - ▶ Konqueror
 - ▶ sonstige ...

Der letzte Listentyp wird von folgenden Tags umklammert:

```
<d1> ... </d1>
```

Dies bezeichnet eine sogenannte *Definitionsliste* (Definition List), auch Glossarliste genannt. In einer solchen Liste wechseln sich ausgerückte Begriffe und ihre eingerückten Definitionen ab.

Ein zu definierender Begriff steht zwischen den folgenden Tags:

```
<dt> ... </dt>
```

dt bedeutet »definition title«. Die anschließende Definition steht zwischen diesen Tags:

```
<dd> ... </dd>
```

Dies ist die Abkürzung für »definition data«.

Der folgende Code zeigt ein Beispiel:

```
<h3>Web-Sprachen</h3>
<d1>
  <dt>HTML</dt>
  <dd>Klassische Auszeichnungssprache für
  Webseiten</dd>
  <dt>XHTML</dt>
  <dd>Moderne, XML-basierte Variante von HTML</dd>
  <dt>JavaScript</dt>
  <dd>Skriptsprache zur dynamischen Manipulation von
  Webseiten-Inhalten</dd>
</d1>
```

Hyperlinks

Hyperlinks sind die wichtigsten Elemente von HTML-Dokumenten. Erst durch diese Verknüpfungen entsteht *Hypertext*, also eine Struktur verknüpfter Dokumente. Sie können sowohl auf andere Dokumente innerhalb Ihrer eigenen Web-

site als auch auf fremde Websites und andere Arten von Internet-Ressourcen wie FTP-Download-Sites, Newsgroups oder E-Mail-Adressen verweisen.

Grundsätzlich wird ein Hyperlink durch das folgende Tag gesetzt:

```
<a href="URL_des_neuen_Dokuments">Anklickbarer Text</a>
```

Innerhalb derselben Website werden meist *relative URLs* eingesetzt. Sie geben den Pfad der verknüpften Ressource relativ zum aktuellen Dokument an. Zwei Punkte als Verzeichnisname (..) bedeuten dabei genau wie im Dateisystem, dass das jeweils übergeordnete Verzeichnis angesprochen werden soll.

Hier sehen Sie ein Beispiel für Dokumente, die in einer Verzeichnishierarchie auf einem Webserver verteilt sind:

```

/      (Dokumentenwurzel der Website)
|
+-- index.html
|
+-- [kontakt]
|   |
|   +-- mail.html
|
+-- [neu]
    |
    +-- news.html

```

Im obersten Verzeichnis befindet sich die Startseite, in diesem Fall heißt sie *index.html*. Welche Namen für Startseiten akzeptiert werden, müssen Sie auf dem Webserver einstellen (siehe Kapitel 15, »LAMP«) beziehungsweise mit Ihrem Hosting-Provider abklären. Eine definierte Startseite für eine Website wird benötigt, damit Besucher einfach den Domainnamen oder einen Verzeichnisnamen aufrufen können, ohne ein bestimmtes Dokument anzufordern.

Um zum Beispiel einen Hyperlink von der Datei *index.html* im Wurzelverzeichnis auf die Datei *mail.html* im Verzeichnis *kontakt* zu erzeugen, gilt folgende Syntax (der Link-Text sei hier »E-Mail-Adressen«):

```
<a href="kontakt/mail.html">E-Mail-Adressen</a>
```

Ein Link von der Datei *mail.html* zurück zu *index.html* funktioniert folgendermaßen (Link-Text: »Homepage«):

```
<a href=" ../index.htm">Homepage</a>
```

Ein Link von *mail.html* auf *news.html* im Verzeichnis *neu* (mit dem Link-Text »Neueste Nachrichten«) lautet dann:

```
<a href="../neu/news.html">Neueste Nachrichten</a>
```

Bei Hyperlinks auf andere Server im Internet müssen Sie eine vollständige URL angeben. Als Beispiel sehen Sie hier einen Hyperlink auf die Website von Galileo Computing:

```
<a href="http://www.galileocomputing.de">Galileo Computing</a>
```

Natürlich kann ein Hyperlink nicht nur allgemein auf eine Website, sondern auch auf ein bestimmtes Dokument auf einem anderen Webserver verweisen, wenn Sie dessen URL kennen. Beispielsweise zeigt folgender Link auf die erste Seite der Präsentation meines Buches »Apache 2« bei Galileo Computing:

```
<a href="http://www.galileocomputing.de/1972">Apache 2</a>
```

Bedenken Sie jedoch, dass nicht alle Webmaster sorgfältig mit den Adressen einzelner Dokumente umgehen – sie können sich häufig ändern, sodass ein solcher Link plötzlich ins Leere zeigt und einen unangenehmen »Fehler 404« (Dokument nicht gefunden) produziert. Falls Sie auf Ihren Websites also Linklisten betreiben, sollten Sie diese regelmäßig überprüfen; hilfreich ist auch eine Feedback-Möglichkeit für User (»defekten Link melden«). Umgekehrt sollten Sie die Adressen der einzelnen Dokumente Ihrer eigenen Sites auch dann nach Möglichkeit nicht ändern (oder unter den alten URLs zumindest Hinweise und Weiterleitungen speichern), wenn Sie wesentliche Änderungen an Design oder Inhalten vornehmen – es ist für andere genauso ärgerlich, wenn ein Link auf eines Ihrer Dokumente plötzlich nicht mehr funktioniert.

Bei einer Datei, die Sie über einen Hyperlink ansprechen, muss es sich nicht unbedingt um ein HTML-Dokument handeln. Sie können etwa andere Arten von Dokumenten verlinken, mit denen Browser üblicherweise selbst zurechtkommen, beispielsweise GIF- oder JPEG-Bilder, MP3-Audiodateien oder PDF-Dokumente. Falls der Browser mit einem Dateityp nichts anfangen kann, fragt er nach, ob er die Datei dem Betriebssystem zum Öffnen mit einem geeigneten Programm übergeben oder ob er sie auf die Festplatte speichern soll.

Genauso funktioniert übrigens ein einfacher *Download-Link*: Es handelt sich einfach um einen Hyperlink auf eine Datei, die ein Browser nicht selbst anzeigen kann, etwa ein ausführbares Programm oder eine ZIP-komprimierte Datei.

Abgesehen davon können Sie nicht nur Links auf andere Websites einrichten, sondern auch auf Ressourcen, die über andere Protokolle angesprochen werden. Voraussetzung ist natürlich, dass es sich um Protokolle handelt, mit denen die Browser der meisten Benutzer umgehen können oder für die ein Protokollhilfsprogramm definiert ist, das Ressourcen dieses Typs verarbeitet. Sehr verbrei-

tet sind beispielsweise Download-Links auf Anonymous-FTP-Server, die fast jeder Browser beherrscht, und mit deren Hilfe Downloads schneller und effizienter funktionieren als über HTTP. Sehen Sie sich zum Beispiel den folgenden Link an:

```
<a href="ftp://ftp.uni-erlangen.de/pub/mirrors/opensuse/">
OpenSUSE-Download</a>
```

Es handelt sich um einen Link auf den Mirror des openSUSE-Download-Verzeichnisses auf dem FTP-Server der Universität Erlangen.

Downloads erzwingen

Wenn Sie Dokumente zum Download anbieten möchten, die der Browser eigentlich anzeigen kann – einschließlich Textdateien oder JPEG-Bildern –, können Sie sie in Archive (ZIP, GNUZip oder ähnliche) verpacken. Eine professionellere Alternative besteht in einer Änderung der Webserver-Konfiguration für das entsprechende Verzeichnis: Es geht darum, für die fraglichen Dokumente den MIME-Type `application/octet-stream` (allgemeine Binärdatei) zu erzwingen. Bei Apache bietet sich hierzu eine `.htaccess`-Datei an, falls Sie keinen Zugriff auf die Hauptkonfigurationsdatei haben.

Kopieren Sie die gewünschten Dateien in ein separates Verzeichnis innerhalb der Website. Erstellen Sie dann innerhalb dieses Verzeichnisses eine Textdatei namens `.htaccess`, und schreiben Sie folgende Zeile hinein:

```
ForceType application/octet-stream
```

Für den entsprechenden Konfigurationskontext muss die `AllowOverride`-Option `FileInfo` gesetzt sein (siehe Kapitel 15, »LAMP«). Außerdem müssen Sie sicherstellen, dass sich in dem entsprechenden Verzeichnis und allen Unterverzeichnissen *nur* solche Download-Dateien befinden.

Ein Browser, der auf eine Datei aus diesem Verzeichnis stößt, zeigt diese nun nicht mehr an, sondern bietet sie wie beschrieben zum Download an. Schwierigkeiten macht nur noch der Microsoft Internet Explorer, der gern »schlauer als der MIME-Type« ist und versucht, den Dateityp selbst zu erkennen.

Eine andere Variante ist ein *Hyperlink auf eine E-Mail-Adresse*. Wenn dieser Link angeklickt wird, öffnet der Browser automatisch ein Mail-Fenster im bevorzugten E-Mail-Programm des Anwenders, und trägt die angegebene Adresse bereits als Empfänger ein. Hierzu dient das »Pseudo-Protokoll« `mailto:`:

```
<a href="mailto:ich@mein-provider.de">Senden Sie mir eine Mail</a>
```

Sie können sogar den Betreff der neu zu erzeugenden E-Mail angeben:

```
<a href="mailto:ich@mein-provider.de?subject=Feedback">
Geben Sie mir Feedback!</a>
```

Der Link erzeugt wiederum ein Mail-Fenster für eine neue E-Mail, in das sowohl der Empfänger als auch der Betreff bereits eingetragen sind.

Bei sehr umfangreichen Seiten, deren Länge sich über mehrere Bildschirmhöhen erstreckt, bietet es sich an, einzelne Passagen der Seite per Hyperlink erreichbar zu machen. Zu diesem Zweck werden sogenannte *Anker* oder *Textmarken* gesetzt, die einen bestimmten Punkt im Dokument als Ziel für einen Hyperlink definieren. Beim Anklicken eines Links, der auf einen Anker verweist, wird der Inhalt des Browserfensters automatisch so gescrollt, dass sich die entsprechende Stelle am oberen Rand befindet.

Die Syntax für einen Ankerpunkt lautet folgendermaßen:

```
<a name="name_des_ankers"></a>
```

Gemäß HTML-Spezifikation ist dieses Tag leer. Dennoch werden in manchen HTML-Referenzen Inhalte wie Text oder Überschriften dazwischengesetzt, wovon dringend abzuraten ist. In jedem Fall sollten Sie das Tag *nicht* mit dem XML-End-Slash schließen, sondern das `` explizit hinschreiben – auch ältere Browser erwarten, dass es geschlossen wird, verstehen aber die XML-Kompakt-schreibweise nicht.

Ein Hyperlink auf einen solchen Ankerpunkt erfolgt innerhalb derselben Datei mithilfe der folgenden Syntax:

```
<a href="#name_des_ankers">Zum Ankerpunkt</a>
```

Sie können auch aus einer anderen Datei heraus direkt auf die entsprechende Dokumentstelle zugreifen:

```
<a href="Dateiname#name_des_ankers">Zum Ankerpunkt</a>
```

Es versteht sich von selbst, dass Dateinamen in Websites aus diesem Grund kein #-Zeichen enthalten dürfen!

Auf sehr vielen Websites ist es gängige Praxis, ganz oben auf jede Seite unmittelbar nach dem `<body>`-Tag einen Anker zu setzen – beispielsweise folgendermaßen:

```
<a name="oben"></a>
```

Am unteren Rand des Dokuments befindet sich dann entsprechend ein Hyperlink, um wieder ganz nach oben zu gelangen:

```
<a href="#oben">nach oben</a>
```

Dies bietet sich insbesondere dann an, wenn sich die Hauptnavigation jeweils ganz oben in den Dokumenten befindet.

Bilder in Webseiten einbetten

Ein Bild ist stets eine externe Datei, die erst zum Zeitpunkt des Ladens dynamisch vom Browser eingebettet wird. Bilder können in den komprimierten Dateiformaten GIF, JPEG oder PNG vorliegen. JPEGs sind im Allgemeinen besser für Fotos und andere halbtone Bilder geeignet, während sich die beiden anderen Formate eher für flächige Grafiken mit wenigen Farben eignen (Näheres siehe Kapitel 8, »Desktop-Software«).

Grundsätzlich wird ein Bild mithilfe des folgenden Tags in eine Webseite eingebettet:

```

```

Für die Angabe der URLs von lokalen oder auf anderen Servern liegenden Bildern gelten die Regeln, die weiter oben bereits für Hyperlinks erläutert wurden.

Für Suchmaschinen, sehbehinderte Benutzer oder reine Textbrowser ist es praktisch, dass sich für jedes Bild ein Alternativtext angeben lässt:

```

```

In manchen Browsern wird dieser Text auch als Tooltip angezeigt, wenn Sie mit der Maus über das Bild fahren – hierzu ist allerdings das Attribut `title` besser geeignet, das Sie für beliebige Tags verwenden können.

Geben Sie im Tag stets die Originalgröße des Bildes an – die Syntax für die entsprechenden Attribute sieht so aus:

```

```

Der vorher geladene Text kann dann bereits dargestellt werden, da klar ist, wie er positioniert werden soll. Eine Skalierung des Bildes über die Attribute `height` und `width` ist hingegen nicht zu empfehlen – wenn auch grundsätzlich möglich. Stattdessen sollte das Bild in einem Bildbearbeitungsprogramm von vornherein auf die gewünschte Größe gebracht werden.

Über den `align`-Parameter können Bilder positioniert werden:

```

```

setzt das Bild an den linken Rand des Fensters; der Text umfließt rechts den rechteckigen Umriss des Bildes.

```

```

platziert das Bild am rechten Rand; der Text umfließt links den Umriss des Bildes.

Bei diesen beiden Formatierungen können Sie das Umfließen jederzeit beenden und weiteren Text unter dem Bild weiterführen, indem Sie ein `
` mit dem speziellen Attribut `clear` einfügen:

- ▶ `<br clear="left" />` beendet nur linksseitiges Umfließen,
- ▶ `<br clear="right" />` hebt rechtsseitiges Umfließen auf, und
- ▶ `<br clear="all" />` beendet beides.

Die Attributwerte "top", "middle" und "bottom" setzen das Bild in den normalen Fluss einer Textzeile, wobei der Text entsprechend an die Oberkante des Bildes, auf die Mittellinie oder an die Unterkante gesetzt wird. Andere Angaben wie "absmiddle" oder "baseline", die Sie manchmal in HTML-Dokumenten oder in der Literatur finden, gehören nicht zum HTML-Standard, sondern werden nur von bestimmten Browsern interpretiert.

Mithilfe der Attribute `hspace="Anzahl_Pixel"` und `vspace="Anzahl_Pixel"` können Sie den Abstand des Bildes nach links und rechts beziehungsweise nach oben und unten festlegen.

Wenn Sie ein Bild als Hyperlink definieren möchten, erreichen Sie dies folgendermaßen:

```
<a href="URL"></a>
```

Hierbei wird um das Bild standardmäßig ein Rahmen in der aktuellen link-beziehungsweise vlink-Farbe gezogen. Durch die Angabe von

```

```

lässt sich dies vermeiden.

Anstatt ein ganzes Bild zu einem durchgehenden Hyperlink zu machen, besteht auch die Möglichkeit, einzelne Bereiche des Bildes anklickbar zu machen und jeweils individuell auf einen Klick in die jeweiligen Regionen zu reagieren. Grundsätzlich werden zwei verschiedene Arten dieser sogenannten Image-Maps unterschieden: die nur noch selten verwendeten *serverseitigen Image-Maps* und die *clientseitigen Image-Maps*.

Bei einer serverseitigen Image-Map wird einfach das Attribut `ismap` in das ``-Tag gesetzt; in HTML 4.0 ohne Wert, in XHTML mit einem beliebigen – die beste Wahl ist `ismap="ismap"`. Das Bild wird als normaler Hyperlink auf ein serverseitiges Skript verwendet, das in der Lage ist, die übertragenen Mauskoordinaten

des Bildes zu verarbeiten. Beispielsweise übergibt der folgende Link die Koordinaten an das Perl-CGI-Skript *coords.pl* im Verzeichnis */cgi-bin*:

```
<a href="/cgi-bin/coords.pl"></a>
```

Die Koordinaten werden einfach an die URL angehängt, es wird also statt */cgi-bin/coords.pl* die URL */cgi-bin/coords.pl?100,100* aufgerufen, wenn der Benutzer genau in der Bildmitte klickt.

Das folgende kleine Perl-Skript erzeugt ein HTML-Dokument, das einfach die Koordinaten ausgibt:

```
#!/usr/bin/perl -w
use strict;
my $coords = $ENV{'QUERY_STRING'};
my ($x, $y) = split (/,/, $coords);
print "Content-type: text/html\n\n";
print "<html><head><title>Koordinaten</title></head>";
print "<body>Koordinaten: x=$x, y=$y</body></html>\n";
```

Bessere Dienste als ein selbstgeschriebenes Skript leistet übrigens das Apache-Standardmodul *mod_imagemap* (bis Version 2.0 *mod_imap*); genau wie die clientseitigen Image-Maps ermöglicht es die Definition von Bereichen mit verschiedenen Formen.

Bei einer *clientseitigen Image-Map* wird ein Bild mit beliebig vielen anklickbaren Bereichen versehen (sogenannten *Hot Spots*), die jeweils getrennt als Hyperlinks angezeigt werden sollen. Eine solche Image-Map wird durch die Angabe des Attributs *usemap="#Map-Name"* erzeugt. Mit diesem Map-Namen muss eine *<map>*-Definition übereinstimmen, die die einzelnen Hot Spots definiert. Das Bild selbst muss (und sollte) dabei kein Hyperlink sein. Eine solche Map-Definition sieht zum Beispiel folgendermaßen aus:

```

<map name="mymap">
  <area shape="rect" coords="10, 10, 100, 100"
  href="rechteck.htm" alt="Rechteckiger Bereich" />
  <area shape="circle" coords="300, 300, 50"
  href="kreis.htm" alt="Kreisförmiger Bereich" />
  <area shape="poly" coords="150, 150, 200, 10, 250, 150"
  href="polygon.htm" alt="Polygonförmiger Bereich" />
</map>
```

Die einzelnen `<area>`-Tags definieren die anklickbaren Bereiche. Das Attribut `shape` gibt die Form des jeweiligen Bereichs an:

- ▶ `shape="rect"`
Gibt mithilfe des Attributs `coords` die linke obere und die rechte untere Ecke eines Rechtecks (in Pixeln) an.
- ▶ `shape="circle"` (Kreis)
Verlangt als `coords` die Angabe des Mittelpunkts (X- und Y-Koordinate) und den Radius in Pixeln.
- ▶ `shape="poly"` (Polygon)
Erwartet eine Reihe aufeinanderfolgender X- und Y-Koordinaten.

`href` definiert wie bei einem Hyperlink die Adresse, auf die ein Mausklick in diesen Bereich verzweigen soll. `alt` erzeugt wie gehabt Alternativtexte und manchmal Tooltips.

Tabellen

Eines der wichtigsten Gestaltungsmittel für Webseiten sind HTML-Tabellen. Früher wurden sie vor allem zur Verteilung beliebiger Inhalte im Dokument verwendet. Heute sollten Sie sie, ihrem eigentlichen Zweck gemäß, nur noch für die tabellarische Anordnung von Informationen verwenden. Das Layout sollte dagegen mithilfe von CSS-Bereichen (siehe unten) erstellt werden.

Eine Tabelle steht im HTML-Dokument zwischen den beiden folgenden Tags:

```
<table> ... </table>
```

Tabellen bestehen aus mehreren Zeilen, die in einzelne Zellen aufgeteilt werden; der eigentliche Inhalt darf nur innerhalb der einzelnen Zellen stehen:

```
<table>
  <tr>
    <td>
      <!-- Hier steht der Zellinhalt -->
    </td>
    <!-- eventuell weitere Zellen -->
  </tr>
  <!-- eventuell weitere Zeilen -->
</table>
```

Die Zeilen mit den `<!-- ... -->`-Bereichen sind übrigens HTML-Kommentare. Sie können sich über beliebig viele Zeilen erstrecken. Bedenken Sie, dass sie zwar vom Browser ignoriert werden, dass aber jeder Benutzer den Quellcode einer Webseite einsehen und damit auch die Kommentare lesen kann.

Zwischen `<tr>` und `</tr>` (Abkürzung für »table row«) steht eine einzelne Zeile, die aus einer beliebigen Anzahl von Zellen bestehen kann. Natürlich sollten alle Zeilen einer Tabelle in der Regel gleich viele Zellen besitzen, anderweitig sind keine Spalten definiert. Dennoch sorgt der Browser dafür, dass untereinanderliegende Zellen mit der gleichen Breite gezeichnet werden.

Die einzelne Tabellenzelle steht innerhalb der Tags `<td> ... </td>` (`td` bedeutet »table data«). Nur zwischen diesen beiden Tags dürfen die eigentlichen Tabelleninhalte stehen. Für *Überschriftzellen* mit Spalten- oder Zeilenüberschriften existiert auch das spezielle Tag `<th> ... </th>` (»table heading«), das an den entsprechenden Stellen statt der `<td>`-Tags gesetzt werden kann und den Text der Zelle in der Regel fett und in der Zelle zentriert setzt.

Zu guter Letzt existiert noch das optionale Tag `<caption> ... </caption>`, das nicht in eine Zeile oder Zelle gesetzt wird, sondern unmittelbar hinter das öffnende `<table>`-Tag. Es handelt sich um ein Tag zur Beschriftung der Tabelle. Sein einziges Attribut `align` kann die Werte "top" (über der Tabelle, der Standardwert) oder "bottom" (unter der Tabelle) annehmen.

Der folgende Codeausschnitt zeigt ein vollständiges Beispiel:

```
<table border="2">
  <caption>Bestandteile von HTML-Tabellen</caption>
  <tr>
    <th>Element</th>
    <th>Erklärung</th>
  </tr>
  <tr>
    <th>table</th>
    <td>Die eigentliche Tabelle</td>
  </tr>
  <tr>
    <th>tr</th>
    <td>Tabellenzeile</td>
  </tr>
  <tr>
    <th>td</th>
    <td>Tabellenzelle</td>
  </tr>
</table>
```

Das im `<table>`-Tag verwendete Attribut `border` sorgt dafür, dass ein sichtbarer Tabellenrahmen gezeichnet wird – hier mit einer Breite von zwei Pixeln.

Tabellen-Attribute

Es gibt verschiedene Attribute, die die einzelnen Tabellenbestandteile auf verschiedene Art und Weise einrichten. Beachten Sie, dass die meisten von ihnen (`align`, `border` und dergleichen) besser mithilfe von CSS gesetzt werden sollten. Zunächst werden hier die Attribute des `<table>`-Tags behandelt:

- ▶ `align="left"|"right"|"center"`
Dieses Attribut richtet die Tabelle im Fenster aus. Bei "left" und "right" wird sie wie ein entsprechend formatiertes Bild von Text umflossen; "center" setzt sie absatzbildend in die Fenstermitte. Wenn Sie gar kein `align` angeben, wird die Tabelle absatzbildend nach links gesetzt.
- ▶ `width="..."|"...%"`
Mit diesem Attribut wird die Breite der Tabelle in Pixeln beziehungsweise in Prozent der Fensterbreite angegeben. Der Wert wird ignoriert, falls die Inhalte der Tabelle nicht in die angegebene Breite hineinpassen.
- ▶ `height="..."|"...%"`
`height` bestimmt die Höhe der Tabelle in Pixeln beziehungsweise in Prozent der Fensterhöhe an. Auch dieser Wert wird ignoriert, falls die Inhalte der Tabelle sonst nicht passen.
- ▶ `border="..."`
Mit diesem Attribut setzen Sie die Breite des Tabellenrahmens in Pixeln. Beachten Sie, dass in modernen Browsern ohne Angabe des `border`-Parameters eine Tabelle ohne Rand erzeugt wird. Da dies bei älteren Browsern jedoch nicht so sein muss, sollte für eine unsichtbare Layout-Tabelle immer ausdrücklich `border="0"` gesetzt werden.
- ▶ `cellpadding="..."`
Gibt die Entfernung des jeweiligen Zellinhaltes zum Zellrand in Pixeln an.
- ▶ `cellspacing="..."`
Gibt die Breite der Zellränder in Pixeln an.

Das `<td>`-Tag unterstützt eine ganze Reihe von Attributen, die hier zum großen Teil aufgelistet werden.

- ▶ `align="center"|"right"|"left"|"justify"`
Dieses Attribut richtet den Inhalt der Zelle horizontal aus (linksbündig, rechtsbündig, zentriert, Blocksatz). Standard ist die linksbündige Ausrichtung.
- ▶ `valign="middle"|"top"|"bottom"`
Richtet den Zellinhalt vertikal aus (oben, mittig, unten). Standard ist die mittige Ausrichtung.

- ▶ `colspan="..."`
Hier wird festgelegt, dass sich diese Zelle über die angegebene Anzahl von Spalten der Tabelle erstrecken soll; natürlich gibt es dann in der entsprechenden Zeile der Tabelle weniger Zellen.
- ▶ `rowspan="..."`
Gibt an, dass sich die Zelle über die angegebene Anzahl von Zeilen der Tabelle erstrecken soll. Die Definition erfolgt in der obersten Zeile, in der die Zelle beginnt; in allen betroffenen Zeilen braucht die betreffende Zelle nicht mehr definiert zu werden.
- ▶ `width="..."|"...%"`
Setzt die Zellbreite in Pixeln beziehungsweise relativ zur gesamten Tabelle. Wird die Breite jeder Zelle in Pixeln angegeben, so sollte eine Angabe der Tabellenbreite unterbleiben. Es genügt übrigens, wenn Sie diese Angaben einmal in den Zellen der obersten Zeile vornehmen (alternativ verstehen neuere Browser auch die im nächsten Unterabschnitt vorgestellten `<colgroup>`-Tags). Generell muss zu `width`-Angaben noch gesagt werden, dass die Breitenangabe ignoriert wird, wenn der Inhalt einer Zelle in der jeweiligen Spalte zu breit ist, um in der gewünschten Zellbreite angezeigt zu werden.
- ▶ `height="..."|"...%"`
Stellt die Höhe der Zelle in Pixeln beziehungsweise in Prozent der Tabellenhöhe ein. Auch hier wird die vollständige Darstellung des Inhalts gegenüber der Höhenangabe bevorzugt.
- ▶ `nowrap="nowrap"`
Dieses Attribut verhindert Zeilenumbrüche innerhalb einer Zelle. Es funktioniert in fast jedem Browser, gilt aber als veraltet und wird durch eine entsprechende Stylesheet-Angabe ersetzt.

Inhalt einer Zelle können beliebige Elemente sein, die HTML unterstützt, also Texte, Bilder und Hyperlinks. Natürlich kann ein `<td> ... </td>` auch wieder eine weitere Tabelle enthalten.

Formulare

Suchmaschinen, Bestellseiten oder Webforen bestehen aus diversen Eingabeelementen wie Textfeldern oder Auswahlknöpfen. Der übergeordnete Container für eine Gruppe solcher interaktiver Elemente ist das *HTML-Formular*. Seine Definition enthält eine URL, an die sämtliche eingegebenen Daten versandt werden. Formulare bilden also das Frontend webbasierter Anwendungen.

Jedes Formular steht zwischen den folgenden Tags:

```
<form action="URL" method="Versandmethode">
  <!-- beliebiges HTML mit Formularelementen -->
</form>
```

Als *Action-URL* wird in aller Regel die Adresse eines Server-Skripts angegeben. Eine verbreitete Technik für solche Skripte, PHP, wird in Kapitel 15, »LAMP«, beschrieben.

Das Attribut `method` gibt an, über welche HTTP-Methode die Daten versendet werden sollen (siehe ebenfalls Kapitel 15): "post" versendet sie separat und sorgt dafür, dass sie vom Server so interpretiert werden, als würden sie dort direkt per Tastatur eingegeben (Standardeingabe). "get" hingegen hängt die Formulardaten direkt an die URL an, in der Form `url?daten`. Beispielsweise verwenden Suchanfragen an Suchmaschinen oft diese Form, sodass zum Beispiel die URL einer Suche nach dem Begriff »html« bei Google (in minimaler Form) folgendermaßen aussieht:

```
http://www.google.de/search?q=html
```

Der Parameter `q=html` wird also an das Programm `search` übermittelt. Hinter den Kulissen befragt dieses Programm daraufhin Googles Datenbank, nimmt die Ergebnisse entgegen und generiert wiederum eine HTML-Seite daraus, die es an den Browser des Anfragenden zurücksendet.

Der Vorteil der Methode "get" besteht darin, dass Sie die URL in der Favoritenliste Ihres Browsers ablegen können. Allerdings ist sie nicht für größere Datenmengen geeignet, da die Maximallänge für URLs 2.000 Zeichen beträgt.

Ein weiteres (optionales) Attribut ist `enctype`; es gibt den MIME-Type der übertragenen Formulardaten an. Der Standardwert ist "application/x-www-form-urlencoded"; dieses Format ersetzt in den Formulardaten alle Zeichen, die in URLs nicht gestattet sind oder in Formulardaten eine Sonderbedeutung haben, durch passende Escape-Sequenzen: Leerzeichen werden zu `+`-Zeichen; die meisten Satzzeichen und alle Nicht-ASCII-Zeichen werden durch ein `%`-Zeichen und ihren zweistelligen hexadezimalen Code im Zeichensatz ersetzt. Falls Sie den Datei-Upload über ein Formular (siehe Kapitel 15, »LAMP«) ermöglichen möchten, müssen Sie den Typ "multipart/form-data" wählen.

Innerhalb des Containers `<form> ... </form>` können Sie eine Reihe unterschiedlicher *Eingabeelemente* definieren. Die meisten von ihnen verwenden das `<input>`-Tag:

```
<input type="Elementtyp" name="Elementname"
value="Wert" />
```

`type` gibt an, um welche Art von Eingabeelement es sich überhaupt handelt – beispielsweise ein Textfeld (`type="text"`) oder ein Absendeknopf (`type="submit"`). `name` und `value` bilden ein Name/Wert-Paar, das als Name=Wert mit den Formular-daten versandt wird. Je nach konkretem Eingabeelement kommt das eine oder andere Spezialattribut hinzu.

- ▶ `<input type="radio" name="..." value="..." />`
 Definiert einen sogenannten *Radio-Button*. Der Name stammt daher, dass dieser Button sich so verhält wie die Knöpfe an alten Radios: Wird einer von ihnen gedrückt, springt der zuvor ausgewählte automatisch heraus. Mit Radio-Buttons können Sie dem Benutzer die Auswahl einer einzigen Option aus mehreren Alternativen ermöglichen. `name` bezeichnet dabei den Namen der Gruppe, zu der der Radio-Button gehört. `value` ist der Inhalt, der als Auswahl für diese Gruppe in den Formular-daten erscheinen soll. Der Text, mit dem der Button beschriftet werden soll, wird einfach hinter das Tag gesetzt.
- ▶ `<input type="checkbox" name="..." value="..." />`
 Definiert eine *Checkbox*, mit der Benutzer mehrere Optionen an- und wieder abwählen können. Der optionale Parameter `checked="checked"` kann bei Checkboxes und Radiobuttons stehen und hat zur Folge, dass die entsprechende Option innerhalb ihrer Gruppe bereits ausgewählt ist.
- ▶ `<input type="text" name="..." size="..." maxlength="..." />`
 Bietet ein Feld zur Texteingabe; optional gibt `size` die Breite in Zeichen und `maxlength` die maximale Eingabelänge an. Der ebenfalls optionale Parameter `value="..."` sorgt dafür, dass der betreffende Text bereits voreingetragen im Textfeld steht.
- ▶ `<input type="password" name="..." size="..." maxlength="..." />`
 Funktioniert im Prinzip genau wie ein Textfeld – mit dem Unterschied, dass die Eingabe als »*****« angezeigt wird.
- ▶ `<input type="submit" value="..." />`
 Stellt einen Button zur Verfügung, der durch Mausklick den Inhalt des Formulars an die URL versendet, die im `<form>`-Tag angegeben wurde. `value` hat hier eine etwas andere Bedeutung: Es enthält den Text, mit dem der Button beschriftet wird.
- ▶ `<input type="reset" value="..." />`
 Setzt alle Einträge, die der Benutzer im Formular vorgenommen hat, auf den Ursprungszustand zurück. `value` enthält wiederum die Beschriftung des Buttons.
- ▶ `<input type="button" value="..." />`
 Stellt eine allgemeine Schaltfläche zur Verfügung, deren Aussehen einem

Absende- oder Lösch-Button entspricht. In Zusammenarbeit mit JavaScript kann sie ein benutzerdefiniertes Ereignis auslösen.

- ▶ `<input type="hidden" name="..." value="..." />`
Dies ist im engeren Sinne kein Eingabefeld, denn es wird vom Browser nicht angezeigt und Benutzer können keine Eingabe vornehmen. Es handelt sich um eine festgelegte Angabe, die zusammen mit den vom Benutzer eingegebenen Formulardaten übertragen wird. Nützlich sind Hidden-Felder etwa für Ordnungszwecke (welches Formular wurde eigentlich versandt?) oder für Zwischenwerte, die von serverseitigen Programmen generiert wurden und wieder mit den Daten verschickt werden müssen, um vom nächsten Skript aus darauf zurückzugreifen.
- ▶ `<input type="file" name="..." />`
Bietet die Möglichkeit, den Pfad einer lokalen Datei einzutippen oder über einen mitgelieferten Button interaktiv auszuwählen. Wenn das Empfängerskript über eine entsprechende Einrichtung verfügt, wird die angegebene Datei zusammen mit den anderen Formulardaten hochgeladen. In Kapitel 15, »LAMP«, finden Sie ein entsprechendes PHP-Skript mit weiteren Hinweisen. Beachten Sie, dass der Formulardatatype über das `<form>`-Attribut `enctype` auf `"multipart/form-data"` gesetzt werden muss, damit es funktioniert.

Neben den diversen `<input>`-Varianten gibt es noch weitere Formularelemente, die über eigenständige Tags gebildet werden.

Außer Checkboxes und Radiobuttons kann ein Formular auch noch Menüs enthalten, aus denen der Benutzer auswählen kann. Diese Menüs haben in HTML folgende Syntax:

```
<select name="..." size="...">
  <option value="..."> 1. Auswahlmöglichkeit </option>
  <option value="..."> 2. Auswahlmöglichkeit </option>
  <!-- bei Bedarf weitere Optionen -->
</select>
```

Diese Struktur stellt ein Menü zur Verfügung, das die verschiedenen Auswahlmöglichkeiten enthält. Die zusätzliche Option `multiple="true"` bei `<select>` ermöglicht die Auswahl mehrerer Felder – allerdings muss dabei die Taste `[Strg]` festgehalten werden, worauf Sie die Benutzer hinweisen sollten.

`name` enthält wiederum den Namen, mit dem der übertragene Wert in den Formulardaten versehen wird. `size` gibt die Anzahl der sichtbaren Zeilen an; wird diese von der Anzahl der Optionen überschritten, erscheint automatisch ein Rollbalken. Der von den meisten Autoren verwendete Sonderfall `size="1"` erzeugt ein relativ elegantes Pulldown-Menü.

Das `<option>`-Tag bestimmt durch das Attribut `value` wiederum den zu übertragenden Wert für die Formulardaten. Darüber hinaus kennt es den zusätzlichen Parameter `selected="selected"`, der innerhalb eines Select-Menüs nur auf ein einziges `<option>`-Tag angewandt werden sollte und dafür sorgt, dass die entsprechende Option vorausgewählt ist.

Die Texteingabe in ein Formular ist auch mehrzeilig möglich, und zwar über das folgende Tag:

```
<textarea name="..." rows="..." cols="..."></textarea>
```

Dieses Tag stellt ein Eingabefeld mit der durch `cols` angegebenen Breite und der mittels `rows` eingestellten Höhe zur Verfügung. Sollte zwischen dem öffnenden und dem schließenden Tag Text stehen, so erscheint er als vorgefertigter Eintrag im Feld. Auch wenn Sie keinen Text hineinschreiben, sollten Sie das Tag explizit und nicht per XML-End-Slash schließen, weil ältere Browser ihn nicht verstehen und in diesem Fall einfach den Rest des Dokuments in das Textfeld schreiben.

Der optionale Parameter `wrap` gibt die Art des Zeilenumbruchs an, der verwendet werden soll:

- ▶ `wrap="none"`
Diese Wertzuweisung setzt keine automatischen Zeilenumbrüche; Benutzer können beliebig in derselben Zeile weiterschreiben (Standard bei älteren Browsern; funktioniert bei moderneren oft gar nicht mehr).
- ▶ `wrap="virtual"`
Bricht am rechten Feldrand um, jedoch nur in der Darstellung; der gespeicherte Text enthält keine Zeilenumbrüche (Standard bei den meisten aktuellen Browsern).
- ▶ `wrap="physical"`
Mit dieser Anweisung werden dagegen Zeilenumbrüche in den eigentlichen Text gesetzt, wenn er am rechten Feldrand umgebrochen wird.

Meta-Tags und Suchmaschinen

Im Head eines HTML-Dokuments können Sie neben dem Titel – übrigens der wichtigsten Information für Suchmaschinen – eine Reihe zusätzlicher Informationen unterbringen, von denen einige für den Browser und andere für Suchmaschinen gedacht sind. Die meisten dieser Informationen stehen in sogenannten `<meta>`-Tags. Der Aufbau dieser Tags ist folgender:

```
<meta name="..." content="..." />
```

oder

```
<meta http-equiv="..." content="..." />
```

Die Attribute `name` beziehungsweise `http-equiv` geben den Namen oder die Kategorie einer zu definierenden Information an, `content` enthält ihren Wert. Der Unterschied zwischen der `name`- und der `http-equiv`-Variante besteht darin, dass eine `http-equiv`-Information (»http equivalent«) einem HTTP-Header-Feld entspricht. In der Regel handelt es sich um wichtige Verarbeitungsoptionen für den Browser. Eine der wichtigsten Informationen dieses Typs gibt den MIME-Type und den Zeichensatz der Webseite an und überschreibt den Standardwert, den der Webserver liefert:

```
<meta http-equiv="content-type" content="text/html;
charset=iso-8859-1" />
```

Andere Zeichensätze sind beispielsweise `iso-8859-9` für türkischen Text oder `iso-8859-7` für Griechisch.

Eine weitere häufig verwendete Variante ist das automatische erneute Laden der aktuellen Seite beziehungsweise die Weiterleitung auf eine andere. Dazu wird folgendes Tag gesetzt:

```
<meta http-equiv="refresh" content="10" />
```

Unter `content` wird die Anzahl der Sekunden angegeben, nach der die Seite erneut geladen werden soll. Nützlich ist dies bei Dokumenten, deren Inhalte häufig durch serverseitige Informationen aktualisiert werden, beispielsweise Sportergebnisse oder Börsenkurse.

Die folgende Variante lädt statt der bisherigen Seite nach fünf Sekunden das Dokument *test.htm*:

```
<meta http-equiv="refresh" content="5;url=test.htm" />
```

Dies ist beispielsweise nützlich, wenn die eigene Site auf eine neue Adresse umgezogen ist, die bekannt gemacht werden soll.

Was Meta-Tags für Suchmaschineninformationen angeht, müssen Sie zunächst wissen, welche Informationen Suchmaschinen überhaupt auswerten. Die wichtigste Angabe ist, wie bereits erwähnt, der weiter oben beschriebene Dokumenttitel:

```
<title>...</title>
```

Weiterhin sind einige Meta-Tags sehr wichtig:

```
<meta name="description" content="Kurze Beschreibung der Website/
einzelnen Seite" />
```

Bei Frame-Sites können Sie aus bereits erläuterten Gründen nur die Startseite mit solchen Informationen versehen; hier sollte das Tag die gesamte Site beschreiben. Bei framelosen Websites sollte jeweils ein kurzer Hinweis auf die gesamte Website und anschließend eine Beschreibung des jeweiligen Dokuments erfolgen. Die meisten Suchmaschinen werten etwa 150 bis 200 Zeichen aus.

Mit dem folgenden Meta-Tag können Sie eine Reihe relevanter Schlüsselwörter angeben, die vor allem als Suchbegriffe für die Site dienen sollen:

```
<meta name="keywords" content="..." />
```

Ein ungefährender Richtwert sind etwa 20 bis 50 Schlüsselwörter. Wichtig ist, dass die meisten Suchmaschinen keine Wiederholungen dulden. Ab der dritten Wiederholung werden Sie womöglich sogar aus der Datenbank einer Suchmaschine entfernt.

Die Begriffe »Künstler, Künstlerin, Künstlerbedarf« stellen beispielsweise keine Wiederholung dar. Verboten wäre dagegen die Liste »Künstler, künstler, KÜNSTLER«. Sie sollten die Schreibweise »Künstler« wählen, denn bei vielen Suchmaschinen funktioniert die Suchbegriffseingabe nach dem folgenden Schema: Kleinschreibung findet alle Varianten, gemischte Groß-/Kleinschreibung findet genau die eingegebene Variante.

Sie sollten die folgenden Arten von Schlüsselwörtern verwenden:

- ▶ Angaben über das Tätigkeitsfeld, den Zweck oder das Interessengebiet der Website
- ▶ geografische Angaben wie Gemeinde oder Bundesland
- ▶ Namen wichtiger Personen

Vermeiden sollten Sie dagegen die folgenden Arten von Wörtern:

- ▶ allgemeine Wörter, Füll- und Flickwörter
- ▶ Wörter, die zwar viele Hits bringen, aber nichts mit der eigenen Site zu tun haben
- ▶ Markennamen, insbesondere der Konkurrenz – sie können zu rechtlichen Problemen führen

Der in älteren Dokumentationen oft empfohlene Text in Hintergrundfarbe ist für viele Suchmaschinen ebenfalls ein guter Grund zum Ignorieren. Auch die schnelle automatische Weiterleitung über mehrere Pseudo-Tunnelseiten hinweg, die vermeintlich das Unterbringen von mehr relevantem Text ermöglicht, wird oft nicht toleriert.

Das Meta-Tag `robots` steuert den Zugriff von Suchmaschinen auf die aktuelle Seite sowie die Beachtung von Hyperlinks:

```
<meta name="robots" content="[no]index, [no]follow" />
```

`index` bedeutet, dass die aktuelle Seite indiziert werden soll, während `noindex` die Seite ignoriert. `follow` folgt automatisch den Hyperlinks auf der aktuellen Seite, `nofollow` folgt den Links nicht. Bei allen normalen Websites sollte auf jeder öffentlichen Einzelseite `index`, `follow` verwendet werden.

Das `robots`-Meta-Tag wird nicht von allen Suchmaschinen unterstützt. Zusätzlich sollte im obersten Verzeichnis der Site die Textdatei `robots.txt` liegen, die folgende Angaben enthalten kann:

- ▶ `User-agent` ist der Name, mit dem sich die Suchmaschinensoftware beim Server meldet. In der Regel wird `User-agent: *` verwendet, was für alle Suchmaschinen steht.
- ▶ `Disallow: /verzeichnis` bedeutet, dass alle Dateien und Unterverzeichnisse unterhalb von `/verzeichnis` für Suchmaschinen gesperrt sind.

Zu guter Letzt sollten Sie daran denken, dass selbst die besten Meta-Tags nicht gegen inhaltsreichen und gut strukturierten Text ankommen. Bild- oder flash-lastige Seiten haben es im Ranking der Suchmaschinen relativ schwer.

Nachdem Sie die genannten Informationen eingetragen haben, müssen Sie sich bei den einzelnen Suchmaschinen anmelden. Dabei ist zwischen echten Suchmaschinen wie Google, AltaVista oder Lycos und redaktionell betreuten Katalogen wie Yahoo! oder web.de zu unterscheiden.

Bei den echten Suchmaschinen gibt es irgendwo einen Link wie »Seite anmelden« oder Ähnliches. Sie müssen hier lediglich die URL der Website und eventuell eine E-Mail-Adresse für Rückfragen eingeben. In den Tagen nach der Anmeldung besucht die Suchmaschine die Website und indiziert sie gemäß des `robots`-Meta-Tags und der Datei `robots.txt`.

Bei den Katalogen müssen Sie in der Regel ein umfangreiches Formular ausfüllen. Der Katalogbetreiber entscheidet daraufhin selbst, ob die Site aufgenommen wird oder nicht.

11.2.2 Cascading Stylesheets (CSS)

CSS sind kein integraler Bestandteil von HTML, gehören aber zu den Neuerungen, die das W3C bereits seit der HTML-4.0-Spezifikation empfiehlt und fördert. Durch Stylesheets wird die Struktur eines HTML-Dokuments vollständig von sei-

nem Inhalt getrennt. Dies macht die Dokumente einerseits erheblich übersichtlicher, weil die unzähligen layoutorientierten Tags entfallen, und ermöglicht zum anderen die saubere Definition des Layouts an zentraler Stelle.

Konkret geht es darum, Stilvorlagen für das Aussehen beliebiger HTML-Elemente zu definieren. Das bedeutet, dass Sie mithilfe von CSS sehr konkret in das Layout einer Webseite oder einer ganzen Site eingreifen können, indem Sie etwa festlegen, dass alle Überschriften vom Typ `<h1> ... </h1>` rot, zentriert und in kursiver 36-Punkt-Helvetica erscheinen sollen.

Die Arbeit mit Stylesheets hat folgende Vorteile:

- ▶ Die Erzeugung eines immer wiederkehrenden, einheitlichen Layouts wird erheblich vereinfacht – einmal festgelegt, können bestimmte Formatierungen automatisch wiederholt werden.
- ▶ Es ist mithilfe von Stylesheets erstmals möglich, Elemente auf einer Webseite pixelgenau zu platzieren – im Extremfall sogar übereinander. Auf diese Weise können Sie beispielsweise Text mit Schatten oder Text auf einem Bild realisieren.
- ▶ In Zusammenarbeit mit JavaScript können Elemente, die CSS zur Stilfestlegung verwenden, dynamisch in ihren Eigenschaften verändert werden. In diesem Zusammenhang spricht man gern von DHTML (Dynamic HTML). Das W3C empfiehlt dazu die Verwendung eines allgemeinen Objektmodells, das alle Elemente des Browsers und des HTML-Dokuments standardisiert (DOM, Document Object Model).

Platzieren von Stylesheets

Stylesheet-Angaben können an verschiedenen Stellen im Dokument oder in einer eigenen Datei stehen. Sie haben folgendes Schema:

```

selektor {
  attribut: wert;
  attribut: wert
}

```

Der *Selektor* gibt an, welches Objekt oder welche Gruppe von Objekten durch diese Stylesheet-Angabe formatiert werden soll. Es werden vier verschiedene Arten von Selektoren verwendet:

▶ *Element*

Hier wird der Name eines HTML-Tags angegeben, und zwar ohne seine Klammern. Beispiele: `h1` oder `td`. Jedes Auftreten des Tags wird automatisch mit den angegebenen Formaten versehen.

Möglich ist auch eine durch Komma getrennte Liste mehrerer HTML-Tags, für die gemeinsame Einstellungen gelten sollen. Es ist kein Problem, differierende Einstellungen noch einmal separat für jedes dieser Tags vorzunehmen.

► *Klasse*

Der Name einer Klasse beginnt mit einem Punkt. Ein HTML-Tag kann mit dem speziellen Attribut `class="klassenname"` versehen werden, damit es zusätzlich zu seinem normalen Stil diese speziellen Einstellungen erhält. Sie können sowohl eine Klasse für ein bestimmtes HTML-Tag einrichten (etwa `h2.blau`), die entsprechend nur für diese Art von Elementen verwendet werden kann, oder aber eine allgemeine Klasse wie `.zitat`, die Sie jedem beliebigen Tag zuweisen können.

Allgemeine CSS-Klassen werden häufig im Zusammenhang mit zwei besonderen Tags verwendet: `<div> ... </div>` ist ein absatzbildendes Element, genauer gesagt sogar eine frei schwebende, beliebig positionierbare Ebene (»DHTML-Layer«). ` ... ` kann dagegen beliebige Elemente innerhalb eines Absatzes umfassen und ihnen auf diese Weise Stylesheet-Formate zuweisen.

► *Unabhängiger Stil*

Ein unabhängiger Stil wird einem einzelnen Element im Dokument zugewiesen. Sein Name beginnt mit einer Raute (`#`); das gewünschte Element erhält diesen Stil über das Universalattribut `id="stilname"` (hier ohne Raute). Am häufigsten werden unabhängige Stile verwendet, um über das Tag `<div>` DHTML-Layer zu erzeugen. In diesem Fall darf eine solche `id`-Definition nur für ein einziges Element im Dokument verwendet werden.

► *Pseudoformate*

Diese Sonderform der Stylesheet-Angabe definiert die verschiedenen Stadien von Hyperlinks oder verschiedene Bereiche von Absätzen. Die Schreibweise für jeden der Hyperlink-Selektoren ist `a:zustand`, wobei der Zustand einen der folgenden Werte annehmen kann:

- `a:link` ist der Grundzustand des Hyperlinks.
- `a:visited` beschreibt bereits besuchte Hyperlinks.
- `a:hover` bezeichnet einen Hyperlink, der gerade vom Mauszeiger berührt wird; dieser Zustand ermöglicht einen Rollover-Effekt für Hyperlinks, allerdings nur im Internet Explorer.
- `a:active` ist der aktuell angeklickte Hyperlink.

Der CSS-Code kann an folgenden Stellen definiert werden:

- ▶ In einem `<style> ... </style>`-Bereich im Head des Dokuments. Konkret sieht eine solche Angabe folgendermaßen aus:

```
<style type="text/css">
<!--
  /* Konkrete CSS-Angaben */
-->
</style>
```

Die Verschachtelung der eigentlichen Stylesheet-Angaben in einen HTML-Kommentar ist üblich und verhindert, dass alte, CSS-unfähige Browser diese Angaben im Body der Seite ausgeben. Innerhalb solcher CSS-Blöcke und in externen CSS-Dateien können Sie Kommentare im C-Stil verwenden, wie oben gezeigt:

```
/* Kommentar */
```

- ▶ In einer externen Datei, einer Textdatei mit der Dateiendung `.css`. Eine solche Datei darf nur CSS und keinerlei HTML enthalten, auch keine `<style>`-Tags. Eingebunden wird eine solche Datei im Head eines HTML-Dokuments, und zwar folgendermaßen:

```
<link rel="stylesheet" href="format.css" />
```

Eine moderne Alternative für das Einbinden von externen CSS-Dateien, die allerdings leider nicht von allen Browsern unterstützt wird, ist die Verwendung einer XML-Steueranweisung. Diese kann nicht nur in HTML-Dokumenten, sondern in beliebigen XML-Dokumenten stehen und ein Stylesheet einbinden, das das Aussehen der diversen XML-Elemente definiert:

```
<?xml-stylesheet type="text/css" href="format.css"?>
```

Die Verwendung externer Stylesheet-Dateien ist besonders vorteilhaft, um einer umfangreichen Website ein einheitliches Layout zuzuweisen, das darüber hinaus an einer zentralen Stelle geändert werden kann.

- ▶ Innerhalb eines einzelnen HTML-Tags. Dazu dient das spezielle HTML-Attribut `style="..."`. Beispiel:

```
<p style="...">
<!-- Absatz mit spezieller Formatierung -->
</p>
<p>
<!-- Hier gilt wieder das normale Absatzformat -->
</p>
```

Stylesheet-Wertangaben

Bevor die einzelnen Attribute beschrieben werden, sollten Sie sich einen Überblick über die Arten von Werten verschaffen, die Attribute überhaupt annehmen können. Sie unterscheiden sich zum Teil erheblich von den Werten, die Sie traditionellen HTML-Attributen zuweisen können: Zum einen haben Sie oft viel mehr Auswahlmöglichkeiten, zum anderen sind die Angaben in der Regel konkreter und eindeutiger.

Im Einzelnen sind die folgenden Arten von Werten möglich:

► *Festgelegte Werte*

Viele Attribute kennen eine Reihe unterschiedlicher konkreter Angaben. Beispielsweise erhält das Attribut `font-family` eine Liste bevorzugter Schriftarten. Ein anderes Beispiel ist die Eigenschaft `text-align`, die die Textausrichtung angibt. Sie kann die festgelegten Werte `left`, `right`, `center` und `justify` annehmen.

► *Numerische Werte*

Die Angabe numerischer Werte erfordert eine Zahl und eine angehängte Maßeinheit ohne trennendes Leerzeichen. Tabelle 11.4 zeigt die möglichen Maßeinheiten.

Maßeinheit	Bedeutung	Hinweise
px	Pixel	Die häufigste und wichtigste Maßeinheit.
pt	Punkt	Der DTP-Punkt, 1/72 Zoll. Wird gern für die Schriftgröße verwendet.
mm	Millimeter	Diese Angaben sind von der Bildschirmgröße und -auflösung abhängig. Gebrochene Werte mit Punkt trennen: 2.5cm, nicht 2,5cm.
cm	Zentimeter	
in	Inch / Zoll (1" = 2,54 cm)	
em	Breite des M	Entspricht dem Geviert, also der Höhe der aktuellen Schriftart.
ex	Breite des x	Wird manchmal als relative Angabe für Wort- oder Zeichenabstände verwendet.
%	Prozent	Im Allgemeinen relativ zum umgebenden Element (Tabellenzelle, Browserfenster). Bei Schriftangaben relativ zur Schriftgröße.

Tabelle 11.4 Maßeinheiten für numerische Stylesheet-Werte

► Farben

Zulässig sind beliebige HTML-Farbangaben – es handelt sich um eine Raute, gefolgt von sechs Hexadezimalstellen (je zwei für Rot, Grün und Blau). Ein Beispiel wäre #FF0000 für Knallrot. Eine andere Möglichkeit ist `rgb(rotwert, grünwert, blauwert)`. Die drei Werte können dezimal zwischen 0 und 255 oder in Prozent angegeben werden.

Stylesheet-Eigenschaften

In diesem Unterabschnitt werden die wichtigsten Stylesheet-Attribute aufgelistet und erläutert. Wichtig ist, dass nicht alle Formatangaben bei jedem HTML-Tag zulässig sind. Bei den meisten Attributen ist relativ offensichtlich, für welche Tags sie geeignet sind; bei anderen finden Sie entsprechende Hinweise.

Die folgenden Attribute beschreiben die verschiedenen Aspekte der typografischen Auszeichnung. Sie sind bei fast jedem beliebigen Tag zulässig.

► font-family

Eine durch Komma getrennte Liste von Schriftarten. Beispiel:

```
font-family: Verdana, Arial, Helvetica, sans-serif
```

► font-size

Die Angabe der Schriftgröße, meist in Punkt (12pt) oder Pixeln (20px). Die Pixel-Angabe ist zu bevorzugen. Sie wird plattformübergreifend identisch interpretiert, während verschiedene Betriebssysteme unterschiedliche Vorstellungen von der Abbildung der Punkt-Angabe auf dem Monitor haben.

► font-style

Dient dazu, Schrift kursiv zu setzen. Die festgelegten Werte sind folgende: `normal`, `italic` (kursiv), `oblique` (elektronisch schräg gestellt, kein separater Kursiv-Schnitt).

► font-weight

Gibt an, wie fett die Schrift dargestellt werden soll. Mögliche Werte sind entweder die festgelegten Angaben `normal`, `bold` (fett) und `extra-bold` (besonders fett) oder glatte numerische Hunderterschritte von 100 (`extra light`) bis 900 (`black`, also ultrafett). Die meisten verfügbaren Schriften definieren allerdings keineswegs neun verschiedene Stufen.

► text-decoration

Gibt an, dass der Text mit Linien versehen, das heißt unterstrichen werden soll und Ähnliches: `none` (keine Linie), `underline` (unterstrichen), `overline` (Linie darüber) oder `line-through` (durchgestrichen).

- ▶ `letter-spacing`
Gibt den Zeichenabstand oder die Laufweite des Textes an. Der Standardwert ist `0pt`; positive Werte erzeugen gesperrten Text, negative verengen die Laufweite.

Die folgenden CSS-Attribute sind zur Formatierung aller absatzbildenden Elemente geeignet, beispielsweise für `<p>`-Elemente, aber auch für Tabellenzellen und ähnliche Tags.

- ▶ `text-align`
Gibt die Textausrichtung mit den möglichen Werten `left` (linksbündig), `right` (rechtsbündig), `center` (zentriert) oder `justify` (Blocksatz) an.
- ▶ `text-indent`
Gibt den Einzug an, also die Einrückung der ersten Zeile eines Bereichs. Der Wert ist numerisch. Positive Werte erzeugen einen eingerückten Einzug, während negative für einen hängenden Einzug sorgen. Für Letzteres ist allerdings eine zusätzliche Verschiebung des linken Innenrandes mittels `margin-left` erforderlich.
- ▶ `line-height`
Bestimmt die Zeilenhöhe durch einen numerischen Wert. Der Standardwert ist etwas größer als die aktuelle Schriftgröße; ein höherer Wert erzeugt einen größeren Zeilenabstand.
- ▶ `vertical-align`
Dies ist die vertikale Ausrichtung des Elements; entspricht dem bekannten `valign` bei Tabellenzellen.
- ▶ `display`
Diese spezielle Angabe ist bei HTML im Grunde überflüssig, da für HTML-Tags klar ist, ob sie absatzbildend sind oder nicht. Wenn Sie CSS dagegen für allgemeine XML-Dokumente einsetzen, ist dieses Format sehr wichtig. Mögliche Werte sind `block`, wenn ein Element absatzbildend sein soll, oder `inline` für ein innerhalb einer Zeile stehendes Element.

Die folgenden CSS-Attribute regeln die Abstände und Rahmeneinstellungen beliebiger Elemente.

- ▶ `margin-top`, `margin-bottom`, `margin-left`, `margin-right`
Diese vier Attribute bestimmen den Abstand des Bereichs zur Umgebung nach oben, unten, links und rechts. Der Wert ist numerisch. Alternativ gibt `margin` einen identischen Abstand auf allen vier Seiten an.
- ▶ `padding-top`, `padding-bottom`, `padding-left`, `padding-right`
Diese Attribute geben den inneren Abstand eines Elements zu seinem Rand

an. Auch hier ermöglicht ein einfaches `padding` eine einheitliche Angabe für alle vier Seiten. Der Unterschied zu den `margin`-Einstellungen zeigt sich, wenn eine sichtbare Rahmenlinie verwendet wird: `padding` setzt den inneren Raum bis zu diesem Rahmen, während `margin` den Außenabstand des Rahmens zur Umgebung festlegt.

- ▶ `border-top`, `border-bottom`, `border-left`, `border-right`

Die Werte für diese Attribute sind eine durch Leerzeichen getrennte Liste von Linienattributen für den äußeren Rand auf der angegebenen Seite eines beliebigen Elements. Stattdessen können Sie mit `border` einen einheitlichen Rahmen auf allen vier Seiten festlegen.

Der erste Wert ist die Linienstärke mit den möglichen Werten `thin` (dünn), `medium` (mittel), `thick` (dick) oder einem beliebigen numerischen Wert. Alternativ kann dieser Wert in Stylesheets auch separat als `border-width` (beziehungsweise `border-top-width` und so weiter) angegeben werden.

Es folgt die Angabe des Linienstils. Möglich sind die folgenden Werte: `solid` (einfache, durchgezogene Linie), `dotted` (gepunktet), `dashed` (gestrichelt), `double` (doppelter Rahmen) sowie die verschiedenen Relief-Effekte `groove`, `ridge`, `inset` und `outset`. Diese Angabe können Sie auch einzeln über das Attribut `border-style` setzen oder Angaben für die Rahmen auf den einzelnen Seiten machen, beispielsweise `border-left-style`.

Schließlich wird die Linienfarbe eingestellt. Hier ist jede beliebige HTML- oder CSS-Farbangabe möglich. Sie lässt sich auch als Einzelwert über `border-color` oder pro Seite durch Attribute wie `border-left-color` festlegen.

Hier zwei Beispiele, um das Ganze zu verdeutlichen:

```
border: medium solid #FF0000
```

Auf allen vier Seiten sollen durchgezogene Randlinien mittlerer Dicke in Rot gesetzt werden.

```
border-top: 4 dashed #0000FF;
border-bottom: 4 dashed #0000FF;
border-left-width: 2;
border-left-style: dotted;
border-left-color: #009900
```

- ▶ Diese verschiedenen Einstellungen sorgen (ohne Berücksichtigung irgendeiner Art von Geschmack) dafür, dass oben und unten 4 Pixel dicke, gestrichelte blaue Randlinien erscheinen, während sich links eine nur 2 Pixel starke, gepunktete blaue Linie befindet.

Fast jedem Tag, zumindest jedem mit Textinhalt, können mithilfe der hier folgenden Angaben Text- und Hintergrundfarbe oder ein Hintergrundbild zugewiesen werden.

- ▶ `color`
Stellt die Vordergrund- beziehungsweise Textfarbe ein.
- ▶ `background-color`
Definiert die Hintergrundfarbe. Jedes beliebige Element kann individuell eingefärbt werden, sowohl absatzbildende Tags als auch Textauszeichnungselemente.
- ▶ `background-image`
Legt ein Hintergrundbild für das Element fest. Der Wert ist `url :`, gefolgt von der URL des Bildes.
- ▶ `background-attachment`
Bestimmt, ob das Hintergrundbild fixiert oder mit der Seite gescrollt wird. Die entsprechenden Werte lauten `fixed` oder `scroll`, wobei Letzteres Standard ist.
- ▶ `background-repeat`
Dieses Attribut bestimmt, ob und in welche Richtung ein Hintergrundbild gekachelt werden soll. Die zulässigen Werte sind `repeat` (in beide Richtungen kacheln), `repeat-x` (nur horizontal wiederholen), `repeat-y` (nur vertikal) oder `no-repeat` (keine Wiederholung, nur Einzelbild).

Layer erzeugen und positionieren

Eine zusätzliche Aufgabe von CSS besteht in der Definition von Eigenschaften für sogenannte Layer. Es handelt sich um unabhängige Ebenen, die frei über der eigentlichen Webseite »schweben« und beliebig positioniert werden können. Nachdem ein Layer erstellt wurde, können seine CSS-Eigenschaften wie Position oder Sichtbarkeit mithilfe der Skriptsprache JavaScript nachträglich geändert werden – im Zusammenhang mit asynchronen XML-HTTP-Anfragen entsteht daraus dann die vor einiger Zeit groß in Mode gekommene *AJAX-Technik* (Asynchronous JavaScript and XML).

Technisch gesehen wird ein Layer durch ein `<div>`-Tag gebildet, das dafür ein `id`-Attribut mit einem eindeutigen Namen und die CSS-Eigenschaft `position` benötigt. Praktischerweise kann das `id`-Attribut auch gleich zur Definition der Stylesheet-Formatierung für den Layer über eine unabhängige Stildefinition verwendet werden.

Für Layer werden die folgenden speziellen Formate definiert:

- ▶ `position`
Abgesehen davon, dass ein `<div>` ohne diese Angabe kein Layer, sondern lediglich eine Art Absatz ist, legt das Attribut über seinen Wert die Art der Positionsangaben für den Layer fest: Der Wert `absolute` bedeutet, dass die Positionsangaben in Pixeln erfolgen, während `relative` sie in Prozent des Browserfensters angibt. Ein dritter möglicher Wert ist `float`, der das Element relativ zum links oder rechts danebenliegenden positioniert und auf diese Weise der wichtigste Layout-Helfer für moderne Websites ist.
- ▶ `left`
Die x-Position des Layers, das heißt sein Abstand vom linken Seitenrand (beziehungsweise vom umgebenden oder – bei `position: float` – vom links danebenliegenden Element). Der Wert wird je nach `position`-Festlegung in Pixeln (zum Beispiel `100px`) oder in Prozent (beispielsweise `30%`) angegeben.
- ▶ `top`
Diese Eigenschaft gibt entsprechend die y-Position des Layers an, also den Abstand vom oberen Rand.

Alternativ können auch `right` und `bottom` (der Abstand vom rechten beziehungsweise unteren Rand) verwendet werden.

- ▶ `z-index`
Dieses Attribut gibt die »Stapelreihenfolge« von Layers an: Je höher der Wert, desto weiter liegt der Layer im Vordergrund. Falls `z-index` weggelassen wird, werden die Layers des Dokuments einfach in der Reihenfolge gestapelt, in der sie im HTML-Code definiert werden.

Ein Beispiel: Auf dem Bild *test.jpg*, das sich 100 Pixel vom oberen und 100 Pixel vom linken Rand entfernt befindet, soll der Text »Hallo« erscheinen. Dazu werden zunächst in einem `<style>`-Block im Head die folgenden unabhängigen Formate definiert:

```
#bild {
  position: absolute;
  top: 100px;
  left: 100px;
  z-index: 1
}
#text {
  position: absolute;
  top: 110px;
  left: 110px;
```

```

z-index: 2;
font-size: 18pt;
color: #FFFF00
}

```

Anschließend werden im Body des HTML-Dokuments die beiden Layer definiert:

```

<div id="bild"></div>
<div id="text">Hallo</div>

```

Das nächste Beispiel versieht einen Text mit einem »Schlagschatten«; der Text wird einmal in Dunkelgrau und einmal in einer anderen Farbe leicht verschoben übereinandergesetzt.

Als Erstes erfolgt wieder die Stylesheet-Definition im Head:

```

#schrift {
  position: absolute;
  top: 10px;
  left: 10px;
  z-index: 2;
  color: #FF0000;
  font-family: Arial, Helvetica, sans-serif;
  font-size: 24pt;
  font-weight: bold
}

```

```

#schatten {
  position: absolute;
  top: 14px;
  left: 7px;
  z-index: 1;
  color: #333333;
  font-family: Arial, Helvetica, sans-serif;
  font-size: 24pt;
  font-weight: bold
}

```

Diese Definitionen können nun folgendermaßen für Layers verwendet werden:

```

<div id="schrift">Wenn wir &uuml;ber den Schatten streiten,<br />
&uuml;bersehen wir das Wesentliche.<br />
-- Aesop</div>
<div id="schatten"> Wenn wir &uuml;ber den Schatten streiten,<br />
&uuml;bersehen wir das Wesentliche.<br />
-- Aesop </div>

```

11.2.3 Der Webseiten-Editor Quanta

Der komfortable HTML- und XML-Editor *Quanta Plus* gehört zum KDE-Projekt. Er ist ein Lieferbestandteil von openSUSE; Sie finden ihn bei der YaST-Software-Suche nach `quanta` unter dem Paketnamen `kdewebdev3`. Gestartet wird der Editor unter **K-MENÜ • INTERNET • ERSTELLEN VON WEBSEITEN • QUANTA PLUS (WEB-ENTWICKLUNGSUMGEBUNG)**.

Das Erste, was Ihnen beim Eingeben von HTML- oder XML-Tags auffallen dürfte, ist das Popup-Menü, das bereits beim Eintippen der öffnenden spitzen Klammer angezeigt wird. Es enthält eine Liste von HTML-Tags, die Sie mit der Maus auswählen können. Während Sie tippen, wird die Auswahl auf die bereits vorhandenen Zeichen beschränkt. Unglücklicherweise enthält dieses Menü aus traditionellen Gründen nur großgeschriebene Tags. Als erste »Amtshandlung« sollten Sie daher **EXTRAS • GROSS-/KLEINSCHREIBUNG DER TAGS UMSCHALTEN** wählen und sowohl Tags als auch Attribute auf Kleinbuchstaben umstellen.

Sobald Sie ein Tag ausgewählt oder eingetippt haben, wird automatisch das zugehörige schließende Tag erzeugt. Geben Sie etwa `<html>` ein, und das passende `</html>` wird hinzugefügt. Der Cursor befindet sich sofort zwischen den beiden Tags. Dies gilt aber nicht nur für HTML-Tags, sondern auch für beliebige frei eingegebene XML-Tags. Geben Sie zum Testen beispielsweise `<cds>` ein – sofort ist `</cds>` zur Stelle.

Über dem eigentlichen Eingabebereich gibt es eine Symbolleiste; ihr Inhalt lässt sich durch die diversen Registerkarten umschalten. Im Einzelnen handelt es sich um folgende Auswahlmöglichkeiten:

► **STANDARD**

Enthält Schaltflächen für **FETT**, **KURSIV** und **UNTERSTRICHEN** (verwenden Sie Letzteres bitte niemals, da alle Welt es für einen Link halten wird!). Es folgen Zeilenumbruch, Absatz und geschütztes Leerzeichen. Danach kommen Anker (für Hyperlinks, benannte Anker und andere Verknüpfungen), Bild, Trennlinie und Kommentar. Den Abschluss bilden die verschiedenen Absatzausrichtungen sowie das ``-Tag.

► **STIL**

Diese Palette ist für CSS und andere Formatierungen gedacht. Nach `` und `<div>` folgen CSS und ein Farbwähler. Der nächste Bereich bietet vorformatierten, tief- und hochgestellten Text. Den Abschluss bilden die ersten fünf der sechs möglichen Überschriften-Hierarchiestufen.

► **TABELLEN**

Die erste Schaltfläche startet den **TABELLEN-ASSISTENTEN**, mit dem sich die Eigenschaften von Tabellen komfortabel festlegen lassen. Danach folgen But-

tons für die diversen Einzelaspekte von Tabellen, die zum Teil weiter oben besprochen wurden.

► LISTEN

Auch hier steht als Erstes ein Assistent zur Verfügung, der das Erstellen von Listen erleichtert. Danach folgen Schaltflächen für die einzelnen Listenarten und ihre diversen Aspekte.

► FORMULARE

In dieser Sektion macht das Formular selbst den Anfang; danach werden die diversen Eingabe- und Auswahlelemente bereitgestellt.

► ANDERE

Hier finden Sie eine Sammlung zusätzlicher Elemente, die in keine andere Kategorie passen.

Unter dem Menüpunkt PLUGINS finden Sie nützliche Zusatz-Software, etwa den KIMAGEMAPEDITOR, mit dem sich komfortabel clientseitige Image-Maps erstellen lassen. Auch die anderen Menüs haben unzählige interessante Funktionen zu bieten. Mit dem Basiswissen aus diesem Abschnitt können Sie ohne Weiteres selbst auf Entdeckungsreise gehen.

11.3 LaTeX

Die WYSIWYG-Dokumentenverarbeitung im Bereich des Textsatzes war nicht zu allen Zeiten selbstverständlich. Fotosatzmaschinen und frühe computerbasierte Satz- und Layoutsysteme verwendeten angepasste Spezialsprachen, um das Layout und den Textinhalt von Druckdokumenten zu beschreiben.

Eine der bekanntesten Textsatzsprachen wurde in den 80er-Jahren von *Donald E. Knuth* entworfen. Hauptberuflich war er ein sehr bekannter Informatikprofessor – sein Hauptwerk ist das bisher dreibändige, auf sieben Bände ausgelegte »The Art of Computer Programming«. 1982 gab er das Manuskript einer Neuauflage des ersten Bandes beim Verlag ab und war enttäuscht über den schlechten Satz, der dabei herauskam. Dies veranlasste ihn, mit der Arbeit an seinem eigenen Textsatzsystem zu beginnen, genannt TeX (gesprochen »Tech« – mit ch wie in »Blech«).

Zusammen mit TeX entwickelte Knuth ein weiteres Programm namens METAFONT zur Definition vektorbasierter elektronischer Schriften. Die wichtigsten Ideen von METAFONT führten später zur Entwicklung der Adobe-PostScript-Schriften. In TeX gesetzte Dokumente bestechen durch ihre hohe Qualität, die nicht einmal von professionellen Layoutprogrammen wie InDesign oder QuarkX-

Press erreicht wird, sondern höchstens von der Kunst des manuellen Bleisatzes. Außerdem ist kein anderes Programm in der Lage, in ähnlicher Perfektion mathematische Formeln zu setzen.

TeX selbst wird nicht mehr weiterentwickelt, sondern es werden nur noch die (verhältnismäßig seltenen) Bugs bereinigt. Deshalb konvergiert die Versionsnummer gegen p ; zurzeit ist es 3.141592.

TeX allein ist relativ schwierig zu benutzen; es müssen Unmengen von Konfigurationsanweisungen geschrieben werden, bevor das eigentliche Dokument gesetzt werden kann. Aus diesem Grund fügten *Leslie Lamport* und andere Entwickler eine Makrosammlung namens LaTeX hinzu, die viele der Konfigurationsarbeiten durch Zusammenfassung erleichtert, sodass Autoren sich mehr auf den eigentlichen Text konzentrieren können als auf die Satzsprache.

openSUSE wird mit LaTeX und umfangreichen Zusatzpaketen geliefert. Die betreffende Software befindet sich in der YaST-Selektion LATEX, SGML UND XML.

11.3.1 Erstes Beispiel

Bevor im nächsten Unterabschnitt einige (wenige) LaTeX-Elemente erläutert werden, wird zunächst ein Einführungsbeispiel präsentiert. Geben Sie den folgenden Text in einem beliebigen Editor ein (Emacs ist ideal, da er einen LaTeX-Modus besitzt):

```
\documentclass{article}
\begin{document}
\section \LaTeX{}
\LaTeX{} ist eine Makrosammlung, die auf der Satzsprache \TeX{} von
\emph{Donald E. Knuth}\footnote{Der Autor des bekannten The Art Of
Computer Programming.} basiert. \LaTeX{} bietet eine Reihe von
Vorteilen:
\begin{itemize}
\item
Gestochen scharfer Textsatz.
\item
Mathematische Formeln in Perfektion.
\item
Listen, Tabellen und Abbildungen ohne Probleme.
\end{itemize}
\end{document}
```

Wie Sie sehen, werden Formatierungsbefehle mit `\befehl` eingeleitet; die Inhalte, die ein Befehl umschließt, stehen in geschweiften Klammern. Im Einzelnen werden hier folgende besondere Formate verwendet:

- ▶ `\documentclass{article}`
Jedes LaTeX-Dokument benötigt eine Dokumentklasse. `article` ist eine der häufigsten und wird für kürzere Texte verwendet.
- ▶ `\begin{document} ... \end{document}`
Bei `\begin{document}` ist die Definition diverser Metainformationen beendet, und der eigentliche Dokumentinhalt beginnt. Mit `\end{document}` können Sie ihn beenden.
- ▶ `\section`
Eine Hauptabsatzüberschrift
- ▶ `\LaTeX{}`, `\TeX{}`
LaTeX und TeX in ihrer eigentlichen Schreibweise mit versetzten Buchstaben
- ▶ `\emph{...}`
Der Text in den geschweiften Klammern wird kursiv gesetzt (`emph` steht für »emphasis«, also Betonung)
- ▶ `\footnote{...}`
Der Inhalt der geschweiften Klammern wird als Fußnote gesetzt; die Fußnotenmarkierung erscheint an der aktuellen Stelle des Dokuments
- ▶ `\begin{itemize} ... \end{itemize}`
Umschließt eine einfache Aufzählung (bullet list)
- ▶ `\item`
Leitet einen Aufzählungspunkt ein

Speichern Sie das Dokument als Datei mit der Endung `.tex`. Nun wird es vom eigentlichen LaTeX-Programm in eine DVI-Datei (Device-Independent Print File) umgewandelt, die über das Programm `dvips` in PostScript umgewandelt werden kann. Die freie PostScript-Implementierung *GhostScript* ermöglicht schließlich die Vorschau und Ausgabe des Dokuments. Geben Sie dazu folgende Befehle ein:

```
$ latex latexinfo.tex
$ dvips latexinfo.dvi
$ gs latexinfo.ps
```

Eine noch praktischere Möglichkeit besteht darin, aus den DVI-Dokumenten direkt das modernere PDF-Format zu generieren. Geben Sie dazu `dvipdf` statt `dvips` ein. Beispiel:

```
$ dvipdf latexinfo.dvi
```

Auf Ihrem openSUSE-System und in anderen modernen Distributionen lässt sich der gesamte Arbeitsaufwand zum Erstellen eines PDF-Dokuments aus der LaTeX-Quelle sogar zu einem einzigen Befehl zusammenfassen:

```
$ pdflatex latexinfo.tex
```

Wie ein solches Dokument formatiert aussieht, sehen Sie im nächsten Unterabschnitt, wo nach der Erläuterung einiger wichtiger Formate noch ein größeres Beispieldokument erstellt wird.

11.3.2 LaTeX-Basisformate

LaTeX besitzt einen unendlichen Vorrat an Formatierungen für Zeichen, Absätze und andere Elemente. Aus diesem Grund kommen hier nur einige Formate exemplarisch zur Sprache. Es gibt aber zahlreiche gute Bücher, die Ihnen weiterhelfen können; empfehlen kann ich Ihnen unter anderem »LaTeX« von Karsten Günther (Bonn 2004, Galileo Press).

Dokumentgliederung

Wie bereits erwähnt, muss jedes LaTeX-Dokument mit einer `\documentclass`-Anweisung beginnen. Diese richtet die spezifische Umgebung für die jeweilige Dokumentsorte ein. Einige wichtige Dokumentklassen sind:

- ▶ `article` – ein kürzerer Text, im Wesentlichen als Zeitschriftenartikel gedacht
- ▶ `report` – Erweiterung von `article`, die zusätzlich über Kapitel (`\chapter`) verfügt
- ▶ `book` – die Dokumentklasse für Bücher. Beispielsweise beginnen Kapitel stets auf einer rechten Seite.
- ▶ `proc` – zweispaltiges Protokoll (`proceedings`)
- ▶ `slides` – Präsentationsfolien
- ▶ `letter` – Klasse für geschäftliche Korrespondenz

Den Platz zwischen der `\documentclass`-Definition und der Anweisung `\begin{document}`, die den eigentlichen Dokumentinhalt einleitet, bildet die sogenannte *Präambel*. Hier stehen zunächst oft mehrere `\usepackage`-Anweisungen zum Einbinden von Erweiterungen (sogenannten Makropaketen), die LaTeX um zahllose Aspekte ergänzen. Hier nur einige wenige im Überblick:

- ▶ `array` – erweiterte Tabellenfunktionen
- ▶ `enumerate` – verbesserte nummerierte Listen
- ▶ `exscale` – Skalierung mathematischer Formeln
- ▶ `fontenc` – Verwendung von Schriften mit besonderer Codierung (wird oft für nicht-englische Dokumente geladen)
- ▶ `fontspec` – Funktionen für Schriftartentabellen

- ▶ `german` – Erweiterung für Deutsch (Umlaute, Trennregeln usw.)
- ▶ `graphics` – diverse Grafikfunktionen
- ▶ `multicol` – flexibler Mehrspaltensatz

Weitere verbreitete Meta-Informationen, die Sie in die Präambel schreiben können, sind Titel und Autor des Dokuments. Der Titel wird wie folgt angegeben:

```
\title{Titeltext}
```

Für den Autor wird `\author{...}` verwendet; innerhalb der geschweiften Klammern dieses Feldes können Sie `\\` verwenden, um Zeilenumbrüche zu benutzen. Das folgende Beispiel setzt unter den Autorennamen die kursive E-Mail-Adresse:

```
T\author{Sascha Kersken\\emph{sk@lingoworld.de}}
```

Wenn das Dokument mehrere Autoren besitzt, können Sie mehrere `\author`-Blöcke verwenden. Zwischen ihnen sollten Sie den speziellen Befehl `\and` anbringen.

Die Autoren- und Titelangaben werden zunächst gespeichert; innerhalb des eigentlichen Dokuments können Sie sie dann mit `\maketitle` ausgeben.

Das eigentliche Dokument wird, wie bereits erwähnt, von `\begin{document}` und `\end{document}` umschlossen. Innerhalb des Dokuments können Sie zahlreiche, hierarchisch verschachtelte Unterteilungen einfügen:

```
\part{Überschrift}
```

Teile bilden die oberste Gliederungsebene.

```
\chapter{Überschrift}
```

Kapitel sind nur in den Dokumentklassen `report` und `book` verfügbar.

```
\section{Überschrift}
```

Beginnt einen neuen Abschnitt mit der angegebenen Überschrift. In den meisten Dokumentklassen werden Abschnitte nummeriert.

```
\subsection{Überschrift}
```

Dies ist entsprechend ein Unterabschnitt; auf diese Weise können Sie Abschnitte weiter unterteilen.

```
\subsubsection{Überschrift}
```

Wenn Sie möchten, können Sie sogar noch eine weitere Gliederungsebene einführen.

Eine besondere Bedeutung besitzt die bereits erwähnte Anweisung `\maketitle`: Sie gibt die in der Präambel definierten Informationen Titel und Autor sowie das aktuelle Datum aus und sollte unmittelbar auf `\begin{document}` folgen. Auf Wunsch können Sie diese Anweisung durch `\begin{titlepage}` und `\end{titlepage}` umschließen, um eine Titelseite anstelle eines Titelblocks zu erzeugen.

Zeichenformatierung und Sonderzeichen

LaTeX wurde ursprünglich für den Umgang mit englischem Text geschrieben und versteht von Hause aus nur reinen ASCII-Text. Wenn Sie deutsche Texte damit verfassen möchten, binden Sie folgendes Makropaket ein:

```
\usepackage{german}
```

Standardmäßig werden Umlaute und ß dann wie folgt eingegeben:

- ▶ "s für ß
- ▶ "a für ä, "A für Ä
- ▶ "o für ö, "O für Ö
- ▶ "u für ü, "U für Ü

Auch für andere Sonderzeichen gibt es spezielle Eingabeformate. Beispielsweise müssen Sie alle Zeichen, die in LaTeX eine besondere Bedeutung haben, speziell codieren, wenn Sie die Zeichen als solche meinen – dies gelingt in vielen Fällen durch Voranstellen des bekannten Backslashes. Beispiele sind etwa die Sequenzen `\{`, `\}`, `\$`, `\%` und `\&`. Das Prozentzeichen leitet normalerweise einen Kommentar ein, der bis zum Ende der aktuellen Zeile reicht und von LaTeX ignoriert wird. `\` erzeugt, wie bereits erwähnt, einen einfachen Zeilenwechsel, während ein neuer Absatz durch zwei Zeilenumbrüche im LaTeX-Code herbeigeführt wird.

Für einige andere Sonderzeichen sind dagegen spezielle Schreibweisen oder Textbefehle definiert. Hier nur einige wenige Beispiele:

- ▶ ```` (zwei Backticks) sind die einleitenden (englischen) Anführungszeichen.
- ▶ `''` (zwei Apostrophe) steht für die ausleitenden (englischen) Anführungszeichen.
- ▶ `--` erzeugt einen Gedankenstrich (–).
- ▶ `\lbrack` erstellt eine öffnende eckige Klammer, also `[`.
- ▶ `\rbrack` sorgt für das Gegenstück, die schließende eckige Klammer `]`.
- ▶ `\textbackslash` stellt einen Backslash (`\`) dar.
- ▶ `\dots` setzt eine Ellipse, also drei Punkte (...).

Eine Fülle weiterer Sonderzeichen erschließt Ihnen das Makropaket `textcomp`, das Sie mithilfe der folgenden Präambel-Anweisung einbinden können:

```
\usepackage{textcomp}
```

Einige wichtige Zeichen aus diesem Paket sind `\texteuro` (Eurosymbol: €), `\textcopyright` (Copyright-Zeichen: ©), `\textquotestraightbase` (einfaches, gerades Anführungszeichen: ') oder `\textquotestraightdblbase` (doppeltes, gerades Anführungszeichen: ").

Wie Sie bereits gesehen haben, gibt es schließlich spezielle Formatbefehle für die eigenwillig gesetzten Eigennamen der Sprache und ihrer Erweiterung: `\TeX{}` und `\LaTeX{}`.

Die wichtigsten Inline-Formatierungen für Zeichen sind `\emph{Text}` für *kursiven Text* und `\textbf{Text}` für **Fettdruck**.

Zum Einstellen der Schriftgröße gibt es (von winzig bis riesengroß) folgende Vorgabewerte:

- ▶ `\tiny`
- ▶ `\scriptsize`
- ▶ `\footnotesize`
- ▶ `\small`
- ▶ `\normalsize`
- ▶ `\large`
- ▶ `\Large`
- ▶ `\huge`
- ▶ `\Huge`

Wenn Sie viel Text in einer bestimmten Größe schreiben möchten, können Sie mit dem passenden Befehl bis auf Weiteres umschalten und später mit `\normalsize` zur Standard-Schriftgröße zurückkehren. Möchten Sie dagegen nur einige Wörter in einer anderen Größe darstellen, dann setzen Sie die entsprechende Größenangabe mitsamt dem Größenbefehl in geschweifte Klammern. Beispiel:

```
Normaler Text, {\small etwas kleinerer Text}, wieder normal.
```

Die drei Grund-Schriftsorten legen Sie folgendermaßen fest:

- ▶ `\rmfamily` (Roman) ist Serifenschrift.
- ▶ `\sffamily` stellt serifenlose Schrift ein.
- ▶ `\ttfamily` ist für Nichtproportionalschrift (Festbreitenschrift) zuständig.

Aufzählungen und Listen

Jeder wissenschaftliche Text und jedes Fachbuch wie das vorliegende macht häufigen Gebrauch von Listen und Aufzählungen. Genau wie HTML versteht sich auch LaTeX vorzüglich auf das Erstellen dieser Elemente. Es gibt im Wesentlichen drei Listenarten: Beschreibungslisten, Aufzählungen und nummerierte Listen.

Eine einfache Aufzählung steht zwischen den Anweisungen `\begin{itemize}` und `\end{itemize}`. Jeder einzelne Aufzählungspunkt beginnt mit der Anweisung `\item`. Hier ein Beispiel:

```
\begin{itemize}
\item Der erste Punkt
\item Ein weiterer Punkt
\item Der letzte Punkt
\end{itemize}
```

Sie können solche Aufzählungen auch ineinander verschachteln, indem Sie unter einem `\item` der äußeren Ebene einen weiteren `\begin{itemize} ... \end{itemize}`-Block einfügen. Auch dafür sehen Sie hier ein Beispiel:

```
\begin{itemize}
\item Erster "au"serer Punkt
  \begin{itemize}
    \item Erster innerer Punkt des ersten "au"seren Punktes
    \item Zweiter innerer Punkt des ersten "au"seren Punktes
  \end{itemize}
\item Zweiter "au"serer Punkt
  \begin{itemize}
    \item Erster innerer Punkt des zweiten "au"seren Punktes
    \item Zweiter innerer Punkt des zweiten "au"seren Punktes
  \end{itemize}
\end{itemize}
```

Nummerierte Listen stehen zwischen `\begin{enumerate}` und `\end{enumerate}`; ansonsten gelten dieselben Bedingungen wie bei Aufzählungen – beispielsweise wird auch hier `\item` für das einzelne Element verwendet. In verschachtelten Nummerierungen verwendet die erste Ebene standardmäßig arabische Ziffern, die zweite Kleinbuchstaben. Mithilfe spezieller Anweisungen, die hier zu weit führen würden, lässt sich dies ändern. Hier ein kurzes Beispiel:

```
\begin{enumerate}
\item Erstens
\item Zweitens
\item Drittens
\end{enumerate}
```

Die dritte Art der Liste, die Beschreibungsliste, ähnelt der Definitionsliste in HTML. Am Anfang der `\item`-Objekte können Sie einen Begriff oder auch mehrere Wörter in eckige Klammern setzen; dieser Text wird dann ausgerückt und fett dargestellt. Beispiel:

```
\begin{description}
\item [\TeX{}}] ist die urspr"ungliche Auszeichnungssprache
von Donald E. Knuth, die er f"ur den Satz seines Werkes
``The Art Of Computer Programming'' -- ein Standardwerk
der Computerliteratur -- entwickelte.
\item [\LaTeX{}}] ist eine Sammlung von \TeX{}-Makros von
Leslie Lamport und anderen, die die Benutzung der
urspr"unglichen Satzsprache erheblich erleichtert.
\end{description}
```

Tabellen

Für den Satz von Tabellen gibt es unzählige Erweiterungspakete, aber hier können natürlich nur die absoluten Basics berücksichtigt werden.

Eine Tabelle wird durch folgenden Block umschlossen:

```
\begin{tabular}{Spalteninfo}
...
\end{tabular}
```

Die Spalteninfo beinhaltet mindestens ein Zeichen für jede geplante Spalte, das deren Ausrichtung angibt: `l` (linksbündig), `c` (zentriert) oder `r` (rechtsbündig). Optional können Sie beispielsweise Pipe-Zeichen (`|`) zwischen die Zellformatierungen setzen, um die Zellen durch Trennlinien zu separieren.

Innerhalb des Tabellenblocks werden Zellen durch `&` separiert und Zeilen durch `\\` abgeschlossen. Hier ein Beispiel für eine einfache Tabelle mit drei linksbündigen Spalten:

```
\begin{tabular}{l|l|l}
Die & erste & Zeile \\
Die & zweite & Zeile \\
Die & dritte & Zeile
\end{tabular}
```

Wenn Sie auch horizontale Trennlinien einfügen möchten, können Sie vor der jeweiligen Zeile ein `\hrline` hinzufügen. Für einen Außenrand kann der gesamte Tabellencode beispielsweise durch `\fbox{...}` umschlossen werden. Beispiel:

```
\fbox{ \begin{tabular}{l|l|l}
\textbf{Software} & \textbf{Autor} & \textbf{Version} \\
\end{tabular}}
```

```
\hline \TeX{} & D.E.Knuth & 3.141592 \\
\hline \LaTeX{} & Leslie Lamport & \LaTeX{}2e
\end{tabular} }
```

Mathematische Formeln

Besonders berühmt ist LaTeX für seine Fähigkeit, mathematische Formeln in Perfektion zu setzen. Dafür ist der spezielle Mathematikmodus zuständig. Sie starten ihn am einfachsten, indem Sie eine Formel zwischen einzelnen Dollarzeichen einschließen, wenn sie innerhalb einer Textzeile stehen soll, oder zwischen doppelten Dollarzeichen, falls sie einen Absatz bilden soll. Beispiele:

Pythagoras lehrte: $a^2+b^2=c^2$

Die erste binomische Formel:
 $(a+b)^2=a^2+2ab+b^2$

Wie Sie wahrscheinlich bereits gemerkt haben, wird $\text{\sp{...}}$ für Exponenten verwendet. Analog existiert das Format $\text{\sb{...}}$ für Indizes, das heißt tiefgestellte Werte.

Daneben besitzt der Mathematikmodus Hunderte von Formatierungsmöglichkeiten und Spezialzeichen. Hier nur eine Handvoll für einen ersten Eindruck:

- ▶ Die für Winkel verwendeten griechischen Buchstaben lassen sich mit ihren lateinischen Bezeichnungen \alpha , \beta usw. bis \omega erzeugen. Beachten Sie dabei, dass es – aufgrund der englischsprachigen Herkunft von TeX – \mu und \nu heißt.
- ▶ Griechische Großbuchstaben sind ebenfalls möglich; ihr Name beginnt einfach mit dem entsprechenden lateinischen Großbuchstaben. Dabei sind nur diejenigen definiert, die nicht auch im lateinischen Alphabet vorkommen, etwa \Delta oder \Xi .
- ▶ \land , \lor und \not stellen die logischen Operatoren Und, Oder sowie Nicht dar.
- ▶ \in und \notin stehen für »ist Element« beziehungsweise »ist nicht Element«.
- ▶ $<$, $>$ und $=$ können Sie einfach verwenden; daneben stehen folgende Relationen zur Verfügung: \neq für ungleich, \leq für kleiner oder gleich und \geq für größer oder gleich.
- ▶ \sin , \cos , \tan , \ln , \log und so weiter erzeugen die gleichnamigen Funktionen.
- ▶ $\text{\sqrt{Wert}}$ setzt den Wert unter ein Wurzelzeichen.

Ein größeres Beispiel

Das nachfolgende Listing fasst die meisten der oben beschriebenen LaTeX-Formatierungen zusammen. In Abbildung 11.2 und Abbildung 11.3 sehen Sie dann das Ergebnis, ein zweiseitiges PDF-Dokument.

```

\documentclass{article}
\usepackage{german}
\title{\LaTeX{}-Beispielsammlung}
\author{Sascha Kersken \ \ \emph{sk@lingoworld.de} }
\begin{document}
\maketitle
\section{Das Wichtigste in K"urze}
Hier erfahren Sie Grundlegendes "uber \TeX{} und \LaTeX{}.
\begin{description}
\item[\TeX{}] wurde von \textbf{Donald E. Knuth} entwickelt,
weil er sich f"ur sein Meisterwerk ``The Art Of Computer
Programming'' einen professionelleren Textsatz w"unschte.
\item[\LaTeX{}] ist eine Erweiterung der Satzsprache
\TeX{}, die Dokumentklassen und andere vorgefertigte
Formatierungen enth"alt und den Anwendern so das Leben
entscheidend erleichtert.
\end{description}
\section{Aufz"ahlungen und Listen}
Dies ist eine verschachtelte Aufz"ahlung/Liste:
\begin{itemize}
\item Textbasierte Auszeichnungssprachen
  \begin{itemize}
\item XML
\item HTML
\item \LaTeX{}
\end{itemize}
\item Inkompatible Bin"arformate
  \begin{itemize}
\item MS Office
\item PDF (ja, leider auch -- rein geht immer, raus
nur sehr notd"urftig!)
\end{itemize}
\end{itemize}
\section{Spa"s mit Tabellen}
\fbbox{
  \begin{tabular}{|l|l|l|}
\textbf{Betriebssystem} & \textbf{Aktuelle Version} &
\textbf{Bemerkungen} \\ \
\hline Linux & Kernel 2.6.16.14 & Open Source-OS \\

```

```

\hline FreeBSD & 6.0 & Open Source-OS \\
\hline Windows & Vista, SP1 & kommerziell, von Microsoft
\end{tabular}
}
\section{Schriftgr"o"sen und -formate}
\rmfamily
Dies ist Serifenschrift (Roman)

\sfamily
Dies ist serifenlose Schrift

\ttfamily
Hier kommt Schreibmaschinenschrift (Nichtproportionalschrift)

\rmfamily
{\tiny Ultrawinzig}, {\scriptsize winzig}, {\footnotesize
sehr klein}, {\small klein}, normal, {\large gr"o"ser},
\Large noch gr"o"ser}, {\huge sehr gro"s}, {\Huge riesig}.

Dann gibt es noch \textbf{fett}, \emph{kursiv} und eine
\textbf{\emph{Mischung aus beidem}}.

\section{Zum Abschluss ein bisschen Mathe}
Pythagoras wusste bereits:  $a^2+b^2=c^2$ 

Das so genannte DeMorgan-Theorem besagt:
 $\neg(a \wedge b) = \neg a \vee \neg b$ 
 $\neg(a \vee b) = \neg a \wedge \neg b$ 
\end{document}

```

Tippen Sie den Code ab, oder laden Sie ihn sich aus dem Listing-Bereich der Website zum Buch herunter. Anschließend können Sie Folgendes eingeben, um ihn zum PDF-Dokument zu kompilieren:

```
$ pdflatex latex_sample.pdf
```

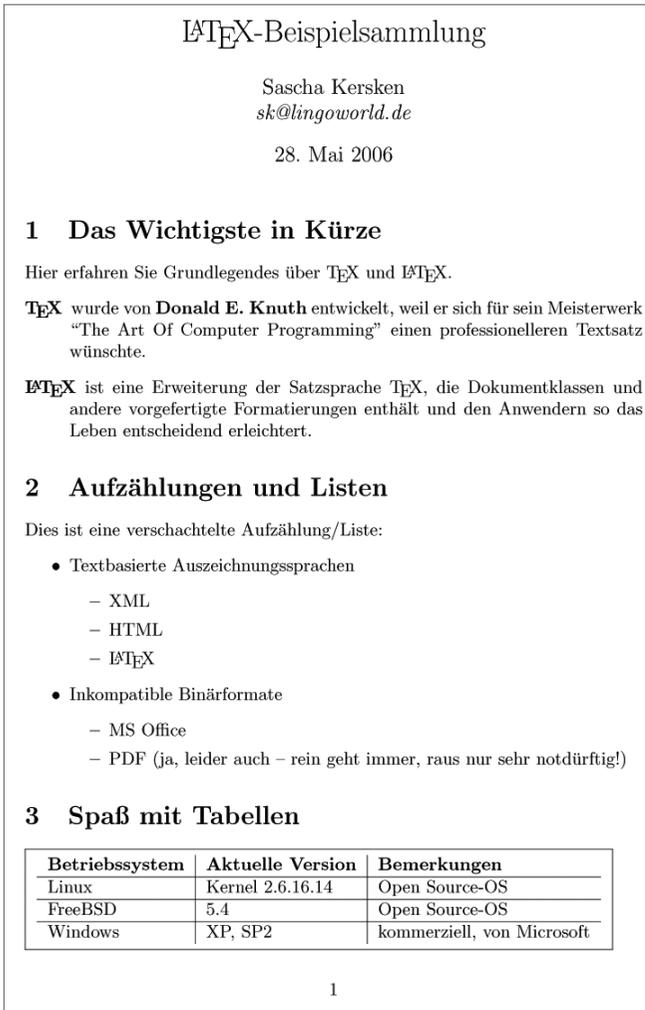


Abbildung 11.2 Die erste Seite des aus dem LaTeX-Dokument erzeugten PDFs (auf den Satzspiegelinhalt reduziert)

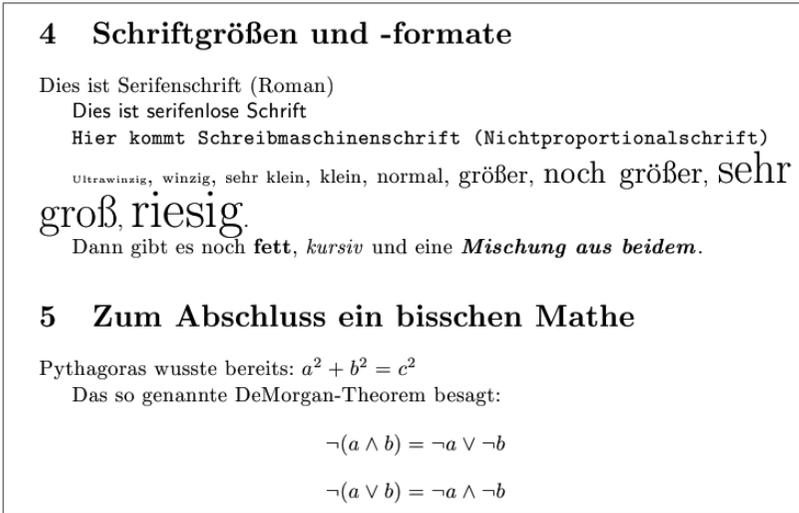


Abbildung 11.3 Die zweite Seite des PDF-Dokuments aus der LaTeX-Vorlage

11.4 Zusammenfassung

Textbasierte Auszeichnungssprachen sind in der IT-Welt und besonders auf UNIX-Systemen wie Linux allgegenwärtig. In diesem Kapitel haben Sie die drei wichtigsten von ihnen kennengelernt.

XML hat sich seit seiner Einführung im Jahr 1998 zum wichtigsten universellen Datenaustauschformat entwickelt. Seine Stärke besteht darin, dass Sie im Grunde beliebige Formate entwerfen können, die dennoch wohlgeordnet gespeichert werden. Neben diesen freien Datenformaten basieren auf *XML* zahllose verbreitete Standards wie *XHTML*, *SVG*, *DocBook* und mehr.

HTML und seine *XML*-basierte Neuentwicklung *XHTML* bilden das Fundament des World Wide Web. Die Sprache bietet einen durchdachten Satz von Formatierungsmöglichkeiten. Ergänzt wird sie heute durch Cascading Stylesheets (*CSS*), um Struktur und Layout eines Dokuments sauber voneinander zu trennen.

LaTeX ist ein leistungsfähiges Textsatzsystem, das professionelle Dokumente für den Druck erzeugt. Wenn Sie sich einmal an die etwas eigenwillige Syntax gewöhnt haben, werden Sie kaum noch etwas anderes für Ihre Drucksachen verwenden wollen.

*Die Erfindung des Problems ist wichtiger als die
Erfindung der Lösung;
in der Frage liegt mehr als in der Antwort.
– Walther Rathenau*

12 Der Datenbankserver MySQL

Datenbank-Serversysteme sind in der Lage, Anwendungen oder anderen Serverdiensten den Zugriff auf eine Datenbank zu gewähren, bei Bedarf auch über lokale Netzwerke oder über das Internet. Mit openSUSE werden zwei wichtige Open-Source-Datenbankserver geliefert: MySQL und PostgreSQL. In diesem Kapitel wird MySQL in der aktuellen Version 5.1 vorgestellt.

12.1 Relationale Datenbanken

Die Bezeichnung *relationale Datenbank* (englisch »relational database«) verweist auf die *relationale Algebra*. Eine Relation ist danach ein Tupel (geordnete Sammlung einer bestimmten Anzahl von Werten), das Attribute bestimmter Datentypen sammelt. Ein Relationenschema $R = (A_1, A_2, \dots, A_n)$ legt Anzahl und Typ dieser Attribute fest.

In der Praxis kommt ein Tabellenmodell zum Einsatz. In den Spalten einer Datenbanktabelle stehen die verschiedenen Informationskategorien. Die einzelnen Zellen – und damit die Attribute der Relation – werden *Datenfelder* oder kurz *Felder* (»fields«) genannt und sind die kleinste Informationseinheit der Datenbank: Sie enthalten je eine Einzelinformation über ein Element der Datenbank. Eine ganze Zeile ist die Kombination aller Informationen über ein Element und wird *Datensatz* (»record«) genannt. Das jeweilige Element, über das ein Datensatz Informationen enthält, wird als *Entity* (manchmal auch Entität) bezeichnet.

In Tabelle 12.1 sehen Sie ein Beispiel für eine Datenbanktabelle, die verschiedene Informationen über Musik-CDs enthält.

Eine relationale Datenbank besteht aus beliebig vielen Einzeltabellen, die auf vielfältige Weise miteinander verknüpft werden können. Dieser Aufbau sorgt dafür, dass die Daten in der Datenbank *konsistent* sind: Jede Information muss

nur ein einziges Mal gespeichert werden, sodass es nicht zu Mehrdeutigkeiten kommen kann.

Interpret	Titel	Genre	Jahr	CDs	Tracks
Red Hot Chili Peppers	Stadium Arcadium	Rock	2007	2	28
Metallica	Ride The Lightning	Metal	1984	1	8
Iron Maiden	Powerslave	Metal	1986	1	8
Pink Floyd	The Dark Side Of The Moon	Rock	1979	1	9

Tabelle 12.1 Beispiel für eine einfache Datenbanktabelle

Die Verknüpfungen zwischen den Spalten einer Tabelle und zwischen den einzelnen Tabellen werden als *Beziehungen* (englisch relationships, daher manchmal auch »Relationen«) bezeichnet. Eine Beziehung entsteht durch die Verwendung von *Schlüsseln*: Der *Primärschlüssel* (primary key) des Datensatzes einer Tabelle wird als Wert in ein Feld einer anderen Tabelle eingetragen. Der Primärschlüssel ist ein spezielles Datenfeld oder eine Kombination der Werte mehrerer Felder, die innerhalb der Tabelle einen einmaligen Wert besitzen und den Datensatz somit eindeutig kennzeichnen. Der Primärschlüssel einer Tabelle, auf den in einer anderen Tabelle verwiesen wird, heißt dort *Fremdschlüssel* (foreign key).

Der Primärschlüssel ist übrigens ein Sonderfall eines sogenannten *Indexes*. Indizes ermöglichen den schnellen Zugriff auf bestimmte Tabelleninhalte, indem sie die Informationen einer Datenbankspalte separat in geordneter Form abspeichern. Um einen Datensatz anhand eines indizierten Felds zu finden, muss nicht die gesamte Datenbanktabelle sequenziell durchsucht werden.

Es gibt drei verschiedene Arten von Beziehungen, die je nach Art der gespeicherten Information nützlich sind:

► *1:1-Beziehung*

Eine 1:1-Beziehung verknüpft einen Datensatz einer Tabelle mit genau einem Datensatz einer anderen Tabelle. Dies ist immer dann nützlich, wenn bestimmte Informationsaspekte über ein Entity nicht so oft benötigt werden wie andere Aspekte. Die seltener oder in einem anderen Zusammenhang verwendeten Informationen lassen sich auf diese Weise einfach ausblenden. Beispielsweise könnte eine Tabelle Informationen über angebotene Artikel wie Bezeichnung, Preis und Mehrwertsteuersatz enthalten, eine andere dagegen den Lagerbestand.

► *1:n-Beziehung*

Eine 1:n- oder Eins-zu-viele-Beziehung verbindet einen Datensatz einer Tabelle mit beliebig vielen Datensätzen einer anderen Tabelle. Dies ist der häufigste und nützlichste Verknüpfungstyp, weil er den größten Beitrag zur Vermeidung von Inkonsistenzen leistet: Detailinformationen über Werte, die in einer Spalte einer Tabelle beliebig oft vorkommen können, werden in einer separaten Tabelle erfasst. Die erste Tabelle enthält in dem entsprechenden Feld nur noch einen Verweis auf einen bestimmten Datensatz der zweiten Tabelle.

► *m:n-Beziehung*

Eine m:n-, auch Viele-zu-viele-Beziehung genannt, kombiniert beliebig viele Vorkommen eines bestimmten Wertes mit beliebig vielen Vorkommen eines anderen. Stellen Sie sich beispielsweise eine Tabelle vor, die eine Liste lieferbarer Waren enthält, und eine weitere Tabelle, in der die Adressen von Kunden erfasst werden. Jeder Artikel kann von beliebig vielen Kunden gekauft werden, und jeder Kunde kann beliebig viele unterschiedliche Artikel kaufen. Das relationale Datenbankmodell verlangt allerdings, dass diese Art von Beziehung indirekt dargestellt wird: Eine dritte Tabelle listet die einzelnen Kaufaktionen auf; jeder Datensatz enthält dabei eine Verknüpfung zu einem bestimmten Kunden und eine weitere zu einem einzelnen Artikel. Jede dieser Beziehungen für sich ist eine 1:n-Beziehung; die m:n-Beziehung zwischen Kunden und Artikeln besteht nicht direkt.

Ein einfaches Beispiel

Das unter den m:n-Beziehungen angedeutete Beispiel soll im Folgenden konkret ausgeführt werden: Eine Tabelle enthält Daten über Käufer, die zweite Informationen über die Artikel, und die dritte listet jeden einzelnen Kauf auf. Abbildung 12.1 zeigt schematisch die Beziehungen zwischen den Tabellen.

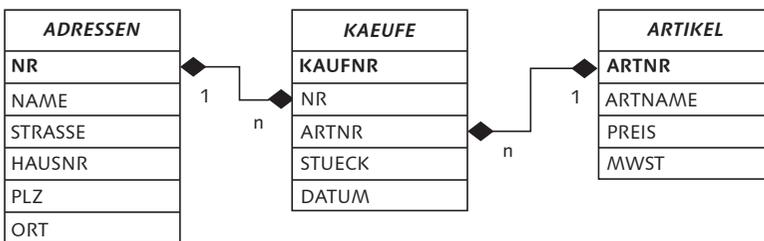


Abbildung 12.1 Beispiel für ein Relationsdiagramm

Das erste, fett gedruckte Feld jeder Tabelle ist der Primärschlüssel. Die Tabelle ADRESSEN enthält die Kundendaten mit dem Primärschlüssel NR. ARTIKEL hat

den Primärschlüssel ARTNR (Artikelnummer). KAEUFE schließlich verwendet einen Primärschlüssel namens KAUFNR. Da keine andere Tabelle auf einzelne Käufe zugreift, benötigt die Tabelle momentan eigentlich keinen Primärschlüssel.

Die Beschriftung an den kleinen Rauten, von denen die Beziehungen ausgehen, zeigt den Beziehungstyp an. Beide Beziehungen in der Datenbank sind 1:n-Beziehungen. Die m:n-Beziehung zwischen den Tabellen ADRESSEN und ARTIKEL kann natürlich nicht eingezeichnet werden, weil sie nur indirekt besteht.

Konkret kann die Tabelle ADRESSEN zum Beispiel mit den Werten aus Tabelle 12.2 gefüllt werden.

NR	NAME	STRASSE	HAUSNR	PLZ	ORT
1	Schmidt	Kleiner Weg	1	50678	Köln
2	Müller	Alte Str.	78	80543	München
3	Becker	Störtebekerweg	45	20567	Hamburg
4	Heinze	Grüne Allee	36	10345	Berlin

Tabelle 12.2 Die Datenbanktabelle ADRESSEN

Natürlich müssen Sie die Kunden nicht durchnummerieren, sondern können sich Ihr eigenes individuelles Schema für Kundennummern ausdenken. Wichtig ist lediglich, dass jede dieser Nummern nur ein einziges Mal in der Tabelle vorkommt und dass jeder Kunde eine Nummer bekommt.

Die Tabelle ARTIKEL enthält Informationen über die angebotenen Artikel (siehe Tabelle 12.3). Beachten Sie, dass das Feld MWST nicht etwa einen konkreten Wert enthält, sondern den Mehrwertsteuersatz, also den Wert 7 oder den Wert 19. Noch praktischer wären allerdings 1:1-Beziehungen zu einer weiteren Tabelle, die die konkreten Mehrwertsteuersätze enthält; schließlich können sich diese ändern. Die Preise werden in Cent angegeben.

ARTNR	ARTNAME	PREIS	MWST
1	Cola	119	19
2	Vollmilch	79	7
3	Toastbrot	149	7
4	Zahnpasta	179	19

Tabelle 12.3 Die Datenbanktabelle ARTIKEL

Nachdem diese beiden Tabellen eingerichtet sind, können nun Käufe getätigt werden. In der Tabelle KAEUFE könnten etwa die folgenden Geschäftsvorfälle erfasst werden:

KAUFNR	NR	ARTNR	STUECK	DATUM
1	3	3	2	2009-09-15
2	2	1	4	2009-09-16
3	2	2	1	2009-09-16
4	1	4	1	2009-09-18
5	1	3	1	2009-09-18

Tabelle 12.4 Die Datenbanktabelle KAEUFE

Die Einträge in der Tabelle KAEUFE sind in dieser Form für Menschen so gut wie unlesbar. Es geht auch gar nicht darum, sie in einer Form zu verwahren, in der sie sofort lesbar sind, sondern darum, die Daten redundanzfrei und kompakt zu speichern. Für die lesbare Ausgabe beherrscht jedes relationale Datenbankverwaltungssystem sogenannte *Auswahlabfragen*, mit deren Hilfe Sie aufgrund der Beziehungen Daten aus verschiedenen Tabellen zusammenstellen können. Wie diese Abfragen in MySQL erstellt werden, erfahren Sie weiter unten in diesem Kapitel.

Tabelle 12.5 zeigt das Ergebnis einer solchen Auswahlabfrage. Hier werden die Käufe mit den eigentlichen Kunden- und Artikelnamen aufgeführt, alphabetisch nach Kunden und anschließend alphabetisch nach Artikelbezeichnungen sortiert. Damit Sie das Ergebnis nachvollziehen können, wird die Kaufnummer aus der Tabelle KAEUFE übernommen. Die letzte Spalte enthält den errechneten Gesamtpreis für jeden einzelnen Kauf (das Produkt aus Stückzahl und Einzelpreis).

KAUFNR	NAME	ARTNAME	STUECK	GESAMTPREIS
1	Becker	Toastbrot	2	298
2	Müller	Cola	4	476
3	Müller	Vollmilch	1	79
5	Schmidt	Toastbrot	1	149
4	Schmidt	Zahnpasta	1	179

Tabelle 12.5 Eine Auswahlabfrage, die die Käufe in lesbarer Form darstellt

Bitte beachten Sie, dass die Ergebnisse von Auswahlabfragen normalerweise nicht abgespeichert werden. Schließlich basieren sie auf Daten, die sich durch die nachträgliche Änderung oder Ergänzung von Datensätzen in den zugrundeliegenden Tabellen jederzeit ändern können.

12.2 Installation und Inbetriebnahme

Das Datenbanksystem MySQL wurde ursprünglich von dem schwedischen Programmierer *Michael »Monty« Widenius* entwickelt. Seit 1995 arbeitet *David Axmark* an dem Projekt mit – er setzte sich dafür ein, MySQL zunächst als binäre Freeware und später als Open-Source-Software unter der GNU GPL zu verbreiten. Für die Weiterentwicklung und Pflege der Software ist die ebenfalls in Schweden ansässige Firma MySQL AB verantwortlich; sie finanziert dies durch kostenpflichtigen Support sowie durch kommerzielle MySQL-Lizenzen. Letztere erlauben einiges, was die GPL ausdrücklich ausschließt – beispielsweise die Einbettung des MySQL-Servers in kommerzielle Software. Die Firma gehört seit 2005 zum Konzern Sun Microsystems (bekannt für Hardware, das Betriebssystem Solaris und die Programmiersprache Java). Dieser wiederum wird zurzeit von dem kommerziellen Datenbankhersteller Oracle übernommen.

In diesem Kapitel erfahren Sie das Wichtigste über die Arbeit mit den MySQL-Tools sowie über die Erstellung und Anwendung von Datenbanken mit der MySQL-spezifischen SQL-Syntax. Als ausführlichere Einführung in MySQL kann ich Ihnen mein Buch »Praktischer Einstieg in MySQL mit PHP« (2. Auflage, Köln 2007, O'Reilly Verlag) empfehlen. Die Druckausgabe ist vergriffen, aber Sie können es unter <http://www.oreilly.de/german/freebooks/einmysql2ger/> kostenlos als PDF herunterladen.

openSUSE 11.2 enthält MySQL in der aktuellen Version 5.1. Falls Sie auf eine neuere Version angewiesen sind, finden Sie weiter unten eine kurze Anleitung zur manuellen Installation.

Der Lieferumfang von MySQL besteht in openSUSE 11.2 aus den in Tabelle 12.6 gezeigten Installationspaketen. Es gibt kein eigenes Schema für MySQL, sodass Sie die Pakete über die Suche nach »mysql« finden müssen. Das Hauptpaket `mysql` finden Sie im Schema *Web- und LAMP-Server* (siehe Kapitel 15).

Für den Betrieb von MySQL selbst benötigen Sie auf jeden Fall die Pakete `mysql`, `mysql-client` und `libmysqlclient15`; wirklich sinnvoll ist eine Datenbank aber nur, wenn Sie Programme schreiben und/oder einsetzen, die darauf zugreifen – die entsprechenden Programmiersprachenbibliotheken sind also auch meistens notwendig.

Paket	Beschreibung
mysql	der MySQL-Server, Version 5.1.36
mysql-client	der offizielle Kommandozeilenclient <code>mysql</code>
mysql-connector-java	der offizielle JDBC-Treiber für MySQL
libmysqlclient15	allgemeine Client-Bibliotheken für MySQL
perl-DBD-mysql	Perl-DBI-Treiber für MySQL
php5-mysql	die <code>mysql</code> -Extension für PHP (siehe Kapitel 15, »LAMP«)
ruby-mysql	MySQL-Treiber für Datenbankverbindungen in Ruby
bytefx-data-mysql	MySQL-Treiber für Microsoft ADO.NET bzw. Mono
qt3-mysql	MySQL-Plug-in für Qt3 (die Bibliothek, auf der KDE basiert; siehe Kapitel 5, »Grafische Oberflächen«)
libqt4-sql-mysql	MySQL-Unterstützung für Qt4
unixODBC	UNIX/Linux-Variante der ursprünglich von Microsoft stammenden Universaldatenbankschnittstelle ODBC (nicht nur für MySQL)

Tabelle 12.6 Die Pakete für MySQL in openSUSE 11.2

Einen kurzen Einstieg in die MySQL-Programmierung am Beispiel PHP finden Sie in Kapitel 15, »LAMP«.

12.3 Eigenschaften von MySQL

MySQL ist ein relationales Datenbankverwaltungssystem (englisch Relational Database Management System, RDBMS) auf Client-Server-Basis. Der Server ist schneller, hat dafür aber einen etwas geringeren Leistungsumfang zu bieten als viele Konkurrenzprodukte. Genauer gesagt unterstützt MySQL unterschiedliche *Storage Engines* für Datenbanktabellen, unter denen zwei besonders hervorzuheben sind: die klassische Engine *MyISAM*, die die höchste Performance besitzt, und der von der Firma InnoDB entwickelte Tabellentyp *InnoDB*, der diverse Funktionen unterstützt, die in anderen SQL-Serversystemen schon seit Längerem selbstverständlich sind. Dazu gehören vor allem *Transaktionen* – die Zusammenfassung beliebig vieler SQL-Anweisungen, die »im Paket« bestätigt oder zurückgesetzt werden können (Commit beziehungsweise Rollback). Da die Firma InnoDB, die InnoDB entwickelt hat, seit einigen Jahren zu Oracle gehört, wurde eine

MySQL-eigene transaktionsfähige Engine namens *Falcon* entwickelt; bisher steht sie aber nur in der Alpha-Version MySQL 6.0 zur Verfügung.

Der Funktionskern des Datenbankservers ist der Daemon `mysqld`. Unter openSUSE befindet er sich im Verzeichnis `/usr/sbin` und wird durch das Startskript `/etc/init.d/mysql` gesteuert. Wie üblich können Sie seinen automatischen Start mithilfe von `chkconfig` oder über den Runlevel-Editor (siehe Kapitel 3, »Systemrundgang und Grundkonfiguration«) beeinflussen; nach der Installation ist der Server zunächst so konfiguriert, dass er in Runlevel 3 und 5 automatisch gestartet wird.

Zum weiteren Lieferumfang von MySQL gehören diverse Tools und Skripten, insbesondere folgende:

- ▶ `/usr/bin/mysql`
Kommandozeilenclient, siehe nächster Unterabschnitt
- ▶ `/usr/bin/mysqladmin`
Konsolen-Administrationsprogramm für den MySQL-Server
- ▶ `/usr/bin/mysqldump`
Hilfsprogramm für den Export von MySQL-Tabellen
- ▶ `/usr/bin/mysql_install_db`
Tool zur Ersteinrichtung der Verwaltungsdatenbank `mysql`; wird beim ersten Aufruf des SUSE-Startskripts für MySQL automatisch aufgerufen.
- ▶ `/usr/bin/mysql_fix_privilege_tables`
Hilfsprogramm, das die User-Tabellen in der Datenbank `mysql` aktualisiert; muss nach einem MySQL-Upgrade ausgeführt werden.

Der MySQL-Server und die diversen Hilfsprogramme verarbeiten beim Start Konfigurationsdateien – standardmäßig die globale Datei `/etc/my.cnf` sowie benutzerspezifische Konfigurationseinstellungen unter `~/.my.cnf`. Einige wenige Optionen für diese Dateien werden weiter unten angesprochen, die meisten sprengen allerdings den Rahmen dieses Kapitels. Für übliche Anwendungsfälle funktioniert MySQL aber auch ganz ohne diese Dateien.

12.3.1 MySQL 5.4 manuell installieren

Falls Sie auf die neuen Features von MySQL 5.4 (zurzeit im Betastadium) angewiesen sind (oder diese einfach einmal ausprobieren möchten), müssen Sie sich diese Version separat herunterladen und von Hand installieren. MySQL gehört zu einer Minderheit von Open-Source-Projekten, die für Linux und andere UNIX-Systeme auch eigene Binaries zur Verfügung stellen. Die hier gezeigte Installationsanleitung gilt übrigens auch für MySQL 6.0.

Besuchen Sie für den Download die Website <http://www.mysql.com>. Klicken Sie auf »Downloads«, dann auf »MySQL Community Server« und unterhalb dieser Kategorie auf »5.4«. Da die MySQL-Website öfter neu strukturiert wird, kann ich hier keine genaueren Hinweise geben. Suchen Sie sich unter den Linux-Downloads Ihre Plattform aus; für die meisten Leser dieses Buches dürften »Linux (x86)« oder »Linux (AMD64 / Intel EM64T)« unter der Rubrik »Linux (non RPM packages) downloads« die passende Wahl darstellen. Die hier beschriebenen Schritte habe ich mit der Version 5.4.3-beta getestet, aber bezüglich ihrer Installation unterscheiden sich die diversen Pakete nicht voneinander.

Bevor Sie MySQL 5.4 installieren, ist es ratsam, einen eventuell bereits laufenden MySQL-Server zu beenden und seinen automatischen Start mittels `chkconfig` oder per Runlevel-Editor zu deaktivieren.

Zuerst sollten Sie mittels `su root` werden. Anschließend wird das soeben heruntergeladene Paket zunächst entpackt:

```
# tar xzvf mysql-5.4.3-beta-linux-i686-glibc23.tar.gz
```

Danach wird das neu erstellte Verzeichnis nach `/usr/local` verschoben, und Sie sollten dorthin wechseln:

```
# mv mysql-5.4.3-beta-linux-i686-glibc23 /usr/local
# cd /usr/local
```

Um die restlichen Schritte einfacher zu bewerkstelligen, sollten Sie nun einen symbolischen Link namens `mysql` erstellen, der auf das Verzeichnis mit dem komplizierten Namen verweist:

```
# ln -s /usr/local/mysql-5.4.3-beta-linux-i686-glibc23 \
    /usr/local/mysql
```

Falls noch kein Benutzer existiert, unter dessen ID der MySQL-Server ausgeführt werden soll, müssen Sie ihn nun mitsamt seiner Gruppe anlegen. Beispiel:

```
# groupadd mysql
# useradd -g mysql mysql
```

Als Nächstes müssen einige Rechte verändert werden, um MySQL abzusichern und sein Funktionieren zu gewährleisten:

```
# chown -R root mysql
# chown -R mysql mysql/data
# chgrp -R mysql mysql
```

Das gesamte Verzeichnis *mysql* gehört also dem Benutzer *root* und der Gruppe *mysql*, während das Unterverzeichnis *data* auf den User *mysql* übertragen wurde, damit der MySQL-Server überhaupt Tabellen erstellen und verändern kann.

Nun müssen Sie das Skript *mysql_install_db* im Unterverzeichnis *scripts* ausführen, um die Verwaltungsdatenbank *mysql* anzulegen:

```
# cd mysql
# scripts/mysql_install_db --user=mysql
```

Damit ist die Installation abgeschlossen, und Sie können den Server starten:

```
# bin/mysqld_safe --user=mysql &
```

Wenn Sie MySQL beim Booten automatisch starten möchten, können Sie das Skript *mysql.server* aus dem Unterverzeichnis *support-files* als Startskript verwenden. Legen Sie dazu als Erstes einen passenden Symlink an:

```
# ln -s /usr/local/mysql/support-files/mysql.server \
  /etc/init.d/mysql154
```

Dieses Skript aktivieren Sie danach mittels *chkconfig* (oder im Runlevel-Editor) für die passenden Runlevels:

```
# chkconfig -a mysql154
```

Als letzten Schritt empfiehlt es sich, einige häufig benutzte Hilfsprogramme aus dem MySQL-Unterverzeichnis *bin* in ein Verzeichnis zu linken, das sich in Ihrem Suchpfad befindet, damit Sie diese Programme aus jedem Arbeitsverzeichnis heraus starten können. Dazu gehören insbesondere der Client *mysql* (siehe nächster Abschnitt) sowie die weiter unten behandelten Administrationstools *mysqladmin* und *mysqldump*. Hier ein Beispiel für *mysql*:

```
# ln -s /usr/local/mysql/bin/mysql /usr/bin/mysql
```

12.4 Der Kommandozeilenclient *mysql*

Der mit der MySQL-Standarddistribution gelieferte Kommandozeilenclient *mysql* eignet sich vor allem zur schnellen Überprüfung von Datenbankstrukturen und -inhalten sowie für Verwaltungsaufgaben. Auch für das interaktive Erlernen und Ausprobieren der Datenbankabfragesprache SQL ist er bestens geeignet. Nach der YaST-Installation von MySQL (oder dem Anlegen eines passenden Symlinks bei manueller Installation) wird er zunächst folgendermaßen gestartet:

```
$ mysql
```

Der MySQL-Standardbenutzer ist auf UNIX-Systemen `root@localhost` – die Authentifizierung erfolgt beim MySQL-Server stets durch die drei Komponenten Benutzername, Hostname und Passwort, wobei Benutzername und Passwort nichts mit den Systembenutzern zu tun haben. Die entsprechenden Daten werden in verschiedenen Tabellen der Verwaltungsdatenbank *mysql* gespeichert (Näheres über die MySQL-Benutzerverwaltung und -Authentifizierung erfahren Sie weiter unten in diesem Kapitel). Dass Sie sich bei einer MySQL-Standardinstallation auf UNIX-Systemen als `root` ohne Passwort anmelden können (und das sogar, ohne im System selbst `root` zu sein!), ist auf Dauer ein ernsthaftes Sicherheitsproblem. Daher sollten Sie zunächst ein Passwort für `root` festlegen. Dies geschieht innerhalb des Kommandozeilenclients mithilfe der folgenden Anweisung:

```
mysql> SET PASSWORD FOR
-> root@localhost=PASSWORD('Passwort');
```

Wie dieses Beispiel zeigt, müssen Sie innerhalb von `mysql` jedes Kommando (außer die unten beschriebenen internen Kürzel) durch ein Semikolon abschließen;  ohne dieses Abschlusszeichen wechselt nur die Zeile und zeigt einen Pfeil (->) als Prompt an, sodass Sie längere Anweisungen übersichtlich in mehreren Zeilen eingeben können.

Anschließend sollten alle Anmeldeöglichkeiten mit leerem Benutzernamen und/oder mit leerem Passwort ausgeschlossen werden. Welche das sind, können Sie mithilfe der folgenden Abfrage auf die Verwaltungstabelle *mysql.user* ermitteln:

```
mysql> SELECT user, host, password
-> FROM mysql.user
-> WHERE user="" OR password="";
```

Nachdem das Passwort für `root@localhost` gesetzt wurde, sollte diese Abfrage bei einer MySQL-Standardinstallation unter Linux/UNIX noch vier Datensätze ergeben. Beispiel:

```
+-----+-----+-----+
| user | host      | password |
+-----+-----+-----+
| root | tux      |          |
|      | tux      |          |
|      | localhost|          |
|      | 127.0.0.1|          |
+-----+-----+-----+
```

Genau diese vier Datensätze werden im nächsten Schritt einfach gelöscht:

```
mysql> DELETE FROM mysql.user
      -> WHERE user="" OR password="";
```

Die hier verwendete SQL-Syntax wird weiter unten erläutert.

Nach der Änderung sollten Sie sich zunächst abmelden und mit Passwort neu anmelden. Geben Sie dazu `exit`, `quit` oder das Kürzel `\q` ein, oder drücken Sie einfach `[Strg]+[D]`:

```
mysql> \q
```

Für den Start von `mysql` mit Passwort gilt danach folgende Syntax:

```
$ mysql -u Benutzername -p
```

Falls der MySQL-Server sich auf einem anderen Rechner im Netzwerk oder Internet befindet, können Sie übrigens die Option `-h Hostname` oder `--host Hostname` hinzufügen. Beispiel:

```
$ mysql -u root -h dbserver.test.local -p
```

Für die Anmeldung als `root` am lokalen MySQL-Server genügt dagegen die folgende Zeile:

```
$ mysql -u root -p
```

Anschließend werden Sie aufgefordert, das Passwort einzugeben. Danach können Sie beliebige SQL-Abfragen sowie interne Befehle des Clients selbst eingeben. Einen Überblick über Letztere liefert das Kommando `help` oder eine seiner Kurzfassungen `\h`, `\?` oder `?`. Beispiel:

```
mysql> \h
```

Jede der eingebauten Anweisungen besitzt ein Kürzel; bei denjenigen, die eine SQL-Anweisungszeile abschließen oder modifizieren, müssen Sie dieses Kürzel sogar standardmäßig verwenden – es sei denn, Sie geben beim Start den Kommandozeilenparameter `--named-commands` ein.

Hier die wichtigsten internen Befehle des Clients im Überblick:

- ▶ `\g (go)`
Dieser Befehl schließt die Eingabe einer Anweisung ab, genau wie das Semikolon. Dies ist besonders praktisch, wenn Sie statt des Semikolons mittels `\d` (siehe unten) ein anderes Zeichen eingestellt haben.
- ▶ `\G (ego)`
Diese Anweisung beendet die Eingabe ebenfalls und stellt die Felder eventu-

eller Ergebnisdatensätze untereinander in einer Tabelle dar. Bei Tabellen mit vielen oder besonders breiten Spalten ist das sehr hilfreich. Ein Beispiel finden Sie im nächsten Abschnitt.

▶ `\c (clear)`

Bricht die Eingabe einer Anweisung ab, ohne diese auszuführen, was nützlich ist, wenn Sie sich bei einer mehrzeiligen Eingabe vertippen.

▶ `\. Dateipfad (source Dateipfad)`

Der Befehl führt den Inhalt der angegebenen Datei als SQL-Anweisungen aus. Derartige Dateien können Sie – beispielsweise als Datenbank-Backups – mithilfe des Dienstprogramms `mysqldump` erstellen. Auf die Syntax dieses Tools wird erst weiter unten näher eingegangen, aber das folgende Beispiel exportiert schon einmal die Datenbank *movies* mit Standardoptionen, sobald Sie das `root`-Passwort eingeben:

```
$ mysqldump -u root -p movies >movies.sql
```

Die Ausgabeumleitung ist hier übrigens notwendig – andernfalls landet der SQL-Code auf der Konsole.

▶ `\! Befehl (system Befehl)`

Führt den angegebenen Shell-Befehl aus. Sehr nützlich ist zum Beispiel `\! clear` zum Löschen des Bildschirms (wobei in der Regel auch `[Strg]+[L]` funktioniert).

▶ `\T Dateipfad (tee Dateipfad)`

Schreibt sämtliche Ein- und Ausgaben in die angegebene Protokolldatei.

▶ `\t (notee)` beendet die Protokollierung wieder.

▶ `\u Datenbank (use Datenbank)`

Legt die angegebene Datenbank als Standard fest. Das bedeutet, dass Sie auf Tabellen dieser Datenbank zugreifen können, ohne *Datenbank.Tabelle* zu schreiben, und dass neue Tabellen automatisch in dieser Datenbank erzeugt werden.

▶ `\d Zeichenfolge (delimiter Zeichenfolge)`

Stellt statt des Semikolons die angegebene Zeichenfolge als Befehlsabschluss ein. Dies ist unter anderem zur Eingabe sogenannter Stored Procedures notwendig, da sie Semikola enthalten – diese automatisierten SQL-Befehlsabfolgen werden ab MySQL 5.0 unterstützt, in diesem Buch aber nicht behandelt.

▶ `\s (status)`

Dieser Befehl zeigt einen Statusüberblick des MySQL-Servers an:

```
mysql> status
mysql Ver 14.12 Distrib 5.0.51a, for suse-linux-gnu (i686) using
```

```

readline 5.2

Connection id:          3
Current database:
Current user:          root@localhost
SSL:                   Not in use
Current pager:         less
Using outfile:         ''
Using delimiter:       ;
Server version:        5.0.51a
Protocol version:      10
Connection:            Localhost via UNIX socket
Server character set:  latin1
Db character set:      latin1
Client character set:  latin1
Conn. character set:   latin1
UNIX socket:           /var/lib/mysql/mysql.sock
Uptime:                1 hour 6 min 20 sec

Threads: 1 Questions: 3 Slow queries: 0 Opens: 11 Flush
tables: 1 Open tables: 0 Queries per second avg: 0.001

```

Hier noch einige nützliche MySQL-Befehle, die Ihnen das Leben mit der Datenbank und dem Kommandozeilenclient erleichtern:

- ▶ `SHOW DATABASES`
Dieser Befehl zeigt eine Liste aller vorhandenen Datenbanken an, darunter auch die Verwaltungsdatenbank *mysql*, die Ausprobierdatenbank *test* und die virtuelle, nur lesbare Informationsdatenbank *INFORMATION_SCHEMA*.
- ▶ `SELECT DATABASE()`
Zeigt an, welche Datenbank zurzeit als Standard ausgewählt ist.
- ▶ `SHOW TABLES`
Zeigt eine Liste aller Tabellen der aktuellen Datenbank an.
- ▶ `SHOW TABLES FROM db`
Mit diesem Befehl wird dagegen eine Liste aller Tabellen der angegebenen Datenbank geliefert.

12.5 MySQL-Datenbanken erstellen und verwalten

Wie der Name des Servers schon vermuten lässt, verwendet MySQL die Standardsprache für relationale Datenbanken: die bereits in den 1970er-Jahren von *Edgar F. Codd* entworfene *Structured Query Language* (SQL). Wie fast jedes

Datenbanksystem besitzt allerdings auch MySQL seinen eigenen SQL-Dialekt mit Besonderheiten und spezifischen Erweiterungen. In diesem Abschnitt lernen Sie die wichtigsten SQL-Anweisungen zum Erstellen von Datenbanken und Tabellen, zum Abrufen ausgewählter Informationen sowie zum Ändern von Daten und Strukturen kennen. Sie können die hier vorgestellten Abfragen in den oben vorgestellten Kommandozeilenclient `mysql` oder in grafische Clients wie das webbasierte `phpMyAdmin` (siehe Kapitel 15, »LAMP«) eingeben, aber auch über die diversen Programmierschnittstellen in eigenen Programmen oder Skripten verwenden. Bei der Eingabe in `mysql` müssen Sie an das abschließende Semikolon denken.

In diesem Buch werden die Namen von SQL-Anweisungen und Funktionen vollständig in Großbuchstaben geschrieben; dies ist allgemein üblich. Eigentlich spielen Groß- und Kleinschreibung dabei aber keine Rolle – anders als bei Datenbank- und Tabellennamen, für die daher ausschließlich Kleinbuchstaben zum Einsatz kommen.

12.5.1 Datenbanken und Tabellen erstellen

Eine kurze Übersicht über die Theorie der relationalen Datenbanken haben Sie bereits zu Beginn dieses Kapitels erhalten. Deshalb finden Sie hier vor allem einen praktischen Überblick über die MySQL-Syntax.

Die grundlegende SQL-Anweisung zur Erstellung einer neuen MySQL-Datenbank lautet:

```
CREATE DATABASE Datenbankname
```

Diese Abfrage kennt nur wenige Optionen, da die meisten Einstellungen auf der Ebene der einzelnen Tabelle vorgenommen werden. Die wichtigsten möglichen Angaben sind folgende:

- ▶ `[DEFAULT] CHARACTER SET Zeichensatz`
Gibt den Standardzeichensatz für die Tabellen der Datenbank an. Eine Übersicht über die möglichen Zeichensätze mit Erläuterungen liefert folgende Anweisung im `mysql`-Client:

```
mysql> SHOW CHARACTER SET;
```

Wenn Sie gar keine Angaben machen, ist der Standardzeichensatz `latin1`; für Datenbanken mit deutschsprachigen Inhalten ist dies in Ordnung.

- ▶ `COLLATE Sortierfolge`
Legt die Standardsortierfolge (Kollation) für Textinhalte fest. Geben Sie im Client `mysql` Folgendes ein, um eine Liste aller Kollationen zu erhalten:

```
mysql> SHOW COLLATION;
```

Da MySQL aus Schweden stammt, ist die Standardkollation `latin1_swedish_ci`; `ci` steht dabei für »case insensitive«, also ohne Unterscheidung von Groß- und Kleinschreibung. Für Deutsch gibt es gleich zwei mögliche Kollationen: `latin1_german1_ci` sortiert nach Wörterbuch – ä, ö, ü und ß werden mit a, o, u beziehungsweise s gleichgesetzt (beziehungsweise in ansonsten identischen Wörtern dahinter einsortiert – zum Beispiel »Bar«, »Bär«). `latin1_german2_ci` ist dagegen die Telefonbuch-Sortierreihenfolge, bei der Umlaute und ß wie ae, oe, ue beziehungsweise ss behandelt werden.

- ▶ `IF NOT EXISTS` (vor dem Datenbanknamen)
Erstellt die Datenbank nur dann, falls noch keine Datenbank mit der angegebenen Bezeichnung existiert.

Das folgende Beispiel erstellt eine Datenbank namens *movies* mit dem Zeichensatz `latin1` und der Sortierfolge Deutsch (Wörterbuch), falls sie noch nicht existiert:

```
CREATE DATABASE IF NOT EXISTS movies
CHARACTER SET latin1 COLLATE latin1_german1_ci
```

Wenn Sie diese Erstellungsabfrage im Kommandozeilenclient `mysql` eingeben (Semikolon nicht vergessen!), sollten Sie die neue Datenbank im nächsten Schritt als Standard auswählen (das wiederum geht ohne Semikolon, weil es ein interner Befehl des Clients ist):

```
mysql> use movies
```

Nachdem die Datenbank existiert, können Sie darin beliebig viele Tabellen erstellen. Dies geschieht mithilfe von `CREATE TABLE`-Abfragen, deren Minimalsyntax schematisch so aussieht:

```
CREATE TABLE Tabellename
(
    Spaltenname Datentyp [Optionen] [,
    Spaltenname Datentyp [Optionen], ...]
)
```

Bevor Details vertieft werden, hier ein erstes Beispiel – es erstellt die Tabelle `mv_actors` mit Informationen über Schauspielerinnen und Schauspieler:

```
CREATE TABLE mv_actors
(
    act_id INT AUTO_INCREMENT,
    act_name VARCHAR(40),
```

```

act_vorname VARCHAR(30),
act_gdatum DATE,
act_geschlecht ENUM('w','m'),
act_land INT,
PRIMARY KEY (act_id),
INDEX (act_name)
);

```

Es ist empfehlenswert, allen Tabellen einer Datenbank sowie allen Spalten einer Tabelle wie hier jeweils ein gemeinsames Namenspräfix zuzuweisen. Dies macht den Zusammenhang zwischen den Daten auf den ersten Blick deutlich. Zudem verhindert es bei gleichartigen Spalten verschiedener Tabellen die Notwendigkeit, aus Gründen der Eindeutigkeit jedes Mal `Tabellenname.Spaltenname` zu schreiben, da Sie hinter dem Präfix einfach denselben Grundnamen verwenden können.

Die Tabelle `mv_actors` enthält folgende Spalten:

- ▶ `act_id`
Stellt den sogenannten Primärschlüssel dar. Dieser ist ein eindeutiger Wert, der den gesamten Datensatz repräsentiert und damit in anderen Tabellen als Verweis auf die jeweiligen Schauspieler dienen kann. Der Datentyp dieser Spalte ist `INT` (Ganzzahl); die Option `AUTO_INCREMENT` sorgt dafür, dass die Felder dieser Spalte automatisch durchnummeriert werden.
- ▶ `act_name`
Der Nachname des jeweiligen Schauspielers vom Typ `VARCHAR(40)`. Es handelt sich dabei um einen String variabler Länge mit bis zu 40 Zeichen.
- ▶ `act_vorname`
Enthält die Vornamen der Schauspieler und muss mit maximal 30 Zeichen auskommen.
- ▶ `act_gdatum`
Speichert die Geburtsdaten; der Datentyp `DATE` kann ein beliebiges Kalenderdatum enthalten.
- ▶ `act_geschlecht`
Diese Spalte ist vom Datentyp `ENUM`, der auf sehr speicherplatzsparende Weise einen Wert aus einer vorgegebenen Liste von Alternativen speichern kann.
- ▶ `act_land`
Enthält das Land, aus dem der Schauspieler stammt. Dabei wird hier nicht der Name des Landes gespeichert, sondern eine ganze Zahl. Diese verweist als sogenannter Fremdschlüssel auf eine noch anzulegende Tabelle, die die Namen und gegebenenfalls weitere Informationen über die einzelnen Länder

speichert. Da auch jeder Film, jeder Regisseur und so weiter mit einem bestimmten Land verbunden ist, bietet es sich an, die Länder auszulagern. Dies vermeidet Redundanzen und damit eventuelle Inkonsistenzen.

Die Tabelle enthält zwei Indizes:

1. Der bereits erwähnte Primärschlüssel ist ein Sonderfall eines eindeutigen Indexes (`UNIQUE`) – jeder Wert darf in der gesamten Tabelle nur einmal vorkommen.
2. Auf das Feld `act_name` wurde ein einfacher Index gesetzt. Dieser beschleunigt die Suche und das Sortieren nach den Nachnamen. Einen solchen Index können Sie sich genauso vorstellen wie den alphabetischen Index am Ende dieses Buches – er speichert die verschiedenen Werte des Feldes erneut und merkt sich dabei die Stellen, an denen sie in der Tabelle vorkommen.

Beachten Sie, dass Indizes die Größe Ihrer Tabellen vervielfachen – daher sollten Sie sie maßvoll einsetzen, das heißt, auf diejenigen Felder beschränken, nach denen besonders häufig gesucht wird.

InnoDB-Tabellen bieten übrigens auch die Möglichkeit, die Bezüge zwischen verschiedenen Tabellen durch definierte Fremdschlüssel – sogenannte Constraints – abzusichern. Dieses Thema würde hier allerdings zu weit führen. Bitte konsultieren Sie dazu die MySQL-Online-Dokumentation oder die in Anhang B empfohlene Literatur.

Im Übrigen wäre es auch Platzverschwendung, hier Erstellungsabfragen für sämtliche Tabellen der Datenbank *movies* abzudrucken. Daher finden Sie unter den Listings auf der Website zum Buch die SQL-Datei *movies.sql*, die die gesamte Datenbank in Ihrem MySQL-Server erzeugt. Innerhalb des Kommandozeilenclients können Sie sie wie folgt importieren:

```
mysql> source movies.sql;
```

oder kurz

```
mysql> \. movies.sql
```

Auch für Tabellen können Sie die Klausel `IF NOT EXISTS` verwenden, um eine Tabelle nur zu erzeugen, falls sie noch nicht besteht. Hier ein verkürztes Beispiel:

```
CREATE TABLE IF NOT EXISTS test
(...)
```

Hinter der schließenden Klammer, die die Auflistung der Spalten und Indizes beendet, können Optionen für die Tabelle als solche folgen. Die wichtigsten sind:

- ▶ CHARACTER SET *Zeichensatz*
Diese Option legt den Standardzeichensatz auf Tabellenebene fest; sie wurde weiter oben für Datenbanken erläutert.
- ▶ COLLATE *Sortierfolge*
Dies ist die Standardsortierreihenfolge der Tabelle; auch diese Einstellung wurde bereits als Datenbankoption vorgestellt.
- ▶ ENGINE= *Tabellentyp*
Mithilfe dieser Option können Sie den Typ (Storage Engine) der Tabelle festlegen. Wie bereits erwähnt, besitzt der Standardtyp¹ MyISAM eine bessere Performance, während InnoDB mehr Funktionen unterstützt; in MySQL 6.0 steht zusätzlich Falcon zur Verfügung. Daneben gibt es noch einige weitere, nicht ganz so wichtige Tabellentypen.

Beispiel: Engine=InnoDB

Wenn Sie eine Tabelle erzeugt haben, können Sie sich im Client `mysql` folgendermaßen ihre Struktur anzeigen lassen (hier um eine Spalte gekürzt):

```
mysql> DESCRIBE mv_actors;
+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default |
+-----+-----+-----+-----+-----+
| act_id         | int(11)       |      | PRI | NULL     |
| act_name       | varchar(40)   | YES  | MUL | NULL     |
| act_vorname    | varchar(30)   | YES  |     | NULL     |
| act_gdatum     | date          | YES  |     | NULL     |
| act_geschlecht | enum('w','m') | YES  |     | NULL     |
| act_land       | int(11)       | YES  |     | NULL     |
+-----+-----+-----+-----+-----+
```

Noch genauer ist eine `SHOW CREATE TABLE`-Abfrage – sie zeigt eine Abfrage an, die die Struktur der entsprechenden Tabelle ergäbe, nicht etwa den Wortlaut der tatsächlich verwendeten Abfrage. Beispiel:

```
mysql> SHOW CREATE TABLE mv_actors \G
***** 1. row *****
Table: mv_actors
Create Table: CREATE TABLE `mv_actors` (
  `act_id` int(11) NOT NULL auto_increment,
  `act_name` varchar(40) collate latin1_german1_ci
  default NULL,
```

1 Vorsicht beim Systemwechsel: In MySQL für Windows ist die Standard-Engine InnoDB. Daher ist es empfehlenswert, auch den vermeintlichen Standardtabellentyp immer explizit anzugeben.

```

`act_vorname` varchar(30) collate
latin1_german1_ci default NULL,
`act_geburtsdatum` date default NULL,
`act_geschlecht` enum('w','m') collate
latin1_german1_ci default NULL,
`act_land` int(11) default NULL,
PRIMARY KEY (`act_id`),
KEY `act_name` (`act_name`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1
COLLATE=latin1_german1_ci

```

Wie Sie sehen, werden hier auch Optionen angezeigt, die aufgrund von Standard-einstellungen der übergeordneten Datenbank oder des Servers selbst eigentlich automatisch zustande kommen.

12.5.2 Daten einfügen

Nachdem eine Tabelle erstellt wurde, können Sie mithilfe von Einfügeabfragen (SQL-Schlüsselwort `INSERT`) neue Datensätze erzeugen. Die Syntax dieser Anweisung lautet wie folgt:

```

INSERT INTO Tabellename [(Spalte1, Spalte2, ...)]
VALUES (Wert1, Wert2, ...)[,
      (Wert1, Wert2, ...), ...]

```

Die Liste der Spaltennamen brauchen Sie nur dann zu verwenden, wenn nicht für alle Spalten Werte eingefügt werden. Das folgende Beispiel fügt einen neuen Datensatz in der Tabelle `mv_actors` hinzu – hier wird die Spaltenliste verwendet, weil `act_id` automatisch einen Wert erhält und daher ausgelassen wird:

```

INSERT INTO mv_actors
(act_name, act_vorname, act_gdatum,
 act_geschlecht, act_land) VALUES
("De Niro", "Robert", "1943-08-17", "m", 2)

```

Dass es funktioniert hat, zeigt im `mysql`-Client eine Meldung wie diese:

```
Query OK, 1 row affected (0.00 sec)
```

Wenn Sie die Liste der Spalten nicht angeben möchten, können Sie für das `AUTO_INCREMENT`-Feld auch einfach den Wert `NULL` angeben. Beispiel:

```

INSERT INTO mv_actors VALUES
(NULL, "Pitt", "Brad", "1963-12-18", "m", 2)

```

Optional lassen sich – durch Komma getrennt – beliebig viele Datensätze gleichzeitig hinzufügen. Hier ein Beispiel mit zwei weiteren:

```
INSERT INTO mv_actors
(act_name, act_vorname, act_gdatum,
 act_geschlecht, act_land) VALUES
("Jolie", "Angelina", "1975-06-04", "w", 2),
("Jackson", "Samuel L.", "1948-12-21", "m", 2)
```

Wie Sie sehen, gehören Strings sowie Datums- und Uhrzeitangaben in Anführungszeichen. Bei MySQL ist es egal, ob Sie einfache oder doppelte Anführungszeichen verwenden, anders als bei vielen anderen SQL-Datenbanken wie etwa PostgreSQL. Der ANSI-SQL-Standard schreibt einfache Anführungszeichen vor, weil doppelte für Tabellen- und Spaltenbezeichnungen reserviert sind, die Leerzeichen enthalten oder genauso heißen wie offizielle SQL-Bezeichner. Dafür verwendet MySQL allerdings ebenfalls nicht standardkonform ``Backticks``.

12.5.3 MySQL-Datentypen

Für die unterschiedlichen Arten von Daten in einer Datenbank sind jeweils andere Datentypen geeignet. MySQL kennt eine Reihe von Typen, die sich zunächst in folgende Kategorien unterteilen lassen:

- ▶ Numerische Typen
- ▶ String-Datentypen
- ▶ Datums- und Uhrzeittypen
- ▶ Text- und Binärblöcke
- ▶ Aufzählungstypen

Numerische Datentypen

Die numerischen Datentypen lassen sich näher in Integer- und Fließkommatypen unterteilen. Die *Integertypen* speichern ganze Zahlen mit unterschiedlicher Bitbreite, aus der sich der speicherbare Wertebereich ergibt. In Tabelle 12.7 finden Sie eine entsprechende Übersicht. Der Unsigned-Wertebereich in der letzten Spalte betrifft die Option `UNSIGNED`, die bei der Definition einer Spalte angegeben werden kann.

Beim Versuch, zu große oder zu kleine Werte in eine Spalte mit Integer-Datentyp einzufügen, wird der größt- beziehungsweise kleinstmögliche Wert verwendet. Der Kommandozeilenclient `mysql` zeigt in einem solchen Fall an, dass es eine Warnung gibt.

Datentyp	Bit	Wertebereich	Unsigned-Wertebereich
TINYINT	8	128 bis +127	0 bis 255
SMALLINT	16	32.768 bis +32.767	0 bis 65.535
MEDIUM-INT	24	8.388.608 bis +8.388.607	0 bis 16.777.215
INT	32	2.147.483.648 bis +2.147.483.647	0 bis 4.294.967.295
BIGINT	64	9.223.372.036.854.775.808 bis + 9.223.372.036.854.775.807	0 bis 18.446.744.073.709.551.615

Tabelle 12.7 Ganzzahlige MySQL-Datentypen

Es folgt ein kurzes Kompletbeispiel. Zunächst wird (in einer beliebigen Datenbank – praktisch ist etwa die automatisch vorhandene »Ausprobierdatenbank« *test* – eine neue Tabelle erstellt:

```
mysql> CREATE TABLE tinytest
-> (
->   i1 TINYINT,
->   i2 TINYINT UNSIGNED
-> );
```

Anschließend werden folgende Daten eingefügt:

```
mysql> INSERT INTO tinytest VALUES
-> (0, 0),
-> (-1, -1),
-> (-129, -1),
-> (128, 256);
```

MySQL beschwert sich mit der folgenden Meldung:

```
Query OK, 4 rows affected, 5 warnings (0.03 sec)
```

Den Inhalt der fünf Warnungen erfahren Sie mithilfe der folgenden Anweisung, falls Sie sie unmittelbar als nächste ausführen:

```
mysql> SHOW WARNINGS;
+-----+-----+-----+
| Level | Code | Message |
+-----+-----+-----+
| Warning | 1264 | Data truncated; out of |
| | | range for column 'i2' at row 2 |
+-----+-----+-----+
```

```
[...]
5 rows in set (0.00 sec)
```

Die Ausgabe passt gerade noch in ein Standard-Terminalfenster, ist aber zu breit für den Satzspiegel dieses Buches, daher habe ich hier ein wenig »gepfuscht« und eine Meldung in zwei Zeilen unterteilt. Falls Tabellen auch im Terminal zu breit sind, können Sie die Daten stattdessen untereinander ausgeben, indem Sie die Anweisung mit `\G` abschließen:

```
mysql> SHOW WARNINGS \G
***** 1. row *****
Level: Warning
Code: 1264
Message: Data truncated; out of range for column 'i1' at row 1
[...]
5 rows in set (0.00 sec)
```

Für *Fließkommazahlen* stehen zwei Typen zur Verfügung: `FLOAT` mit 32 Bit und `DOUBLE` (Synonym `REAL`) mit 64 Bit. Der Unterschied liegt hier in der Genauigkeit, das heißt, es werden unterschiedlich viele Stellen gespeichert. Intern werden Fließkommazahlen nämlich im dualen Exponentialformat ($x * 2^n$) abgelegt. Beachten Sie, dass statt des Kommas, genau wie in Programmiersprachen, ein Punkt verwendet werden muss (entsprechend lautet die englische Bezeichnung »Floating Point Numbers«).

Eine Besonderheit stellt der Datentyp `DECIMAL` dar: Er bietet eine Speichermöglichkeit für *Festkommazahlen*, bei denen die Anzahl der Nachkommastellen nicht variiert. Sie können (und sollten) bei der Definition von `DECIMAL`-Feldern die Gesamtzahl der Stellen und die Anzahl der Nachkommastellen angeben – dies geschieht mithilfe der Schreibweise `DECIMAL(m,n)`. Ein Feld mit der Angabe `DECIMAL(5,2)` kann beispielsweise Werte zwischen $-999,99$ und $9.999,99$ speichern – die zusätzliche Stelle, die das Minuszeichen einnimmt, wird nicht mitgezählt, obwohl sie im positiven Bereich eine weitere Ziffernstelle bietet. Ein naheliegendes Anwendungsbeispiel für `DECIMAL` sind Währungsbeträge.

String-Datentypen

Für (relativ kurze) *Strings* gibt es die beiden Datentypen `CHAR(n)` und `VARCHAR(n)`. `CHAR`-Felder können bis zu 255 Zeichen enthalten und belegen stets die bei der Definition angegebene Zeichenanzahl, auch wenn ihr konkreter Inhalt kürzer ist. `VARCHAR`-Felder verwenden dagegen nur den Speicher, den die aktuell enthaltenen Zeichen benötigen; der Zahlenwert (1 bis 65.535) gibt hier die maximal mögliche Anzahl von Zeichen an. Ein Feld mit der Definition `CHAR(100)` ist also immer 100 Zeichen breit, während `VARCHAR(100)` Platz für bis zu 100 Zei-

chen bietet. Im Allgemeinen lässt sich feststellen, dass CHAR-Felder auch ohne Index schneller verarbeitet werden, dafür spart der Einsatz von VARCHAR Speicherplatz.

Datums- und Uhrzeitdatentypen

Für Datum und Uhrzeit stehen folgende Typen zur Verfügung:

- ▶ DATETIME
Speichert eine vollständige Zeitangabe aus Datum und Uhrzeit im Format "JJJJ-MM-TT hh:mm:ss". Beispiel: "2007 - 09 - 10 10:59:17". Wenn Sie beim Einfügen von Werten die Uhrzeit weglassen, wird sie auf 00:00:00 gesetzt.
- ▶ DATE
Dieser Typ enthält nur ein Datum wie "2007 - 09 - 10".
- ▶ TIME
Speichert eine Uhrzeit, zum Beispiel "10:59:17".
- ▶ TIMESTAMP
TIMESTAMP besitzt dasselbe Format wie DATETIME. Die Besonderheit besteht darin, dass TIMESTAMP-Felder beim Einfügen und bei jeder Änderung des entsprechenden Datensatzes mit der aktuellen Systemzeit aktualisiert werden.

Text- und Binärblocktypen

Für umfangreichere *Text- oder Binärinhalte* stehen die in Tabelle 12.8 gezeigten TEXT- beziehungsweise BLOB-Typen (*Binary Large Objects*) zur Verfügung. Die angegebene Maximalgröße bezieht sich bei den Textblöcken auf Zeichen und bei den Binärblöcken auf Bytes.

Textblocktyp	Binärblocktyp	Maximale Länge
TINYTEXT	TINYBLOB	255
TEXT	BLOB	65.535
MEDIUMTEXT	MEDIUMBLOB	16.777.215
LONGTEXT	LOB	4.294.967.295

Tabelle 12.8 TEXT- und BLOB-Datentypen

Aufzählungstypen

Die letzte Gruppe, die *Aufzählungstypen*, sind in dieser Form eine Besonderheit von MySQL und gehören nicht zum SQL-Standard. Es gibt die beiden Varianten ENUM und SET. Wenn Sie eine Spalte als ENUM('Wert1', 'Wert2', 'Wert3', ...)

definieren, können die Felder dieser Spalte jeweils einen der angegebenen Werte annehmen. Intern werden sie als Verweise auf eine durchnummerierte Liste gespeichert und verbrauchen so nur wenig Speicher. Im vorigen Abschnitt wurde etwa das Beispiel `act_geschlecht ENUM('m', 'w')` verwendet. Hier würde 'm' der Wert »1« und 'w' der Wert »2« zugewiesen.

Bei Spalten vom Typ `SET('Wert1', 'Wert2', 'Wert3', ...)` kann ein Feld mehrere der Vorgabewerte annehmen. Dazu werden die Werte als einzelne Bits gespeichert, sodass sich die gewählten oder nicht gewählten Werte als Bitmuster darstellen. Angenommen, eine Spalte beschreibt mögliche Tonspurformate für eine Video-DVD und besitzt die Definition `SET('mono', 'stereo', '5.1')`. Für eine DVD mit zwei Audiotracks in den Formaten `mono` und `5.1` würde der Wert als 'mono', '5.1' angegeben; intern würde der Binärwert 101 (dezimal 5) gespeichert.

12.5.4 Auswahlabfragen

Mithilfe der SQL-Anweisung `SELECT` können Sie Informationen aus den Tabellen Ihrer Datenbanken ermitteln oder auch beliebige Ausdrücke auswerten lassen. Die einfachste Form einer `SELECT`-Abfrage lautet:

```
SELECT * | Spalte1[, Spalte2, ...] FROM Tabelle;
```

Ein `*` steht für alle Spalten; alternativ können Sie eine Liste der gewünschten Spalten angeben. Hier eine Abfrage, die nur einige Spalten der weiter oben abgedruckten Beispieleinträge aus der Tabelle `mv_actors` anzeigt:

```
mysql> SELECT act_name, act_vorname, act_gdatum
-> FROM mv_actors;
```

```
+-----+-----+-----+
| act_vorname | act_name | act_gdatum |
+-----+-----+-----+
| Robert      | De Niro  | 1943-08-17 |
| Brad        | Pitt     | 1963-12-18 |
| Angelina    | Jolie    | 1975-06-04 |
| Samuel L.   | Jackson  | 1948-12-21 |
+-----+-----+-----+
```

Kriterien mit WHERE festlegen

Das Verhalten von `SELECT` lässt sich durch zahlreiche Klauseln verändern. Die wichtigste ist `WHERE`; sie gibt Kriterien an, auf deren Basis die Auswahl bestimmter Datensätze erfolgt. Hier ein Beispiel, das die älteren (vor 1950 geborenen) Schauspieler anzeigt:

```
mysql> SELECT act_name, act_vorname, act_gdatum
      -> FROM mv_actors WHERE act_gdatum < "1950";
+-----+-----+-----+
| act_vorname | act_name | act_gdatum |
+-----+-----+-----+
| Robert      | De Niro  | 1943-08-17 |
| Samuel L.   | Jackson | 1948-12-21 |
+-----+-----+-----+
```

Zur Formulierung der WHERE-Klauseln werden am häufigsten – wie hier – Vergleichsoperatoren verwendet. Dies sind unter anderem = (gleich), != (ungleich), < (kleiner), > (größer), <= (kleiner oder gleich) und >= (größer oder gleich). Neben Zahlen und Datums- sowie Uhrzeitangaben können Sie auch Strings vergleichen; zur Ermittlung der Reihenfolge wird die jeweilige Kollation berücksichtigt.

Einen Schritt weiter gehen Mustervergleiche mit LIKE. Ein Suchmuster kann die beiden Platzhalter % (beliebig viele beliebige Zeichen) und _ (genau ein beliebiges Zeichen) enthalten – damit entsprechen sie den Shell-Wildcards * beziehungsweise ?. Das folgende Beispiel wählt die beiden Schauspieler aus, deren Nachnamen mit J anfangen:

```
mysql> SELECT act_vorname, act_name
      -> FROM mv_actors WHERE act_name LIKE "J%";
+-----+-----+
| act_vorname | act_name |
+-----+-----+
| Samuel L.   | Jackson |
| Angelina    | Jolie    |
+-----+-----+
```

Der Operator REGEXP ermöglicht sogar den Vergleich mit POSIX-konformen regulären Ausdrücken. Dies ist eine MySQL-eigene Erweiterung, die nicht zum ANSI-SQL-Standard gehört.

Operatoren und Funktionen

Im Übrigen gibt es unzählige MySQL-Operatoren und -Funktionen, deren Ergebnisse Sie zu Ausdrücken verbinden können. Hier nur einige Beispiele:

- ▶ *Arithmetische Operationen*
+, -, *, / und % (Modulo, also der Rest einer ganzzahligen Division)
- ▶ *Mathematische Funktionen*
Zum Beispiel POW(x,y) für die Potenz x^y , SIN(x) für den Sinus, ROUND(x,n) zum Runden auf n Nachkommastellen oder PI() für den Wert der Kreiszahl p.

► *String-Funktionen*

Dazu gehören `CONCAT(str1, str2, ...)` zum Verbinden beliebig vieler Einzelstrings (Synonym: `str1 || str2`), `LENGTH(str)` zur Ermittlung der String-Länge oder `SUBSTRING(str,pos[,n])` für einen (n Zeichen langen) Teilstring von `str` ab `pos`.

► *Datums- und Uhrzeitfunktionen*

Zu den wichtigen gehören: `DATE_FORMAT(Datum,Formatstring)` zur Formatierung eines Datums per `date(1)`-kompatiblem Formatstring, `UNIX_TIMESTAMP(Datum)` zur Umwandlung einer MySQL-Datumsangabe in Sekunden seit EPOCH sowie `FROM_UNIXTIME(sek_seit_epoch[, Formatstring])` für die umgekehrte Berechnung.

► *Logische Operatoren*

Ausdrücke lassen sich per `AND` (alle müssen zutreffen) und `OR` (mindestens einer muss wahr sein) kombinieren oder durch ein vorangestelltes `NOT` negieren.

Achtung! Beachten Sie bei SQL-Funktionen, dass zwischen dem Funktionsnamen und der öffnenden Klammer kein Leerzeichen stehen darf! Es muss beispielsweise `UNIX_TIMESTAMP("2007-09-15")` heißen.

Komplexe Ausdrücke können nicht nur in der `WHERE`-Klausel, sondern auch in der `SELECT`-Anweisung selbst verwendet werden. Das folgende Beispiel kombiniert die Namen der ersten beiden Schauspieler im Format »Vorname Nachname«:

```
mysql> SELECT CONCAT(act_vorname, " ", act_nachname)
      -> FROM mv_actors LIMIT 0,2;
+-----+
| CONCAT(act_vorname, " ", act_name) |
+-----+
| Robert De Niro                      |
| Brad Pitt                           |
+-----+
```

Etwas unschön ist hier die Spaltenüberschrift im Ergebnis. Für solche Fälle steht die Klausel `AS` bereit, die Ergebnisse umbenennt. Hier als Beispiel die beiden nächsten Schauspieler:

```
mysql> SELECT CONCAT(act_vorname, " ", act_name)
      -> AS Schauspieler FROM mv_actors LIMIT 2,2;
+-----+
| Schauspieler                        |
+-----+
| Angelina Jolie                      |
| Samuel L. Jackson                  |
+-----+
```

In den meisten Fällen können Sie das `AS` selbst sogar weglassen und das Alias einfach hinter die Tabellen- oder Spaltenbezeichnung schreiben:

```
mysql> SELECT CONCAT(act_vorname, " ", act_name)
      -> Schauspieler FROM mv_actors LIMIT 2,2;
+-----+
| Schauspieler |
+-----+
| Angelina Jolie |
| Samuel L. Jackson |
+-----+
```

Sobald die gewünschte Beschriftung Leerzeichen enthält, müssen Sie sie übrigens in Anführungszeichen setzen.

Joins

Eine besonders komplexe Form der Auswahlabfrage sind die sogenannten *Joins*, die Werte aus mehreren Tabellen miteinander verknüpfen. Im Grunde ist dies der Hauptnutzen relationaler Datenbanken. Joins lassen sich entweder durch die spezielle `JOIN`-Syntax oder mittels `WHERE` beschreiben. In beiden Fällen muss eine Spalte einer Tabelle zum Primärschlüssel einer anderen Tabelle in Beziehung gesetzt werden.

Eine einfache 1:n-Beziehung, die sich durch einen Join zwischen zwei Tabellen ausdrücken lässt, ist in der Filmdatenbank die Ausgabe des Regisseurs jedes Films – der Sonderfall mehrerer Regisseure für einen Film wurde einfach weggelassen.² Die folgende Abfrage wählt die Regisseure aller in der Datenbank gespeicherten Filme aus, beschränkt die Ausgabe aber mithilfe einer `LIMIT`-Klausel auf die ersten zwei Filme:

```
mysql> SELECT mov_titel Film,
      -> CONCAT(dir_vorname, " ", dir_name) Regisseur
      -> FROM mv_movies JOIN mv_directors
      -> ON mov_director=dir_id
      -> LIMIT 0,2;
```

2 Das Dilemma, einen meiner Lieblingsfilme – »The Matrix« – dadurch nicht aufnehmen zu können, habe ich umgangen: Die Gebrüder Wachowski werden einfach als »ein Regisseur« geführt.

```

+-----+-----+
| Film           | Regisseur      |
+-----+-----+
| Star Wars Episode III | George Lucas |
| Brazil         | Terry Gilliam  |
+-----+-----+

```

Mithilfe der `WHERE`-Syntax lässt sich diese Abfrage auch so schreiben:

```

SELECT mov_titel Film,
CONCAT(dir_vorname, " ", dir_name) Regisseur
FROM mv_movies, mv_directors
WHERE mov_directors=dir_id
LIMIT 0,2

```

In der Datenbank *movies* enthält die Tabelle `mv_actors_movies` die Beziehungen zwischen Schauspielern und Filmen: Da jeder Schauspieler in mehreren Filmen mitspielen kann und jeder Film mehrere Darsteller hat, handelt es sich um eine m:n-Beziehung, die weder in der Schauspielertabelle `mv_actors` noch in der Filmtabelle `mv_movies` stehen kann. Um nun alle Filme auszugeben, in denen Robert De Niro mitspielt, muss folgende komplexe Abfrage verwendet werden:

```

mysql> SELECT mov_titel Film, act_name Schauspieler
-> FROM mv_movies JOIN mv_actors_movies JOIN mv_actors
-> ON mov_id=mva_movie AND mva_actor=act_id
-> WHERE act_name="De Niro";
+-----+-----+
| Film          | Schauspieler  |
+-----+-----+
| Analyze This  | De Niro       |
| Analyze That  | De Niro       |
| Brazil        | De Niro       |
+-----+-----+

```

Aggregatfunktionen

Ein letztes wichtiges Element von Auswahlabfragen sind die *Aggregatfunktionen*. Sie wählen keine einzelnen Feldwerte aus, sondern fassen diese nach diversen Kriterien zusammen. Die einfachste Aggregatfunktion ist `COUNT()`; sie zählt einfach die Ergebnisdatensätze einer Abfrage. Als Feldangabe genügt hier in der Regel das `*`. Hier ein Beispiel, das die Anzahl der gespeicherten Filme ausgibt:

```
mysql> SELECT COUNT(*) AS "Anzahl Filme" FROM mv_movies;
+-----+
| Anzahl Filme |
+-----+
|           33 |
+-----+
```

Andere Aggregatfunktionen benötigen die Angabe einer bestimmten Spalte, um korrekt zu funktionieren. Hier die wichtigsten im Überblick:

- ▶ SUM(Spalte)
Die Summe aller Felder einer Spalte in der Auswahl
- ▶ MIN(Spalte)
Der kleinste Feldwert in der Spalte
- ▶ MAX(Spalte)
Der größte Feldwert in der Spalte
- ▶ AVG(Spalte)
Der Mittelwert aller Felder in der Spalte

Das folgende Beispiel wählt das Jahr des ältesten Films in der Tabelle `mv_movies` aus:

```
mysql> SELECT MIN(mov_jahr) FROM mv_movies;
+-----+
| MIN(mov_jahr) |
+-----+
|           1964 |
+-----+
```

Der folgende – in gewisser Weise durchaus plausibel erscheinende – Versuch, den passenden Film und das Jahr zusammen auszugeben, scheitert übrigens mit einer Fehlermeldung:

```
mysql> SELECT mov_titel, MIN(mov_jahr) AS Jahr
-> FROM mv_movies;
ERROR 1140 (42000): Mixing of GROUP columns
(MIN(),MAX(),COUNT(),...)
with no GROUP columns is illegal if there is no GROUP BY clause
```

Aggregatfunktionen können nicht ohne Weiteres mit der Auswahl normaler Felder gemischt werden. Hier muss eine Unterabfrage verwendet werden:

```
mysql> SELECT mov_titel, mov_jahr FROM mv_movies
-> WHERE mov_jahr=
-> (SELECT MIN(mov_jahr) FROM mv_movies) \G
```

```
***** 1. row *****
mov_titel: Dr. Seltsam oder wie ich lernte, die Bombe zu lieben
mov_jahr: 1964
```

Eine weitere Möglichkeit besteht darin, Aggregatfunktionen mithilfe der Klausel `GROUP BY` gruppenweise auf Feldinhalte anzuwenden. Das folgende Beispiel zeigt die Anzahl der gespeicherten Filme der einzelnen Jahre seit 2000:

```
mysql> SELECT mov_jahr, COUNT(*) FROM mv_movies
-> WHERE mov_jahr>=2000 GROUP BY mov_jahr
-> ORDER BY mov_jahr ASC;
```

Die Klausel `ORDER BY` sortiert übrigens nach der angegebenen Spalte, und zwar entweder aufsteigend (`ASC`) oder absteigend (`DESC`). Sie können auch – durch Komma getrennt – mehrere Feldnamen angeben. In diesem Fall wird nach dem zweiten Feld sortiert, wo das erste identisch ist; ein typisches Beispiel wäre das Sortieren nach Vornamen bei gleichen Nachnamen.

12.5.5 Änderungsabfragen

Eine weitere Klasse von SQL-Abfragen nimmt Änderungen an Ihren Datenbanken vor – entweder an den Dateninhalten oder an der Struktur.

Datenänderungsabfragen

Mithilfe der SQL-Anweisung `UPDATE` lässt sich der Inhalt bestehender Tabellen modifizieren. Die grundlegende Syntax sieht so aus:

```
UPDATE Tabelle SET Spalte=Wert[, Spalte=Wert, ...]
[WHERE Kriterium]
```

Um Änderungen gefahrlos ausprobieren zu können, sollten Sie zunächst eine Kopie einer bestehenden Tabelle anfertigen. Die folgende Anweisung erstellt eine identische Kopie von `mv_actors` mit dem neuen Namen `mv_actors2`:

```
CREATE TABLE mv_actors2 LIKE mv_actors;
INSERT INTO mv_actors2 SELECT * FROM mv_actors;
```

Die `LIKE`-Klausel kopiert die Struktur, während die `INSERT`-Abfrage für die Übernahme der Daten zuständig ist.

In `mv_actors2` wird nun ein neuer Schauspieler eingefügt, zunächst mit falschem Geburtsdatum:

```
INSERT INTO mv_actors2
(act_name, act_vorname, act_gdatum, act_geschlecht, act_land)
VALUES ("DiCaprio", "Leonardo", "1974-11-10", "m", 2)
```

Die folgende Abfrage korrigiert das Datum auf den 11. November:

```
UPDATE mv_actors2 SET act_gdatum="1974-11-11"
WHERE act_name="DiCaprio"
```

Achtung! Beachten Sie, dass die WHERE-Klausel hier sehr wichtig ist – wenn Sie sie weglassen, wird das entsprechende Feld in *allen* Datensätzen auf den angegebenen Wert gesetzt!

Strukturänderungsabfragen

Die Anweisung ALTER TABLE ändert die Struktur einer Tabelle. Sie bietet zahlreiche komplexe Möglichkeiten – hier nur die wichtigsten:

► ADD COLUMN

Diese Anweisung fügt eine weitere Spalte hinzu. Das folgende Beispiel fügt eine Spalte namens act_bemerk in mv_actors2 ein:

```
ALTER TABLE mv_actors2
ADD COLUMN act_bemerk VARCHAR(100)
```

Dies fügt die Spalte am Ende ein; wenn Sie die Position festlegen möchten, können Sie FIRST (ganz links) oder AFTER *Spaltenname* hinzufügen.

► CHANGE COLUMN

Damit können Sie den Namen oder die Definition einer Spalte ändern. Sie müssen stets den bisherigen und einen neuen Namen angeben; falls der Name gleich bleibt, muss er zweimal genannt werden. Hier ein Beispiel, das act_bemerk in act_kommentar umbenennt und maximal 200 statt 100 Zeichen zulässt:

```
ALTER TABLE mv_actors2
CHANGE COLUMN act_bemerk act_kommentar VARCHAR(200)
```

Wichtig: Wenn Sie den Speicherplatz einer Spalte einschränken, beispielsweise durch Verkürzen der String-Länge oder Wechsel auf einen numerischen Datentyp mit geringerer Bit-Breite, werden die Informationen, die nicht mehr in den verkleinerten Bereich passen, kommentarlos gelöscht.

► DROP COLUMN

Dieser Befehl löscht die angegebene Spalte mitsamt all ihren Inhalten. Die folgende Abfrage entfernt act_kommentar wieder:

```
ALTER TABLE mv_actors2
DROP COLUMN act_kommentar
```

12.5.6 Löschabfragen

Die SQL-Anweisung `DELETE` löscht Datensätze; auch hier gibt die `WHERE`-Klausel an, welche Sie entfernen möchten. Das folgende Beispiel löscht den weiter oben hinzugefügten Leonardo DiCaprio wieder:

```
DELETE FROM mv_actors2 WHERE act_name="DiCaprio"
```

Wenn Sie `WHERE` weglassen, wird der gesamte Inhalt der Tabelle gelöscht – also Vorsicht!

`DROP TABLE` löscht eine ganze Tabelle mitsamt ihren Inhalten. Hier ein Beispiel, das die Kopie `mv_actors2` wieder entfernt:

```
DROP TABLE mv_actors2
```

Mit `DROP DATABASE` können Sie sogar eine ganze Datenbank löschen. Hier eine Beispielabfrage, die eine Datenbank namens *muell* löscht:

```
DROP DATABASE muell
```

12.6 MySQL-Administration

Die Administration von MySQL-Servern ist ein sehr umfangreiches Thema, das in einem Buch wie diesem nur angeschnitten werden kann. Bei näherem Interesse konsultieren Sie die MySQL-Online-Dokumentation oder eines der in Anhang B empfohlenen Bücher.

Für die MySQL-Administration stehen im Wesentlichen drei Arten von Werkzeugen zur Verfügung:

1. Mit dem Server gelieferte Konsolen-Tools wie `mysql` (siehe oben), `mysqladmin` und `mysqldump`
2. *MySQL Administrator*, ein grafisches Administrationstool der MySQL AB. Es kann auf der Website *mysql.com* heruntergeladen werden, wird aber hier nicht beschrieben, da es weitgehend intuitiv und selbsterklärend ist.
3. Grafische Clients, die nicht auf die Administration spezialisiert sind (zum Beispiel `phpMyAdmin`, siehe Kapitel 15, »LAMP«)

12.6.1 mysqladmin

Einige Verwaltungsaufgaben lassen sich mithilfe des Kommandozeilentools `mysqladmin` erledigen. Genau wie den Konsolencient `mysql` müssen Sie es mit `-u`

Benutzername (in der Regel `root`) und `-p` für die Passwortanforderung aufrufen, sodass seine Syntax so aussieht:

```
mysqladmin Befehl -u Benutzername -p
```

Die wichtigsten Befehle, die Sie eingeben können, sind:

- ▶ `create Datenbank`
Dieser Befehl erzeugt die angegebene Datenbank wie `CREATE DATABASE`.
- ▶ `drop Datenbank`
Damit wird die gewünschte Datenbank gelöscht
- ▶ `extended-status`
Mit diesem Befehl erhalten Sie erweiterte Statusinformationen. Da die Liste sehr lang ist, sollten Sie sie durch `|less` filtern.
- ▶ `ping`
Überprüfen Sie, ob der MySQL-Server läuft.
- ▶ `reload`
Mit diesem Befehl werden die Benutzerinformationen neu geladen (siehe nächster Unterabschnitt).
- ▶ `shutdown`
Beendet den MySQL-Server
- ▶ `version`
Diese Anweisung gibt die Version des MySQL-Servers aus.

Hier ein Beispiel, das den Server beendet:

```
$ mysqladmin shutdown -u root -p
```

12.6.2 Benutzerverwaltung

Wie bereits erwähnt, erfolgt die Benutzeridentifikation in MySQL anhand der drei Komponenten Host, Benutzername und Passwort. Die entsprechenden Daten werden in der Verwaltungsdatenbank *mysql* gespeichert. Hier dienen insbesondere folgende Tabellen der Verwaltung von Benutzerrechten:

- ▶ `user`
Enthält Benutzernamen und verschlüsselte Passwörter zur Überprüfung der Anmeldung sowie globale Benutzerrechte.
- ▶ `host`
Diese Tabelle enthält hostbasierte Berechtigungen.
- ▶ `db`
Hier werden die Berechtigungen an einzelnen Datenbanken gespeichert.

- ▶ `tables_priv`
Diese Tabelle verwaltet die Rechte an einzelnen Tabellen.
- ▶ `columns_priv`
In dieser Tabelle finden Sie die Benutzerrechte an einzelnen Tabellenspalten.

Sie können sich die Inhalte und Strukturen dieser Tabellen mithilfe der weiter oben vorgestellten Abfragen in Ruhe anschauen. Wie Sie sehen, besitzt in `user`, `host` und `db` jedes Recht eine eigene `ENUM`-Spalte mit den möglichen Werten 'Y' (Recht gewährt) oder 'N' (Recht verweigert – dies ist aus naheliegenden Gründen der Standardwert). In `tables_priv` und `columns_priv` gibt es dagegen für alle Rechte je eine einzelne `SET`-Spalte; die darin für den einzelnen User aufgelisteten Rechte werden gewährt, alle anderen verweigert.

Die Überprüfung jedes Datenbankzugriffs erfolgt in zwei Stufen:

1. Prüfung, ob der fragliche Benutzer sich vom entsprechenden Host aus anmelden darf (Tabellen `user` und `host`). Scheitert dies aus einem der möglichen Gründe (unbekannter Benutzer, unberechtigter Host, falsches Passwort), dann bricht der Anmeldeversuch mit einer Fehlermeldung ab. Beispiel:

```
$ mysql -u wronguser -p
ERROR 1045 (28000): Access denied for 'wronguser'@'localhost'
(using password: YES)
```

2. Wenn die grundlegende Legitimationsprüfung erfolgreich war, wird als Nächstes getestet, ob der nunmehr angemeldete Benutzer die gewünschte Operation durchführen darf. Dies geht schrittweise von oben nach unten: Hat er das entsprechende Recht global (in der Tabelle `user`), dann ist bereits alles klar. Wenn nicht, wird in der Tabelle `db` überprüft, ob der Benutzer die entsprechende Berechtigung für die gesamte aktuelle Datenbank besitzt. Ist auch dies nicht der Fall, dann geht es mit der Tabelle und schließlich mit den einzelnen Spalten weiter. Erst wenn alle diese Prüfungen versagen sollten, erhalten Sie eine Fehlermeldung wie diese:

```
mysql> SHOW TABLES FROM test;
ERROR 1044 (42000): Access denied for user 'darfnix'@'localhost'
to database 'test'
```

Achtung!

Es ist *sehr* wichtig, dieses Verfahren richtig zu verstehen: Wenn ein Recht auf einer übergeordneten Ebene besteht, ist die Berechtigungsüberprüfung unwiderruflich abgeschlossen. Es gibt also *keine* Möglichkeit, einem Benutzer zuerst eine allgemeine Erlaubnis zu erteilen und diese dann im Einzelnen wieder einzuschränken. Es ist also umso unerlässlicher, hier »geizig« zu sein und Benutzern stets nur die unbedingt notwendigen Berechtigungen zu erteilen.

Um einen neuen Benutzer zu erzeugen, starten Sie zunächst den Kommandozeilenclient als `root`:

```
$ mysql -u root -p
```

Wie es nun weitergeht, hängt von der MySQL-Version ab. Ab MySQL 5.0 wird die Anweisung `CREATE USER` verwendet, um ein Benutzerkonto anzulegen. Die allgemeine Syntax lautet:

```
CREATE USER username@hostname IDENTIFIED BY 'Passwort'
```

Hier ein konkretes Beispiel:

```
mysql> CREATE USER movieuser@localhost
-> IDENTIFIED BY 'MyMovieZ';
```

Diese Anweisung erzeugt einen neuen Datensatz in der Verwaltungstabelle `mysql.user`, in der User- und Hostname, verschlüsseltes Passwort und keinerlei Rechte (alles auf 'N') eingetragen sind:

```
mysql> SELECT * from mysql.user WHERE user='movieuser'\G
***** 1. row *****
      Host: localhost
      User: movieuser
      Password: *3FDFD7AFF25ABF4050FF0F8047BA911125F63DE5
      Select_priv: N
      Insert_priv: N
      Update_priv: N
      [...]
```

Falls Sie es – etwa in gemietetem Webspacer – mit MySQL 4.1 oder einer älteren Version zu tun haben, müssen Sie beachten, dass `CREATE USER` dort nicht zur Verfügung steht. Stattdessen wird wie zum Erteilen von Rechten eine `GRANT`-Abfrage verwendet. Benutzen Sie dazu das spezielle »Recht« `USAGE`; es erstellt den Eintrag in der Tabelle `user`, erlaubt die grundsätzliche Anmeldung am MySQL-Server und setzt alle globalen Privilegien auf 'N'. Die allgemeine Schreibweise sieht so aus:

```
GRANT USAGE ON *.* TO username@hostname
IDENTIFIED BY 'Passwort'
```

Für den neuen Benutzer `movieuser@localhost` sieht die entsprechende Abfrage so aus:

```
mysql> GRANT USAGE ON *.* TO movieuser@localhost
-> IDENTIFIED BY 'MyMovieZ';
```

Sollten Sie mit relativ alten Clients oder Programmierschnittstellen arbeiten (beispielsweise PHP-Versionen vor 4.4), dann funktioniert die Anmeldung dieses neuen Benutzers nicht. Das Problem besteht darin, dass das Standardverfahren zur Passwortverschlüsselung in MySQL 4.1 geändert wurde; die alten MySQL-Client-Bibliotheken verstehen das neue Verfahren nicht. In einem solchen Fall müssen Sie `IDENTIFIED BY 'Passwort'` im jeweiligen Befehl weglassen – was zunächst einen passwortlosen Benutzer erzeugt – und das Passwort danach folgendermaßen festlegen:

```
mysql> SET PASSWORD FOR movieuser@localhost =
      -> OLD_PASSWORD('MyMovieZ');
```

`SET PASSWORD` kann auch verwendet werden, um ein Passwort nachträglich zu ändern; im Normalfall kommt dann allerdings die Verschlüsselungsfunktion `PASSWORD()` statt `OLD_PASSWORD()` zum Einsatz.

Um bestehenden Benutzern neue Rechte zuzuweisen, wird in jedem Fall die Anweisung `GRANT` verwendet, deren allgemeine Syntax so aussieht:

```
GRANT Recht [(Spalte)][, Recht [(Spalte)] ...]
ON Datenbank.Tabelle
TO username@hostname
```

Die wichtigsten Rechte, die Sie Benutzern erteilen können, werden in Tabelle 12.9 aufgelistet. Eine Liste *aller* möglichen Privilegien erhalten Sie übrigens mit folgender MySQL-Anweisung:

```
mysql> SHOW PRIVILEGES;
```

MySQL-Benutzerrecht	Bedeutung
USAGE	nur Anmeldung; keine Rechte
SELECT	Auswahlabfragen
INSERT	Einfügeabfragen
UPDATE	Datenänderungsabfragen
DELETE	Datenlöschabfragen
CREATE	Tabellenerstellungsabfragen
DROP	Tabellenlöschabfragen
ALTER	Strukturänderungsabfragen
INDEX	Indexverwaltung mit <code>CREATE / DROP INDEX</code>

Tabelle 12.9 Benutzerrechte für `GRANT`- /`REVOKE`-Anweisungen

MySQL-Benutzerrecht	Bedeutung
CREATE VIEW	Erstellen von Views
FILE	Import/Export mit Textdateien
SHOW DATABASES	Anzeigen der Datenbankliste
SHOW VIEW	Anzeigen einer View-Definition (SHOW CREATE VIEW)
CREATE ROUTINE	Erstellen von Stored Procedures/Functions
ALTER ROUTINE	Ändern von Stored Procedures/Functions
EXECUTE	Ausführen von Stored Procedures/Functions
CREATE USER	Benutzer erstellen
SHUTDOWN	MySQL-Server beenden
REPLICATION CLIENT	Replikationseinstellungen ermitteln
REPLICATION SLAVE	als Replikations-Slave fungieren
SUPER	Prozessverwaltung (CHANGE MASTER, KILL usw.)
ALL [PRIVILEGES]	alle Rechte außer GRANT
GRANT OPTION	Rechteverwaltung mit GRANT / REVOKE

Tabelle 12.9 Benutzerrechte für GRANT- /REVOKE-Anweisungen (Forts.)

Das folgende Beispiel erteilt dem weiter oben erstellten Benutzer `movieuser@localhost` das Recht, Daten aus beliebigen Datenbanken und Tabellen auszuwählen:

```
mysql> GRANT SELECT ON *.* TO movieuser@localhost;
```

Zusätzlich soll er die Berechtigung erhalten, Datensätze in beliebige Tabellen der Datenbank `movies` einzufügen:

```
mysql> GRANT INSERT ON movies.* TO movieuser@localhost;
```

In der Tabelle `mv_actors` soll er auch Datensätze löschen dürfen:

```
mysql> GRANT DELETE ON movies.mv_actors
-> TO movieuser@localhost;
```

Schließlich soll er auch noch das Recht haben, den Inhalt der Spalte `mov_titel` in der Tabelle `mv_movies` (den Filmtitel) zu ändern:

```
mysql> GRANT UPDATE (mov_titel) ON movies.mv_movies
-> TO movieuser@localhost;
```

Angenommen, dieser Benutzer meldet sich an und führt folgende Abfrage durch:

```
mysql> UPDATE mv_movies SET mov_titel="Reine Nervensache"
      -> WHERE mov_titel="Analyze This";
```

Dann überprüft der MySQL-Server nacheinander folgende Werte:

1. Besitzt der User `movieuser@localhost` das globale Recht zum Ändern (`Update_priv`) in der Tabelle `user`? – Nein.
2. Besitzt er das allgemeine Änderungsrecht (`Update_priv`) für die Datenbank `movies` in der Tabelle `db`? – Nein.
3. Besitzt er das Recht, beliebige Spalten in der Tabelle `mv_movies` zu ändern (Wert `Update` in der Spalte `Table_priv` der Tabelle `Tables_priv`)? – Nein.
4. Besitzt er die Berechtigung, die Spalte `mov_titel` in der Tabelle `mv_movies` zu ändern (Wert `Update` in der Spalte `Column_priv` – sowohl in der Tabelle `Tables_priv` als auch in `Columns_priv`; in Letzterer mit dem Spaltennamen)? – Ja.

Um einem User dieselben Privilegien wie `root` zu erteilen (wovon Sie in der Praxis im Allgemeinen absehen sollten), genügt die folgende Anweisung übrigens nicht:

```
mysql> GRANT ALL PRIVILEGES ON *.*
      -> TO verwalter@localhost;
```

Dieser Benutzer besitzt nämlich nicht das Recht, seinerseits mittels `GRANT` Benutzerrechte zu erteilen. Soll dies gewünscht sein, ist folgende Variante erforderlich:

```
mysql> GRANT ALL PRIVILEGES ON *.*
      -> TO verwalter@localhost WITH GRANT OPTION;
```

Um einem Benutzer ein zuvor erteiltes Recht wieder zu entziehen, wird eine `REVOKE`-Anweisung verwendet. Ihre allgemeine Syntax lautet:

```
REVOKE Recht [(Spalte)][, Recht [(Spalte)] ...]
ON datenbank.tabelle FROM username@hostname
```

Das folgende Beispiel entzieht dem Benutzer `movieuser@localhost` das weiter oben erteilte Recht, in alle Tabellen der Datenbank `movies` Daten einzufügen:

```
mysql> REVOKE INSERT ON movies.*
      -> FROM movieuser@localhost;
```

Möchten Sie dagegen einen Benutzer mit allen seinen Berechtigungen entfernen, können Sie in MySQL 5 folgende Anweisung verwenden:

```
mysql> DROP USER ex_user@localhost;
```

In 4er-Versionen von MySQL ist es dagegen am gründlichsten und empfehlenswertesten, Zeilen mit dem fraglichen Usernamen per SQL-Abfrage aus den Berechtigungstabellen zu löschen. Beispiel:

```
mysql> DELETE FROM user
      -> WHERE user="ex_user" AND host="localhost";
```

Falls Sie den Client nach dem Ändern von Benutzerrechten nicht beenden (oder zur Sicherheit sogar auch dann), sollten Sie die folgende Anweisung ausführen:

```
mysql> FLUSH PRIVILEGES;
```

Dies lädt die Benutzerinformationen neu; alternativ funktioniert auch folgende Konsolenanweisung:

```
$ mysqladmin reload -u root -p
```

12.6.3 Import und Export von Daten, Backups

Für einfache, manuelle Backups von MySQL-Datenbanken und -Tabellen kann das Konsolenprogramm `mysqldump` verwendet werden. Es erzeugt SQL-Dumps der entsprechenden Strukturen und Daten, das heißt Dateien mit SQL-Anweisungen, die die entsprechenden Tabellen erstellen und die Daten einfügen.

Die Syntax dieser Anweisung ist etwas unterschiedlich, je nachdem, ob Sie einzelne Tabellen, einzelne Datenbanken oder alle Datenbanken Ihres Servers exportieren möchten. Für einzelne Tabellen lautet die Syntax:

```
mysqldump [Optionen] Datenbank [Tabelle ...]
```

Wenn Sie keine Tabellenbezeichnungen angeben, wird die gesamte Datenbank exportiert.

Falls Sie mehrere Datenbanken exportieren möchten, müssen Sie folgende Schreibweise verwenden:

```
mysqldump [Optionen] --databases Datenbank
          [Datenbank ...]
```

Für einen Export des gesamten Datenbestandes des Servers gilt schließlich diese Syntax:

```
mysqldump [Optionen] --all-databases
```

Wie alle MySQL-Konsolenhilfsprogramme benötigt auch `mysqldump` Benutzername und Passwort, sodass Sie unter den Optionen auf jeden Fall `-u Benutzername` (hier meist `root`) und `-p` (Passworteingabeaufforderung) angeben müssen.

In 4er-Versionen von MySQL ist es dagegen am gründlichsten und empfehlenswertesten, Zeilen mit dem fraglichen Usernamen per SQL-Abfrage aus den Berechtigungstabellen zu löschen. Beispiel:

```
mysql> DELETE FROM user
      -> WHERE user="ex_user" AND host="localhost";
```

Falls Sie den Client nach dem Ändern von Benutzerrechten nicht beenden (oder zur Sicherheit sogar auch dann), sollten Sie die folgende Anweisung ausführen:

```
mysql> FLUSH PRIVILEGES;
```

Dies lädt die Benutzerinformationen neu; alternativ funktioniert auch folgende Konsolenanweisung:

```
$ mysqladmin reload -u root -p
```

12.6.3 Import und Export von Daten, Backups

Für einfache, manuelle Backups von MySQL-Datenbanken und -Tabellen kann das Konsolenprogramm `mysqldump` verwendet werden. Es erzeugt SQL-Dumps der entsprechenden Strukturen und Daten, das heißt Dateien mit SQL-Anweisungen, die die entsprechenden Tabellen erstellen und die Daten einfügen.

Die Syntax dieser Anweisung ist etwas unterschiedlich, je nachdem, ob Sie einzelne Tabellen, einzelne Datenbanken oder alle Datenbanken Ihres Servers exportieren möchten. Für einzelne Tabellen lautet die Syntax:

```
mysqldump [Optionen] Datenbank [Tabelle ...]
```

Wenn Sie keine Tabellenbezeichnungen angeben, wird die gesamte Datenbank exportiert.

Falls Sie mehrere Datenbanken exportieren möchten, müssen Sie folgende Schreibweise verwenden:

```
mysqldump [Optionen] --databases Datenbank
          [Datenbank ...]
```

Für einen Export des gesamten Datenbestandes des Servers gilt schließlich diese Syntax:

```
mysqldump [Optionen] --all-databases
```

Wie alle MySQL-Konsolenhilfsprogramme benötigt auch `mysqldump` Benutzername und Passwort, sodass Sie unter den Optionen auf jeden Fall `-u Benutzername` (hier meist `root`) und `-p` (Passwortheingabeaufforderung) angeben müssen.

```
mysql -e "flush tables with read lock" -u root -p
mysqldump -u root -p DATENBANK >DATEI
mysql -e "unlock tables" -u root -p
```

Die `mysql`-Option `-e` startet den `mysql`-Client nicht interaktiv, sondern führt die angegebene Anweisung aus. Die MySQL-Anweisung `FLUSH TABLES WITH READ LOCK` schreibt alle zurzeit »schwebenden« Änderungen ordnungsgemäß auf die Festplatte und sperrt dann alle Tabellen aller Datenbanken vollständig. `UNLOCK TABLES` setzt die Sperren wieder zurück.

Leider müssen Sie bei dieser Anweisungsfolge dreimal das `root`-Passwort des MySQL-Servers eingeben. Daher lässt es sich in dieser Form auch nicht per Cron-job automatisieren (siehe Kapitel 17, »System-Automatisierung«). Zwar können Sie das Passwort notfalls im Klartext ohne Abstand hinter die Option `-p` schreiben – aber das sollten Sie auf keinen Fall mit dem `root`-Passwort machen! Dafür empfiehlt es sich eher, einen speziellen Backup-Benutzer mit eingeschränkten Rechten zu erstellen. Er benötigt die globalen Rechte `RELOAD` (für `FLUSH TABLES`), `LOCK TABLES` (für das Sperren und Entsperren) sowie `SELECT` für das Auslesen der Daten zum eigentlichen Backup:

```
mysql> CREATE USER backupuser@localhost
-> IDENTIFIED BY '84ckUp2';
mysql> GRANT RELOAD, LOCK TABLES, SELECT ON *.*
-> TO backupuser@localhost;
```

Anschließend können Sie die drei Anweisungen für MyISAM-Backups wie folgt als Shell-Skript speichern:

```
mysql -e "flush tables with read lock" \
-u backupuser -p84ckUp2
mysqldump -u backupuser -p84ckUp2 Datenbank >Datei
mysql -e "unlock tables" -u backupuser -p84ckUp2
```

Alternativ lassen sich MyISAM-Tabellen auch geschützt sichern, indem Sie `mysqldump` mit der zusätzlichen Option `--lock-all-tables` (Kurzfassung `-x`) aufrufen. Da dies jedoch alle Tabellen auf einmal sperrt, darf es nur zu Zeiten mit geringem Datenbankzugriff passieren, oder indem Sie die Daten von einem nicht anderweitig eingesetzten Replikations-Slave (siehe unten) abgreifen. Die benötigten Benutzerrechte bleiben dabei dieselben, da auch hier das Locking durchgeführt wird.

Um eines der weiter oben angelegten Backups wieder zurückzuspielen, gibt es zwei Möglichkeiten. Die erste wird auf der Konsole angewendet:

```
$ mysql -u root -p <Datei
```

Innerhalb des `mysql`-Clients können Sie dagegen die bereits besprochene Syntax `mysql> \. Datei`

verwenden.

Eine etwas andere Möglichkeit bietet das SQL-Anweisungspaar `SELECT ... INTO OUTFILE` und `LOAD DATA INFILE`: Hier werden die Daten in sogenannte CSV-Textdateien (Comma Separated Values) exportiert beziehungsweise aus diesen importiert. CSV ist ein beliebtes Datenaustauschformat für Tabellenkalkulationsprogramme wie OpenOffice.org Calc (siehe Kapitel 7, »OpenOffice.org«).

Um Daten in eine CSV-Datei zu exportieren, hängen Sie an eine beliebige `SELECT`-Abfrage `INTO OUTFILE Dateiname` an. Beispiel:

```
mysql> SELECT * FROM mv_movies INTO OUTFILE "movies.csv";
```

Der Import aus einer entsprechenden Datei erfolgt dagegen so:

```
mysql> LOAD DATA INFILE "movies.csv"
-> INTO TABLE mv_movies;
```

Über diverse Parameter, die hier allerdings zu weit führen würden, können Sie jeweils den genauen Aufbau der CSV-Dateien bestimmen.

12.6.4 Konfigurationsdateien

Wie bereits erwähnt, kommt MySQL im Standardbetrieb recht gut ohne Konfigurationsdateien aus. Sollten Sie dennoch irgendwelche Aspekte seines Verhaltens ändern wollen, dann können Sie eine der folgenden Dateien anlegen:

- ▶ `/etc/my.cnf`
Dies ist die globale Konfigurationsdatei für Server und Clients.
- ▶ `~/.my.cnf`
Dies ist die benutzerspezifische Konfigurationsdatei nur für Clients.

Der Aufbau dieser Datei entspricht den bekannten Windows-INI-Dateien (ähnlich wie `smb.conf` oder `php.ini`, siehe Kapitel 14, »Datei- und Verzeichnisdienstserver«, beziehungsweise Kapitel 15, »LAMP«): Abschnitte in eckigen Klammern kennzeichnen die Themen (in diesem Fall die einzelnen MySQL-Programme); die einzelnen Konfigurationsanweisungen werden im Format `Parameter = Wert` angegeben.

Der wichtigste Abschnitt ist `[mysqld]` mit Parametern für den Server selbst. Aber auch alle anderen mit MySQL gelieferten Programme lesen die für sie geschrie-

benen Abschnitte in diesen Dateien aus, zum Beispiel `[mysqladmin]`, `[mysqldump]` oder der Client `[mysql]`.

Hier einige wenige Parameter für den Server im Überblick:

- ▶ `port`
Dieser Parameter bezeichnet den TCP-Port, an dem der Server lauscht (Standard 3306; kann geändert werden, um mehrere MySQL-Server auf einem Host zu betreiben).
- ▶ `socket`
Das UNIX-Domain-Socket (Dateipfad) für die lokale Kommunikation (Standard: `/tmp/mysql.sock`)
- ▶ `character-set-server`
Dieser Parameter verweist auf den Zeichensatz des Servers.
- ▶ `collation-server`
Die Sortierfolge des Servers
- ▶ `Language`
Die Sprache für Fehlermeldungen und Warnungen (Sie können beispielsweise auf `german` umschalten).
- ▶ `sql-mode`
Der SQL-Kompatibilitätsmodus; zum Beispiel `mysql` (Standard) oder `ansi` (ändert etwa die weiter oben diskutierte Bedeutung der verschiedenen Anführungszeichen).

In den Konfigurationsdateien können Sie jeden Wert ändern, dessen aktuelle Einstellung die folgende MySQL-Anweisung zeigt:

```
mysql> SHOW VARIABLES;
```

MySQL wird mit einigen Vorlagen für mögliche Konfigurationsdateien geliefert, von *my-small.cnf* (optimiert für sehr kleine Datenbanken) bis hin zu *my-huge.cnf* für gigantische Datenbanken. Sie können bei Bedarf eine dieser Dateien nach */etc/my.cnf* kopieren und noch etwas an Ihre Bedürfnisse anpassen.

12.6.5 Logdateien

MySQL ist in der Lage, verschiedene Logdateien anzulegen, die eine wichtige Basis zur Wiederherstellung verlorener Daten, für die Replikation (siehe nächster Unterabschnitt) oder zur Fehlersuche bilden.

Standardmäßig wird nur eine Error-Logdatei angelegt. Sie befindet sich – wie alle Logdateien – im MySQL-Datenverzeichnis (weil das `mysql`-Systembenutzerkonto

nur dort Schreibrechte besitzt) und trägt den aktuellen Hostnamen und die Endung *.log* (Beispiel: *tux.log*). Mit folgendem Eintrag unter dem Abschnitt `[mysqld]` in der *my.cnf* können Sie ihren Ort ändern:

```
log-error = Pfad
```

Eine weitere sehr wichtige Logdatei ist das binäre Update-Log. Es protokolliert alle Änderungsabfragen und dient somit als Basis für die Replikation oder auch für die Wiederherstellung seit dem letzten richtigen Backup.

Der entsprechende Eintrag in */etc/my.cnf* lautet:

```
[mysqld]
...
log-bin = mylog
```

Dies legt im MySQL-Datenverzeichnis die Dateien *mylog.000001* und *mylog.index* an (statt *mylog* können Sie auch einen beliebigen anderen Namen eingeben). Bei jedem Neustart des Servers, bei explizitem Flush oder wenn die Datei eine bestimmte Größe überschreitet, wird die nächste (*mylog.000002* und so weiter) angelegt.

Um geplant die nächste Logdatei zu beginnen, können Sie im `mysql`-Client Folgendes eingeben:

```
mysql> FLUSH LOGS;
```

Eine Konsolenalternative ist:

```
$ mysqladmin flush-logs -u root -p
```

Wenn Sie eine automatische Log-Rotation per Cronjob (siehe Kapitel 17, »System-Automatisierung«) durchführen möchten, legen Sie zunächst einen speziellen MySQL-User an, der nur das globale Recht `RELOAD` hat. Danach können Sie ein Skript mit folgender Zeile erstellen:

```
mysqladmin -u reloaduser flush-logs -p Reloadpasswort
```

Da die Update-Logdatei binär ist, können Sie sie nicht in einem Texteditor lesen. Wenn Sie sie im Klartext einsehen möchten, müssen Sie das mit MySQL gelieferte Tool `mysqlbinlog` verwenden. Beispiel:

```
$ mysqlbinlog -u root -p mylog.000001
```

Es gibt noch einige andere mögliche Logs, die Sie durch folgende Einträge in */etc/my.cnf* erzeugen können (die Bedeutung wird jeweils durch `#`Kommentare beschrieben):

```
[mysqld]
...
# Zu langsame Abfragen unter HOSTNAME-slow.log
# protokollieren:
log-slow-queries
# Dazu muss definiert werden, wie lange ZU LANGE ist
# (variiert stark je nach Datenbankgröße):
long_query_time = SEKUNDEN
# Ganz allgemein Abfragen protokollieren, die
# keinen Index verwenden (und daher oft optimierbar sind)
log-queries-not-using-indexes
```

12.6.6 Replikation

Die MySQL-Replikation ermöglicht die automatische Übernahme aller Datenbankänderungen von einem MySQL-Server, dem Master, auf beliebig viele andere MySQL-Server, die Replikations-Slaves. Dies dient zum einen der Ausfallsicherheit – Sie haben stets ein oder gar mehrere sekundenaktuelle Backups zur Hand –, zum anderen auch dem Load-Balancing (Lastverteilung) frequentierter Datenbankanwendungen: Schreibvorgänge müssen zwar weiterhin auf den Master erfolgen, aber Auswahlabfragen können auf die Slaves verteilt werden. In Kapitel 15, »LAMP«, erfahren Sie beispielsweise, wie Sie mit PHP eine Zufallsauswahl aus mehreren MySQL-Servern treffen können.

Zur Einrichtung der Replikation sind einige Schritte erforderlich, aber danach läuft diese vollautomatisch. Selbst wenn ein Slave vorübergehend ausfällt, bringt er sich nach Neustart des MySQL-Servers selbstständig wieder auf den neuesten Stand.

Auf dem Rechner, der Master werden soll (dieser kann seinerseits auch durchaus Slave eines anderen Masters sein), müssen Sie folgende Schritte vornehmen:

1. Erstellen Sie einen User für den Replikations-Slave:

```
mysql> CREATE USER repl_user@Slave-Host
-> IDENTIFIED BY "Passwort";
```

Ob Sie den Slave-Host als einfachen Hostnamen (etwa `heartofgold`) angeben können oder ob Sie seinen FQDN (`heartofgold.test.local`) benötigen, hängt von Ihrer Netzwerkkonfiguration ab; im Zweifelsfall funktioniert Letzteres immer.

2. Erteilen Sie dem neuen User das Recht `REPLICATION SLAVE`:

```
mysql> GRANT REPLICATION SLAVE ON *.*
-> TO repl_user@Slave-Host;
```

3. Richten Sie eine Update-Logdatei ein (siehe oben), falls Sie noch keine haben. Führen Sie in beiden Fällen folgende Schritte durch, um sich die aktuelle Datei und ihre Position zu notieren:

```
mysql> FLUSH TABLES WITH READ LOCK;
mysql> SHOW MASTER STATUS;
```

Merken Sie sich die Angaben `File` (zum Beispiel `mylog.000003`) und `Position` (etwa 345).

4. Stoppen Sie den MySQL-Server, zum Beispiel so:

```
# mysqladmin shutdown -u root -p
```

5. Erstellen Sie einen Snapshot des gesamten Datenverzeichnisses:

```
# cd <MySQL-Datenverzeichnis>
# tar czvf snapshot.tgz */* ibdata*
```

Die `ibdata*`-Dateien existieren übrigens nur, wenn Sie InnoDB-Tabellen verwenden.

6. Führen Sie die folgende Änderung in `/etc/my.cnf` durch:

```
[mysqld]
...
server-id = 1
```

Statt der 1 ist auch eine beliebige andere Nummer möglich, solange sie sich von den IDs aller Slaves unterscheidet; beim ersten Master ist 1 aber Standard.

7. Zum Schluss müssen Sie den MySQL-Server wieder starten.

Auf dem Slave sind dagegen folgende Vorbereitungen erforderlich:

1. Stoppen Sie den MySQL-Server.
2. Sichern Sie Ihre bisherigen Daten:

```
# mv <MySQL-Datenverzeichnis> <NeuerVerzeichnisName>
```

3. Spielen Sie den Snapshot des Masters ein (den Sie zuvor per `scp`, `NIS`, `ftp`, `E-Mail` oder wie auch immer auf den Slave kopiert haben):

```
# cd <Verzeichnis-über-MySQL-Datenverzeichnis>
# tar xzvf snapshot.tgz
```

4. Passen Sie die Rechte des neuen Datenverzeichnisses an:

```
# chown -R mysql:mysql <MySQLDaten>
```

Führen Sie die folgende Änderung in `/etc/my.cnf` durch:

```
[mysqld]
...
server-id = 2
```

Auch hier kann die ID im Grunde beliebig sein; sie muss sich nur vom Master und allen anderen Slaves unterscheiden.

5. Nun können Sie den MySQL-Server wieder starten.
6. Geben Sie im MySQL-Kommandozeilenclient schließlich folgende Anweisungen ein:

```
mysql> CHANGE MASTER TO
-> MASTER_HOST = 'Master-Host',
-> MASTER_USER = 'repl_user',
-> MASTER_PASSWORD = 'Passwort',
-> MASTER_LOG_FILE = 'ermittelte Logdatei'
-> MASTER_LOG_POS = ermittelte Position;
mysql> START SLAVE;
```

Natürlich müssen Sie die Platzhalter wie 'Master-Host' oder 'ermittelte Logdatei' durch konkrete Werte ersetzen.

Ob die Replikation nun funktioniert, können Sie leicht überprüfen, indem Sie auf dem Master Daten ändern und dann versuchen, diese auf dem Slave zu lesen. Wenn Sie alles richtig gemacht haben, müsste es funktionieren. Falls nicht, überprüfen Sie noch einmal alle Schritte; gegebenenfalls müssen Sie auch nachschauen, ob die Firewall auf einem der beteiligten Hosts den MySQL-Port (standardmäßig 3306) blockiert.

12.7 Zusammenfassung

Datenbankserver bilden das Herzstück jeder modernen Enterprise- oder Webanwendung, sodass sie zur Grundausstattung einer ernsthaften Linux-Distribution gehören. MySQL ist der verbreitetste und beliebteste Open-Source-Datenbankserver und wird daher seit Jahren in der jeweils neuesten Version mit SUSE Linux beziehungsweise openSUSE geliefert.

Nach der Installation über YaST – oder auch manuell, falls Sie die neuen Funktionen einer aktuelleren MySQL-Version benötigen – sollten Sie den Server stets absichern: Weisen Sie dem MySQL-Benutzer ein gutes Passwort zu, und löschen Sie dann alle weiteren Zugangsmöglichkeiten ohne Benutzernamen und/oder Passwort.

Für die grundlegende Verwaltung von MySQL-Datenbanken ist der Kommandozeilencient `mysql` geeignet. Hier können Sie wie in einer Shell beliebige SQL-Abfragen eingeben, die sofort ausgeführt und beantwortet werden. Darüber hinaus besitzt der Client einige interne Befehle, etwa zum Anlegen einer Protokoll-datei, zum Wechsel der Standarddatenbank oder zum Ausführen einzelner Shell-Befehle.

Wie beinahe alle relationalen Datenbanksysteme versteht auch MySQL die Abfragesprache SQL (worauf in diesem Fall schon der Name hindeutet). Diese Sprache ist allerdings weniger einheitlich, als wünschenswert wäre, und so besitzt MySQL – wie jede Datenbank – ihren spezifischen SQL-Dialekt mit Abweichungen in Syntax und Datentypen.

Schließlich gibt es noch eine Reihe mitgelieferter Tools zur Administration. Dazu gehört zunächst eine sehr fein einstellbare Benutzer- und Rechteverwaltung. Darüber hinaus können Sie den Serverprozess selbst steuern, Datenbanken exportieren oder sichern, Konfigurationsdateien anlegen, um das Verhalten von MySQL und seinen Teilprogrammen anzupassen und Logdateien erstellen zu lassen. Besonders interessant in puncto Ausfallsicherheit und Lastverteilung ist schließlich die MySQL-Replikation.

*Alle Dinge sind miteinander verbunden.
– Häuptling Seattle*

13 Netzwerkgrundlagen

Da das Internet auf UNIX-Systemen (besonders BSD) konzipiert wurde, sind Linux-Systeme aufgrund ihres entsprechenden Erbes prädestiniert für den Netzwerkeinsatz. In diesem Kapitel und den drei nächsten wird daher ausführlich erläutert, wie Sie den Netzwerkzugriff für Ihr openSUSE-System einrichten und welche Dienste und Möglichkeiten Ihnen dadurch zur Verfügung stehen. Das vorliegende Kapitel behandelt dabei zunächst allgemeine Konzepte und dann die Einrichtung der Netzwerkkomponenten und -protokolle, während sich die nächsten drei Kapitel auf diverse Netzwerkservers konzentrieren.

13.1 TCP/IP-Grundlagen

Der Vorläufer des Internets, das sogenannte ARPANet, wurde 1969 in Betrieb genommen. Aber erst knapp zehn Jahre später wurden die heutigen Netzwerkprotokolle eingeführt. Ein Netzwerkprotokoll ist ein Standard, der einen bestimmten Aspekt der Datenübertragung über ein Netzwerk regelt. »TCP/IP« ist der Name für eine ziemlich große Familie solcher Protokolle, benannt nach zwei ihrer wichtigsten Mitglieder: dem Transmission Control Protocol (TCP) und dem Internet Protocol (IP).

Bis vor einigen Jahren wurden neben TCP/IP noch zahlreiche andere Netzwerkprotokollfamilien eingesetzt – beispielsweise boten verschiedene Betriebssystemhersteller wie Apple, Microsoft oder Novell jeweils eigene, proprietäre Lösungen an. Im Lauf der 1990er-Jahre, während das Internet immer populärer wurde, verloren diese Protokolle mehr und mehr an Bedeutung. Novell beispielsweise kaufte SUSE und arbeitet heute vorwiegend im Linux-Enterprise-Umfeld, wo selbstverständlich TCP/IP eingesetzt wird und nicht etwa Novells eigene, inzwischen veraltete IPX/SPX-Protokolle.

13.1.1 Das Internet-Schichtenmodell

Wie bei allen Computernetzen lassen sich auch beim TCP/IP-Netzwerk verschiedene Funktionsebenen oder Schichten unterscheiden. Diese Schichten betreffen jeweils eine andere Facette des Netzwerks.

Nicht nur Netzwerke lassen sich schichtweise betrachten. Auch dieses Buch besitzt beispielsweise verschiedene »Schichten« im Sinne von Abstraktionsebenen:

1. Das Buch besteht aus bedrucktem Papier als »Trägermedium«.
2. Es enthält Linien, Kurven und Punkte, die von jemandem, der die lateinische Schrift lesen kann, als Buchstaben interpretiert werden.
3. Die Buchstaben ergeben Wörter und diese wiederum Sätze, die jeder versteht, der der deutschen Sprache mächtig ist.
4. Die Sätze verbinden sich schließlich für diejenigen zu sinnvollen Informationen, die grundlegende IT-Kenntnisse haben und alles über openSUSE-Systeme wissen möchten.

Dass diese Ebenen voneinander getrennt betrachtet werden können, zeigt sich daran, dass jede für sich austauschbar ist, ohne die anderen zu beeinflussen:

1. Der Text muss nicht auf Papier gedruckt sein: Ich schreibe ihn beispielsweise an einem Rechner und sehe ihn auf dem Monitor; daneben könnte er auch mit Zuckerguss auf eine (hinreichend große) Torte gespritzt oder in Stein gemeißelt werden.
2. Zur lateinischen Schrift gibt es bekanntermaßen zahlreiche Alternativen. Natürlich ist nicht jede Schrift gleich gut zur Wiedergabe jeder Sprache geeignet, aber prinzipiell kann man die Schrift wechseln und die Sprache beibehalten. Zum Beispiel wurde Türkisch früher mit arabischen Buchstaben geschrieben, heute dagegen mit leicht angepassten lateinischen (was übrigens besser funktioniert).
3. Bücher können in andere Sprachen übersetzt werden – und wenn dieses Buch (lizenziert) übersetzt würde, hätte ich überhaupt nichts dagegen.
4. Nicht alle Leser benötigen ein openSUSE-Handbuch. Vielleicht möchten Sie zuvor ein Einsteigerbuch lesen, oder Sie interessieren sich für ganz andere Themen und lesen beispielsweise lieber einen guten, spannenden Thriller.

Mit Netzwerken verhält es sich so ähnlich. Das bekannte OSI-Referenzmodell – auf das hier nicht näher eingegangen wird – besitzt sieben solcher Schichten; zur Beschreibung des Internets und sonstiger TCP/IP-Netze reichen dagegen vier aus. Nach dem DDN Standard Protocol Handbook sind diese vier Schichten folgende:

1. Die *Netzzugangsschicht* (Network Access Layer) beschreibt, auf welche Weise die angeschlossenen Rechner auf das Übertragungsmedium zugreifen, sich über die Sendereihenfolge einigen und mit Datenkollisionen umgehen.
2. Die *Internetschicht*¹ (Internet Layer) beschreibt die eindeutige Adressierung der Rechner und die Weiterleitung von Daten über mehrere miteinander verbundene Netze hinweg, das sogenannte *Routing*.
3. Auf der *Host-zu-Host-Transportschicht* (Host-to-Host Transport Layer) werden die Daten in kleine Einheiten unterteilt, die Datenpakete, und von einer Anwendung auf dem einen Rechner (Host) zu einer Anwendung auf dem anderen geschickt.
4. Die *Anwendungsschicht* (Application Layer) schließlich definiert, wie sich verschiedene Anwendungsprogramme über das Netzwerk »unterhalten«.

Abbildung 13.1 zeigt das Schichtenmodell der TCP/IP-Netzwerke mit zahlreichen Protokollen und Diensten, von denen viele in diesem und den nachfolgenden Kapiteln besprochen werden.

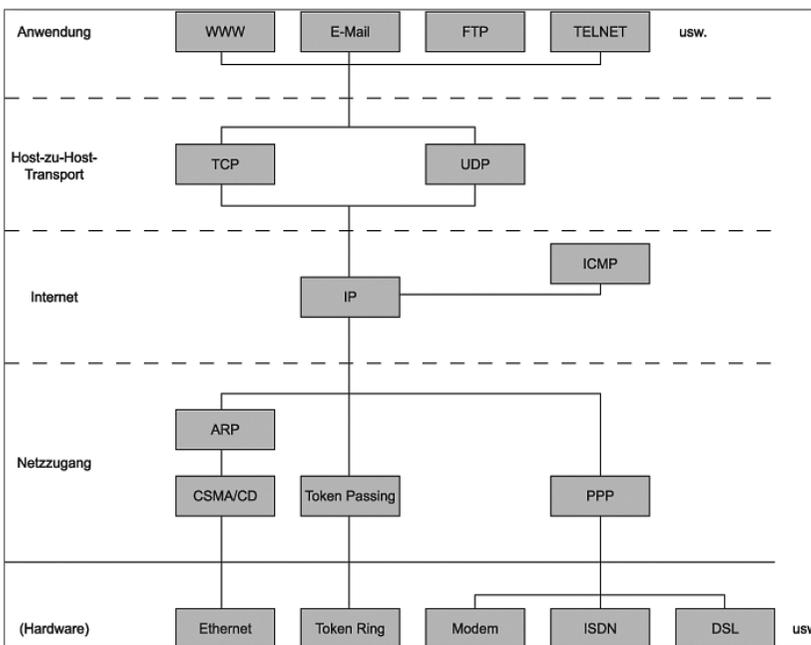


Abbildung 13.1 Der TCP/IP-Protokollstapel mit wichtigen Protokollen und Diensten

1 Beachten Sie, dass der Begriff »Internetschicht« älter ist als »das Internet«. Er besagt, dass Pakete *zwischen* verschiedenen physischen Netzwerken transportiert werden können.

Der Ablauf der Netzwerkkommunikation ist immer gleich: Eine Anwendung auf einem Rechner – die sich natürlich auf der Anwendungsschicht befindet – übergibt Daten an den passenden Dienst der Host-zu-Host-Transportschicht. Dieser teilt die Daten in kleinere Einheiten ein, die sogenannten Datenpakete, und versieht jedes Paket mit einer speziellen Transportinformation, dem Transport-Header. Diese vorbereiteten Pakete werden wiederum an die Internetschicht hinuntergereicht, die einen weiteren Header hinzufügt. Dieser enthält unter anderem die Absender- und die Empfängeradresse. Die Pakete, die mit all diesen zusätzlichen Informationen versehen wurden, heißen Datagramme. Erst in dieser Form werden sie tatsächlich der Netzzugangsschicht und damit letztlich der Netzwerkhardware anvertraut.

Der Empfängerrechner verfährt nun umgekehrt: Nachdem eine Funktion der Internetschicht festgestellt hat, dass die Daten tatsächlich für diesen Host bestimmt sind, reicht sie die Datagramme an den jeweils angesprochenen Transportdienst weiter. Dieser erkennt aus den Header-Informationen, für welche Anwendung die Daten bestimmt sind, und übergibt sie an diese.

Netzzugangsarten für TCP/IP-Netzwerke gibt es wie Sand am Meer; da sich die Internetprotokolle als der einzige ernstzunehmende Standard für größere heterogene Netze durchgesetzt haben, kann TCP/IP mit beinahe jeder Art von Netzwerk- und DFÜ-Hardware unter so gut wie allen Betriebssystemen verwendet werden. Die wichtigsten Zugangsformen und ihre Konfiguration werden im nächsten Abschnitt behandelt.

Ähnlich sieht es übrigens mit der obersten Schicht aus, der Anwendungsschicht. Hier tummeln sich unzählige verschiedene Anwendungsprotokolle; in diesem Buch kommen unter anderem verschiedene Dateiserver-Dienste sowie HTTP, FTP und diverse E-Mail-Protokolle zur Sprache.

Die folgenden Unterabschnitte konzentrieren sich zunächst auf die zwei mittleren Schichten, auf denen die beiden Protokolle arbeiten, die der ganzen Protokollfamilie den Namen gegeben haben: die Internetschicht mit dem Internet Protocol (IP) und die Host-zu-Host-Transportschicht mit dem Transmission Control Protocol (TCP) und seinem Konkurrenten, dem User Datagram Protocol (UDP).

13.1.2 Das Internet Protocol (IP)

Das Internet Protocol, die »Monopollösung« für die Internetschicht, löst vor allem zwei wichtige Aufgaben:

1. eindeutige Adressierung der Rechner und Netze
2. Weiterleitung der Datenpakete vom Absenderrechner an den Empfänger, auch über mehrere Netze hinweg. Letzteres wird als Routing bezeichnet.

Zur Adressierung besitzt jeder Rechner im TCP/IP-Netz eine sogenannte *IP-Adresse*. Es gibt zwei verschiedene Varianten des IP, die sich vor allem durch die Länge (und damit die verfügbare Anzahl) der Adressen unterscheiden. Das klassische IPv4-Protokoll verwendet 32 Bit lange Adressen, während die neuen IPv6-Adressen eine Länge von 128 Bit besitzen. IPv6 setzt sich erst allmählich durch; der weitverbreitete Standard ist nach wie vor IPv4. Zwar wird dessen Adressraum im Internet allmählich knapp, aber der flächendeckende Umstieg auf die neue Version scheitert bisher an der Inkompatibilität der beiden Varianten zueinander sowie an der mangelnden IPv6-Unterstützung durch manche Netzwerkhard- und -software.

Wie alle Internetprotokolle und -standards ist auch das IP öffentlich verfügbar in sogenannten RFC-Dokumenten (Request for Comments) beschrieben. Alle Personen, Institutionen und Firmen, die an der Weiterentwicklung des Internets beteiligt sind, stellen ihre Vorschläge und Lösungen seit über 30 Jahren in solchen freien Dokumenten zur Verfügung; es gibt inzwischen über 5.700 davon. Sie können sie zum Beispiel unter <http://www.faqs.org/rfcs> lesen. IPv4 ist in RFC 791 dokumentiert, die neue Variante IPv6 in RFC 2460.

IPv4

Die 32 Bit langen IPv4-Adressen werden üblicherweise dezimal als einzelne, durch Punkte getrennte Bytes geschrieben, zum Beispiel »137.51.8.41«. Ein Teil dieser Adresse bezeichnet das Netzwerk, in dem sich der Rechner befindet, der Rest den Rechner selbst innerhalb dieses Netzes. Da es unterschiedlich große Netzwerke gibt, wurden ursprünglich verschiedene Arten – sogenannte Klassen – von IPv4-Netzen erfunden. Die Bits am Anfang der Adresse gaben darüber Auskunft, zu welcher Klasse eine IP-Adresse gehörte, und damit, wie viele Bits der Adresse das Netz und wie viele den Rechner (Host) kennzeichneten:

- ▶ Wenn das erste Bit 0 war, gehörte die Adresse zur Klasse A. Das erste der vier Bytes lag also im Bereich 0 bis 127. In dieser Klasse kennzeichneten die ersten 8 Bit das Netz und die restlichen 24 den Host. Ein Klasse-A-Netz bot jeweils 16,7 Millionen Host-Adressen.
- ▶ Wenn die ersten beiden Bits 10 waren, handelte es sich um die Klasse B; das erste Byte hatte einen Wert zwischen 128 und 191. Hier standen die ersten 16 Bit für das Netz und die anderen 16 für den Host. Ein Klasse-B-Netz enthielt demnach 65.536 einzelne Adressen.

- ▶ Waren die ersten drei Bits 110, dann gehörte das Netz zur Klasse C. Das erste Byte hatte also einen Wert zwischen 192 und 223. 24 Bit adressierten das Netz, und nur acht waren für den Host zuständig, sodass es nur 256 Adressen in einem Netz gab.
- ▶ Adressen, bei denen die ersten vier Bits 1110 waren und deren erstes Byte damit einen Wert zwischen 224 und 239 hatte, gehörten zur Klasse D. Sie dienten nicht der Adressierung einzelner Rechner, sondern waren sogenannte Multicast-Adressen. Multicast ermöglicht (auch heute noch) die Verteilung von Daten an mehrere Rechner, denen dieselbe Multicast-Adresse zugewiesen wird. Dies ist beispielsweise ideal für Videokonferenzen, Video-on-Demand oder andere Streaming-Dienste.

Die Klasseneinteilung ist zwar inzwischen veraltet, aber da sie die Aufteilung zwischen Netzwerk- und Host-Teil der Adressen verdeutlicht, sehen Sie in Tabelle 13.1 eine genaue Übersicht über die IP-Adressklassen. Die Anzahl der Netzwerk-Bits ist doppelt angegeben: Der erste Wert gibt die Gesamtzahl an, während der Wert in Klammern die relevante Anzahl von Netz-Bits abzüglich der Klassenkennzeichnung angibt – bei der Klasse A beispielsweise 8 (7), da das erste Bit 0 lauten *muss*. Die Anzahl der pro Netz verfügbaren Hosts ist um zwei geringer, als sich rechnerisch ergeben würde. Das liegt daran, dass die niedrigste Adresse eines Netzes das Netzwerk selbst kennzeichnet und die höchste – die Broadcast-Adresse – dazu dient, innerhalb des Netzes Daten an alle angeschlossenen Rechner zu versenden.

Klasse	Start-Byte	Netzwerk-Bits	Host-Bits	Anzahl Netze	Anzahl Hosts
A	0–127	8 (7)	24	128	16.777.214
B	128–191	16 (14)	16	16.384	65.534
C	192–223	24 (21)	14	2.097.152	254
D	224–239	Multicast-Adressen			

Tabelle 13.1 Die IP-Adressklassen und ihre Eigenschaften

Mithilfe des Dienstes NAT (Network Address Translation) können diese privaten IP-Adressen automatisch in öffentlich gültige Adressen umgewandelt werden und umgekehrt. Dies gehört zu den Standardfunktionen moderner DSL-Router und ähnlicher Geräte. Ihr Linux-System kann ebenfalls als NAT-Gateway fungieren; dies erledigt die auch für die lokale Firewall zuständige Funktion `iptables` (siehe Abschnitt 13.6, »Grundlagen der Netzwerksicherheit«).

Als das Internet Protocol entwickelt wurde, schien die Anzahl der verfügbaren Netze, die sich aus der Klassenlogik ergibt, locker auszureichen. Da die Anzahl der Rechner und Netze im Internet jedoch seit Beginn exponentiell zunimmt, erweist sich das Klassensystem schon seit langer Zeit als zu starr und verschwenderisch: Allein die Hälfte aller rechnerisch verfügbaren Adressen wird für die Klasse A benutzt, obwohl selbst die größten Netze der Welt wohl kaum aus 16,7 Millionen Hosts bestehen. Aus diesem Grund wurde ein flexibleres System namens *CIDR* (Classless Inter-Domain Routing) eingeführt. Es ermöglicht das Setzen der Grenze zwischen Netz- und Host-Teil der Adresse bei einem beliebigen Bit. Bei einer CIDR-Adresse muss diese Grenze deshalb angegeben werden. Dafür gibt es zwei verschiedene Schreibweisen:

1. Die Anzahl der Netzwerk-Bits wird durch einen Slash getrennt hinter die Adresse geschrieben: 192.78.16.97/24 ist beispielsweise die CIDR-Angabe für die Klasse-C-Adresse 192.78.16.97.
2. Alternativ kann die Grenze zwischen Netz- und Host-Bits separat durch eine sogenannte Subnet Mask (deutsch Teilnetzmaske) angegeben werden: Die Maske wird wie die IP-Adresse selbst dezimal in vier Einzel-Bytes geschrieben; Bits der Netzwerkadresse werden auf den Wert 1 gesetzt, Bits der Host-Adresse auf 0. Die Maske für ein Klasse-C-Netz ist demnach 255.255.255.0.

Die Besonderheit von CIDR wird deutlicher, wenn man eine Adresse betrachtet, die nicht zu einer der alten Klassen gehört, sondern eine individuelle Bit-Grenze besitzt: 160.76.153.12/19 gehört nicht zu dem Klasse-B-Netz 160.76.0.0 (CIDR-Adresse 160.76.0.0/16), sondern zu dem Netz 160.76.128.0/19. Es handelt sich um eines der Netze, die durch die Aufteilung des Netzes 160.76.0.0/16 in acht Teilnetze entstehen. Diese Netze werden in Tabelle 13.2 gezeigt. Mit einem Host-Teil von 13 Bit enthält jedes dieser Netze 8.190 Hosts (8.192 Einzeladressen inklusive Netzwerk- und Broadcast-Adresse). Die Subnet Mask lautet 255.255.224.0.

Netzwerk	Erster Host	Letzter Host	Broadcast-Adresse
160.76.0.0/19	160.76.0.1	160.76.31.254	160.76.31.255
160.76.32.0/19	160.76.32.1	160.76.63.254	160.76.63.255
160.76.64.0/19	160.76.64.1	160.76.95.254	160.76.95.255
160.76.96.0/19	160.76.96.1	160.76.127.254	160.76.127.255
160.76.128.0/19	160.76.128.1	160.76.159.254	160.76.159.255

Tabelle 13.2 Alle Netze, die durch CIDR-Aufteilung (Subnetting) des Netzes 160.76.0.0/16 in acht Teilnetze entstehen

Netzwerk	Erster Host	Letzter Host	Broadcast-Adresse
160.76.160.0/19	160.76.160.1	160.76.191.254	160.76.191.255
160.76.192.0/19	160.76.192.1	160.76.223.254	160.76.223.255
160.76.224.0/19	160.76.224.1	160.76.255.254	160.76.255.255

Tabelle 13.2 Alle Netze, die durch CIDR-Aufteilung (Subnetting) des Netzes 160.76.0.0/16 in acht Teilnetze entstehen (Forts.)

Noch flexibler als CIDR ist die sogenannte VLSM-Adressierung (Variable Length Subnet Mask). Durch dieses Adressierungsschema lässt sich ein Netzwerk in Teilnetze unterschiedlicher Größe unterteilen, was beispielsweise in einer Firma mit unterschiedlich großen Abteilungen sehr nützlich ist.

Jede IP-Adresse muss im Internet einmalig sein. Allerdings hat die IANA (Internet Assigned Numbers Authority) einige Adressen zur Verwendung in privaten, nicht direkt mit dem Internet verbundenen Netzwerken freigegeben:

- ▶ das Netz 10.0.0.0/8,
- ▶ die 16 Netze 172.16.0.0/16 bis 172.31.0.0/16,
- ▶ die 256 Netze 192.168.0.0/24 bis 192.168.255.0/24.

Eine besondere Bedeutung haben darüber hinaus die folgenden IPv4-Adressen:

- ▶ 127.0.0.1 ist die sogenannte Loopback-Adresse, über die ein Host mit sich selbst TCP/IP-Kommunikation betreiben kann.² Beim Testen von Netzwerkdiensten und -anwendungen werden Sie diese Adresse oft verwenden, um Client und Server auf demselben Rechner zu betreiben.
- ▶ 255.255.255.255 ist die allgemeine Broadcast-Adresse. Sie kann zum Beispiel von einem Host verwendet werden, der beim Booten seine eigene IP-Adresse im Netz erfragt, weil sie über den Dienst DHCP dynamisch zugewiesen wird.
- ▶ Das 169.254.0.0/16 wird für APIPA (Automatic Private IP Addressing) oder »link local« verwendet – über diesen Dienst kann sich ein DHCP-Client selbst eine Adresse zuweisen, wenn aus irgendeinem Grund kein DHCP-Server verfügbar ist.

IPv4-Datagramme bestehen aus den Daten, die die übergeordnete Transportschicht dem IP-Protokoll anvertraut, plus dem IP-Header.

² Genau genommen ist sogar das gesamte Netz 127.0.0.0/8 für Loopback vorgesehen, aber die traditionelle Adresse ist 127.0.0.1, und nur dieser wird für gewöhnlich der Hostname *localhost* zugewiesen.

Dieser Header besitzt den folgenden Aufbau:

- ▶ *Version* (4 Bit)
Die Versionsnummer des IP-Protokolls, die das Paket verwendet. Bei IPv4, wie der Name schon sagt, die Version 4.
- ▶ *IHL* (4 Bit)
Internet Header Length; die Länge des Internet-Headers in 32-Bit-Worten. Der kleinste mögliche Wert beträgt 5.)
- ▶ *Type of Service* (8 Bit)
Ein Code, der die Art des Datenpakets bestimmt. Bestimmte Sorten von Paketen, etwa für den Austausch von Routing- oder Status-Informationen, werden von bestimmten Netzen bevorzugt weitergeleitet. In ihrem Aprilscherz des Jahres 1999 bot die Computerzeitschrift c't ein angebliches Tool zum Download an, das diese Quality-of-Service-Informationen manipulieren könne, um die Geschwindigkeit von Internet-Verbindungen zu erhöhen.
- ▶ *Paket-Gesamtlänge* (16 Bit)
Die Gesamtlänge des Datagramms in Bytes, Header und Nutzdaten
- ▶ *Identifikation* (16 Bit)
Ein durch den Absender frei definierbarer Identifikationswert, der beispielsweise das Zusammensetzen fragmentierter Datagramme ermöglicht.
- ▶ *Flags* (3 Bit)
Kontrollflags, die die Paketfragmentierung regeln. Das erste Bit ist reserviert und muss immer 0 sein, das zweite (DF) bestimmt, ob das Paket fragmentiert werden darf (Wert 1) oder nicht (0), das dritte (MF) regelt, ob dieses Paket das letzte Fragment (0) ist oder ob weitere Fragmente folgen (1).
- ▶ *Fragment-Offset* (13 Bit)
Dieser Wert (angegeben in 64-Bit-Blöcken) legt fest, an welcher Stelle in einem Gesamtpaket dieses Paket steht, falls es sich um ein Fragment handelt. Das erste Fragment oder ein nicht fragmentiertes Paket erhält den Wert 0.
- ▶ *Time to Live* (8 Bit)
Der TTL-Mechanismus sorgt dafür, dass Datagramme nicht endlos im Internet weitergeleitet werden, falls die Empfängerstation nicht gefunden wird. Jeder Router, der ein Datagramm weiterleitet, zieht von diesem Wert 1 ab; wird der Wert 0 erreicht, dann leitet der betreffende Router das Paket nicht weiter, sondern verwirft es.
- ▶ *Protokoll* (8 Bit)
Die hier gespeicherte Nummer bestimmt, für welches Transportprotokoll der Inhalt des Datagramms bestimmt ist. Die beiden wichtigsten Transportprotokolle, TCP und UDP, werden im nächsten Abschnitt beschrieben.

- ▶ *Header-Prüfsumme* (16 Bit)
Die Prüfsumme stellt eine einfache Plausibilitätskontrolle für den Datagramm-Header zur Verfügung. Ein Paket, dessen Header-Prüfsumme nicht korrekt ist, wird nicht akzeptiert und muss erneut versendet werden.
- ▶ *Quelladresse* und *Zieladresse* (je 32 Bit)
Die IP-Adressen von Absender und Empfänger. IP-Adressen wurden oben ausführlich behandelt.
- ▶ *Optionen* (variable Länge)
Die meisten IP-Datagramme werden ohne zusätzliche Optionen versandt, da Absender- und Empfänger-Host sowie alle auf dem Weg befindlichen Router die jeweils verwendeten Optionen unterstützen müssen. Zu den verfügbaren Optionen gehören unter anderem Sicherheitsfeatures und spezielle Streaming-Funktionen.

IPv6

IPv6-Adressen sind, wie erwähnt, 128 Bit lang. Dies ergibt einen rechnerischen Adressraum von unvorstellbaren $3,4 \times 10^{38}$ Adressen. Auch wenn diese Anzahl von Hosts wohl niemals erreicht werden wird, besitzt eine so große Adressenzahl andere Vorteile: Erstens sind Unmengen von Netzen mit zahlreichen verschiedenen Größen möglich, und zweitens passt die IPv6-Adressierung zu den zahlreichen Geräten, die inzwischen neben den klassischen Allround-Computern ins Internet drängen: Handys, PDAs, Armbanduhren und sogar Kühlschränke und Mikrowellengeräte.

Da es etwas unhandlich wäre, die IPv6-Adressen in 16 dezimalen Byte-Gruppen zu schreiben, werden sie stattdessen in acht vierstelligen Hexadezimalblöcken dargestellt. Eine IPv6-Adresse lautet also beispielsweise so:

09AC:7B03:CC3D:BD88:0000:0037:6293:CCF1. Sie lässt sich abkürzen, indem die führenden Nullen jeder Gruppe weggelassen werden und indem eine zusammenhängende Gruppe aus reinen Nullenblöcken durch `::` dargestellt wird. Die Beispieladresse lautet in dieser Schreibweise also folgendermaßen: 9AC:7B03:CC3D:BD88::`37:6293:CCF1`.

Die Grenze zwischen Netzwerk- und Host-Teil der IPv6-Adressen wurde von Anfang an flexibel gesetzt; wie bei IPv4-Adressen wird die Bit-Anzahl des Netzwerk-Teils durch einen `/` getrennt hinter die Adresse geschrieben. Dabei kann der Rest der Adresse weggelassen werden, wenn das Netzwerk als solches gemeint ist, zum Beispiel 9AC:7B03/32.

Während die automatische Zuweisung der Adressen sich bei IPv4 nur durch nachträglich erfundene Dienste wie DHCP oder APIPA realisieren lässt, ist sie bei

IPv6 die Regel. Aus diesem Grund wird der Host-Teil der Adresse bei einem Ethernet-LAN beispielsweise automatisch aus der 48 Bit langen MAC-Adresse (Media Access Control; eindeutige Hardware-Adresse) der Netzwerkschnittstelle – ergänzt um einen 16 Bit langen Kennzeichnungsblock – gebildet.

In der Praxis wird IPv6 heute vornehmlich durch *Tunnelung* benutzt: Die IPv6-Datenpakete werden in IPv4-Datenpakete »verpackt« (erhalten also einen IPv4-Paket-Header) und über das IPv4-adressierte Internet zum Ziel geleitet – ein Netzwerk, das intern die entsprechende IPv6-Adressierung versteht. Es gibt zahlreiche kommerzielle und freie Anbieter solcher IPv6-Tunneldienste.

Der Header von IPv6-Datagrammen ist weniger komplex aufgebaut als der IPv4-Header. Er besitzt folgende Struktur:

- ▶ *Version* (4 Bit)
Die Versionsnummer des IP-Protokolls; hier natürlich 6.
- ▶ *Klasse* (8 Bit)
Dieses Feld gibt die Priorität an, mit der das Datagramm übertragen werden soll. Es ist noch nicht abschließend geklärt, wie die entsprechenden Werte aussehen sollen.
- ▶ *Flow Label* (20 Bit)
Ein Erweiterungsfeld, in das ein von 0 verschiedener Wert eingetragen wird, wenn IPv6-Router das Datagramm auf besondere Weise behandeln sollen. Es dient vor allem der Implementierung der Quality-of-Service-Funktionalität, mit deren Hilfe Paketsorten voneinander unterschieden werden, um beispielsweise Echtzeitanwendungen wie Streaming Multimedia oder Videokonferenzen zu unterstützen.
- ▶ *Payload Length* (16 Bit)
Die Länge der Nutzdaten, die auf den Header folgen.
- ▶ *Next Header* (8 Bit)
Der Wert in diesem Feld gibt den Typ des ersten Erweiterungs-Headers an, falls einer vorhanden ist. Es gibt bisher sechs Arten von Erweiterungs-Headern; eine Übersicht finden Sie in Tabelle 13.3.
- ▶ *Hop Limit* (8 Bit)
Ein neuer Name für die Time-to-Live-Funktionalität: Jeder Router zieht von dem ursprünglichen Wert 1 ab; bei Erreichen des Wertes 0 wird das Paket verworfen.
- ▶ *Quell- und Zieladresse* (je 128 Bit)
Die Adresse des Absenders und des Empfängers; genau wie bei IPv4, nur entsprechend der Protokollspezifikation 128 statt 32 Bit lang.

Header	Next-Header-Code	Beschreibung
Hop-by-Hop Options Header	0	Optionen, die bei jedem Routing-Schritt ausgeführt werden müssen
Routing Header	43	Festlegung der Router, über die das Paket geleitet werden soll
Fragment Header	44	Der Absender muss bei IPv6 die MTU herausfinden und Fragmente selbst bilden; die Fragment-Informationen befinden sich hier.
Authentication Header	51	Authentifizierung des Absenders gegenüber dem Empfänger
Encapsulating Security Payload Header	50	Dient der Verschlüsselung des Datagramms (IPv6).
Destination Options Header	60	Optionen, die nur für den Ziel-Host bestimmt sind
Upper-Layer Header	59	Header einer höheren Schicht; aus IPv6-Sicht also Nutzdaten

Tabelle 13.3 Die verschiedenen Typen von IPv6-Erweiterungs-Headern

IP-Routing

Das Wichtigste am Konzept des IPs ist die Fähigkeit, Daten über mehrere miteinander verbundene Netze weiterzuleiten. Diese Aufgabe übernehmen spezialisierte Rechner, die *Router*. Sie sind jeweils mit mindestens zwei verschiedenen Netzwerken verbunden und verfügen über Informationen darüber, welche Datenpakete sie wohin weiterleiten müssen, damit diese letztlich ihr Ziel erreichen. Während diese Routing-Informationen innerhalb kleinerer Firmennetze statisch konfiguriert werden, führen die Router der Internet-Provider und anderer Betreiber großer Netze sogenannte Routing-Protokolle aus, mit deren Hilfe die Routing-Konfiguration automatisch veröffentlicht und verbreitet wird.

Auch jeder Host im TCP/IP-Netzwerk benötigt grundlegende Routing-Informationen, um Daten jeweils über die richtige Netzwerkschnittstelle an den korrekten Router weiterzuleiten. Aus der Sicht eines Hosts gibt es zwei Sorten von Routern: Einen »normalen« Router, an den der Host Daten weitergibt, deren Ziel ein bestimmtes Netzwerk ist, und das Default Gateway (der Standard-Router), dem alle Datenpakete übergeben werden, für deren Ziel eben kein spezieller Router konfiguriert wurde.

Angenommen, das (rein interne) Netzwerk 192.168.0.0/24 einer kleinen Firma mit der Domain *test.local* ist über den Router *intern* (192.168.0.2) mit dem internen Nachbar-Teilnetz 192.168.1.0/24 verbunden, während *extern* (192.168.0.1) das Default Gateway ist. Im Teilnetz 192.168.1.0/24 besitzt *intern* die IP-Adresse 192.168.1.1 und ist dessen einziger Router und damit Default Gateway.

Aus diesem Szenario ergibt sich für einen Host im Netz 192.168.0.0/24 die folgende Routing-Situation:

- ▶ Datenpakete für Rechner im Netzwerk 192.168.0.0/24 werden ohne Umweg über einen Router direkt an den Empfänger versandt.
- ▶ Daten, die für Hosts im Netz 192.168.1.0/24 bestimmt sind, werden dem Router *intern* (192.168.0.2) zur Weiterleitung übergeben.
- ▶ Daten für alle anderen Netze, also für das gesamte Internet, werden an den Router *extern* (192.168.0.1) weitergegeben.

Ein Host im Netzwerk 192.168.1.0/24 besitzt dagegen eine einfachere Routing-Konfiguration; da es in diesem Teilnetz nur einen Router gibt, brauchen nur zwei Fälle unterschieden zu werden:

- ▶ Daten für das Netz 192.168.1.0/24 werden ohne Routing unmittelbar zugestellt.
- ▶ Die restlichen Datenpakete werden dem Router *intern* (hier: 192.168.1.1) übergeben; dieser weiß natürlich, dass er Daten für das Netz 192.168.0.0/24 über seine Schnittstelle 192.168.0.2 unmittelbar zustellen kann, während er alle anderen an den nächsten Router *extern* (192.168.0.1) weiterleiten muss.

In der Praxis führen Router eigene, komplexe Protokolle aus, um Informationen über die jeweils erreichbaren Netze miteinander auszutauschen. Man unterscheidet zwei Arten: Interne Routing-Protokolle wie das Routing Information Protocol (RIP) arbeiten innerhalb der Netzwerke eines einzelnen Anbieters (autonomes System), während externe Routing-Protokolle – etwa das Border Gateway Protocol (BGP) – verschiedene autonome Systeme miteinander verbinden.

Damit ein IP-Datagramm nicht endlos im Netz kreist, wenn sein Bestimmungsort nicht gefunden werden kann, enthält sein Header ein 8 Bit breites Feld namens TTL (Time To Live). Jeder einzelne Router, den das Paket passiert, zieht von diesem Wert 1 ab. Sobald der Wert 0 geworden ist, verwirft der Router das Paket. Ein Datagramm kommt also entweder innerhalb von 255 Hops (Routing-Schritten) an oder gar nicht. In der Regel sind allerdings erheblich weniger Hops erforderlich.

13.1.3 Transportprotokolle

Aufgabe der Host-zu-Host-Transportschicht ist die direkte Lieferung der Datenpakete von einer Anwendung auf einem Host zu einer Anwendung auf einem anderen Host. Ein Transportprotokoll nimmt die Daten von einem Anwendungsprogramm entgegen, das über das Netzwerk kommunizieren möchte. Es teilt die Daten in Datenpakete auf, versieht sie mit einer eindeutigen Nummer, die die entsprechende Anwendung kennzeichnet, und reicht sie an das IP weiter, das die tatsächliche Datenweiterleitung über das Netzwerk durchführt.

Im Internet-Protokollstapel gibt es zwei verschiedene Transportprotokolle. Jede Netzwerkanwendung kann sich dasjenige aussuchen, das besser zu ihr passt: Das TCP garantiert die Auslieferung jedes einzelnen Datenpakets und sorgt durch eine Datenpaketnummerierung dafür, dass die Pakete am Ziel wieder in der richtigen Reihenfolge zusammengesetzt werden können. Das UDP ist dagegen erheblich schneller als TCP, kann aber nur einzelne Pakete ohne Reihenfolge übertragen und überprüft auch nicht, ob sie tatsächlich ankommen. Während TCP also datenstromorientiert ist, ist UDP nachrichtenorientiert.

TCP

Das TCP (RFC 793) sorgt für die Übertragung eines kontinuierlichen Datenstroms. Obwohl netzwerktypisch noch immer jedes einzelne Paket einen beliebigen Routing-Weg zum Ziel wählen kann, wird durch die eingebaute Flusskontrolle eine virtuelle Punkt-zu-Punkt-Verbindung zwischen den beiden Hosts eingerichtet.

Auch TCP-Pakete besitzen einen eigenen Header, dessen Struktur aus folgenden Feldern besteht:

- ▶ *Quell-Port* (16 Bit)
Ports stellen eine Methode zur Identifikation der konkreten Anwendungen zur Verfügung, die auf den beteiligten Hosts miteinander kommunizieren. Der Quell-Port ist die Portnummer des Absenders.
- ▶ *Ziel-Port* (16 Bit)
Dies ist entsprechend die Portnummer des Empfängers.
- ▶ *Sequenznummer* (32 Bit)
Normalerweise gibt diese Nummer an, dem wievielten Byte der zu übertragenden Sequenz das erste Nutzdaten-Byte des Pakets entspricht. Ausnahme: Ist das SYN-Flag gesetzt, so wird die Anfangs-Sequenznummer (Initial Sequence Number, ISN) angegeben.

- ▶ *Bestätigungsnummer* (32 Bit)
Das Startbyte der Sequenz, deren Übertragung als Nächstes erwartet wird; ist nur bei gesetztem ACK-Bit von Bedeutung.
- ▶ *Offset* (4 Bit)
Anzahl der 32-Bit-Wörter, aus denen der Header besteht; gibt entsprechend den Beginn der Nutzdaten im Paket an.
- ▶ *Reserviert* (6 Bit)
Dieser Header ist reserviert für zukünftige Anwendungen; muss 0 sein.
- ▶ *Flags* (6 Bit)
Verschiedene Statusbits; im Einzelnen:
 - ▶ *URG*: Urgent Data wird versandt; der Inhalt des Urgent-Zeigers muss beachtet werden.
 - ▶ *ACK*: Acknowledgement – das Bestätigungsfeld muss berücksichtigt werden.
 - ▶ *PSH*: Push-Funktion – ist dieses Bit gesetzt, dann wird die Pufferung des Pakets verhindert; es wird unmittelbar gesendet.
 - ▶ *RST*: Reset – Verbindung zurücksetzen
 - ▶ *SYN*: Sequenznummern synchronisieren
 - ▶ *FIN*: Ende der Sequenz; keine weiteren Daten vom Absender
- ▶ *Fenster* (16 Bit)
Die Anzahl von Datenbytes, die der Absender des Pakets zu empfangen bereit ist; basiert unter anderem auf der IP-MTU der verwendeten Schnittstelle.
- ▶ *Prüfsumme* (16 Bit)
Anhand dieser einfacheren Plausibilitätskontrolle kann die Korrektheit der übertragenen Daten überprüft werden.
- ▶ *Urgent-Zeiger* (16 Bit)
Ein Zeiger auf das Byte der aktuellen Sequenz, das Urgent Data enthält. Wird nur ausgewertet, wenn das URG-Flag gesetzt ist.
- ▶ *Optionen* (variable Länge)
Verschiedene hersteller- und implementierungsabhängige Zusatzinformationen; stets ein Vielfaches von 8 Bit lang.

Die TCP-Verbindung beginnt durch den sogenannten Drei-Wege-Handshake:

- ▶ Der Host, der die Verbindung initiiert (üblicherweise der Client, denn er »will etwas« vom Server), sendet dem Empfänger ein Datenpaket, in dessen TCP-Header ein spezielles Bit namens SYN gesetzt ist.

- ▶ Der Empfänger antwortet mit einem Paket, in dem die beiden Flags SYN und ACK gesetzt sind.
- ▶ Der ursprüngliche Absender beantwortet dieses Paket noch einmal mit einem Datenpaket, in dem nur ACK gesetzt ist.

Diese Vorgehensweise garantiert, dass beide Seiten sowohl senden als auch empfangen können. Die Flusskontrolle kommt dadurch zustande, dass der Header jedes Pakets eine Sequenznummer enthält. Sie gibt den Offset des Datenbytes für den Teil des Gesamtdatenstroms an, der mit diesem Paket versandt wird. Der Absender erwartet dann für jedes einzelne Paket eine Bestätigung, in der das ACK-Bit gesetzt ist und die eine Bestätigungsnummer enthält: den Byte-Offset des nächsten erwarteten Pakets. Falls eine Bestätigung innerhalb einer bestimmten Zeit ausbleibt, sendet der Absender das Paket einfach erneut.

Die Identifikation der beteiligten Anwendungen erfolgt durch ein Paar 16 Bit langer Portnummern. Da eine bestimmte Verbindung durch beide Portnummern gekennzeichnet wird, ist es möglich, dass ein TCP-Server jeweils eine feste Portnummer hat, während ein Client eine zufällige Nummer (»Ephemeral Ports«) erhält. Die wichtigsten Server-Dienste haben festgelegte Ports (»Well-known Ports«) zwischen 0 und 1.023 – ein Webserver hat zum Beispiel standardmäßig die Portnummer 80. Auf UNIX-Systemen dürfen diese Ports lokal nur durch root verwendet werden. Daneben gibt es auch registrierte Server-Ports über 1.024, zum Beispiel 3306 für MySQL. Alle Zuweisungen finden Sie im Internet unter <http://www.iana.org/assignments/port-numbers>. Auf Ihrem System gibt es zudem die Datei `/etc/services`, in der alle bekannten Zuordnungen stehen. Tabelle 13.4 zeigt einige weitere Beispiele, die teilweise auch für UDP gelten.

Nummer	Transportprotokoll	Name	Beschreibung
7	TCP, UDP	echo	genaue Rückgabe der übermittelten Daten zur Kontrolle
13	TCP, UDP	daytime	Uhrzeitsynchronisation (RFC 867)
20	TCP	ftp	FTP-Datenstrom
21	TCP	ftp	FTP-Steuerung
22	TCP	ssh	Secure Shell – Telnet-Alternative mit Verschlüsselung
23	TCP	telnet	Terminal-Emulation
25	TCP	smtp	E-Mail-Versand

Tabelle 13.4 Einige wichtige TCP-Port- beziehungsweise UDP-Dienstnummern

Num-mer	Transport-protokoll	Name	Beschreibung
53	TCP, UDP	domain	Nameserver-Abfragen
80	TCP	http	Webserver
110	TCP	pop3	E-Mail-Postfach-Server (klassisch)
142	TCP	imap	E-Mail-Postfach-Server (modern)
443	TCP	https	gesicherte HTTP-Verbindungen über SSL/TLS
3128	TCP	proxy	Proxy-Server (squid)
3306	TCP	mysql	MySQL-Datenbankserver

Tabelle 13.4 Einige wichtige TCP-Port- beziehungsweise UDP-Dienstnummern (Forts.)

Interessant ist schließlich auch das Thema *Urgent Data*: Manchmal muss ein Host einen anderen über einen besonderen Zustand informieren, beispielsweise eine Konfigurationsänderung, oder einen vom Benutzer initiierten Abbruch mitteilen. Zu diesem Zweck wird das URG-Flag gesetzt; der Empfänger ermittelt daraufhin aus dem Urgent-Zeiger-Feld des Paket-Headers die Bytenummer innerhalb der Sequenz, in der sich diese dringlichen Daten befinden. Es handelt sich stets nur um ein einziges Byte, das auch als »Out of Bound-Byte« bezeichnet wird, weil es nicht zum gewöhnlichen Datenstrom gehört. Es ist also unmöglich, auf diesem Weg eine längere dringende Mitteilung zu versenden, aber immerhin besteht die Möglichkeit, bestimmte zwischen den Anwendungen vereinbarte Signale zu versenden.

UDP

Wenn Einzeldaten ihr Ziel so schnell wie möglich erreichen sollen, es aber nicht auf Vollständigkeit oder auf die richtige Reihenfolge einer größeren Datenmenge ankommt, kann eine Anwendung statt TCP das UDP verwenden. Dieses Protokoll ist in RFC 768 spezifiziert. Der Name hat damit zu tun, dass die IP-Datagramme der Anwendung ohne viel zusätzlichen Aufwand zur Verfügung stehen.

Der Hauptgrund, warum UDP schneller ist als TCP, liegt am wesentlich kleineren Paket-Header. Das Verhältnis zwischen Konfigurations- und Nutzdaten ist also erheblich günstiger als beim TCP. Der Header enthält nämlich nichts weiter als die beiden bei TCP beschriebenen Portnummern (hier Service-Nummern genannt), Paketlänge und Prüfsumme; jedes der Felder hat eine Länge von 2 Byte.

Die einzelnen Header-Felder haben dieselbe Bedeutung wie die gleichnamigen Felder beim TCP-Protokoll. Mit der Länge ist hier die Länge des gesamten Pakets inklusive dieses Headers gemeint. Der Quell-Port wird häufig einfach auf 0 gesetzt: Da UDP dem schnellen Versand einer einzelnen Nachricht dient, auf die in der Regel keine Antwort erwartet wird, ist es nicht nötig, diese Information festzulegen. Der Ziel-Port ist dagegen meist der festgelegte Port des UDP-Servers, an den das Paket verschickt wird. UDP wird für viele einfache Internetdienste verwendet: die Uhrzeitsynchronisation über ein Netzwerk, den Echo-Dienst zur Kontrolle der Funktionstüchtigkeit von Verbindungen oder entfernten Hosts und so weiter.

Ein UDP-Datenpaket ist seiner Natur gemäß eine einzelne Nachricht, die so schnell wie möglich ans Ziel befördert wird. Der Absender kümmert sich nicht darum, ob sie ankommt oder nicht, und erwartet auch keine Empfangsbestätigung. Man könnte TCP gewissermaßen als »Einschreiben mit Rückschein« bezeichnen, während UDP eher einer Postkarte entspricht.

13.2 Lokale Netzwerkschnittstellen konfigurieren

Man kann Netzwerke anhand ihrer Reichweite grob in die beiden Varianten lokales Netzwerk (Local Area Network, kurz LAN) und Fernnetzwerk (Wide Area Network oder WAN) unterteilen. In diesem Abschnitt werden mit Ethernet und Wireless LAN die beiden wichtigsten lokalen Netzwerkschnittstellen behandelt; um die Konfiguration des Fernzugriffs geht es dann im nächsten Abschnitt.

13.2.1 Ethernet

Der verbreitetste Standard für lokale Netzwerke wird als *Ethernet* bezeichnet (der Namensbestandteil »Äther« stammt daher, dass es sich anfangs um einen Standard für Datenfunk handelte). Ethernet beschreibt keine bestimmte Verkabelungstechnik (folgerichtig wurden im Laufe der Jahre mehrere entwickelt), sondern vor allem ein einheitliches Netzzugangsverfahren namens *CSMA/CD* (Carrier Sense Multiple Access with Collision Detection), das als Standard Nummer 802.3 vom *IEEE* (Institute of Electrical and Electronical Engineers) zertifiziert wurde. Dieses Verfahren funktioniert im Prinzip folgendermaßen:

1. Ein Gerät, das Daten senden möchte, lauscht den Netzabschnitt ab, um festzustellen, ob dieser gerade frei ist, das heißt, ob gerade kein anderes Gerät sendet (Carrier Sense).
2. Wurde in Schritt 1 festgestellt, dass der Netzabschnitt frei ist, beginnt die Station mit dem Senden der Daten. Möglicherweise hat auch eine andere Station

festgestellt, dass das Netz frei ist, und beginnt gleichzeitig mit dem Senden (Multiple Access).

3. Falls auf die beschriebene Art und Weise zwei Stationen gleichzeitig mit dem Senden begonnen haben, findet eine sogenannte Datenkollision statt, die von den beteiligten Stationen entdeckt wird (Collision Detection). Eine Station, die eine Kollision bemerkt, stellt das Senden von Nutzdaten ein und versendet stattdessen eine Warnmeldung (Jam Signal).
4. Eine Station, die wegen einer Datenkollision das Senden abgebrochen hat, beginnt nach einer zufällig gewählten Zeitspanne von wenigen Millisekunden erneut mit dem Senden. Genau diese Zufälligkeit der Zeitspanne, die nach einem komplizierten Verfahren berechnet wird, ist enorm wichtig, damit die beiden Stationen beim nächsten Versuch nicht wieder genau gleichzeitig mit dem Senden beginnen.

Jede Ethernet-Schnittstelle, also die Netzwerkkarte beziehungsweise der fest eingebaute Anschluss, ist mit einer weltweit einmaligen Identifikationsnummer ausgestattet, der sogenannten *MAC-Adresse* (Media Access Control). Es handelt sich um eine 48 Bit lange Zahl, die in sechs hexadezimalen Blöcken zwischen 0 und 255 (00 bis FF hex) geschrieben wird, zum Beispiel 00-A0-C9-E8-5F-64. Ein Protokoll namens ARP (Address Resolution Protocol) übernimmt die Umsetzung der MAC-Adressen in die willkürlich zugewiesenen IP-Adressen und umgekehrt.

Die Datenpakete – auf der Netzzugangsschicht Frames genannt – werden mit den MAC-Adressen der sendenden und der empfangenden Station versehen und in der Regel an alle Stationen im Segment versandt. Jede Station überprüft daraufhin, ob die Daten für sie bestimmt sind oder nicht. Im Übrigen kann man Ethernet-Schnittstellen auch in den »promiscuous mode« schalten, in dem sie ohne Unterschied alle Daten entgegennehmen – auf diese Weise kann der gesamte Datenverkehr in einem Netzsegment überwacht werden.

Die MAC-Adresse wird normalerweise nicht über das jeweilige Teilnetz hinaus verbreitet. Nach außen würde ihre Verwendung auch keinen Sinn ergeben, da das nächste Teilnetz auf einer Route womöglich noch nicht einmal zum Ethernet-Standard gehört.

Heutzutage wird Ethernet in aller Regel über *Twisted-Pair-Kabel* betrieben (die diversen Methoden mit Koaxialkabeln sind aus der Mode gekommen). Bei dieser Kabelsorte handelt es sich um einen verdrehten Kupfer-Zweidrahtleiter: Je zwei isolierte Kupferdrähte werden umeinandergewickelt – dies verhindert die gegenseitige Beeinträchtigung der Signalqualität, die bei parallel zueinander verlaufenden Kabeln durch die elektromagnetischen Felder auftreten würde. In einem Twisted-Pair-Kabel verlaufen üblicherweise vier, manchmal auch acht solcher

Doppeladern nebeneinander. Sie enden auf beiden Seiten in einem RJ-45-Stecker, der auch für ISDN-Anschlüsse verwendet wird. Bekannt sind solche Kabel vor allem durch ihre Verwendung als Telefonleitungen.

Alle Rechner werden jeweils über ein eigenständiges Kabel an einen zentralen Verteiler angeschlossen. Der Vorteil dieser Form der Vernetzung besteht grundsätzlich darin, dass der Ausfall einer einzelnen Verbindung zwischen einem Rechner und dem Verteiler nicht zur Unterbrechung des gesamten Netzes führt, wie es beim busförmigen Koaxialkabel-Ethernet der Fall ist.

Der zentrale Verteiler wird in seiner einfacheren Form *Hub* genannt, die etwas teurere, aber leistungsfähigere Bauweise heißt *Switching Hub* oder kurz *Switch*. Die innere Struktur des Hubs ist letztlich busförmig, sodass es genau wie bei der klassischen Vernetzung über Koaxialkabel zu Datenkollisionen kommen kann. Ein Switch stellt dagegen für zwei Stationen, die miteinander kommunizieren möchten, eine exklusive Punkt-zu-Punkt-Verbindung bereit. Dies geschieht dadurch, dass ein Switch die MAC-Adressen aller Schnittstellen zwischenspeichert, an die er bereits Daten ausgeliefert hat, und auf diese Weise die restlichen Stationen nicht mehr mit Daten behelligen muss, die gar nicht für sie bestimmt sind. Da die Preise für Netzwerkzubehör in den letzten Jahren allgemein stark gesunken sind, gibt es eigentlich keinen Grund mehr, etwas anderes als einen Switch einzusetzen.

Bei einem Hub teilen sich alle beteiligten Rechner die gesamte Übertragungsgeschwindigkeit, beim Switch steht sie dagegen jeder einzelnen Verbindung zur Verfügung.

Hubs oder Switches weisen in der Regel 5 bis 24 Anschlüsse (Ports) auf, an die jeweils ein Gerät angeschlossen werden kann. Um Netzwerke mit mehr Geräten zu betreiben, sind diese Geräte kaskadierbar: Die meisten Hubs oder Switches besitzen einen speziellen Port, den sogenannten Uplink-Port, der über ein Kabel mit einem normalen Port eines weiteren Verteilers verbunden werden kann. Bei vielen Hubs/Switches kann ein einzelner Port über einen Schalter zwischen Normal und Uplink umgeschaltet werden.

Die einzige Ausnahme von der allgemeinen Regel, dass ein Hub oder Switch benötigt wird, bildet der Sonderfall, in dem nur zwei Rechner miteinander vernetzt werden sollen: Die beiden Stationen können unmittelbar über ein sogenanntes Crosslink-Kabel verbunden werden. Dieses spezielle Kabel besitzt überkreuzte Anschlusspaare anstelle der geradlinig verlaufenden bei normalen Twisted-Pair-Kabeln.

Es existieren zwei Arten von Ethernet über Twisted Pair, die unterschiedliche Übertragungsgeschwindigkeiten unterscheiden: Bei *10 Base T* beträgt die Datenübertragungsrate 10 MBit/s; bei *100 Base T* (auch Fast Ethernet genannt) sind es 100 MBit/s.

Die meisten Netzwerkkarten, Hubs und Switches, die heute verkauft werden, unterstützen beide Übertragungsraten. Der zu verwendende Wert kann bei vielen Netzwerkkarten per Software eingestellt werden, häufiger wird er automatisch gewählt. Natürlich sollten Sie prinzipiell darauf achten, keine reine 10-MBit-Hardware mehr zu kaufen. Aber möglicherweise hat 100-MBit-Hardware der ersten Generation, die nicht auf 10 MBit/s heruntergeschaltet werden kann, sogar noch schlimmere Einschränkungen zur Folge. Zwar ist es bei normalen Standard-PCs ein Leichtes, die Netzwerkkarte gegen ein neueres Modell auszutauschen, um die Kompatibilität zu einer aktualisierten Netzwerkumgebung aufrechtzuerhalten, aber bei anderen Geräten wie beispielsweise Netzwerkdruckern oder kompakten Router-Boxen ist das eventuell nicht möglich. Solche Geräte könnten zu einem reinen 100er-Netz eventuell nicht mehr kompatibel sein.

Noch neuere Formen von Ethernet erreichen Übertragungsraten von 1.000 MBit/s (*Gigabit Ethernet*), entweder über Lichtwellenleiter (*1000 Base FL* für »Fiber Logic«) oder über mehradrige Twisted-Pair-Kabel (*1000 Base TX*). Bereits entwickelt, aber noch nicht weitverbreitet ist Ethernet mit 10 GBit/s.

ifconfig

Wenn Sie einer Ethernet-Schnittstelle (oder auch einigen anderen Schnittstellenarten) eine IP-Adresse zuweisen möchten, funktioniert dies am einfachsten über NetworkManager oder YaST (siehe unten). Traditionell können Sie aber auch den Befehl `ifconfig` (für »interface configuration«) verwenden. Wenn Sie `ifconfig` ohne Parameter eingeben, erhalten Sie eine Übersicht über die aktuelle Konfiguration, die zum Beispiel folgendermaßen aussieht:

```
$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:00:CB:53:17:64
          inet addr:192.168.1.2  Bcast:192.168.1.255
           Mask:255.255.255.0
          inet6 addr: fe80::200:cbff:fe53:1764/10
           Scope:Link
          inet6 addr: fe80::cb53:1764/10 Scope:Link
          EtherTalk Phase 2 addr:65280/53
          UP BROADCAST RUNNING MULTICAST  MTU:150
           Metric:1
```

```

RX packets:8908250 errors:0 dropped:8242
  overruns:0 frame:0
TX packets:7453081 errors:0 dropped:0
  overruns:0 carrier:0
collisions:2829807

lo      Link encap:Local Loopback
        inet addr:127.0.0.1  Mask:255.0.0.0
        inet6 addr: ::1/128 Scope:Host
        EtherTalk Phase 2 addr:0/0
        UP LOOPBACK RUNNING MTU:3924  Metric:1
        RX packets:994 errors:0 dropped:0 overruns:0
          frame:0
        TX packets:994 errors:0 dropped:0 overruns:0
          carrier:0
        collisions:0

```

Wie Sie sehen, besitzt der Computer eine Ethernet-Karte mit dem Gerätenamen `eth0` und der IP-Adresse `192.168.1.2` sowie natürlich ein Loopback-Interface. Wenn der Rechner noch weitere Ethernet-Karten hätte, würden ihre Gerätenamen entsprechend `eth1`, `eth2` und so weiter lauten.

Wenn Sie der Schnittstelle `eth0` eine andere IP-Adresse zuweisen möchten, können Sie Folgendes eingeben:

```
# ifconfig eth0 192.168.0.7 netmask 255.255.255.0
```

Beachten Sie aber, dass eine solche Zuweisung nur für die Dauer der aktuellen Sitzung gilt. Damit sie dauerhaft gültig bleibt, müsste dieser Konfigurationsbefehl in einem Skript stehen, das beim Systemstart automatisch ausgeführt wird. Es ist viel einfacher, die grafische Methode zu verwenden – insbesondere für erweiterte TCP/IP-Optionen wie Nameserver, den Bezug der Konfigurationsdaten über DHCP oder den Netzzugang über Wählleitungen.

Ethernet-Schnittstellen mit YaST konfigurieren

Die Erstkonfiguration einer Netzwerkkarte mit YaST wurde bereits in Kapitel 2, »openSUSE installieren«, im Rahmen der Systeminstallation erläutert. Öffnen Sie den Punkt **NETZWERKEINSTELLUNGEN** in der YaST-Kategorie **NETZWERKGERÄTE**. Der Dialog besitzt vier Registerkarten.

Auf der Registerkarte **Globale Optionen** können Sie zunächst wählen, ob Sie **NetworkManager** verwenden möchten. Dies ist ideal für die Arbeit mit den Desktop-Umgebungen **KDE** oder **GNOME** – überhaupt ist der **NetworkManager**

für Desktop-Rechner optimiert; auf einem typischen Serverrechner sollten Sie ihn abschalten. Als Nächstes können Sie sich entscheiden, ob IPv6 aktiviert werden soll – dies ist standardmäßig der Fall, und es spricht auch nichts dagegen.

Darunter finden Sie den Abschnitt **OPTIONEN FÜR DHCP-CLIENT**. Wenn die Schnittstelle über DHCP konfiguriert wird, können Sie hier einige erweiterte Einstellungen vornehmen. Als **KENNUNG FÜR DHCP-CLIENT** wird standardmäßig die MAC-Adresse der vorliegenden Netzwerkkarte gesendet. Diese ist eindeutig, sodass dies normalerweise kein Problem darstellt. Die einzige Ausnahme besteht darin, wenn Sie über Xen, VMware oder ähnliche Software mehrere virtuelle Maschinen auf dem Rechner betreiben, die sich dieselbe physische Netzwerkkarte teilen – in diesem Fall müssen Sie hier selbst einen beliebigen, eindeutigen Wert eingeben.

Unter **ZU SENDENDER HOSTNAME** können Sie schließlich eine Kennung eingeben, die ein DHCP-Server verwenden kann, um Nameserver-Einträge dynamisch zu aktualisieren. Wenn dies bei Ihrem DHCP-Server nicht der Fall ist (Standard), lassen Sie einfach den Wert **AUTO** stehen.

Auf der Registerkarte **ÜBERSICHT** werden zunächst die verfügbaren Netzwerkkarten angezeigt. Wenn Sie eine von ihnen auswählen und auf **BEARBEITEN** klicken (oder **HINZUFÜGEN** wählen, um eine nicht automatisch erkannte Karte zu konfigurieren), gelangen Sie in den Dialog, der bereits in Kapitel 2 ausführlich beschrieben wurde. Die Registerkarte **ALLGEMEIN** wurde noch nicht behandelt; die Einstellungen bedeuten Folgendes:

► **GERÄT AKTIVIEREN**

Dieser Punkt bestimmt, wann die Schnittstelle aktiviert wird. Hier die möglichen Methoden:

- **BEIM SYSTEMSTART** führt selbst dann die automatische Inbetriebnahme durch, wenn keine Verbindung besteht.
- **BEI KABELANSCHLUSS** nimmt die Netzwerkkarte automatisch in Betrieb, sobald ein Netzkabel angeschlossen wird.
- **FALLS HOT-PLUGGED** ist ideal für USB-Netzwerk-Adapter oder ähnliche hot-plugging-fähige Geräte.
- **MANUELL** aktiviert die Netzwerkkarte nur auf Benutzerwunsch.
- **NIEMALS** deaktiviert die Schnittstelle ganz.
- **BEI NFSROOT** nimmt die Schnittstelle in Betrieb, falls ein Dateisystem von einem NFS-Netzwerkvolume gemountet werden soll.

▶ FIREWALL-ZONE

Hier wird die Konfiguration der openSUSE-Firewall für die jeweilige Schnittstelle vorgenommen (siehe Abschnitt 13.6, »Grundlagen der Netzwerksicherheit«, weiter unten). Sie haben die Auswahl zwischen folgenden Einstellungen:

- ▶ AUTOMATISCH ZUGEWIESENE ZONE – es wird die allgemeine Voreinstellung der Firewall verwendet.
 - ▶ INTERNE ZONE (NICHT GESCHÜTZT) – die Schnittstelle erhält keinen Schutz, da sie definitionsgemäß ausschließlich zu einem LAN ohne direkte Internetverbindung gehört.
 - ▶ ENTMILITARISIERTE ZONE – die Netzwerkkarte gehört zu einer speziellen Netzwerkzone zwischen zwei Firewalls, der entmilitarisierten Zone (englisch De-Militarized Zone oder kurz DMZ). Eine solche separate Zone wird oft zur sicheren Bereitstellung von Internetdiensten über Web-, Mail- oder Proxyserver verwendet.
 - ▶ EXTERNE ZONE – der Rechner besitzt eine eigene Internetverbindung, beispielsweise per DSL, und ist damit durch Angriffe gefährdet. Die Firewall wird auf maximalen Schutz voreingestellt.
- ▶ AKTIVIERE GERÄTE-KONTROLLE FÜR NICHT-ROOT NUTZER ÜBER KINTERNET
Erlaubt auch gewöhnlichen Benutzern die Steuerung der Schnittstelle; normalerweise darf dies nur root.
- ▶ MTU FESTLEGEN
Die MTU (Maximum Transfer Unit) ist die maximale Datenpaketgröße, die die Schnittstelle transportieren darf. Diesen Wert brauchen Sie nur einzutragen, wenn es Probleme geben sollte. Manchmal kann eine bestimmte Einstellung auch die Performance steigern; beachten Sie dazu die Dokumentation Ihrer Netzwerkkarte.

Auf der Registerkarte ADRESSE in der Konfiguration einer Netzwerkkarte wird die IP-Adresse konfiguriert, wie in Kapitel 2 beschrieben, entweder manuell oder per DHCP. Unter ZUSÄTZLICHE ADRESSEN können Sie derselben Netzwerkkarte zusätzliche IP-Adressen zuweisen. Auf diese Weise kann der Rechner beispielsweise ein Teil verschiedener logischer Netzwerke mit anderen IP-Adressbereichen werden. Wählen Sie HINZUFÜGEN, um eine neue Adresse einzutragen. Geben Sie unter ALIASNAME eine beliebige, aber eindeutige Bezeichnung ein – diese darf nicht den Schnittstellennamen wie etwa `eth0` enthalten, weil dieser automatisch zu Ihrer Eingabe hinzugefügt wird. Geben Sie danach eine freie IP-ADRESSE und die passende NETZMASKE ein.

Die dritte und letzte Registerkarte der Netzwerkkartenkonfiguration ist HARDWARE. Hier brauchen Sie nur einzugreifen, wenn Ihre Netzwerkkarte nicht auto-

matisch erkannt wurde oder trotz erfolgter Erkennung merkwürdige Probleme bereitet. Der GERÄTENAME ist zunächst die Bezeichnung des Geräts im `/dev`-Verzeichnisbaum. Wenn Sie auf **ÄNDERN** klicken, können Sie die Netzwerkkarte über MAC-Adresse, BusId, oder Gerätenamen ändern. Der MODULNAME bezeichnet das für die konkrete Karte zuständige Kernel-Modul, also den Treiber – gängig sind etwa `8139too` für Karten mit den Chipsätzen Realtek RTL 8129 und 8139, `ne2k-pci` für NE2000-kompatible PCI-Karten. Unter **OPTIONEN** tragen Sie gegebenenfalls Parameter zum Laden des Moduls ein, etwa `io=220 irq=7` für die I/O-Basisadresse `0x220` und den IRQ 7 – solche Einstellungen waren jedoch fast ausschließlich bei den heute ausgestorbenen ISA-Karten erforderlich.

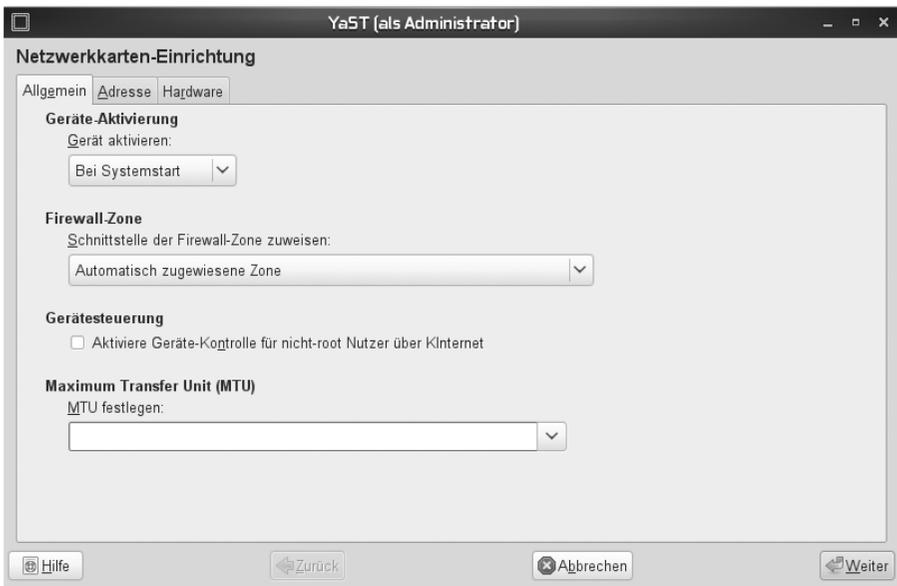


Abbildung 13.2 Die Registerkarte »Allgemein« der Netzwerkkarten-Konfiguration

Auf der Registerkarte **HOSTNAME/DNS** der **NETZWERKEINSTELLUNGEN** stehen folgende Optionen zur Verfügung:

► **HOSTNAME UND DOMÄNENNAME**

Geben Sie hier den **HOSTNAMEN** und den **DOMÄNENNAMEN** ein, die Sie dem Rechner (auf der aktuellen Netzwerkschnittstelle) zuweisen möchten. In einem lokalen Netz ohne permanente Internetverbindung sollte der Domainname keine offizielle Endung wie `.de` oder `.com` besitzen, sondern beispielsweise auf `.local` enden. Das durchgehende Beispiel dieses Buches sind der Hostname `tux` und der Domainname `test.local`.

Darunter können Sie ankreuzen, ob Hostnamen über DHCP geändert werden sollen – wenn DHCP zum Einsatz kommt, ist dies in der Regel sinnvoll. Außerdem können Sie sich entscheiden, ob der Hostname in die Datei */etc/hosts* geschrieben werden soll – dafür spricht einiges.

► NAMESERVER UND DOMÄNENSUCHLISTE

Hier können Sie einen oder mehrere Nameserver angeben, über die der Rechner die Auflösung von Domainnamen in IP-Adressen durchführen soll. Damit dies sinnvoll ist, müssen Sie natürlich die IP-Adressen der Nameserver und nicht etwa Ihre eigenen Domainnamen eingeben, weil Letztere ohne Nameserver eben keine Bedeutung haben. Wenn sich in Ihrem eigenen Netzwerk keine Nameserver befinden, können Sie diejenigen Ihres Providers angeben – dies ist allerdings sinnvoller für eine externe Netzwerkschnittstelle, etwa für DSL.

Unter DOMÄNENSUCHE können Sie Domainnamen angeben, die eventuell automatisch an interne Hostnamen angehängt werden sollen, um deren IP-Adressen zu ermitteln. Angenommen, in Ihrem lokalen Netzwerk mit dem Domainnamen *test.local* befindet sich ein zweiter Rechner namens *geeko*, und Sie betreiben einen Nameserver in Ihrem LAN. Dieser benötigt für die Namensauflösung vollständige Hostnamen (*Full Qualified Domain Names*, kurz FQDN), in diesem Fall also *geeko.test.local*. Damit innerhalb des lokalen Netzes auch die Kurznamen gültig sind, sollten Sie in dieses Feld den oder die Domainnamen Ihres Netzwerks eintragen, für das vorliegende Beispiel also *test.local*.

Falls die Netzwerkkarte ihre Konfiguration über DHCP erhält, sollten Sie die Option DNS-KONFIGURATION ÄNDERN auf den Wert STANDARDRICHTLINE VERWENDEN setzen – in diesem Fall kann auch die Nameserverkonfiguration per DHCP modifiziert werden. Da dies nichts schadet, wenn *kein* DHCP verwendet wird, ist diese Option standardmäßig ausgewählt.

Die letzte Registerkarte der Netzwerkkonfiguration ist ROUTING. Wenn Ihr Netzwerk über einen Router mit dem Internet oder anderen Netzwerken verbunden ist, müssen Sie den Router hier eintragen. Die Standardgateways (STANDARD-IPV4-GATEWAY und STANDARD-IPV6-GATEWAY) dienen dabei der Weiterleitung aller Daten, die weder für das eigene Teilnetz noch für ein Netzwerk mit explizitem eigenem Router bestimmt sind. Im Fall eines privaten Netzes oder eines kleinen Firmennetzwerks gibt es in der Regel nur einen (meist DSL-)Router, dessen IP-Adresse hier eingetragen wird. Falls DHCP eingesetzt wird, ist es nicht nötig, hier etwas einzutragen, da der DHCP-Server auch die Routing-Informationen automatisch liefert.

Für den selteneren Fall, dass zusätzlich Router für Verbindungen zu bestimmten anderen Teilnetzen vorhanden sind, können Sie die Optionen unter ROUTING-TABELLE nutzen. Klicken Sie hier auf HINZUFÜGEN, und geben Sie die passenden Werte für den neuen Router ein: ZIEL ist das konkrete Netzwerk, zu dem der Router eine Verbindung bereitstellt; geben Sie hier die Netzwerkadresse des betreffenden Netzes ein, etwa 192.168.1.0. Unter GATEWAY müssen Sie die IP-Adresse des Routers eintragen – wie weiter oben beschrieben, besitzt dieser natürlich mindestens zwei; hier geht es um diejenige innerhalb Ihres eigenen Teilnetzes, beispielsweise 192.168.0.2. Die NETZMASKE müssen Sie nach dem weiter oben beschriebenen Klassen- oder auch CIDR-Schema selbst eintragen; in kleinen Netzen mit den hier verwendeten privaten Klasse-C-Adressen ist dies stets 255.255.255.0. GERÄT (OPTIONAL) gibt die MAC-Adresse der Netzwerkkarte an. Das Feld OPTIONEN ist für Spezialeinträge der Routing-Tabelle reserviert, deren Betrachtung hier zu weit führen würde.

Wenn Ihr Rechner über mehrere Netzwerkschnittstellen verfügt, können Sie die Option IP-WEITERLEITUNG AKTIVIEREN ankreuzen, damit er selbst als Router fungiert. Auf diese Weise können Sie anderen Rechnern in Ihrem lokalen Netzwerk beispielsweise Ihren DSL-Internetzugang zur Verfügung stellen. Beachten Sie aber bitte die Vertragsbedingungen Ihres Providers; einige Privattarife gestatten unter Umständen den Einsatz von Routern nicht. In diesem Fall müssen Sie stattdessen einen Proxyserver aufsetzen (siehe nächstes Kapitel).

NetworkManager verwenden

Falls Sie in der Netzwerkkonfiguration den *NetworkManager* aktiviert haben, wird er verwendet, um die Schnittstelle zu konfigurieren. Diese Aufgabe erledigt er weitgehend automatisch – je nachdem, an welches kabelbasierte Netzwerk Sie den Rechner anschließen oder in den Einflussbereich welches drahtlosen Netzwerks Sie gelangen, wird die entsprechende Schnittstelle ohne Ihr Zutun konfiguriert. Dazu ist allerdings in der Regel die Verfügbarkeit von DHCP-Servern oder anderen Konfigurationshilfsmitteln erforderlich. Andernfalls müssen Sie YaST erneut aufrufen und auf die oben beschriebene manuelle Konfiguration umschalten.

Der einzige Hinweis auf die Anwesenheit des NetworkManagers ist ein kleines Tray-Icon (Symbol im rechten Bereich des Panels). Wenn Sie es mit der rechten Maustaste anklicken, können Sie verschiedene Informationen abrufen; eine WLAN-Verbindung lässt sich darüber gegebenenfalls auch sofort deaktivieren. Einige innerhalb des Menüs verfügbare Konfigurationsbefehle starten hingegen wiederum die oben beschriebenen YaST-Module.

13.2.2 Wireless LAN

Die zweite Schnittstellenart für lokale Netzwerke, die neben Ethernet eine größere praktische Bedeutung besitzt, ist die Drahtlostechnik WLAN. Nach einem kurzen theoretischen Überblick wird im Folgenden die Konfiguration beschrieben.

Theoretische Grundlagen

Wireless LAN (kurz WLAN) nach dem Standard IEEE 802.11 besteht aus mehreren Unterstandards, die sich in den Punkten Frequenzspektrum, Übertragungsrate und Funktechnologie unterscheiden. Sie alle werden jedoch über Funk betrieben; eine ursprünglich ebenfalls spezifizierte Infrarot-Variante hat sich nicht durchgesetzt. Infrarot wird größtenteils für den drahtlosen Anschluss von Peripheriegeräten wie Mäusen oder Tastaturen verwendet. Tabelle 13.5 zeigt eine Übersicht über die wichtigsten gebräuchlichen 802.11-Varianten.

Standard	Frequenzbereich	Übertragungsrate	Funktechnik
802.11	2,4 GHz	1 oder 2 MBit/s	FHSS/DSSS
802.11a	5 GHz	bis zu 54 MBit/s	OFDM
802.11b	2,4 GHz	5,5/11/22 MBit/s	HR/DSSS
802.11g	2,4 GHz	bis zu 54 MBit/s	OFDM
802.11n ³	2,4 und 5 GHz	bis zu 600 MBit/s	MIMO

Tabelle 13.5 Verschiedene Varianten von IEEE 802.11

Die Trägerfrequenz von 2,4 GHz wird vor allem deshalb am häufigsten verwendet, weil sie nicht lizenzpflichtig ist – es handelt sich nämlich um die Frequenz, mit der Mikrowellenherde arbeiten, da diese Wellenlänge Wassermoleküle am effektivsten erhitzt.

Die diversen Funkverfahren arbeiten alle mit verschiedenen Abarten der *Frequency-Hopping-Methode*, die auch im Mobilfunk eingesetzt wird: Nach einem bestimmten Schema werden die Funkwellen über mehrere Frequenzen übertragen, die mehrmals in der Sekunde wechseln. Dies ist erheblich weniger stör anfällig als die Verwendung einer einzelnen Frequenz. Die grundlegende Technik wurde Mitte der 30er-Jahre von der österreichischen Schauspielerin *Hedy Lamarr* erfunden. Ihr damaliger Ehemann war Rüstungsfabrikant, und diese Funktechnik sollte helfen, Torpedos der Alliierten fernzusteuern, ohne dass die

3 Dieser Standard wurde erst im September 2009 endgültig verabschiedet und ist in der Praxis noch kaum verbreitet.

Signale abgefangen und verfälscht werden könnten. Im Einzelnen werden folgende Verfahren unterschieden:

- ▶ *FHSS* (Frequency Hopping Spread Spectrum)
Die Frequenzen wechseln nach einem zufälligen Muster.
- ▶ *DSSS* (Direct Sequence Spread Spectrum)
Es werden erheblich mehr Einzelfrequenzen verwendet; die Verteilung wird nach einem komplexen mathematischen Verfahren durchgeführt.
- ▶ *HR/DSSS* (High-Rate Direct Sequence Spread Spectrum)
DSSS mit speziellen Erweiterungen, die eine höhere Übertragungsrate ermöglichen
- ▶ *OFDM* (Orthogonal Frequency Division Multiplexing)
Jeder Kanal wird in mehrere Teilkanäle unterteilt, die Signale werden über alle Teilkanäle parallel übertragen. Aus diesem Grund ist OFDM das Übertragungsverfahren mit der höchsten Datenrate, andererseits aber auch das aufwendigste, sodass die entsprechende Hardware vergleichsweise teuer ist.
- ▶ *MIMO* (Multiple Input Multiple Output) bündelt mehrere OFDM-Übertragungen, um eine höhere Gesamtübertragungsgeschwindigkeit zu erreichen.

Der größte Teil der Wireless-LAN-Hardware, der momentan verkauft wird, basiert auf dem Standard 802.11g; manchmal ist auch noch das ältere und langsamere 802.11b im Einsatz. Die Preise für Hardware dieser Variante sind in den letzten Jahren stark gefallen. Außerdem sind immer mehr Notebooks und Desktop-PCs ab Werk mit einer WLAN-Schnittstelle ausgestattet. Vorreiter dürften das PowerBook und das iBook (die noch PowerPC-basierten Vorläufer der MacBooks) von Apple gewesen sein – Apple fördert diese Technologie unter dem Namen AirPort seit Jahren.

Als Netzzugangsverfahren in 802.11-Netzen kommt *CSMA/CA* zum Einsatz (Carrier Sense Multiple Access with Collision Avoidance) – wie der Name vermuten lässt, werden Datenkollisionen von vornherein vermieden. Anders als bei *CSMA/CD* sendet eine Station, die ein freies Übertragungsmedium (in diesem Fall den entsprechenden Funkkanal) vorfindet, nicht einfach ihre Daten, sondern eine Sendeaufforderung (RTS). Daraufhin warten andere sendebereite Stationen, und die erste Station, die das RTS gesendet hat, schickt ihre Daten, nachdem ihr die Empfängerstation ihre Empfangsbereitschaft (CTS) signalisiert hat. Abgeschlossen wird die Datenübertragung durch ein ACK-Signal; daraufhin kann die nächste Station ihren Sendewunsch bekanntgeben.

Das einfachste denkbare 802.11-WLAN besteht nur aus mehreren Rechnern mit entsprechender Schnittstelle, die auf direktem Weg miteinander kommunizie-

ren. Ein solcher Aufbau wird als *Basic Service Set* (BSS) bezeichnet. Die Entfernung zwischen zwei beliebigen Stationen darf die maximale Reichweite des Funksignals nicht überschreiten, da jede Station die Signale nur senden und empfangen, aber nicht verstärken und weiterleiten kann. Da ein solches Netzwerk nicht mit anderen Netzen kommunizieren kann, wird es als *unabhängiges BSS* (Independent BSS oder kurz IBSS) bezeichnet. Derartige Netzwerke sind sinnvoll für die sogenannte Ad-hoc-Vernetzung temporärer Zusammenkünfte wie Messen oder LAN-Partys.

Ein wenig komplexer wird der Aufbau eines BSS, wenn ein Access Point hinzugefügt wird. Im Grunde funktioniert ein Access Point wie ein Ethernet-Hub, denn sobald er vorhanden ist, kommunizieren die Stationen nicht mehr direkt miteinander, sondern senden die Frames an den Access Point, der sie an den gewünschten Empfänger weitergibt. Die Identifikation der einzelnen Stationen erfolgt wie bei Ethernet anhand einer 48 Bit langen MAC-Adresse. Ein BSS mit einem Access Point wird als *Infrastruktur-BSS* bezeichnet (und natürlich nicht IBSS abgekürzt). Für die Reichweite des Netzes ist nur noch die Entfernung zwischen einer Station und dem Access Point ausschlaggebend.

Die wichtigste Aufgabe eines Access Points besteht in seiner Funktion als Bridge. Er verbindet das WLAN mit einem Backbone-Netzwerk – meistens Twisted-Pair-Ethernet. Auf diese Weise kann das WLAN mit stationären Teilen des Netzes verbunden werden oder Zugang zu Servern und Routern erhalten, ohne dass diese selbst mit WLAN-Schnittstellen ausgestattet werden müssten.

Im Übrigen bildet ein Verbund aus miteinander vernetzten Access Points (entweder ebenfalls über Funk oder über Ethernet) ein sogenanntes *Extended Service Set* (ESS). Eine Station kann sich innerhalb eines ESS frei bewegen, weil die Access Points einander darüber auf dem Laufenden halten, welche Stationen sich gerade in ihrem Bereich befinden. Eine Station kann immer nur genau mit einem Access Point verbunden sein; sobald das Signal eines anderen Access Points stärker wird als das des bisherigen, meldet die Station sich bei ihrem alten Access Point ab und bei dem neuen an. Auf diese Weise werden Frames immer über den jeweils aktuellen Access Point an eine Station gesendet.

Ein zusätzlicher Nutzen von Access Points besteht darin, dass sie in der Lage sind, Frames zu puffern, die an bestimmte Stationen adressiert sind. Gerade Notebooks schalten im Standby-Modus oft auch die WLAN-Schnittstelle ab, um Strom zu sparen; sobald die Verbindung wieder aufgebaut wird, werden die zwischengespeicherten Frames ausgeliefert.

Das ESS-Modell wird immer häufiger für öffentlich verfügbare Netzwerkzugänge eingesetzt; in Bahnhöfen, Flughäfen oder Gaststätten stehen zunehmend öffent-

liche WLAN-Access-Points zur Verfügung, in die sich Notebook-Benutzer ohne Weiteres einbuchten können. Mittlerweile werden sogar die ersten Innenstädte fast flächendeckend mit einander überlappenden Access Points ausgestattet. Irgendwann könnte ein ähnlich dichtes Netz entstehen, wie es die Mobilfunk-Zellen inzwischen bilden.

Eine der größten Herausforderungen beim Einsatz von Wireless-Technologien bleibt die Sicherheit. Es ist zwar auch nicht weiter schwierig, das Signal von Ethernet-Kabeln abzuhören, aber immerhin ist es vergleichsweise einfach, den physikalischen Zugang zu ihnen zu kontrollieren.

Beim WLAN kann dagegen im Grunde genommen jeder die Signale mit einer kompatiblen Antenne auffangen und analysieren, um unberechtigt Informationen zu erhalten oder gar zu manipulieren. Das gilt umso mehr, als man die Grenzen der Funkreichweite niemals ganz genau auf die Größe des zu vernetzenden Gebäudes oder Geländes abstimmen kann; es ist also durchaus möglich, die Funkwellen außen zu empfangen. Daher ist es wichtig, moderne WLAN-Access-Points oder -Router mit dem Verschlüsselungsstandard WPA2 zu verwenden.

WLAN-Konfiguration

WLAN-Schnittstellen werden in YaST als Netzwerkkarten geführt. Dies dient dazu, dass die Konfiguration im Wesentlichen genauso funktioniert wie bei den oben beschriebenen Ethernet-Karten. Es betrifft die TCP/IP-Einstellungen wie die IP-Adresse (falls Ihr WLAN-Access-Point oder -Router kein DHCP verwendet).

Als besondere Option existiert zusätzlich der Dialog KONFIGURATION DER DRAHTLOSEN NETZWERKKARTE. Dieser besitzt folgende Einstellungen:

► **BETRIEBSMODUS**

Hier können Sie gemäß der obigen theoretischen Beschreibung die Art Ihres WLANs auswählen: AD-HOC (PEER TO PEER) ist für IBSS-Netze gedacht, während VERWALTET versucht, die Verbindung mit einem Access Point herzustellen. MASTER schließlich macht Ihren Rechner selbst zum WLAN-Access-Point.

► **NETZWERKNAME (ESSID)**

Falls Sie einen Access Point verwenden, müssen Sie hier die ESSID eintragen, die Sie dort festgelegt haben. Andernfalls können Sie dieses Feld frei lassen.

► **AUTHENTIFIZIERUNGSMODUS**

Hier können Sie einstellen, auf welche Weise die Authentifizierung im drahtlosen Netzwerk erfolgen soll. Auch dieser Wert wird durch Ihren WLAN-Router oder -Access Point bestimmt. Wie oben erwähnt, sind Access Points ohne Verschlüsselung oder mit dem veralteten WEP (Wired Equivalent Privacy) aus

Sicherheitsgründen nicht zu empfehlen; idealerweise verwenden Sie ausschließlich WPA2-fähige Geräte.

► **SCHLÜSSELART**

Aus den oben diskutierten Gründen sollten Sie Wireless LAN ausschließlich mit Verschlüsselung verwenden. Falls Sie den Schlüssel bereits bei der Konfiguration des Routers festgelegt haben, wählen Sie ASCII beziehungsweise HEXADEZIMAL (je nachdem, wie der Router den Schlüssel anzeigt) und geben Sie ihn unter **VERSCHLÜSSELUNGS-KEY** exakt ein. Falls Sie noch keinen Schlüssel konfiguriert haben, können Sie auch **PASSPHRASE** wählen und sich ein beliebiges, möglichst langes Passwort (daher »-phrase«!) neu verschlüsseln lassen.

Speziell für den WLAN-Betrieb existieren zusätzlich zwei Kommandozeilen-Tools: `iwconfig` funktioniert genau wie `ifconfig`, zeigt aber zusätzliche Wireless-Optionen wie die **ESSID** und den Schlüssel an. `iwlist` liefert weitere Detailinformationen über die Konfiguration der WLAN-Schnittstelle sowie eine Liste der erreichbaren Drahtlosnetzwerke.

Inkompatible Netzwerkhardware mit Windows-Treibern betreiben

Für manche besonders günstige Netzwerkhardware, beispielsweise USB-WLAN-Adapter, gibt es partout keine Linux-Treiber. In diesen Fällen kann das Programm `ndiswrapper` helfen, denn es bindet Windows-Netzwerktreiber in Ihr Linux-System ein. Zunächst müssen Sie `ndiswrapper` selbst installieren, da es nicht zur openSUSE-Standardinstallation gehört. Suchen Sie das gleichnamige Paket in YaST oder geben Sie als root folgenden Konsolenbefehl ein:

```
# zypper install ndiswrapper
```

Danach wird der Datenträger mit dem Windows-Treiber benötigt. Nachdem er gemountet wurde, geben Sie Folgendes ein, um den Treiber mithilfe seiner `inf`-Datei zu installieren:

```
# ndiswrapper -i <Dateiname>.inf
```

Es besteht keine Garantie dafür, dass das betreffende Gerät auf diese Weise funktioniert, aber der Versuch kann nicht schaden.

13.3 Wählverbindungen ins Internet

Die wichtigste Technologie für den Internetzugang sind heute Hochfrequenz-Verbindungen über verschiedene *DSL*-Dienste. Daneben spielen *Modems* für den Zugang über analoge Telefonleitungen sowie die digitale Telefonleitung *ISDN* noch eine gewisse Rolle. Diese verschiedenen Zugangsverfahren werden im Folgenden dargestellt.

13.3.1 PPP

Eine Gemeinsamkeit aller DFÜ-Netzwerkverbindungen besteht in der Notwendigkeit, die Datenübertragung über diese Leitungen zu standardisieren und bestimmte Grundlagen für die Protokolle der Vermittlungsschicht zu schaffen. Dafür werden spezielle Protokolle verwendet, die den Netzzugang über relativ langsame serielle Leitungen ermöglichen. Das traditionelle Protokoll für die Vernetzung über Wählleitungen war *SLIP* (Serial Line Interface Protocol); allerdings besitzt es eine Reihe organisatorischer und technischer Mängel und wurde deshalb weitgehend durch das Point-to-Point Protocol (*PPP*) ersetzt.

PPP kümmert sich um die Authentifizierung des Benutzers nach der Einwahl, indem Benutzername und Passwort übermittelt werden; anschließend verhandeln die beiden direkt miteinander verbundenen Punkte die eigentlichen Netzwerkdetails. Eine der wesentlichsten Fähigkeiten des Protokolls für Internetverbindungen besteht darin, dass der Einwahlknoten dem anwählenden Rechner automatisch eine IP-Adresse zuweisen kann, über die diese Netzwerkschnittstelle im gesamten Internet identifiziert wird.

Im Einzelnen erfolgen bei PPP also die folgenden Schritte:

1. Wird eine Wählleitung (analog oder ISDN) verwendet, so stellt der Rechner des Benutzers über die entsprechende Schnittstelle eine Telefonverbindung her. Falls die Leitung besetzt sein sollte, werden spezielle frei konfigurierbare Maßnahmen getroffen; in der Regel erfolgt nach einer gewissen Wartezeit ein erneuter Wählversuch. Bei DSL-Leitungen wird ebenfalls die Verbindung aktiviert, auch wenn man dies nicht als Wählen im klassischen Sinne bezeichnen kann.
2. Der Einwahlknoten verlangt eine Authentifizierung, in der Regel in Form von Benutzername und Passwort. Dazu wird intern entweder das veraltete *PAP* (Password Authentication Protocol) oder das modernere Protokoll *CHAP* (Challenge Handshake Authentication Protocol) verwendet. Die PPP-Module in SUSE Linux übermitteln diese Daten auf Wunsch nach einmaliger Konfiguration ohne Ihr Zutun automatisch.
3. Nachdem die Daten überprüft wurden, erfolgt entweder die Ablehnung des Benutzers und der Verbindungsabbau, oder die Netzwerkparameter werden ausgehandelt. Auch wenn PPP als Netzzugangsgrundlage für alle möglichen Protokolle der Vermittlungsschicht dienen kann, wird heute fast nur noch TCP/IP aufgesetzt. Zu diesem Zweck weist der PPP-Knotenpunkt des Internetproviders der seriellen Verbindung auf der Einwahlseite eine IP-Adresse zu.

13.3.2 Internetzugang über DSL

DSL ist die Abkürzung für Digital Subscriber Line (etwa »digitale Abonnement-Leitung«); der Name soll verdeutlichen, dass es sich de facto um eine Standleitung anstelle einer Wählleitung handelt. Zur Einführung von DSL kam es, da es durch die allmähliche Verbesserung der Qualität von Telefonleitungen möglich wurde, Signale hoher Frequenz zu übertragen. Die meisten DSL-Dienste verwenden also genau wie Modem- und ISDN-Verbindungen die klassischen Kupferleitungen der Telefongesellschaften.

Es existieren zwei grundsätzliche Varianten von DSL: Bei Symmetric DSL (*SDSL*) sind die Übertragungsraten für ankommende und ausgehende Daten identisch, bei Asymmetric DSL (*ADSL*) ist die ankommende Übertragungsrate höher als die ausgehende. *SDSL* ist eher für mittlere bis große Unternehmen geeignet, die nicht nur permanent auf das Internet zugreifen, um im Web zu recherchieren oder ihre E-Mails zu lesen, sondern bei denen auch eine Menge Datenausgänge stattfinden. *ADSL* dagegen wird häufiger von Privatkunden oder kleineren Firmen eingesetzt, die recht hohe Datenmengen aus dem Internet herunterladen, aber nur verhältnismäßig wenige und eher kleine Uploads durchführen.

ADSL wird inzwischen von so gut wie jedem Internetprovider angeboten; die Leitungen selbst werden dabei meist von der Telekom bereitgestellt. Bei manchen günstigen Privatkunden-DSL-Angeboten ist der Zugang nur einem einzelnen Rechner gestattet; der Betrieb eines DSL-Routers, der ein ganzes LAN mit dem Internet verbindet, ist unter Umständen nicht erlaubt. Hier können Sie sich durch die Einrichtung eines Proxys behelfen. Die *ADSL*-Leitungen sind mit verschiedenen Übertragungsraten verfügbar; natürlich sind schnellere etwas teurer. Gängige Werte sind 4 oder 6 MBit/s im Download; neuere Technik wie *ADSL2+* erreicht auch 16 oder gar 25 MBit/s. Glasfaser-Kunden können sich sogar über Downloadraten bis zu 100 MBit/s freuen.

SDSL-Lösungen werden von sehr vielen kommerziellen Providern angeboten und stellen je nach Bedarf viele verschiedene Übertragungsraten zur Verfügung. Sie sind deutlich teurer als *ADSL*-Angebote mit der gleichen oder gar mit einer höheren Übertragungsrate, weil die entsprechende Technik aufwendiger ist.

Die Gebühren für DSL-Anschlüsse werden in der Regel nicht nach der Nutzungsdauer berechnet; sogenannte Flatrates sind üblich. Einige Provider verwenden allerdings eine Volumenbeschränkung, das heißt, ohne Aufpreis darf monatlich nur eine bestimmte Datenmenge transferiert werden.

Neben den DSL-Angeboten, die über normale Telefonleitungen laufen, werden seit einiger Zeit auch spezielle Lösungen angeboten. Eine davon ist die Internetverbindung über das Glasfaserkabel des Kabelfernsehens. Da dieses Kabel für das

Passivmedium Fernsehen erfunden wurde, besitzt es in seiner ursprünglichen Version keine Rückkanal-Fähigkeit – es können Daten empfangen, aber nicht gesendet werden; noch nicht einmal die Anforderung einer URL kann abgesetzt werden. Erst allmählich wird der Rückkanal derjenigen Kabelnetze nachgerüstet, die die Deutsche Telekom bereits verkauft hat.

Eine andere Lösung ist besonders interessant für kleine Gemeinden, die so weit von der nächsten größeren Stadt entfernt liegen, dass sich eine Nachrüstung der bestehenden Telefonleitungen oder Fernsehkabelnetze nicht lohnt: die Kommunikation mit einem Satelliten über eine Parabolantenne.

ADSL anschließen

An Ihre Telefonsteckdose, den sogenannten TAE-Anschluss, müssen Sie einen sogenannten *Splitter* anschließen – eine Frequenzweiche, die die hochfrequenten DSL-Signale und die niedrigfrequenten normalen Telefonsignale voneinander trennt. Den Ausgang für die Telefonsignale bietet wiederum ein TAE-Anschluss, an den entweder ein Analog-Telefon oder ein NTBA angeschlossen wird, je nachdem, ob ADSL mit einem Analog- oder mit einem ISDN-Telefonanschluss kombiniert wird.

Den Ausgang für die speziellen ADSL-Signale bietet eine Twisted-Pair-Buchse vom Typ RJ-11. An diesen Anschluss wird in der Regel ein ADSL-Modem angeschlossen, das dann über USB oder Twisted-Pair-Ethernet mit dem Computer verbunden wird. Natürlich ist die Bezeichnung »ADSL-Modem« technisch gesehen Unfug; bei DSL findet keinerlei Analog-Digital-Umwandlung statt. Dennoch ist der Begriff »Modem« für das Gerät weitverbreitet, weil es den Computer mit einer seriellen Fernleitung verbindet. Beim Anschluss über eine Ethernet-Schnittstelle kommt eine spezielle PPP-Variante namens *PPPoE* (PPP over Ethernet) zum Einsatz.

DSL mit YaST konfigurieren

Nachdem Sie Ihre DSL-Geräte verkabelt haben, können Sie die zugehörige Softwarekonfiguration durchführen. Auch dafür gibt es ein komfortables YaST-Modul – wählen Sie DSL in der Abteilung NETZWERKGERÄTE. Möglicherweise wird Ihr DSL-Anschluss automatisch erkannt, in diesem Fall können Sie auf BEARBEITEN klicken, um die Einstellungen zu überprüfen. Wählen Sie andernfalls HINZUFÜGEN, um Ihre Verbindung manuell einzurichten. In beiden Fällen landen Sie im gleichen Dialog. Hier sind folgende Eingaben möglich:

► PPP-MODUS

Geben Sie an, wie Ihr Rechner an das DSL-Modem angeschlossen ist. Wenn Sie einen Ethernet-Anschluss verwendet haben, stellen Sie PPP ÜBER ETHERNET

ein; für DSL-Modems mit USB-Anschluss ist die korrekte Einstellung dagegen meist PPP ÜBER ATM (konsultieren Sie gegebenenfalls die Dokumentation des Geräts). CAPI FÜR ADSL verwendet einen ISDN-ähnlichen Treiber und kommt seltener zum Einsatz. TUNNEL-PROTOKOLL FÜR POINT-TO-POINT schließlich ist die Einstellung der Wahl für aktive DSL-Modems, die zugleich als Router fungieren.

▶ VPI/VCI

Hier können Sie nur für den Modus PPP über ATM etwas eingeben. Falls dies erforderlich ist, steht auch dieser konkrete Wert im Handbuch des DSL-Modems.

▶ ETHERNETKARTE

Wählen Sie hier eventuell diejenige Netzwerkkarte aus, an der DSL angeschlossen ist. Falls Sie die Karte selbst noch nicht konfiguriert haben, klicken Sie auf NETZWERKKARTEN KONFIGURIEREN, und nehmen Sie die weiter oben beschriebenen Arbeitsschritte vor.

▶ SERVER-NAME ODER IP-ADRESSE

Geben Sie an dieser Stelle die IP-Adresse oder den Hostnamen Ihres aktiven DSL-Modems ein, falls Sie weiter oben den Modus TUNNEL-PROTOKOLL gewählt haben.

▶ GERÄT AKTIVIEREN

In diesem Menü können Sie steuern, wann die DSL-Verbindung hergestellt werden soll. Hier die möglichen Werte:

- ▶ BEIM SYSTEMSTART stellt die Verbindung automatisch nach dem Booten her.
- ▶ FALLS HOT-PLUGGED aktiviert die Verbindung beim Anschließen; dies ist besonders nützlich für PCMCIA- oder USB-basierte DSL-Adapter von Notebooks.
- ▶ MANUELL überlässt es dem Benutzer, die DSL-Verbindung herzustellen. Dafür gibt es diverse Kommandozeilentools oder das KDE-Programm KPPP. Empfehlenswert ist diese Einstellung, wenn Sie keine Flatrate haben.
- ▶ NIEMALS deaktiviert den Anschluss.
- ▶ AKTIVIERE GERÄTE-KONTROLLE FÜR NICHT-ROOT NUTZER ÜBER KINTERNET Kreuzen Sie diese Option an, wenn nichtprivilegierte User den DSL-Anschluss steuern dürfen. Dies ist besonders wichtig, wenn Sie unter GERÄTE-AKTIVIERUNG den Wert MANUELL einstellen.

Wenn Sie diesen Dialog fertig ausgefüllt haben, klicken Sie wie gewohnt auf WEITER. Daraufhin gelangen Sie auf die Seite INTERNET SERVICE PROVIDER (ISP) WÄHLEN. Die Auswahl LÄNDER enthält einige vorgefertigte Einträge bekannter Provi-

der. Falls Ihr Provider in der Liste steht, können Sie ihn hier auswählen und WEITER anklicken, andernfalls müssen Sie auf NEU klicken, um Ihren Provider manuell einzutragen.

In beiden Fällen landen Sie auf der Seite PROVIDER-PARAMETER. Geben Sie unter BENUTZERNAME den Wert aus Ihren DSL-Vertragsunterlagen ein – bei vorgefertigten Providerprofilen können Sie meist den Vorgabewert akzeptieren. Wenn mehrere Personen Zugriff auf den Computer haben, sollten Sie aus Sicherheitsgründen IMMER PASSWORT ABFRAGEN ankreuzen; dadurch werden Sie bei jedem Verbindungsaufbau aufs Neue nach dem Passwort gefragt. Bei einem reinen Privatrechner können Sie das Feld dagegen deaktivieren und Ihr Passwort ein für allemal eintippen.

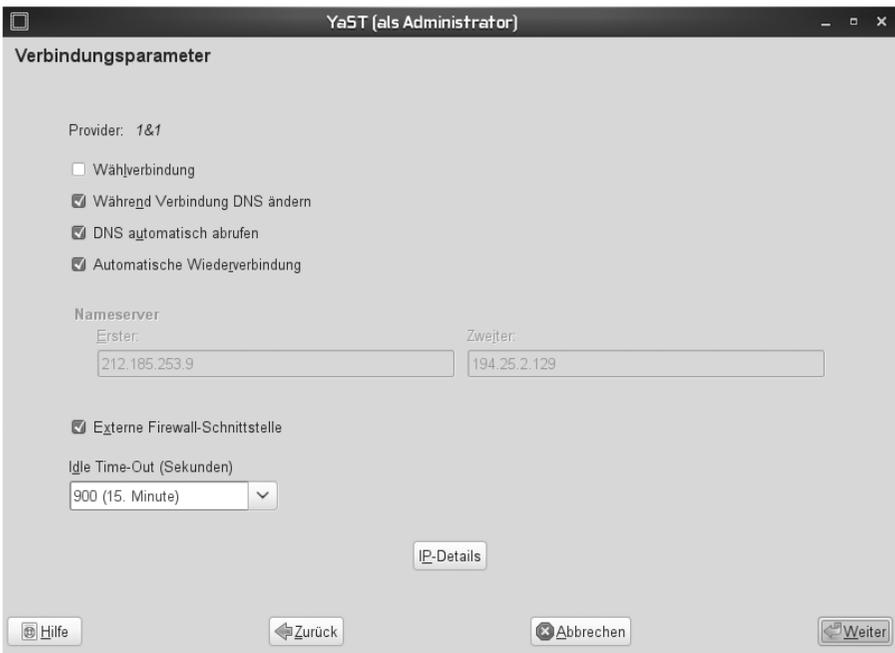


Abbildung 13.3 Einstellen der DSL-Verbindungsparameter

Auf dem nächsten Bildschirm werden die VERBINDUNGSPARAMETER eingestellt (siehe Abbildung 13.3):

- ▶ **WÄHLVERBINDUNG**
Stellt automatisch eine Verbindung her, sobald ein Programm Daten aus dem Internet anfordert. In diesem Fall müssen Sie mindestens einen Nameserver Ihres Providers (IP-Adresse) eintragen.

- ▶ **WÄHREND VERBINDUNG DNS ÄNDERN**
Erlaubt dem Provider, nach dem Verbindungsaufbau die Daten alternativer oder zusätzlicher Nameserver einzustellen. Dies ist in der Regel empfehlenswert.
- ▶ **DNS AUTOMATISCH ABRUFEN**
Stellt die Nameserver beim Verbindungsaufbau dynamisch ein. Solange Sie nicht den Punkt **WÄHLVERBINDUNG** aktivieren, ist diese Einstellung meist unproblematisch; falls es Probleme gibt, deaktivieren Sie sie und geben Sie im nächsten Feld mindestens einen Nameserver manuell ein.
- ▶ **NAMESERVER**
Wie bereits erwähnt, können Sie in diese Felder die IP-Adressen von einem oder zwei Nameservern Ihres Providers eintragen, wenn Sie **WÄHLVERBINDUNG** auswählen und/oder **DNS AUTOMATISCH ABRUFEN** deaktivieren.
- ▶ **EXTERNE FIREWALL-SCHNITTSTELLE**
Diese Option sollte im Regelfall angekreuzt werden. Sie bedeutet, dass diese Schnittstelle über die openSUSE-Firewall besonders geschützt wird. Näheres erfahren Sie in Abschnitt 13.6, »Grundlagen der Netzwerksicherheit«.
- ▶ **IDLE TIME-OUT (SEKUNDEN)**
Damit legen Sie fest, nach wie vielen Minuten Inaktivität die Verbindung unterbrochen werden soll. Bei einer Flatrate ist die Voreinstellung 300 Sekunden (fünf Minuten) vernünftig; wenn Sie eine minutenbasierte Abrechnung verwenden, sollten Sie dagegen einen viel kürzeren Wert wählen.
- ▶ **IP-DETAILS**
Diesen Parameter brauchen Sie nur in Ausnahmefällen einzustellen, etwa bei einer Einwahlverbindung von zu Hause in Ihr Firmennetzwerk. Bei normalen Providerverbindungen sind die voreingestellten Werte **DYNAMISCHE IP-ADRESSE** und **STANDARDROUTE** (alle Ziele, die nicht im lokalen Teilnetz liegen, über diese Verbindung suchen) nämlich in Ordnung.

Nachdem Sie das letzte Mal auf **WEITER** geklickt haben, erscheint die Frage, ob Sie sofort den E-Mail-Zugang konfigurieren möchten. Näheres darüber lesen Sie in Kapitel 16, »Weitere Internet-Serverdienste«.

13.3.3 Internetzugang per Modem (analoge Telefonleitung)

Einige User – vor allem Bewohner ländlicher Gebiete, in denen kein DSL verfügbar ist – verwenden nach wie vor analoge Modems für den Internetzugang. Selbstverständlich werden auch diese von openSUSE unterstützt; in diesem Unterabschnitt erfahren Sie nach einem theoretischen Kurzüberblick, wie Sie Ihr Modem einrichten können.

Theoretische Grundlagen

Für Modems wurden im Laufe der Zeit viele verschiedene Standards entwickelt, die sich insbesondere bezüglich ihrer maximalen Datenübertragungsrate voneinander unterscheiden. Der aktuelle Standard heißt V.90 und überträgt bis zu 56.600 Bit/s – gemessen an den üblichen Geschwindigkeiten von DSL oder gar Ethernet ist das natürlich sehr langsam, aber kein Vergleich zu den Modem-Geschwindigkeiten vergangener Jahrzehnte. Rein physikalisch scheint mit 56,6 kBit/s die Leistungsgrenze erreicht zu sein; etwas höhere Übertragungsraten lassen sich durch die heutzutage häufig verwendete Datenkomprimierung erzielen – je nach übertragener Datenart etwa bis zur doppelten Leistung.

Das Wort *Modem* ist eine Zusammensetzung aus den Abkürzungen für Modulator und Demodulator, weil es die digitalen Signale des Computers in frequenzmodulierte Analogimpulse umwandelt, diese über die Telefonleitung überträgt und am Ziel wieder zurückverwandelt. Zu diesem Zweck muss es auf der einen Seite mit dem Computer verbunden werden, zum Beispiel über USB, klassisch auch über die alte serielle Schnittstelle oder als PCI-Steckkarte. Auf der anderen Seite wird das Modem über einen TAE-Stecker an die Telefonbuchse angeschlossen. Praktisch sind in diesem Zusammenhang Dreifach-TAE-Dosen, die leicht nachgerüstet werden können: Sie verfügen über einen speziellen Anschluss (TAE-F) für ein Telefon in der Mitte und zwei Anschlüsse (TAE-N) für Zusatzgeräte – Faxgerät, Anrufbeantworter oder eben Modem – außen.

Das Modem wird vom Computer über ein einfaches ASCII-basiertes Protokoll gesteuert; heutzutage verwenden praktisch alle Modems den sogenannten *Hayes-Befehlssatz* (benannt nach einem längst vergessenen Modem-Hersteller). Da die Befehle dieses Protokolls alle mit der Zeichenfolge »AT« beginnen, wird er mitunter auch als AT-Befehlssatz bezeichnet. Wichtige Befehle sind etwa folgende:

- ▶ ATDT <Rufnummer>
DT steht für »Dial Tone« – eine Rufnummer wird im Tonwahlverfahren (Mehrfrequenzverfahren) angewählt.
- ▶ ATDP <Rufnummer>
»Dial Pulse« – eine Rufnummer wird im Pulswahlverfahren angewählt (heute sehr selten).
- ▶ ATH
»Hangup« – die Telefonverbindung wird unterbrochen, es wird »aufgelegt«.
- ▶ ATZ
Das Modem wird auf den Einschaltzustand zurückgesetzt (Reset).

Das Tonwahlverfahren verwendet mehrere Töne unterschiedlicher Frequenzen, die zusammen die verschiedenen Ziffern und Funktionen des Telefons repräsentieren. Das Pulswahlverfahren sendet dagegen eine Reihe von »Klicktönen« – einen für eine 1, zwei für eine 2 und so weiter, bis zehn für eine 0. Seitdem alle Vermittlungsstellen in den deutschen Telefonnetzen digital sind, benötigt niemand mehr das langsamere und unzuverlässigere Pulswahlverfahren. Verwechseln Sie übrigens digitale Vermittlung nicht mit digitaler Signalübertragung – Letztere findet beispielsweise bei ISDN statt, das im nächsten Unterabschnitt behandelt wird.

Bevor eine Datenkommunikation überhaupt denkbar ist, müssen sich beide Seiten darüber einig sein, auf welche Weise sie die aufeinanderfolgenden einzelnen Bits überhaupt als Datenbytes interpretieren sollen, was Datenbits, Stoppbits und eventuelle Parity-Bits angeht. Die meisten Internet-Einwahlpunkte verwenden heutzutage den Standard 8N1 (8 Datenbits, kein Parity, 1 Stoppbit). Dies muss in den Modem-Konfigurationsdaten eingetragen werden.

Nach der Herstellung der eigentlichen Telefonverbindung findet der sogenannte *Handshake* (Handschlag) zwischen den beiden Gegenstellen statt – es wird eine Übertragungskapazität ausgehandelt, die beide Seiten verwenden können. Erst nachdem die grundlegende Datenübertragung funktioniert, wird PPP eingesetzt, um die eigentliche Netzwerkverbindung über die Telefonverbindung herzustellen.

Den Modem-Zugang mit YaST einrichten

Wählen Sie **MODEMS** in der YaST-Kategorie **NETZWERKGERÄTE**, um Ihr Modem zu konfigurieren. Klicken Sie auf **BEARBEITEN**, wenn ein Modem automatisch erkannt wurde, oder auf **HINZUFÜGEN**, um es andernfalls manuell einzurichten. Daraufhin wird der Bildschirm **MODEMPARAMETER** angezeigt. In diesem Dialog, der in Abbildung 13.4 gezeigt wird, sind folgende Felder auszufüllen:

- ▶ **MODEMGERÄT**
Wählen Sie den Hardwaretyp und -anschluss Ihres Modems aus. `/dev/modem` beschreibt PCI-Modem-Karten. `/dev/ttyACM0` bis `/dev/ttyACM3` steht für die USB-Anschlüsse, während `/dev/ttyS0` bis `/dev/ttyS3` die traditionellen seriellen Anschlüsse (COM-Ports) bezeichnet.
- ▶ **AMTSHOLUNG**
Falls Ihr Modem an eine Telefonanlage angeschlossen ist, müssen Sie möglicherweise eine Ziffer vorwählen, um ein Freizeichen zu erhalten – meistens die 0. Tragen Sie hier gegebenenfalls die betreffende Ziffer ein.

- ▶ **WÄHLMODUS**
Wie oben beschrieben, gibt es prinzipiell TONWAHL oder IMPULSWAHL; Letzteres ist zumindest in Deutschland nicht mehr nötig.
- ▶ **LAUTSPRECHER AN**
Aktiviert den in vielen externen Modems eingebauten Lautsprecher, sodass Sie die typischen Datenübertragungsgeräusche hören können.
- ▶ **WAHLTON ABWARTEN**
Führt den Einwahlversuch erst durch, wenn das Freizeichen ertönt. Im Zusammenhang mit einem Präfix zur Amtsholung ist dies nicht empfehlenswert, in anderen Fällen kann es sinnvoll sein.

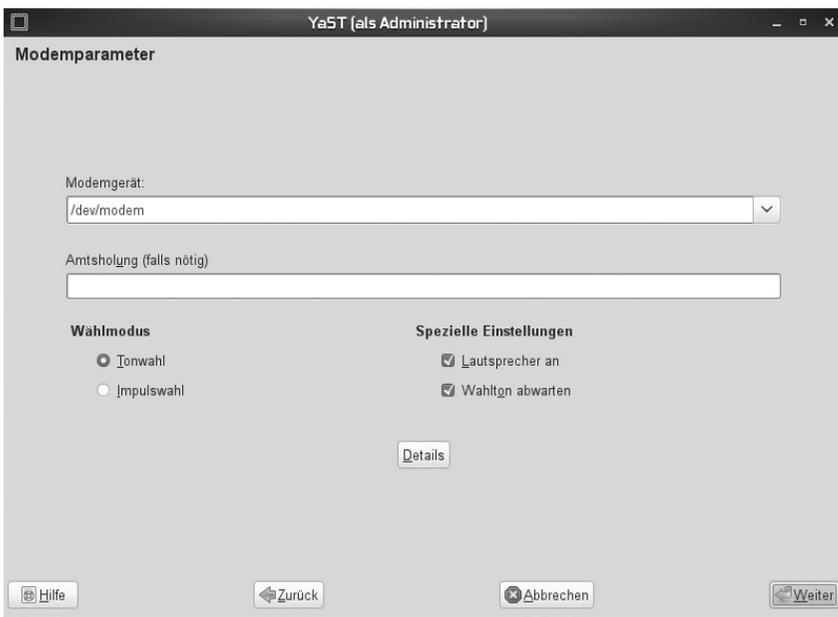


Abbildung 13.4 Einstellen der Modemparameter in YaST

- ▶ **DETAILS**
Diese Schablfläche öffnet einen weiteren Dialog mit Feineinstellungen, die Sie gegebenenfalls in der Dokumentation Ihres Modems und/oder in den Unterlagen Ihres Providers nachschlagen müssen:
 - ▶ Die BAUDRATE ist die Übertragungsgeschwindigkeit des Modems; voreingestellt ist der V.90-Standardwert 57.600.
 - ▶ Die INITIALISIERUNGS-STRINGS FÜR DAS MODEM gehören zu den oben angesprochenen AT-Befehlen. Der erste ist ATZ (Reset), der zweite stellt die Modemparameter ein.

- ▶ AKTIVIERE GERÄTE-KONTROLLE FÜR NICHT-ROOT NUTZER ÜBER KINTERNET ermöglicht es gewöhnlichen Benutzern, das Modem über KInternet zu verwalten.
- ▶ REGULÄRE AUSDRUCK DER VORWAHL ZUR AMTSHOLUNG legt fest, welche Zeichen normale Benutzer für das Amtsholungspräfix verwenden dürfen. Verwenden Sie beispielsweise [0-9]? für »eine oder keine Ziffer«.

Als Nächstes folgt der Dialog INTERNET SERVICE PROVIDER (ISP) WÄHLEN. Dieser Dialog entspricht dem weiter oben für DSL beschriebenen – abgesehen davon, dass eine echte Telefonnummer eingegeben werden muss (oder bei einem bereits in der Liste vorhandenen Provider automatisch eingetragen wird).

13.3.4 ISDN

Das damalige SUSE Linux gehörte zu den ersten Linux-Distributionen, die ISDN unterstützten. Wenngleich dieser Dienst im DSL-Zeitalter eine geringere Bedeutung besitzt als früher, funktionieren die ISDN-Treiber in openSUSE noch immer prima. In diesem Unterabschnitt wird wie gehabt zuerst kurz auf die Theorie eingegangen, danach erfolgt die Beschreibung der Konfiguration.

Theoretische Grundlagen

Das *Integrated Services Digital Network* (etwa »Digitalnetzwerk mit integrierten Diensten«) oder kurz ISDN wurde in den 80er-Jahren von verschiedenen europäischen Telefongesellschaften eingeführt; die aktivste von ihnen dürfte die in Deutschland damals noch zuständige Deutsche Bundespost gewesen sein. Es handelt sich im Prinzip um die Übertragung digitaler Signale über klassische Kupferdraht-Telefonleitungen. Einem reinen Telefonkunden bietet ISDN zunächst die folgenden unmittelbaren Vorteile:

1. Es werden zwei voneinander unabhängige Kanäle zur Verfügung gestellt; über beide kann gleichzeitig telefoniert, gefaxt oder Datenfernübertragung betrieben werden.
2. Die Rufnummer eines Anrufers, der ebenfalls ISDN verwendet, wird übermittelt und im Display eines entsprechend ausgerüsteten Telefons angezeigt.
3. Während eine Verbindung besteht, kann ein weiterer Anruf angenommen werden; entweder wird der jeweils andere Gesprächspartner in den Wartezustand versetzt (Makeln), oder der neue Anrufer wird mit dem bisherigen in dieselbe Verbindung aufgenommen (Dreierkonferenz).

Inzwischen stehen die meisten dieser Dienste dank der flächendeckend digitalen Vermittlung in Deutschland auch Analog-Kunden zur Verfügung; lediglich die

beiden unabhängigen Kanäle bleiben ISDN vorbehalten. Im Übrigen erhalten ISDN-Benutzer üblicherweise drei unabhängige Rufnummern (bei einigen Telefongesellschaften sogar noch mehr), die frei auf die jeweiligen Geräte verteilt werden können.

Technisch betrachtet werden sogar drei Kanäle zur Verfügung gestellt; die beiden *B-Kanäle* übertragen Telefongespräche, Faxe oder Daten mit jeweils 64 kBit/s, während der *D-Kanal* Dienstinformationen wie Rufnummernübermittlung oder Anklopfen mit 16 kBit/s überträgt. Für Internetverbindungen und andere Datenübertragungsmethoden besteht die Möglichkeit, beide B-Kanäle zu bündeln und auf diese Weise insgesamt eine Datenübertragungsrate von 128 kBit/s zu gewährleisten; natürlich entstehen dafür auch doppelte Kosten.

In der Praxis funktioniert ISDN folgendermaßen: An die normale TAE-Telefonsteckdose (oder den Telefonausgang eines DSL-Splitters, falls auch DSL vorhanden ist) wird ein spezielles ISDN-Endgerät namens *NTBA* angeschlossen. Es stellt zwei sogenannte *S0-Basisanschlüsse* zur Verfügung; sie verwenden die auch vom Twisted-Pair-Ethernet bekannten RJ-45-Stecker. An jeden dieser Anschlüsse kann ein ISDN-Endgerät angeschlossen werden, beispielsweise ein Telefon, ein Faxgerät oder ein ISDN-Adapter zur Computerdatenübertragung. Damit niemand seinen kompletten Telekommunikations-Gerätepark umstellen muss, werden als spezielle Form von ISDN-Endgeräten sogenannte *TK-Anlagen* angeboten, die wiederum den Anschluss analoger Endgeräte ermöglichen. Einige TK-Anlagen bieten auch durchgeschleifte *S0-Anschlüsse* an, um beispielsweise ISDN-Geräte an einer praktischeren Stelle anzuschließen.

Natürlich sollten Sie nicht versuchen, ein Analogmodem an eine TK-Anlage anzuschließen – erstens würde das den eigentlichen Vorteil der ISDN-Datenübertragung zunichtemachen, und zweitens funktioniert es nicht: Da die übertragenen Daten nach außen wie ISDN aussehen, würde die Gegenstelle ihre Antworten mit einer Übertragungsgeschwindigkeit übermitteln, die das Modem nicht verarbeiten kann.

Um einen Computer mit ISDN zu verbinden, werden stattdessen verschiedene Formen von ISDN-Adaptern angeboten: als PCI-Steckkarten oder externe USB-Geräte. Am verbreitetsten sind die Fritz!-Produkte der Berliner Firma AVM.

Eine interne ISDN-Karte sieht genauso aus wie eine moderne Ethernet-Karte, und es kann leicht passieren, dass man das Twisted-Pair-Netzwerkkabel mit seinem baugleichen Stecker in die ISDN-Karte steckt und umgekehrt (selbstverständlich geschieht in diesem Fall gar nichts). Jedenfalls muss normalerweise ein ISDN-Kabel vom Anschluss des ISDN-Adapters zu einem *S0-Anschluss* verlaufen.

Der Unterschied zwischen den Übertragungsraten eines heutigen Modems (56,6 kBit/s) und Ein-Kanal-ISDN (64 kBit/s) mag Ihnen nicht besonders groß erscheinen. Als jedoch das Internet für Privatkunden interessant zu werden begann, lag die Übertragungsrate der meisten Modems bei 9.600 oder 14.400 Bit/s; erst allmählich kamen Geräte mit 28.800 Bit/s hinzu. Abgesehen davon besitzt ISDN noch heute einen weiteren Vorteil gegenüber Modem-Verbindungen: Der Verbindungsaufbau geht ohne Verzögerung vonstatten, während es bei Modems zu Wartezeiten von etlichen Sekunden kommen kann, bis die Leitung bereit ist.

ISDN-Konfiguration mit YaST

Auch das Konfigurationselement ISDN befindet sich in der YaST-Kategorie NETZWERKGERÄTE. Sie können wie gehabt auf BEARBEITEN klicken, wenn Ihr ISDN-Adapter automatisch erkannt wurde; andernfalls müssen Sie HINZUFÜGEN wählen. Letzteres führt zunächst zur Registerkarte MANUELLE AUSWAHL DER ISDN-KARTE. Wählen Sie links den Hersteller Ihrer Karte und dann rechts das konkrete Modell. Unter SUCHE können Sie zudem eine inkrementelle Suche nach dem Namen einer Karte durchführen. Wenn Ihre Karte nicht dabei ist, sollten Sie sich im Handbuch oder auf der Website des Herstellers informieren, ob sie mit einem anderen Adapter baugleich oder kompatibel ist; andernfalls müssen Sie sich auf die Suche nach einem passenden Kernelmodul machen.

Die nächste Seite ist die ISDN-LOW-LEVEL-KONFIGURATION. Je nach Karte und gewähltem Treiber sind verschiedene Optionen verfügbar. Die vorliegende beispielhafte Beschreibung wurde mit der recht verbreiteten Fritz!Card PCI v2.0 getestet.

Für besagte Fritz!Card stehen zwei verschiedene TREIBER zur Verfügung: der nur binär verfügbare CAPI 2.0 DRIVER (leistungsfähiger und mit Fax-Funktionen) sowie der Open-Source-Treiber HISAX DRIVER. Bitte beachten Sie, dass der CAPI 2.0-Treiber und andere rein binäre ISDN-Treiber kein Bestandteil der openSUSE-Distribution sind, sondern sich auf der Nicht-OSS-Zusatz-CD befinden; der Dialog enthält in diesen Fällen direkt an Ort und Stelle einen Hinweis, wo Sie sie herunterladen können.

Das ISDN-PROTOKOLL regelt Übertragungsformat-Details; in ganz Europa wird heute EURO-ISDN (EDSS1) verwendet. 1TR6 ist ein historischer Standard, der keine Rolle mehr spielt; die restlichen Optionen sind für seltene Speziallösungen reserviert. Viele neuere Treiber unterstützen ohnehin nur noch Euro-ISDN. Zusätzlich müssen Sie Ihr LAND (oder, falls es nicht in der Liste auftaucht, Ihre LANDESVORWAHL), Ihre ORTSKENNZIFFER (*ohne* führende 0, zum Beispiel 221 für Köln) und eventuell die weiter oben im Abschnitt über Modems diskutierte VORWAHL ZUR AMTSHOLUNG eingeben.

Die Option ISDN-PROTOKOLLIERUNG STARTEN sorgt dafür, dass Logdateien über Ihre ISDN-Kommunikation geführt werden, die Sie etwa zur Überprüfung Ihrer Abrechnungen einsetzen können. Die GERÄTE-AKTIVIERUNG schließlich wurde bereits ausführlich für andere Netzwerk- und Wählschnittstellen diskutiert.

Es folgt der Bildschirm ISDN-SERVICE-AUSWAHL. Hier wählen Sie aus, über welches Datenprotokoll die Verbindung mit Ihrem Provider erfolgen soll. RAWIP wird kaum noch verwendet, weil die Authentifizierung ausschließlich anhand der anwählenden ISDN-Mehrgerätenummer (MSN) durchgeführt wird – dieses Merkmal lässt sich relativ leicht fälschen und ist somit unsicher. Zudem können IP-Adressen über RawIP nicht dynamisch zugewiesen werden. In fast allen Fällen werden Sie daher wahrscheinlich die Option NEUE SYNCPPP-SCHNITTSTELLE HINZUFÜGEN wählen.

Für die SyncPPP-Schnittstelle sind folgende Einstellungen vorzunehmen:

- ▶ **EIGENE TELEFONNUMMER**
Geben Sie diejenige ISDN-Rufnummer (MSN) ein, die Sie der ISDN-Karte zuordnen möchten (ohne Vorwahl). Falls Sie einen Fax-fähigen Treiber gewählt haben, wird dies auch Ihre Faxnummer.
- ▶ **GERÄTE-AKTIVIERUNG**
Regelt wie gehabt, wie die Verbindung aktiviert werden soll; die Details können Sie weiter oben im DSL-Abschnitt nachlesen. Da es für ISDN-Internetzugänge oft keine Flatrates gibt, sollten Sie bevorzugt MANUELL und nicht etwa BEIM SYSTEMSTART wählen – ansonsten kann es eventuell teuer werden. Die manuelle Aktivierung erfolgt mithilfe des Kommandos `ifup Schnittstellenname als root, also meistens ifup ipp0`. Um die Verbindung wieder zu unterbrechen, wird entsprechend `ifdown ipp0` verwendet.
- ▶ **BENUTZERGESTEUERT**
Im Zusammenhang mit der manuellen Geräte-Aktivierung ist diese Option sinnvoll, damit Sie nicht jedes Mal `root` werden müssen, um die ISDN-Anwahl durchzuführen.
- ▶ **CHARGEHUP**
Auflegen (HangUP) vor der nächsten Gebühreneinheit. Wenn Ihr ISDN-Anschluss das Merkmal »Gebühreninfo« während der Verbindung (und nicht nur *nach* der Verbindung) besitzt, kann dies auf Dauer zu einer geringfügigen Kostenersparnis führen.
- ▶ **KANALBÜNDELUNG**
Beide B-Kanäle sollen gleichzeitig verwendet werden; dies ist doppelt so schnell, aber meistens auch doppelt so teuer. Zudem können Sie in diesem Fall nicht mehr nebenher telefonieren.

▶ EXTERNE FIREWALL-SCHNITTSTELLE

Die ISDN-Verbindung soll von der openSUSE-Firewall als externe Verbindung ins »feindliche« Internet betrachtet werden – dies ist in aller Regel empfehlenswert.

▶ FIREWALL NEU STARTEN

Die openSUSE-Firewall soll beim ISDN-Verbindungsaufbau neu gestartet werden. Im Zusammenhang mit der Option EXTERNE FIREWALL-SCHNITTSTELLE ist diese Einstellung sinnvoll, weil sie garantiert, dass die Änderung der Firewall-Optionen aktiv wird.

▶ DETAILS

In diesem Zusatzdialog können Sie einige Feineinstellungen vornehmen. Hier ein kurzer Überblick:

- ▶ Die TELEFONNUMMERNLISTE ENTFERNTER RECHNER bestimmt, welche Nummern überhaupt von außen eine Wählverbindung zu Ihrer ISDN-Schnittstelle aufbauen dürfen.
- ▶ Aktivieren Sie aus Sicherheitsgründen die Option NUR NUMMERN AUS DER LISTE ERLAUBT, falls Sie die ISDN-Verbindung nur für Internetverbindungen, aber nicht für Faxe oder klassische ISDN-Punkt-zu-Punkt-Datenübertragung verwenden.
- ▶ Die RÜCKRUF-FUNKTIONEN bestimmen, wie Ihr Rechner auf eingehende Anrufe reagieren soll: RÜCKRUF AUS nimmt Anrufe normal entgegen. RÜCKRUF SERVER ruft einwählende Rechner automatisch zurück (gehen Sie vorsichtig damit um, denn das kann teuer werden!). RÜCKRUF CLIENT erwartet dagegen umgekehrt einen Rückruf, wenn Sie eine ISDN-Anwahl durchführen.
- ▶ Die RÜCKRUFVERZÖGERUNG bestimmt die Anzahl der Sekunden, nach denen Ihre ISDN-Schnittstelle den Rückruf durchführt (Server), beziehungsweise wie lange sie darauf wartet (Client).
- ▶ ZUSÄTZLICHE IPPPD-OPTIONEN: In dieses Feld können Sie manuelle Optionen für den ISDN-PPP-Dienst eintragen. Details erfahren Sie gegebenenfalls in der Dokumentation Ihrer ISDN-Karte oder bei Ihrem Provider.

Als Nächstes erscheint der Bildschirm EINSTELLUNGEN FÜR DIE ISDN-IP-ADRESSE. Sofern Ihr Provider Ihnen keine feste IP-Adresse zugewiesen hat (dies ist bei Wählverbindungen fast immer der Fall), kreuzen Sie einfach DYNAMISCHE IP-ADRESSE an. Wenn ISDN Ihre einzige beziehungsweise Ihre Haupt-Internetverbindung ist, müssen Sie zudem STANDARDROUTE ankreuzen, damit alle Datenpakete mit unbekanntem Ziel über diese Schnittstelle weitergeleitet werden.

Zum Schluss wird wiederum der Provider gewählt oder neu eingetragen – halten Sie sich an die weiter oben nachzulesende Beschreibung für DSL (außer dass Sie zusätzlich eine Rufnummer eingeben müssen).

13.4 Das Domain Name System (DNS)

Für die Netzwerkkommunikation zwischen Computern sind IP-Adressen eine hervorragende Einrichtung, und zwar sowohl IPv4- als auch die längeren IPv6-Adressen. Menschen dagegen bevorzugen den Umgang mit Namen gegenüber der Verwendung von Nummern. Aus diesem Grund ist es nützlich, wenn die Hosts im Internet neben der numerischen Adresse auch einen für Menschen leichter zu merkenden Namen besitzen. Das Problem ist nur, dass dieser Name auf effiziente und wenig störanfällige Weise in die IP-Adresse umgewandelt werden muss.

Als das ARPANet in Betrieb genommen wurde, bestand es aus sehr wenigen großen Rechnern an verschiedenen amerikanischen Universitäten. Zwar wuchs es von Anfang an recht schnell, aber selbst mit einigen Hundert beteiligten Knoten war es noch kein allzu großes Problem, eine Liste der Hostnamen mit den entsprechenden Adressen als Textdatei zu pflegen, weil sich Änderungen eher in Grenzen hielten. Diese Datei hieß *hosts.txt* und wurde regelmäßig an die beteiligten Institutionen verteilt.

Noch heute enthält jeder Computer mit einem Betriebssystem aus der UNIX-Familie eine Datei namens */etc/hosts*, die zur lokalen Namensauflösung dient, auch Ihr openSUSE-System. Diese Datei besitzt einen Aufbau wie diesen:

```
127.0.0.1      localhost
192.168.0.1   defaultgw    defaultgw.test.local
192.168.0.2   mailserver   mailserver.test.local
```

In jeder Zeile steht also eine IP-Adresse, gefolgt von einem oder mehreren Hostnamen, die in diese Adresse aufgelöst werden sollen. In den meisten Systemen lässt sich konfigurieren, ob Einträge in dieser Datei beim Nachschlagen von Namen Vorrang vor externen Nameservern haben sollen oder nicht.

Im Laufe der Jahre wuchs das Internet immer weiter, und irgendwann war der Austausch der stets aktualisierten *hosts.txt*-Datei nicht mehr effektiv genug. Aus diesem Grund machte man sich an die Arbeit, ein neues System mit einer verteilten Datenbank aufzubauen. Dieses System heißt *DNS* (Domain Name System); das Grundkonzept wird in RFC 1034 und 1035 beschrieben.

13.4.1 Das DNS-Konzept

DNS besteht im Wesentlichen aus zwei wichtigen Elementen: zum einen den Nameservern, die eine DNS-Server-Software ausführen und auf Anfrage Namensinformationen herausgeben, und zum anderen aus dem hierarchischen System der Domainnamen selbst.

Das System der Domainnamen ist in etwa mit einem Dateisystem vergleichbar, das Verzeichnisse, Unterverzeichnisse und Dateien enthält. Angenommen, ein UNIX-Dateisystem enthält die folgende Datei:

```
/home/sascha/books/opensuse/kap01.txt
```

Dies bedeutet, dass sich die Datei *kap01.txt* im Verzeichnis *opensuse* befindet. Dieses Verzeichnis ist ein Unterverzeichnis von *books*; dies wiederum ist ein Unterverzeichnis von *sascha* im Verzeichnis *home*. Das Verzeichnis *home* schließlich befindet sich unterhalb des Wurzelverzeichnisses */*.

Bei einem DNS-Domainnamen verhält es sich ähnlich, allerdings ist die Reihenfolge der Hierarchie umgekehrt. Betrachten Sie etwa den folgenden Domainnamen:

```
www.buecher.lingoworld.de
```

Es handelt sich um den Host/Dienst *www* in der Subdomain *buecher*, die sich in der Second-Level-Domain *lingoworld* unterhalb der Top-Level-Domain (TLD) *de* – ISO-Länderkürzel für Deutschland – befindet. Die eigentliche Wurzel des DNS ist an dieser Stelle nicht sichtbar; ihr Name ist der leere String "".

Wer seine Rechner im Internet unter einem Domainnamen betreiben möchte, muss zunächst eine Second-Level-Domain unter der passenden Top-Level-Domain registrieren. Je nach TLD ist jeweils eine andere Registrierungsstelle zuständig. Um die Top-Level-Domain *de* kümmert sich beispielsweise die DENIC (*www.denic.de*). In der Praxis ist es jedoch einfacher und meist billiger, einen Domainnamen über einen Provider zu registrieren.

Für typische kleine bis mittlere Unternehmen, die nichts weiter als eine Website und vielleicht ein paar E-Mail-Adressen benötigen, kommt hier am ehesten ein virtueller Host infrage: Ein Domainname wie *www.meine-site.de* verweist auf eine Website, die sich in Wirklichkeit zusammen mit Hunderten anderer Sites auf demselben Server-Rechner bei dem Provider befindet. Darüber hinaus lassen sich natürlich E-Mail-Adressen nach dem Schema *user@meine-site.de* anlegen. In diesem Fall ist der Hoster für die DNS-Konfiguration verantwortlich.

Eine DNS-Domain, die in der Obhut eines bestimmten Netzbetreibers steht, wird als *DNS-Zone* bezeichnet. Im Beispiel *www.buecher.lingoworld.de* ist *lingo-*

world.de eine solche Zone. Der Administrator von *lingoworld.de* kann Hosts innerhalb dieser Zone einrichten, indem er DNS-Daten mit ihren Hostnamen und IP-Adressen erstellt. Sollte er eine Subdomain wie *buecher.lingoworld.de* anlegen, kann er die Hosts und sonstigen Informationen darin entweder selbst konfigurieren oder aber die Administration dieser Subdomain delegieren und somit zur neuen, unabhängigen Zone machen. Dieses System der optionalen Delegation garantiert größtmögliche Flexibilität.

Interessant ist schließlich, wie eine DNS-Anfrage funktioniert: Ein Host, der Daten an einen bestimmten Empfänger im Internet senden möchte, aber nur dessen Hostnamen kennt, muss einen Nameserver befragen, um die zugehörige IP-Adresse zu erhalten.

Jeder Host, der direkt mit dem Internet verbunden ist, kennt mindestens einen Nameserver – entweder wurde dieser bei der Netzwerkkonfiguration manuell angegeben, oder er wurde bei der automatischen Schnittstellenkonfiguration über DHCP (Dynamic Host Configuration Protocol; im LAN) oder PPP (Point-to-Point Protocol; bei Modem, ISDN, DSL) vom entsprechenden Anmelde-Server übermittelt.

Der Host, der eine Namensauskunft wünscht, befragt nun die ihm bekannten Nameserver der Reihe nach. Angenommen, es wird die IP-Adresse zu *www.othernet.com* gesucht. Ein Nameserver, der diese Anfrage erhält, versucht nun, diese Information zu ermitteln und an den Anfragenden zurückzugeben. Damit er dieselbe Antwort beim nächsten Mal schneller geben kann, speichert er die ermittelten Daten in einem Cache (Zwischenspeicher). Wenn der befragte Nameserver bereits den Nameserver kennt, der für die Zone *othernet.com* zuständig ist, wendet er sich direkt an diesen, nimmt die Antwort in Empfang und gibt sie aus. Andernfalls muss er »eine Etage höher« nachfragen: Die Root-Nameserver für die Top-Level-Domain *de* kennen auf jeden Fall mindestens einen Nameserver für jede unterhalb dieser TLD registrierte Domain. Mit dieser Information kann der ursprünglich beauftragte Nameserver also weiterforschen und erhält schließlich die passende Antwort.

Es gibt im Grunde drei Arten von Nameservern:

1. Primäre Master-Nameserver
2. Slave-Nameserver
3. Caching-Only-Nameserver

Auf einem primären Master-Nameserver werden die maßgeblichen – die sogenannten *autoritativen* – Informationen für eine DNS-Zone konfiguriert. Dieser Server ist »der Weisheit letzter Schluss« für die entsprechende Zone.

Ein Slave-Nameserver dient der *Replikation*, enthält also eine Kopie der Daten eines primären Master-Nameservers. Dies dient sowohl der Ausfallsicherheit als auch der Entlastung des Master-Servers. Aus Sicherheitsgründen ist es ratsam, die DNS-Daten des eigenen primären Master-Nameservers auf mindestens einem Slave-Nameserver zu speichern, der sich nicht innerhalb der eigenen Netzwerk-Infrastruktur befindet. Größere Unternehmen und Internet-Provider stellen einander diese Dienstleistung oft gegenseitig zur Verfügung.

Caching-Only-Nameserver werden oft in kleineren Unternehmen eingesetzt, die ihre öffentlichen Server nicht innerhalb ihrer Geschäftsräume betreiben. Diese Server enthalten keinerlei autoritative Informationen und sind auch keine direkten Replikationspartner primärer Master-Nameserver, sondern beschleunigen durch die Zwischenspeicherung lediglich wiederholte Zugriffe auf externe DNS-Daten, oder sie dienen dem Zugriff auf Namensinformationen über Firewalls hinweg.

13.4.2 Der DNS-Server BIND

Es gibt verschiedene Implementierungen von Server-Software, die den DNS-Dienst versieht. Die wichtigste von ihnen ist BIND (Berkeley Internet Naming Domain). Es handelt sich um Open-Source-Software des Internet Software Consortiums (ISC). Sie können BIND und entsprechende Informationen darüber von der Website www.isc.org/products/BIND und zahlreichen dort verzeichneten Mirror-Sites herunterladen. Die zurzeit aktuelle Version ist BIND 9.6.1; die kurze Übersicht in diesem Unterabschnitt bezieht sich auf diese Version. Wenn Sie ausführlichere Informationen benötigen, finden Sie eine praxisorientierte Anleitung in Büchern wie »Das DNS und BIND Kochbuch« von Cricket Liu (Köln 2003, O'Reilly Verlag).

BIND konfigurieren

Als Erstes müssen Sie sicherstellen, dass BIND installiert ist. Das betreffende YaST-Paket *bind* befindet sich in der Selektion NETZWERK UND SERVER; installieren Sie es gegebenenfalls nach. Wenn Sie eine neuere Version als die in der Distribution enthaltene benötigen, können Sie die Software auch manuell kompilieren und installieren; dies wurde bereits als Beispiel für die Behandlung von Auto-make-Softwarepaketen in Kapitel 3, »Systemrundgang und Grundkonfiguration«, erläutert.

Danach können Sie sich an die Konfiguration von BIND machen. An dieser Stelle wird die manuelle Vorgehensweise beschrieben; alle hier behandelten Einstellungen finden Sie gegebenenfalls auch im YaST-Modul DNS-SERVER unter der Kategorie NETZWERKDIENTE wieder.

Die Konfigurationsdaten von BIND bestehen aus zwei verschiedenen Dateien: Die Datei `/etc/named.conf` enthält die Konfigurationsinformationen über den Nameserver selbst, während sogenannte Zonendaten-Dateien die eigentlichen Namensdaten in Form von *Resource Records* enthalten. Beide Dateiarnten sind einfache Textdateien, die Sie mit Ihrem bevorzugten Editor (siehe Kapitel 10, »Klassische Texteditoren«) bearbeiten können.

Die erste wichtige Angabe in der Datei `named.conf` beschreibt das Arbeitsverzeichnis, in dem sich die Zonendaten-Dateien befinden. Da diese Anweisung global ist und keine bestimmte Zone betrifft, wird sie in einen `options`-Block gesetzt:

```
options {
    directory "/var/named";
};
```

Anschließend folgen ein oder mehrere `zone`-Blöcke. Sie enthalten die Konfiguration der Zonen, für die der Nameserver zuständig sein soll.

Falls Ihr Nameserver beispielsweise primärer Master-Nameserver für die Zone `test.local` sein soll, sieht der entsprechende `named.conf`-Eintrag so aus:

```
zone "test.local" {
    type master;
    file "db.test.local";
};
```

Der Dateiname `db.test.local` ist nicht vorgeschrieben. Sie können jeden beliebigen Namen wählen, aber `db.name-der-zone` ist üblich. Natürlich müssen Sie die entsprechende Datei auch anlegen.

Für jede Zone, für die Ihr Nameserver als primärer Master dient, benötigen Sie zusätzlich eine Reverse-Lookup-Zone. Dies ist eine Zone, die den umgekehrten Dienst leistet: die Umwandlung einer gegebenen IP-Adresse in einen Domainnamen. Der Name dieser Zone hängt von der Größe des Subnets ab, in dem sie sich befindet: Es handelt sich um den umgekehrten Netzwerk-Teil der IP-Adresse mit dem reservierten Domainnamen `in-addr.arpa`. Angenommen, `test.local` verwendet das Netz `192.168.0.0/24`, dann wird die folgende Reverse-Lookup-Zone konfiguriert:

```
zone "0.168.192.in-addr-arpa" {
    type master;
    file "db.192.168.0";
};
```

Wenn Ihr Nameserver als Slave dienen soll, beispielsweise für die Zone `othernet.local`, sieht die entsprechende `zone`-Anweisung so aus:

```
zone "othernet.local" {
    type slave;
    masters { 192.168.4.3; };
    file "bak.othernet.local";
};
```

Unter `masters` wird eine Liste von Nameservern angegeben, von denen der Slave die Replikationsdaten erhält. Sie brauchen keine primären Master-Nameserver zu sein; der Begriff »masters« bezieht sich auf übergeordnete Nameserver aus der Perspektive des Slaves.

Schließlich benötigen Sie auf jeden Fall noch einen Eintrag für die Root-Hints-Datei. Diese spezielle Datei enthält Informationen über die Server, die den Stamm des DNS bilden und auf die Root-Nameserver der verschiedenen TLDs verweisen. Die entsprechende Datei (zum Beispiel `named.root`) ist entweder in Ihrer BIND-Distribution enthalten, oder Sie müssen sie per FTP unter der Adresse `ftp.rs.internic.net/domain/named.root` herunterladen. Der Eintrag in `named.conf` sieht so aus:

```
zone "." {
    type hint;
    file "named.root";
};
```

Für einen Caching-Only-Nameserver sieht die `named.conf`-Datei etwa folgendermaßen aus:

```
forwarders {
    194.8.194.70;
    194.8.194.71;
    194.8.194.83
};
forward only;
```

Die Nameserver in den geschweiften Klammern bestimmen, woher noch nicht bekannte Namensauflösungen bezogen werden sollen. Im vorliegenden Fall stammen sie von NetCologne; Sie sollten diejenigen Ihres eigenen Providers oder Netzwerkbetreibers eintragen. Die Einstellung `forward only` legt fest, dass Ihr Server sich ausschließlich auf diese Quellen verlässt und nicht selbstständig versucht, Namen aufzulösen.

Bitte beachten Sie, dass die Syntax der Datei *named.conf* ganz exakt eingehalten werden muss – aus eigener Erfahrung weiß ich, dass Programmierer besonders gern das Semikolon hinter der schließenden geschweiften Klammer vergessen.

Zonendaten

Falls Ihr Nameserver als primärer Master für eine Zone dient, müssen Sie nun die entsprechende Zonendaten-Datei anlegen. Sie enthält Resource Records für die einzelnen Hosts, Server-Dienste und andere Elemente der Zone. Die erste Zeile einer Zonendaten-Datei ist eine *\$TTL*-Anweisung (Time To Live) – sie gibt an, wie lange andere Nameserver die aus dieser Datei enthaltenen Informationen maximal im Cache halten dürfen. Sie können den Wert entweder komplett in Sekunden angeben oder mit den folgenden Maßeinheiten arbeiten: *w* (Wochen), *d* (Tage), *h* (Stunden), *m* (Minuten) oder *s* (Sekunden). Wenn der Wert 28 Stunden betragen soll, gibt es beispielsweise die folgenden drei Möglichkeiten:

```
$TTL 100800
$TTL 28h
$TTL 1d4h
```

Die nächste Information ist ein SOA-Record (Start of Authority). Er enthält die folgenden Konfigurationsinformationen über die Zone selbst:

- ▶ *MNAME*
der Hostname des primären Master-Nameservers
- ▶ *RNAME*
Die E-Mail-Adresse des Verantwortlichen; das @ wird durch einen Punkt ersetzt.
- ▶ *die Seriennummer der Zone*
Sie sollte bei jeder Aktualisierung erhöht werden; ein praktisches Format für manuell gepflegte Zonendaten ist *JJJMMTTVV* (die letzten beiden Stellen bilden eine Versionsnummer, die bei 00 beginnt und wichtig ist, falls einmal mehrere Änderungen an einem Tag stattfinden).
- ▶ *der Refresh-Wert*
Gibt das Intervall an, in dem die Slave-Nameserver anfragen sollen, ob die Zonendaten aktualisiert wurden.
- ▶ *der Retry-Wert*
Bestimmt, wie lange ein Slave warten soll, bevor er nach einem Verbindungsfehler erneut nach Aktualisierungen fragt.
- ▶ *der Expire-Wert*
Gibt an, wie lange die Slaves antworten sollen, wenn der primäre Master nicht

erreichbar ist. Dieser Wert sollte recht hoch sein, weil er für Ausfallschutz sorgt.

► *der Negative-Caching-Wert*

Gibt an, wie lange die Slaves negative Antworten («nicht gefunden» und so weiter) im Cache speichern dürfen. Der Wert sollte recht klein sein, da es sich um einen vorübergehenden Ausfall halten könnte.

Ein vollständiger SOA-Record für `test.local` sieht zum Beispiel so aus:

```
test.local.  IN  SOA  ns1.test.local. (
                hostmaster.test.local.
                2003112501
                30m
                10m
                30d
                30m )
```

Die (runden!) Klammern sind nur dann erforderlich, wenn Daten der Übersicht halber auf mehrere Zeilen verteilt werden.

Wenn Sie einen Host zu einer Zonendaten-Datei hinzufügen möchten, benötigen Sie einen A-Record (Address). Für `pc1.test.local` mit der IP-Adresse `192.168.0.24` sieht ein solcher Eintrag so aus:

```
pc1.test.local.  IN  A   192.168.0.24
```

Außerdem benötigen Sie einen PTR-Record (Pointer) in der Zonendaten-Datei der Reverse-Lookup-Zone (in diesem Beispiel in `db.192.168.0`). Für `pc1.test.local` sieht dieser Eintrag so aus:

```
24.0.168.192.in-addr.arpa.  IN  PTR  pc1.test.local.
```

Auf diese Weise müssen Sie alle Rechner in Ihrer Domain, die öffentlich über Hostnamen verfügbar sein sollen, angeben – natürlich auch Ihre eigenen Name-server, Webserver, Mailserver und so weiter.

Für Webserver-Betreiber ist es oft nützlich, wenn der reine Domainname (`test.local`) ebenso auf den Webserver verweist wie `www.test.local`; viele Benutzer versuchen den Zugriff ohne das Präfix `www`, weil sie das von großen Sites wie *google.com* gewohnt sind. Zu diesem Zweck brauchen Sie nur einen zusätzlichen A-Record einzurichten, der auf die IP-Adresse des Webserver verweist. Beide Möglichkeiten zusammen sehen also so aus:

```
www.test.local.  IN  A   192.168.0.3
test.local.     IN  A   192.168.0.3
```

Einen CNAME-Record (Canonical Name; ein Alias, der auf einen anderen Hostnamen verweist) dürfen Sie in diesem Zusammenhang nicht verwenden. Nützlich ist der CNAME-Record aber beispielsweise dann, wenn Ihr Webserver intern anders heißt als `www`:

```
www.test.local.    IN    CNAME    gandalf.test.local
```

Wenn das Ziel des Alias-Records (der Hostname auf der rechten Seite) sich in einer anderen Zone befindet als der Aliasname, muss dieser Record in der Zone stehen, in die der Aliasname gehört.

Sie können sogar über die Zonendaten bereits eine triviale Form von Load-Balancing betreiben. Wenn Sie mehrere Webserver betreiben, können Sie für jeden von ihnen einen A-Record schreiben:

```
www.test.local.  IN    A    192.168.0.3
www.test.local.  IN    A    192.168.0.4
www.test.local.  IN    A    192.168.0.5
```

Die IP-Adressen der Webserver werden dann im sogenannten Round-Robin-Verfahren (reihum) ausgegeben. Professionelles Load-Balancing benötigt allerdings eine komplexere Konfiguration, die nicht nur DNS betrifft, weil eine gerechte Verteilung der Webserver-Ressourcen beispielsweise auch von der Dateigröße der ausgelieferten Dokumente abhängt und nicht nur von der reinen Anzahl der Namensanfragen.

Ein weiterer wichtiger Typ von Resource Records sind die NS-Records (Nameserver). Sie geben sämtliche autoritativen Nameserver für die Zone an – wie bereits erwähnt sollte mindestens einer dabei sein, der sich nicht innerhalb des eigenen Netzwerks befindet. Für die Domain `test.local` könnten die NS-Records also beispielsweise so aussehen:

```
test.local.     IN    NS    ns1.test.local.
test.local.     IN    NS    ns2.test.local.
test.local.     IN    NS    ns1.provider.de.
```

Mithilfe von NS-Records wird übrigens auch die Delegation untergeordneter Zonen durchgeführt: Für eine Subdomain wird einfach ein anderer Nameserver angegeben. Sofern dieser Nameserver sich innerhalb der Zone befindet, die in der aktuellen Zonendaten-Datei konfiguriert wird, muss zusätzlich ein A-Record für diesen Nameserver angegeben werden:

```
accounting.test.local.    IN    NS    ns.accounting.test.local.
accounting.test.local.    IN    NS    ns2.provider.de.
ns.accounting.test.local. IN    A    192.168.0.9
```

Zu guter Letzt benötigen Sie noch MX-Records (Mail Exchange) für die Angabe von Mail-Zielen. Schließlich sollen Benutzer E-Mails an Adressen wie *user@test.local* schicken können statt an *user@mail.test.local*. MX-Records enthalten eine Prioritätsangabe, die festlegt, welcher Mailserver bevorzugt werden soll. Ein Server mit einem höheren Wert wird nur gewählt, wenn derjenige mit dem nächstkleineren offline oder aus anderen Gründen nicht verfügbar ist:

```
test.local.  IN  MX    0 mail.test.local.
test.local.  IN  MX   10 snailmail.test.local.
test.local.  IN  MX   20 mail.provider.de.
```

Es gibt noch zahlreiche andere Typen von Resource Records. Beispielsweise wurde hier nicht auf die recht komplexe DNS-Konfiguration für IPv6 eingegangen. Wenn Sie selbst für die Verwaltung Ihrer DNS-Zonen verantwortlich sind und öffentliche Nameserver betreiben müssen, kommen Sie nicht umhin, sich entsprechende Literatur zu beschaffen.

13.5 TCP/IP-Diagnose und -Fehlersuche

Beim Arbeiten mit TCP/IP-Netzwerken kommt es mitunter zu Ausfällen und sonstigen Problemen. Einige einfache Programme, die zur TCP/IP-Implementierung fast aller Betriebssysteme gehören, ermöglichen die Diagnose solcher Schwierigkeiten. In diesem Abschnitt werden die wichtigsten von ihnen kurz vorgestellt.

13.5.1 ping

Das einfachste, aber eines der wichtigsten TCP/IP-Dienstprogramme ist `ping`. Der Name ist kein Akronym, sondern ahmt das Geräusch eines Echolots nach. Genau diese Aufgabe erfüllt `ping` auch: An einen beliebigen Host gesendete Kontrolldatenpakete werden von diesem zurückgesendet. Wenn jedes Paket ordnungsgemäß zurückgesendet wird, ist die Verbindung gewährleistet. Zusätzlich gibt das Programm die Antwortgeschwindigkeit in Millisekunden aus.

Die Syntax von `ping` ist sehr einfach: Sie brauchen nur `ping hostname` beziehungsweise `ping ip-adresse` einzugeben. Das Programm sendet ein Datenpaket nach dem anderen an den Empfänger, bis Sie es mittels `[Strg]+[C]` abbrechen. Anschließend wird eine Statistik mit mittlerer Dauer und dem Prozentsatz verlorener gegangener Pakete angezeigt. Die `ping`-Ausgabe sieht beispielsweise so aus:

```
$ ping www.heise.de
PING www.heise.de (193.99.144.71): 56 data bytes
```

```

64 bytes from 193.99.144.71: icmp_seq=0 ttl=255 time=0.172 ms
64 bytes from 193.99.144.71: icmp_seq=1 ttl=255 time=0.099 ms
64 bytes from 193.99.144.71: icmp_seq=2 ttl=255 time=0.095 ms
64 bytes from 193.99.144.71: icmp_seq=3 ttl=255 time=0.093 ms
64 bytes from 193.99.144.71: icmp_seq=4 ttl=255 time=0.094 ms
64 bytes from 193.99.144.71: icmp_seq=5 ttl=255 time=0.093 ms
64 bytes from 193.99.144.71: icmp_seq=6 ttl=255 time=0.093 ms
64 bytes from 193.99.144.71: icmp_seq=7 ttl=255 time=0.098 ms
-- www.heise.de ping statistics --
8 packets transmitted, 8 packets received, 0% packet loss
round-trip min/avg/max = 0.093/0.104/0.172 ms

```

ping verwendet übrigens kein TCP oder UDP, sondern ein eigenes, noch leichtgewichtigeres Protokoll namens ICMP (Internet Control Message Protocol). Es wird in RFC 792 beschrieben.

Beachten Sie, dass die Erreichbarkeit eines Hosts per ping nichts weiter bedeutet, als dass er prinzipiell mit dem Netzwerk beziehungsweise Internet verbunden ist. Dies sagt nichts über seine Aufgabe im Netzwerk oder über die Funktionsbereitschaft von Server-Diensten aus.

Dennoch ist ping ein sehr praktisches Hilfsmittel. Beispielsweise können Sie durch »Anpingen« verschiedener Hosts unterscheiden, welche Netzwerkschnittstelle oder welcher Router unter Umständen ausgefallen ist.

13.5.2 traceroute

Das Programm `traceroute` verfolgt automatisch die Route zum angegebenen Host über das Internet. Dazu setzt es den TTL-Wert seiner Testpakete nacheinander auf 1, 2, 3 und so weiter. Ein Router, bei dem das aktuelle Paket den Wert 0 erreicht, verwirft es aber nicht wie üblich automatisch, sondern sendet eine Meldung zurück. `traceroute` sortiert die Antworten nach dem ursprünglichen TTL-Wert und gibt so nacheinander die Router aus, die bis zum Erreichen des Ziels passiert werden. Das Ganze sieht zum Beispiel so aus:

```

$ traceroute www.heise.de
traceroute to www.heise.de (193.99.144.71), 30 hops max, 40 byte
packets
 1 erx-maw1.netcologne.de (195.14.247.95) 71.634 ms
 2 swrt-maw1-g34.netcologne.de (213.196.239.169) 60.253 ms
 3 cat6509-pgl-v1200.netcologne.de (195.14.195.145) 58.283 ms
 4 rtint4-ge11.netcologne.de (81.173.192.2) 72.938 ms
 5 rtdecix-g01.netcologne.de (81.173.192.85) 56.923 ms
 6 de-cix2.ffm.plusline.net (80.81.193.132) 73.637 ms

```

```
7 c22.f.de.plusline.net (213.83.57.53) 64.342 ms
8 www.heise.de (193.99.144.71) 70.263 ms
```

traceroute ist beispielsweise dann nützlich, wenn ping keinen Erfolg gezeigt hat: Sie können erkennen, ob der Fehler in Ihrem eigenen Netzwerk oder »irgendwo da draußen« liegt.

13.5.3 netstat

Das Programm `netstat` liefert zahlreiche Informationen über den Netzwerkzustand. Wenn Sie den Befehl ohne weitere Angaben eingeben, erhalten Sie eine Übersicht über sämtliche aktuellen Netzwerkverbindungen. Sie können diese Liste mithilfe einer Option einschränken, sinnvollerweise auf TCP. Unter Linux heißt die entsprechende Option `-tcp` oder `-t`, unter vielen anderen UNIX-Versionen sowie unter Windows `-p tcp`. Andere Angaben wie UDP sind nicht sonderlich sinnvoll, da es keine »UDP-Verbindungen« gibt – in der Praxis würden die Kanäle angezeigt, über die vor Kurzem UDP-Datagramme versandt wurden.

Für TCP sieht die Ausgabe zum Beispiel so aus:

```
$ netstat -t
Aktive Internetverbindungen (ohne Server)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp    0      0 tux.test.local:18964    192.168.16:microsoft-ds VERBUNDEN
tcp    0      0 tux.test.local:8659     192.168.168.3:www-http  VERBUNDEN
```

Diese Ausgabe enthält keine lokalen Serverdienste. Wenn Sie diese einschließen möchten, müssen Sie die Option `-a` hinzufügen, das heißt `netstat -ta` eingeben.

Die Option `-r` erfüllt eine andere Aufgabe Sie gibt die aktuellen Routing-Tabellen für alle Schnittstellen aus. Das sieht beispielsweise folgendermaßen aus:

```
$ netstat -r
Aktive Routen:
Netzwerk Ziel           Netzmaske      Gateway         Schnittst. Metrik
      0.0.0.0      0.0.0.0 213.196.251.226 213.196.251.226 1
      127.0.0.0      255.0.0.0      127.0.0.1      127.0.0.1      1
      192.168.0.0    255.255.255.0 192.168.0.2    192.168.0.2    2
      192.168.0.2    255.255.255.255 127.0.0.1      127.0.0.1      1
      192.168.0.255 255.255.255.255 192.168.0.2    192.168.0.2    1
```

Ein vergleichbares Ergebnis liefert übrigens die Eingabe `route`.

13.5.4 nslookup

Das Programm `nslookup` befragt einen Nameserver nach der IP-Adresse zum angegebenen Hostnamen oder umgekehrt. Dabei muss die IP-Adresse des befragten Nameservers angegeben werden, sofern in der TCP/IP-Konfiguration des Betriebssystems keine festen Nameserver eingetragen wurden.

Das Ergebnis sieht beispielsweise wie folgt aus, wenn der erste Nameserver von NetCologne (194.8.194.70) nach der IP-Adresse für `www.heise.de` befragt wird:

```
$ nslookup www.heise.de 194.8.194.70
Server: ns1.netcologne.de
Address: 194.8.194.70
```

```
Nicht-autorisierte Antwort:
Name: www.heise.de
Address: 193.99.144.71
```

Die umgekehrte Art der Abfrage sieht etwa so aus:

```
$ nslookup 193.99.144.71 194.8.194.70
Server: ns1.netcologne.de
Address: 194.8.194.70
```

```
Name: www.heise.de
Address: 193.99.144.71
```

Das Programm `dig` dient ebenfalls dem Befragen von Nameservern, gibt aber ausführlichere Antworten. Beispielsweise kann die Art von DNS-Resource-Records angegeben werden, nach denen gesucht wird. Mit `dig` lautet die oben gezeigte Suche nach der IP-Adresse (also einem A-Record) für den Host `www.heise.de` und ihre Ausgabe so:

```
# dig @194.8.194.70 a www.heise.de
; <<>> DiG 2.2 <<>> @194.8.194.70 a www.heise.de

; (1 server found)
;; res options: init recurs defnam dnsrch
;; got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 10
;; flags: qr rd ra; Ques: 1, Ans: 1, Auth: 3, Addit: 3
;; QUESTIONS:
;;      www.heise.de, type = A, class = IN

;; ANSWERS:
www.heise.de. 65987 A 193.99.144.85
```

```
;; AUTHORITY RECORDS:
heise.de.      65987  NS      ns2.pop-hannover.net.
heise.de.      65987  NS      ns.heise.de.
heise.de.      65987  NS      ns.pop-hannover.de.

;; ADDITIONAL RECORDS:
ns2.pop-hannover.net.  65987  A       62.48.67.66
ns.heise.de.          4714   A       193.99.145.37
ns.pop-hannover.de.   9785   A       193.98.1.200

;; Total query time: 321 msec
;; FROM: prefect to SERVER: 194.8.194.70
;; WHEN: Sat Oct 15 16:16:05 2005
;; MSG SIZE sent: 30 rcvd: 175
```

13.5.5 telnet

Das Programm `telnet` beherrscht nicht etwa nur das gleichnamige Protokoll zur Terminalemulation, sondern ermöglicht auch die Kommunikation mit sämtlichen textbasierten Server-Diensten. Dabei können Sie die Eingaben, die normalerweise von einer Client-Software vorgenommen werden, manuell vornehmen. Auf diese Weise erhalten Sie einen direkten Einblick in Vorgänge, die normalerweise hinter den Kulissen stattfinden, und können so mögliche Kommunikationsfehler aufspüren.

Das Folgende ist beispielsweise der Mitschnitt einer Telnet-Session mit der Site *de.opensuse.org*. Die manuellen Eingaben in `telnet` (die Zeilen von `GET / HTTP/1.1` bis `Connection: close` sowie eine zusätzliche Leerzeile) entsprechen dem wichtigsten Teil der Daten, die normalerweise Ihr Browser an den Webserver übermittelt – Details darüber stehen in Kapitel 15, »LAMP«. Die Ausgabe ist keine Webseite (Statuscode 200), sondern eine Weiterleitung mit dem Statuscode 301, der ein Browser normalerweise automatisch folgt.

```
$ telnet de.opensuse.org 80
Trying 130.57.4.24...
Connected to de.opensuse.org.
Escape character is '^]'.
GET / HTTP/1.1
Host: de.opensuse.org
Connection: close

HTTP/1.1 301 Moved Permanently
Date: Mon, 01 May 2006 14:39:12 GMT
Server: Apache
```

```

X-Powered-By: PHP/4.4.0
Vary: Accept-Encoding, Cookie
Expires: -1
Cache-Control: private, must-revalidate, max-age=0
Last-Modified: Mon, 01 May 2006 14:39:12 GMT
Location: http://de.opensuse.org/Willkommen_auf_openSUSE.org
Content-Type: text/html
Connection: close
Content-Length: 1
Set-Cookie: ZNPCQ002-opende=V0012d871c0a; path=/
P3p: CP="NOI"
Via: 1.1 ICS_SERVER (iChain 2.3.300)

```

Connection closed by foreign host.

Statt der Portnummer 80 hätten Sie auch einfach `http` schreiben können, da dieser Well-known Port in `/etc/services` eingetragen ist.

Für die Terminalemulation, für die `telnet` ursprünglich entwickelt wurde, sollten Sie das Programm übrigens nach Möglichkeit nicht einsetzen. Das Problem ist nämlich, dass alle Daten – einschließlich Passwörtern – im Klartext über das Netzwerk beziehungsweise Internet übertragen werden. Eine vernünftige Alternative ist SSH (Secure Shell). Dieses Protokoll verwendet eine recht sichere Verschlüsselung; allerdings können Sie nicht das Programm `telnet` dafür verwenden. Näheres über SSH-Server und -Client erfahren Sie im nächsten Kapitel.

13.5.6 tcpdump

Das Tool `tcpdump` ist ein sogenannter *Packet-Sniffer*. Er gibt Informationen über sämtliche TCP-Pakete aus, die über eine bestimmte Schnittstelle übertragen werden. Zusätzlich schaltet er die betreffende Schnittstelle in den Promiscuous Modus, falls sie dies unterstützt, sodass Sie zumindest bei einem Hub den gesamten Traffic in Ihrem Netzabschnitt mitlesen können. Hier als Beispiel der Mitschnitt einer einzigen HTTP-Anforderung an einen Webserver im lokalen Netzwerk:

```

# tcpdump
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
12:34:31.809476 IP tux.test.local.13890 > 192.168.0.3.http: S
3366695844:3366695844(0) win 5840 <mss 1460,sackOK,timestamp 34513262
0,nop,wscale 2>
12:34:31.810173 arp who-has tux.test.local tell 192.168.0.3
12:34:31.810474 arp reply tux.test.local is-at 00:0c:29:df:76:f2 (oui Unknown)

```

```

12:34:31.810557 IP 192.168.0.3.http > tux.test.local.13890: S
336115303:336115303(0) ack 3366695845 win 33580 <mss 1460,nop,wscale
0,nop,nop,sackOK>
12:34:31.810748 IP tux.test.local.13890 > 192.168.0.3.http: . ack 1 win 1460
12:34:37.074324 IP tux.test.local.13890 > 192.168.0.3.http: P 1:17(16) ack 1
win 1460
12:34:37.222246 IP 192.168.0.3.http > tux.test.local.13890: . ack 17 win 33564
12:34:37.506413 IP tux.test.local.13890 > 192.168.0.3.http: P 17:19(2) ack 1
win 1460
12:34:37.573510 IP 192.168.0.3.http > tux.test.local.13890: . 1:1461(1460) ack
19 win 33562
12:34:37.573661 IP tux.test.local.13890 > 192.168.0.3.http: . ack 1461 win 2190
12:34:37.575766 IP 192.168.0.3.http > tux.test.local.13890: FP 1461:1975(514)
ack 19 win 33562
12:34:37.576675 IP tux.test.local.13890 > 192.168.0.3.http: F 19:19(0) ack 1976
win 2920
12:34:37.576774 IP 192.168.0.3.http > tux.test.local.13890: . ack 20 win 33562
Strg + C
13 packets captured
26 packets received by filter
0 packets dropped by kernel

```

Jede Ausgabezeile besitzt grundsätzlich folgendes Format:

1. Zeitstempel im Format Stunde:Minute: Sekunde.Mikrosekunde (zum Beispiel 12:34:31.810173)
2. Protokoll (hier jeweils IP für das IP-Protokoll oder arp für ARP-Anfragen, das heißt Ermitteln der MAC-Adresse)
3. Quelladresse.Port (192.168.0.3.http für den Server, tux.test.local.13890 – ein Ephemeral Port – für den Client). Die ARP-Kommunikation besitzt ein anderes Format: who-has Hostname tell Anfragender-Host für die Anfrage beziehungsweise reply Hostname is-at MAC-Adresse für die Antwort.
4. Zieladresse.Port (Details wie bei der Quelladresse)
5. TCP-Flags (einige wie etwa ack wurden in der Einführung dieses Kapitels behandelt)

Beachten Sie, dass `tcpdump` mit geeigneten Parametern auch als leistungsfähiger Ersatz für `ping`, `traceroute` und andere weiter oben vorgestellte Tools dienen kann. Lesen Sie für Einzelheiten die Manpage von `tcpdump(1)`.

Noch komfortabler sind übrigens Packet-Sniffer mit grafischer Oberfläche. openSUSE wird mit dem Klassiker dieser Gattung, *Wireshark*, ausgeliefert, der bis vor Kurzem noch Ethereal hieß. Sie finden das Tool im K-Menü oder im GNOME-

Anwendungsmenü unter SYSTEM • NETZWERK. Um es zu starten, müssen Sie das root-Passwort eingeben. Wählen Sie zunächst unter CAPTURE • INTERFACES eine Schnittstelle aus. Während die Überwachung läuft, wird eine Statistik angezeigt. Sobald Sie STOP drücken, erhalten Sie die Ergebnisse in Tabellenform und können diese in aller Ruhe analysieren – beispielsweise nach bestimmten Kriterien sortieren oder filtern.

13.6 Grundlagen der Netzwerksicherheit

Open-Source-Software wie Linux und die zugehörigen Netzwerkdienste gelten im Vergleich zur kommerziellen Konkurrenz als sicherer. Das liegt zum einen an der Tatsache, dass die Quellcodes von vielen verschiedenen unabhängigen Entwicklern begutachtet werden, sodass Fehler leichter gefunden und oft auch schneller behoben werden. Zusätzlich spielt natürlich die Tatsache eine Rolle, dass die meisten Autoren von Schadsoftware wie Viren oder Würmern sich auf die verbreitetste Plattform – nach wie vor Windows – konzentrieren. Zu sicher sollten Sie sich allerdings nicht fühlen, da mit der zunehmenden Verbreitung von Linux auch die Anzahl schädlicher Programme wächst. Zudem benötigt auch ein potenziell sicheres Open-Source-Betriebssystem eine anständige Konfiguration, damit seine Sicherheit in der Praxis funktioniert.

In diesem Abschnitt erhalten Sie zunächst einen Überblick über diverse Gefahren; anschließend werden Gegenmaßnahmen vorgestellt, vor allem die Konfiguration der openSUSE-Firewall.

13.6.1 Sicherheitsrisiken

UNIX- und Linux-Systeme sind weniger häufig von Viren, Würmern und ähnlichen Schädlingen betroffen als Windows-Rechner. Die wichtigsten Gefahren sind folgende:

► *Exploits*

Die Basis der meisten Angriffe auf Server ist die Ausnutzung (Exploit) bekannter Sicherheitslücken in Programmen. Solche Anfälligkeiten, beispielsweise für Overflows oder für die illegale Eingabe von Programmcode, kommen immer wieder vor. Die meisten Entdecker nutzen diese Lücken glücklicherweise nicht aus, sondern informieren den Hersteller der Software sowie bekannte Sicherheits-Mailinglisten wie Bugtraq. Bei Open-Source-Programmen steht oft innerhalb weniger Tage ein Patch zur Verfügung, der die Sicherheitslücke schließt; bei kommerzieller Software kann es mitunter einige Wochen dauern oder gar erst in der nächsten Hauptversion behoben

werden. Wichtig ist, dass Sie *verfügbare* Patches oder Upgrades immer sofort installieren; wenn Sie für einen Server verantwortlich sind, sollten Sie Bugtraq (<http://www.securityfocus.com>), den Heise-Sicherheits-Newsletter (<http://www.heise.de>) und andere Sicherheitsinformationen abonnieren.

► *DoS-Attacken*

Eine der häufigsten Arten von Angriffen auf Server sind die sogenannten Denial-of-Service-Attacken (DoS). Dabei wird der Serverdienst durch Überlastung zum Absturz gebracht. DoS benötigt nicht unbedingt Sicherheitslücken, sondern sendet oft technisch korrekte (aber viel zu viele) Anfragen an den Server. Gerade das macht es schwer, den Angriff abzuwehren; viele prominente Sites wie Amazon, Yahoo oder heise.de waren schon betroffen. Solange der Angriff aus einer einzigen Quelle stammt, lässt er sich bereits am Router herausfiltern. Häufiger verwenden DoS-Attacken aber gespoofte (gefälschte) Absender-IP-Adressen oder greifen gar von vielen Hunderten, durch Trojaner gesteuerten PCs gleichzeitig an – dies wird als Distributed Denial of Service (DDoS) bezeichnet.

► *Crackerangriffe*

Einige Angriffe auf Server erfolgen mit dem Ziel, die Kontrolle zu übernehmen und den Server für eigene Zwecke zu missbrauchen, beispielsweise als Spam-Relay oder Lagerstätte für illegale Inhalte. Beachten Sie, dass Sie technisch und vor allem rechtlich für Ihren Server verantwortlich sind; die Entschuldigung, dass mehrere Hundert Hollywood-Blockbuster oder widerliche Nazi-Propaganda durch einen Angriff dort gelandet seien, zählt vor Gericht möglicherweise nicht. Wenn Sie also einen öffentlichen Server betreiben, sind Sie in der Pflicht, sich intensiv um dessen Sicherheit zu kümmern – regelmäßige Updates und Patches installieren, Passwörter ändern, Logdateien untersuchen und so weiter (siehe unten).

► *Rootkits*

Bei der gefährlichsten Variante eines Crackerangriffs wird ein sogenanntes Rootkit installiert. Es ersetzt gezielt Systemdateien und -programme durch manipulierte Versionen, um den Angriff zu verschleiern. Beispielsweise könnte eine Rootkit-Version von ps (siehe Kapitel 4, »Mit der Shell arbeiten«) einen laufenden Backdoor-Prozess ausblenden. Die beste Chance, Rootkits zu erkennen, besteht in der Installation eines Intrusion Detection Systems (siehe unten).

► *Web-Attacken*

Besonders verbreitet sind Angriffsversuche auf Web-Transaktionen. Da sie im Klartext ablaufen, ist es relativ leicht, Session-IDs, Formulardaten oder Anmeldeinformationen auszuspiönieren und einzusetzen. Ein Beispiel ist das soge-

nannte *Cross-Site-Scripting* (XSS). Es verwendet einfache Tricks wie zum Beispiel Forenbeiträge, um mit JavaScript angereicherte Hyperlinks zu verbreiten, die nicht nur eine bestimmte URL anfordern, sondern dieser gleichzeitig spezielle Daten zusenden, die dazu führen, dass der Angreifer aus dem Besuch des Opfers auf der entsprechenden Site Informationen abgreifen kann.

► *Man-in-the-Middle-Attacks*

Eine Weiterentwicklung des XSS-Szenarios sind Man-in-the-Middle-Attacks wie etwa das Session-Hijacking. Sie machen sich einen gravierenden Sicherheitsmangel vieler Sites zunutze, die auf Verschlüsselung verzichten, nachdem sich ein Benutzer einmal gegenüber dem Webserver identifiziert hat. Gelingt es einem Angreifer, die URL einer Client-Anfrage aus einer solchen Session zu stehlen, kann er die nächste Anfrage unter der angeblichen Identität des ursprünglichen Users senden und erhält so möglicherweise private Informationen; im schlimmsten Fall handelt es sich bei der angegriffenen Site um das Homebanking-Interface eines Geldinstituts.⁴

► *Angriffe auf CGI-Skripte und Webanwendungen*

Weit verbreitet sind Angriffe auf CGI-Skripte und Webanwendungen. Das liegt nahe, weil diese oft überaus schlampig und ohne Sicherheitsbewusstsein programmiert werden. Wenn es genügt, an eine Formulareingabe ein Semikolon und einen Shell-Befehl anzuhängen, um den Server anzugreifen, dann läuft etwas schief. Für Webanwendungen gilt daher noch mehr als für alle anderen Computerprogramme: Misstrauen Sie jeder Benutzereingabe, und überprüfen Sie sie gründlich. Hinzu kommen natürlich auch hier Exploits; Webserver und Sprachen-Interpreter sind – wie jede Software – nicht fehlerfrei.

► *SQL-Injection*

Sehr viele Webanwendungen setzen auf eine relationale Datenbank auf – in Kapitel 15, »LAMP«, lernen Sie das beliebte Beispiel PHP/MySQL kennen. Da Eingaben aus Webformularen besonders häufig für SQL-Abfragen eingesetzt werden, kann es leicht passieren, dass Angreifer versuchen, diese Abfragen ein wenig zu verlängern.

Stellen Sie sich zum Beispiel vor, ein Formular enthält das folgende Feld:

```
Suche nach: <input type="text" name="such" />
```

4 Einer Bank, die solche hochsensiblen Transaktionen auf so unsichere Art und Weise handhabt, sollten Sie als Kunde allerdings sofort den Rücken kehren. Schließlich würden Sie auch keinen Schuhkarton statt eines stählernen Schließfachs mieten.

Ein PHP-Skript verwendet den Wert aus diesem Feld folgendermaßen:

```
$_such = $_GET['such'];
$result = mysql_query ("select * from info where
    beschreibung like '%$such%';");
```

Jemand gibt in das Feld `such` Folgendes ein:

```
test%'; delete from info;
```

Die SQL-Abfrage wird also folgendermaßen ergänzt:

```
select * from info
where beschreibung like '% test%';
delete from info; %';
```

Der Inhalt der Tabelle `info` wird vollständig gelöscht; der »Müll« hinter dem Semikolon nach `info` wird ignoriert. Natürlich können Angreifer den Quellcode des PHP-Skripts nicht sehen und wissen daher zum Beispiel nicht, wie die Datenbanken und Tabellen heißen. In der Praxis werden solche Angriffsversuche aber auch nicht manuell in die Formulare eingetippt, sondern durch automatisierte Skripte ausgeführt.

Auch hier gilt, dass Sie Formulareingaben dringend genau untersuchen und säubern müssen, bevor Sie Ihre Datenbank damit füttern.

13.6.2 Allgemeine Gegenmaßnahmen

Um sich vor den genannten Gefahren zu schützen, sollten Sie eine Reihe kombinierter Sicherheitsmaßnahmen durchführen. Nur eine sorgfältige Beachtung der nachfolgenden Punkte kann Sie einigermaßen vor Schäden bewahren.

► *Verwenden Sie Antiviren-Software*

Für Ihr Linux-System selbst mag dieser Punkt bisher nicht ganz so wichtig sein. Aber spätestens, wenn Sie einen Samba-Server für Windows-Clients betreiben (siehe nächstes Kapitel), sollten Sie Software installieren, die mit Windows-Viren fertig wird. openSUSE enthält das Paket `samba-vscan`, das die Dateien bei jedem Zugriff auf Ihre Samba-Freigaben überprüft. Achten Sie zusätzlich darauf, dass die entsprechenden Windows-PCs selbst mit aktueller, autoupdate-fähiger Antiviren-Software ausgestattet sind.

► *Installieren Sie regelmäßig alle Patches*

Fehlerfreie Software gibt es nicht. Viel mehr kommt es darauf an, dass bekanntgewordene Sicherheitslücken möglichst schnell geschlossen werden. Aufgrund der aktiven Communities und des Viele-Augen-Prinzips klappt dies

in der Welt der freien Software meist rascher und zuverlässiger als bei kommerziellen Programmen, bei denen Sie den Launen des Herstellers ausgeliefert sind – beispielsweise wird ein dringend nötiger Bugfix aus geschäftspolitischen Gründen manchmal erst mit der nächsten kostenpflichtigen Vollversion geliefert.

Umso wichtiger ist es, dass Sie jeden erhältlichen Sicherheitspatch sofort installieren. Schalten Sie zu diesem Zweck automatische Online-Updates ein (siehe Kapitel 3, »Systemrundgang und Grundkonfiguration«). Zusätzlich sollten Sie regelmäßig zuverlässige IT-News wie *heise.de* oder *golem.de* verfolgen. Allerdings sollten Sie *niemals* ein angebliches Sicherheitsupdate installieren, das per E-Mail gesendet oder verlinkt wird. In den meisten Fällen handelt es sich dabei um eine Phishing-Attacke (siehe oben).

► *Verwenden Sie eine Firewall*

Da viele Bedrohungen von offenen Netzwerkschnittstellen abhängen, lohnt es sich, diese zu kontrollieren und so weit wie möglich dicht zu machen. Im übernächsten Unterabschnitt wird die Konfiguration der in Ihre Distribution integrierten openSUSE-Firewall beschrieben.

► *Sichern Sie wichtige Daten regelmäßig*

Irgendwann erfolgt möglicherweise ein besonders heftiger Angriff, der allen Vorsichtsmaßnahmen trotz und tatsächlich Daten vernichtet. Deshalb ist es überaus wichtig, dass Sie regelmäßig – am besten täglich – alle wichtigen Daten sichern. Wenn Sie wiederbeschreibbare CDs oder DVDs dafür verwenden, verursacht dies nicht einmal laufende Kosten. Bequemerweise lässt sich Brennsoftware zudem so einstellen, dass sie beim Abmelden alle Daten aus den angegebenen Ordnern auf einen Datenträger brennt und anschließend den Rechner herunterfährt.

► *Öffnen Sie niemals unerwartete E-Mail-Anhänge*

Neuere E-Mail-Würmer sind polymorph; sie generieren bei jedem automatischen Versand neue Betreffzeilen und Texte. Zuweilen bedienen sie sich dabei sogar vorhandener Mails, sodass sie immer schlechter von harmlosen, absichtlich versandten Nachrichten zu unterscheiden sind. Wenn Ihnen jemand ein Attachment schickt, sollten Sie sich stets vergewissern, dass es absichtlich versandt wurde.

► *Arbeiten Sie im Alltag nicht mit root-Rechten*

Sie sollten – wie bereits in Kapitel 2 im Rahmen der Installation beschrieben – auf jeden Fall ein Benutzerkonto mit eingeschränkten Rechten einrichten und sich nur dann als `root` einloggen, wenn Sie Systemverwaltungsaufgaben zu erledigen haben.

► *Verwenden Sie Intrusion Detection Systems (IDS)*

Angriffe oder Angriffsversuche sind nicht immer leicht zu erkennen. Am besten ist es, Sie installieren eine Software, die unerwünschte Änderungen erkennt und Alarm gibt. Beachten Sie, dass Sie ein IDS unbedingt installieren müssen, bevor der betreffende Rechner an das Netzwerk oder Internet angeschlossen wird. openSUSE enthält ein relativ neues Tool namens `aide` (Advanced Intrusion Detection Environment), das automatisch Änderungen von Datei-Integritäten protokolliert und eventuell Alarm schlägt. Näheres erfahren Sie in der Manpage des Programms `aide(1)`.

► *Betreiben Sie Server in einer chroot-Umgebung*

Eine interessante Sicherheitsmaßnahme für Serverdienste besteht darin, diese in einen `chroot`-»Käfig« zu sperren. Das bedeutet, dass der Software vorgegaukelt wird, ihr lokales Unterverzeichnis sei das Wurzelverzeichnis des Dateisystems. Beachten Sie allerdings, dass auch dies keine absolute Sicherheit gewährleistet – es gibt immer wieder Sicherheitslücken in den eingesperrten Programmen oder in den `chroot`-Implementierungen, die es einem Angreifer ermöglichen, aus der `chroot`-Umgebung zu entkommen.

Die Einrichtung einer solchen Umgebung ist vor allem deshalb nicht ganz einfach, weil Sie herausfinden müssen, welche Verzeichnisse und Dateien das Programm außer seinem eigenen Installationsumfang benötigt, um ausgeführt werden zu können. Die genaue Konfiguration hängt stark vom jeweiligen Serverdienst und von Ihrem konkreten System ab.

► *Testen Sie Ihr System mit »Crackertools«*

Im ersten Moment hört es sich vielleicht merkwürdig an, aber eine der beste Schutzmaßnahmen vor Angriffen besteht darin, die Werkzeuge der potenziellen Angreifer zu benutzen, um Sicherheitslücken zu erkennen und nach Möglichkeit zu schließen. Eines der wertvollsten Tools in dieser Sparte ist *Nessus*. Dieses freie Programm funktioniert unter Linux und zahlreichen anderen UNIX-Varianten; Sie können es unter <http://www.nessus.org> herunterladen.

► *Geschulte Mitarbeiter*

Die beste technische Sicherheitsmaßnahme nützt nichts, wenn die User nicht darüber Bescheid wissen. Menschliches Versagen ist immer die größte Sicherheitslücke; einige der erfolgreichsten Cracker haben nicht nur durch ausgeklügelte technische Maßnahmen Zugriff auf geschützte Systeme erlangt, sondern vor allem durch »Social Engineering« – siehe beispielsweise »Die Kunst der Täuschung« von Kevin Mitnick (Bonn 2003, vmi), einem verurteilten Cracker und heutigen Sicherheitsfachmann.

Beispielsweise ist der vernünftige Umgang mit Passwörtern weithin unbekannt. Beachten Sie deshalb besonders die folgenden Punkte, und weisen Sie auch Ihre Kollegen darauf hin:

- ▶ Passwörter dürfen niemals sinnvolle Wörter sein, sondern müssen möglichst wirre Kombinationen aus Groß- und Kleinbuchstaben, Ziffern und Sonderzeichen sein.
- ▶ Passwörter müssen regelmäßig gewechselt werden. Wenn Ihr Betriebssystem eine Möglichkeit bietet, die Benutzer nach einer bestimmten Zeit zur Änderung zu zwingen, sollten Sie sie nutzen.
- ▶ Ein Passwort wird niemals aufgeschrieben! Ein sicheres Passwort, das man sich dennoch leicht merken kann, ergibt sich am einfachsten aus den Anfangsbuchstaben eines möglichst sinnlosen Satzes; einige Zeichen sollten allerdings keine Buchstaben sein. Ein Beispiel finden Sie in Kapitel 2, »openSUSE installieren«.
- ▶ Administratoren, Mitarbeiter von Internet-Providern, Polizisten oder Staatsbeamte fragen *niemals* nach dem Passwort. Wer auch immer Sie also danach fragt, ist ein Betrüger.
- ▶ Passwörter sind erst der Anfang – die vorgestellten Maßnahmen zur Passwortsicherheit sind wichtig. Für sich allein sind sie aber noch lange keine ausreichende Absicherung; das sollten sowohl Sie als auch die restlichen Benutzer wissen.

13.6.3 Grundbegriffe der Kryptografie

Die Verschlüsselung von Nachrichten als Strategie zum Überleben in einer oft feindlichen Umwelt ist Tausende von Jahren alt. Politische, wirtschaftliche oder militärische Informationen mussten geheim gehalten werden. Das Aufkommen moderner Massenkommunikationsmittel, die Informationen unabhängig von Menschen und schneller als diese reisen lassen, machte die sichere und unverfälschte Datenübertragung zu einem allgegenwärtigen Problem. Gerade das Internet als allgemein nutzbares Kommunikationsnetz macht Kryptografie für beinahe jeden interessant.

Trivialbeispiel Cäsar-Code

Die Codes früherer Zeiten waren eher einfach: Bekannt ist zum Beispiel der Cäsar-Code, bei dem jeder Buchstabe im Alphabet durch lineare Verschiebung durch einen anderen ersetzt wird. Der »Schlüssel« ist hier die Anzahl der Stellen, um die die Buchstaben verschoben werden. Tabelle 13.6 zeigt das bekannte Beispiel ROT13, das sie um 13 Stellen verschiebt.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M

Tabelle 13.6 Cäsar-Code oder ROTn: Verschiebung der Buchstaben um n Stellen; hier das Beispiel ROT13

Mithilfe der Tabelle dürfte es Ihnen keine Schwierigkeiten bereiten, den folgenden Satz zu lesen: »Irefpuyhrffryhat ybuag fvpu, qraa avpug nyryf trug wrqra rgjnf na«. Das Praktische an den 13 Stellen ist, dass das Alphabet genau halbiert wird, sodass Verschlüsselung und Entschlüsselung derselbe Vorgang sind. Das folgende Perl-Skript leistet entsprechend beides:

```
#!/usr/bin/perl -w
use strict;
print "Nachricht: ";
my $msg = <>;
$msg =~ tr/a-z/n-za-m/;
$msg =~ tr/A-Z/N-ZA-M/;
print "ROT13: $msg\n";
```

Natürlich ist ROT13 in den Augen der Kryptoanalyse (der Wissenschaft von der Entschlüsselung) ein Witz; der schnellste Ansatz ist eine Häufigkeitsanalyse (welcher Buchstabe kommt in einer Sprache am häufigsten vor?) – je länger der Text, desto schneller führt sie zum Ziel.

Moderne Verfahren

Durch die moderne Mathematik und vor allem die Einführung der Computertechnik wurden erheblich komplexere Verschlüsselungsverfahren geschaffen, die sich nicht mehr ohne Weiteres – zum Teil sogar gar nicht – knacken lassen. Das wichtigste Merkmal für eine gute Verschlüsselung ist allerdings eine ausreichende Schlüssellänge; das ausgeklügeltste Verfahren nützt nichts, wenn sich der Schlüsselraum, also die Gesamtheit aller möglichen Schlüssel, in wenigen Minuten durch einen Brute-Force-Angriff durchprobieren lässt.

Zunächst sollten Sie einige Grundbegriffe kennen, die im Zusammenhang mit der modernen Kryptografie im IT-Bereich häufig verwendet werden:

► *Symmetrische Verschlüsselung*

Bei dieser klassischen Form der Verschlüsselung wird zur Verschlüsselung und zur Entschlüsselung derselbe Schlüssel verwendet. Damit dies sicher ist, muss der Schlüssel selbst vor Außenstehenden verborgen werden. Für Internetverbindungen ist dieses Verfahren deshalb nicht geeignet.

► *Asymmetrische Verschlüsselung*

Diese Form der Kryptografie wird auch als Public-Key-Verfahren bezeichnet. Für die Verschlüsselung wird ein anderer Schlüssel verwendet als zur Entschlüsselung. Einer der beiden Schlüssel ist öffentlich (Public Key), der andere geheim (Private Key). Jeder, der an einem solchen Verschlüsselungsverfahren teilnehmen möchte, kann seinen öffentlichen Schlüssel verteilen. Wer diesem Teilnehmer eine Nachricht senden möchte, kann sie mit dessen öffentlichem Schlüssel verschlüsseln. Der andere Teilnehmer kann sie daraufhin mit seinem privaten Schlüssel dechiffrieren. Genau dieses Verfahren liegt Internet-Kryptografielösungen wie SSL oder PGP zugrunde.

► *Einwegverschlüsselung*

Bei dieser besonderen Form der Verschlüsselung ist die Dechiffrierung nicht vorgesehen und (sofern der Algorithmus korrekt arbeitet) auch nicht möglich. Einwegverschlüsselung wird beispielsweise für Passwörter verwendet: Anstatt das Passwort für den Vergleich mit der Benutzereingabe zu entschlüsseln (was ein erhebliches Sicherheitsrisiko wäre), wird die Eingabe nach demselben Verfahren chiffriert wie das verschlüsselt gespeicherte Passwort.

► *Digitale Signatur*

Eine Signatur wird benutzt, um die Identität eines Absenders zu gewährleisten. Damit dies wirklich sichergestellt ist, muss die Signatur von einer vertrauenswürdigen Zertifizierungsstelle beglaubigt werden. Anschließend kann der Anbieter Inhalte mit seinem privaten Schlüssel signieren; mit seinem öffentlichen Schlüssel können Empfänger seine Identität überprüfen.

► *Message-Digest*

Ein Message-Digest-Algorithmus (auch Message Authentication Code oder kurz MAC genannt) berechnet aus einem recht langen Inhalt eine möglichst kurze Zeichenfolge, die die Eindeutigkeit des ursprünglichen Inhaltes gewährleistet. Eine Änderung des Inhaltes sollte zuverlässig auch den Digest-Wert verändern. Dies gewährleistet die Integrität von Inhalten wie E-Mails, Webseiten oder Downloads.

Es gibt im Computer- und Netzwerkbereich zahlreiche Einsatzmöglichkeiten für Kryptografie. Hier nur einige der wichtigsten:

► *E-Mail-Verschlüsselung*

Die Verschlüsselung von E-Mail-Nachrichten mithilfe von PGP (Pretty Good Privacy) oder dessen Open-Source-Variante GnuPG ist eines der beliebtesten Einsatzgebiete für Kryptografie. Es handelt sich um ein Public-Key-Verfahren. Wenn ich jemandem eine verschlüsselte Nachricht senden möchte, kann ich sie mit *seinem* öffentlichen Schlüssel verschlüsseln, der allgemein verfügbar sein darf. Der Empfänger kann die Mail dann ausschließlich mit seinem pri-

vaten Schlüssel entschlüsseln. In modernen E-Mail-Clients lässt sich PGP oder GPG leicht integrieren, sodass es auf Knopfdruck zur Verfügung steht.

▶ *SSL/TLS für Webseiten*

Sicherheitsrelevante Web-Transaktionen sollten nicht im Klartext durchgeführt werden, sondern ebenfalls verschlüsselt ablaufen. Dazu wird gesichertes HTTP (HTTPS) über SSL oder das modernere TLS verwendet. Neben der Verschlüsselung wird die Identität des Anbieters garantiert; er muss ein digitales Zertifikat verwenden, das idealerweise von einer anerkannten Zertifizierungsstelle (CA, Certification Authority) beglaubigt wurde.

▶ *SSH (Secure Shell)*

Die Fernadministration von UNIX-Servern ist eine angenehme und oft zeitsparende Angelegenheit. Dazu wird traditionellerweise das Protokoll und Programm Telnet verwendet. Es ist allerdings extrem unsicher, weil sämtliche Daten im Klartext übertragen werden. SSH ist eine verschlüsselte Lösung für dieses Problem. Um mit einem SSH-Server zu kommunizieren, benötigen Sie ein entsprechendes Client-Terminal. Näheres dazu in Abschnitt 14.3.3, »SSH«.

▶ *VPN (Virtual Private Networks)*

Verteilte IT-Infrastrukturen mit mehreren, weit voneinander entfernten Standorten sind ein Problem: Eine dedizierte Standleitung zwischen ihnen ist oft teuer, das Internet ist dagegen zu unsicher und verlangt lästige Einschränkungen durch die mangelnden Fähigkeiten der verfügbaren Protokolle. Die Lösung heißt VPN: Ein voll ausgestattetes, frei konfigurierbares Netzwerk wird – vorzugsweise verschlüsselt – über das Internet »getunnelt«. openSUSE enthält zu diesem Zweck die Software openVPN.

13.6.4 Die openSUSE-Firewall konfigurieren

In YaST können Sie unter SICHERHEIT UND BENUTZER den Punkt FIREWALL wählen. Dies ermöglicht Ihnen die komfortable Konfiguration der SuSEfirewall2. In Wirklichkeit handelt es sich um ein benutzerfreundliches Frontend für die in den Linux-Kernel integrierte `netfilter`-Funktion. Diese kann auch über das Konsolenkommando `iptables` gesteuert werden. Damit Sie die grundlegende Funktionsweise kennenlernen, wird hier zunächst der manuelle Einsatz von `iptables` und anschließend die YaST-Konfiguration der Firewall beschrieben.

iptables

`iptables` steht seit der Linux-Kernel-Version 2.4 zur Verfügung. In 2.2er-Kernen gibt es einen weniger leistungsfähigen Vorläufer namens `ipchains`. Die Aufgabe von `netfilter` und dessen Konsolen-Frontend `iptables` besteht darin, IP-Datagramme schon vor der Auslieferung an die zuständigen Anwendungen

zu analysieren und gegebenenfalls auszusortieren, umzuleiten oder zu modifizieren.

Es gibt fünf eingebaute *Ketten* (Chains), in die Sie durch die Definition eigener `iptables`-Regeln eingreifen können:

- ▶ `PREROUTING`
unmittelbar nach dem Empfang durch eine Netzwerkschnittstelle
- ▶ `INPUT`
ein eingehendes Paket vor der Auslieferung an die zuständige Anwendung
- ▶ `OUTPUT`
ein ausgehendes Paket direkt nach seiner Erzeugung durch eine lokale Anwendung
- ▶ `POSTROUTING`
unmittelbar vor der Ausgabe über eine Netzwerkschnittstelle
- ▶ `FORWARD`
Behandelt Pakete, die nicht für den eigenen Rechner bestimmt sind, sondern die er in seiner Eigenschaft als Router nur weiterleitet.

Die Regeln zur Filterung von Datenpaketen stehen je nach Protokollen und Verwendungszwecken in verschiedenen *Tabellen*. Drei von ihnen sind vorgefertigt:

- ▶ `nat` (Network Address Translation)
Dient der Umschreibung von IP-Adressen und Ports, um Pakete des internen Netzwerks (in der Regel mit privaten Adressen) in das externe weiterzuleiten. Besitzt standardmäßig die Ketten `OUTPUT`, `POSTROUTING` und `PREROUTING`.
- ▶ `filter`
Die Standardtabelle; sie legt Regeln für erlaubte und verbotene Datenpakete fest: für eingehende Pakete wegen möglicher Angriffe, für ausgehende als Schutz vor Trojanern und Ähnlichem, aber auch, damit Anwendungen nicht »nach Hause telefonieren« (was bei Open-Source-Software allerdings kaum je der Fall sein wird), und schließlich auch für weitergeleitete Pakete (Routing). Die betreffenden Ketten sind dementsprechend `INPUT`, `OUTPUT` und `FORWARD`. Die Tabelle `filter` ist mithin die eigentliche Firewall.
- ▶ `mangle`
Diese Tabelle ermöglicht beliebige Änderungen der Paket-Header, sodass sich selbst die ausgefallensten Modifikationen vornehmen lassen. In dieser Tabelle werden alle fünf Ketten unterstützt.

Für jede Regel kann ein Kriterium (*Match*) hinzugefügt werden, um festzulegen, für welche Pakete sie gelten soll. Schließlich benötigen die Regeln noch ein Ziel (*Target*); standardmäßig sind vier definiert:

- ▶ ACCEPT
Das Paket wird akzeptiert und somit zum nächsten Verarbeitungsschritt weitergeleitet (wo es wiederum durch eine andere Regel beeinflusst werden könnte).
- ▶ DROP
Das Paket wird nicht mehr weiterverarbeitet, sondern vollständig verworfen. Die Variante REJECT sendet dem Paketabsender zusätzlich eine Ablehnungsmeldung.
- ▶ QUEUE
Das Paket wird nicht weiter im Kernel verarbeitet, sondern an den Userspace weitergeleitet und kann hier von einer spezialisierten Anwendung weiterverarbeitet werden.
- ▶ RETURN
Kommt in benutzerdefinierten Verarbeitungsketten zum Einsatz und bedeutet, dass die Verarbeitung in der ursprünglichen, aufrufenden Kette weitergehen soll (ähnelt also der return-Anweisung in vielen Programmiersprachen).

iptables besitzt zahlreiche Parameter. Einige dienen der Angabe der oben genannten Komponenten, andere liefern etwa spezielle Vergleichswerte. Hier nur einige der wichtigsten:

- ▶ -t Tabelle oder --table Tabelle
Gibt eine der oben genannten Tabellen (oder gegebenenfalls eine selbst definierte Tabelle) an. -t nat arbeitet beispielsweise in der nat-Tabelle.
- ▶ -A Kette beziehungsweise --append Kette
Hängt die Regel an die angegebene Kette an. Ein Beispiel wäre -A INPUT für die Kette INPUT. Da manche Regeln nur dann sinnvoll sind, wenn eine bestimmte Reihenfolge eingehalten wird (etwa das Verbot für Pakete von einer bestimmten Quelladresse, während alle anderen erlaubt sind), existiert alternativ die Option -I Kette [Regelnummer]. Sie fügt die neue Regel vor der angegebenen Regel ein oder an den Anfang der Kette, falls Sie keine Nummer angeben. Daneben gibt es übrigens weitere Optionen wie -R (--replace) oder -D (--delete), um Regeln für eine Kette zu ersetzen beziehungsweise zu löschen.
- ▶ -j Ziel
Gibt eines der oben aufgeführten Standard-Targets oder ein anderes Ziel an, bestimmt also, was mit Paketen geschehen soll, wenn die Regel zutrifft. -j DROP verwirft beispielsweise die betreffenden Pakete.
- ▶ -s [!]Quelle[adresse[/Maske]] (auch --src oder --source)
Gibt die Quelladresse beziehungsweise den Quelladressbereich eines Pakets

an. Wenn Sie der Quellangabe ein Ausrufezeichen voranstellen, gilt die Regel für alle Pakete, die *nicht* aus der angegebenen Quelle stammen. `-s 192.168.17.0/24` definiert beispielsweise eine Regel für Pakete, die aus dem angegebenen Netz stammen, während Regeln mit `-s !192.168.0.0/24` für alle Pakete außer denjenigen aus dem betreffenden Netz gelten.

- ▶ `-d [!]Zieladresse[/Maske]` (alternativ `--dst` oder `--destination`)
Legt Regeln für Pakete fest, die eine bestimmte Zieladresse beziehungsweise einen Zieladressbereich haben. Das Ausrufezeichen negiert auch hier die Angabe. `-d 205.18.41.27` betrifft zum Beispiel alle Pakete, die für die angegebene Einzel-IP-Adresse bestimmt sind.
- ▶ `-p [!]Protokoll` (Langform `--protocol`)
Definiert eine Regel für Pakete, die das betreffende Transportprotokoll verwenden beziehungsweise – mit Ausrufezeichen – nicht verwenden. Sie können alle Angaben aus der Datei `/etc/protocols` verwenden, sowohl numerisch als auch mit Namen. Einige wichtige Protokolle sind beispielsweise `tcp`, `udp` und `icmp` (Ping-Pakete).
- ▶ `--source-port` Portnummer oder `--sport` Portnummer
Beschreibt den Quell-Port (TCP) beziehungsweise die Quell-Dienstnummer (UDP) von Paketen, etwa `--sport 80` oder `--sport http` für Pakete, die von einem (je nach Kette lokalen oder entfernten) Webserver stammen. Zulässige Angaben sind alle Dienstnummern von 0 bis 65.535 sowie alle Dienstnamen aus `/etc/services`. Diese Option benötigt die TCP- beziehungsweise UDP-Erweiterung von `iptables`.
- ▶ `--destination-port` Portnummer oder `--dport` Portnummer
Entspricht dem Ziel-Port des Pakets. Ein Beispiel ist `--dport 25` oder `--dport smtp` für Pakete, die an einen SMTP-Server gesendet werden.
- ▶ `-i` Schnittstelle
Beschränkt die Regel auf eine bestimmte Eingabe-Netzwerkschnittstelle, beispielsweise `eth0` für die erste Ethernet-Karte.
- ▶ `-o` Schnittstelle
Ist entsprechend für die ausgehende Schnittstelle gedacht. Ein Beispiel wäre `-o ds10` für Pakete, die über den ersten DSL-Anschluss ausgegeben werden.
- ▶ `-L [Kette]` beziehungsweise `--list [Kette]`
Listet alle Regeln auf, die für die jeweilige Kette definiert wurden. Wenn Sie keine Kette angeben, werden alle vorhandenen `netfilter`-Regeln ausgegeben.
- ▶ `-h` oder `--help`
Zeigt die verfügbaren Optionen an.

Für eine umfangreiche Auflistung möglicher Regeln ist an dieser Stelle natürlich kein Platz, zumal die Standardkonfiguration der openSUSE-Firewall schon einmal fundamentalen Schutz bietet. Hier nur einige Beispiele:

- ▶ `# iptables -t filter -I INPUT -p tcp -s 202.72.18.0/24 -j DROP`
Alle eingehenden TCP-Pakete (`INPUT`) aus dem Netzwerk `202.72.18.0/24` sollen verworfen werden. Wenn Sie die IP-Adressen bekannter Spammer, Virenschleudern oder anderer unerwünschter Gäste kennen, können Sie sie auf diese Weise leicht abweisen. Da in vielen Fällen bereits eine Regel existiert, die alle TCP-Pakete erlaubt, ist es erforderlich, diese Einstellung mit `-I` an den Anfang der `INPUT`-Kette zu stellen.
- ▶ `# iptables -t filter -I INPUT -p ICMP -j DROP`
Diese Regel verhindert grundsätzlich, dass Ihr Rechner mit `ping` überprüft wird. Dies kann sinnvoll sein, wenn Sie Opfer eines DoS-Angriffs mit falsch formatierten ICMP-Paketen werden (»Ping of Death«); manche Netzwerkbetreiber stellen diese Regel auch vorbeugend auf.
- ▶ `# iptables -t filter -O OUTPUT -d 192.168.168.7 -j DROP`
Hier eine Regel, die alle Pakete an eine bestimmte Zieladresse ablehnt. Damit könnten Sie es einer Software abgewöhnen, Verbindungen zum Server ihres Herstellers herzustellen.
- ▶ `# iptables -t nat -A POSTROUTING -o eth0 -j SNAT \`
`> --to 196.48.17.6`
Ein Beispiel für Source-NAT: Die IP-Adresse von Paketen, die über die Schnittstelle `eth0` ausgegeben werden, wird auf einen anderen (externen) Wert geändert.
- ▶ `# iptables -t nat -A POSTROUTING -o ds10 -j MASQUERADE`
IP-Masquerading ist eine spezielle Form von Source-NAT, die die lokale IP-Adresse mit der dynamisch zugewiesenen überschreibt. Auf diese Weise können Sie beispielsweise einen DSL-Anschluss für mehrere Rechner in einem lokalen Netzwerk nutzen.
- ▶ `# iptables -t nat -A PREROUTING -i eth0 -j DNAT \`
`> --to 192.168.0.7`
Destination-NAT ist das Gegenteil von Source-NAT. Hier wird die IP-Adresse eingehender Datenpakete in eine lokale Adresse geändert.

Die Firewall mit YaST konfigurieren

Die Firewall-Konfiguration in YaST besitzt sieben Einstellungskategorien, die hier gemäß der Listenreihenfolge dargestellt werden:

- ▶ **START**
Hier können Sie zunächst festlegen, ob die Firewall automatisch **BEIM SYSTEM-START** oder **MANUELL** gestartet werden soll. Zu empfehlen ist in aller Regel der standardmäßig voreingestellte automatische Start. Zusätzlich gibt es in dieser Rubrik drei Schaltflächen, über die Sie die Firewall starten, stoppen oder neu starten können. Letzteres ist erforderlich, um veränderte Einstellungen sofort wirksam werden zu lassen.
- ▶ **SCHNITTSTELLEN**
Hier können Sie auswählen, zu welcher Netzwerkzone die einzelnen Schnittstellen jeweils gehören sollen. Die Bedeutung der vier möglichen Zonen wurde bereits weiter oben im Rahmen der Ethernet-Konfiguration beschrieben.
- ▶ **ERLAUBTE DIENSTE**
Da die Firewall zumindest in der externen Zone standardmäßig alle Dienste sperrt, werden hier explizit diejenigen Dienste festgelegt, die ausnahmsweise erlaubt sind. Wählen Sie einen Dienst aus dem Pulldown-Menü **ZU ERLAUBENDER DIENST**, und klicken Sie auf **HINZUFÜGEN**. Mit **ENTFERNEN** können Sie einen vorhandenen Dienst wieder loswerden. Die Schaltfläche **ERWEITERT** ermöglicht es Ihnen, auch Dienste freizugeben, die nicht in der Liste vorkommen. Dies geschieht im Wesentlichen anhand ihrer TCP-Ports beziehungsweise UDP-Dienstnummern.
- ▶ **MASQUERADING**
Hier können Sie die oben beschriebene dynamische Änderung der IP-Adresse von Datenpaketen durchführen. Nutzen Sie diese Option, falls Sie Ihren Rechner als Router konfiguriert haben, um Ihren Internetzugang auch anderen Rechnern im Netzwerk zur Verfügung zu stellen, obwohl Sie private IP-Adressen verwendet haben.
- ▶ **BROADCAST**
Dieser Dialog steuert das IP-Broadcasting. Normalerweise besitzt jedes IP-Teilnetz, wie in der Einführung beschrieben, eine eindeutige Broadcast-Adresse, sodass Sie hier nichts Besonderes einzustellen brauchen. Die Option **NICHT BENÖTIGTE BROADCAST-PAKETE PROTOKOLLIEREN** ist nützlich, um eventuelle Angriffe über fehlerhafte Broadcast-Versuche anhand von Logdateien zu erkennen.
- ▶ **PROTOKOLLIERUNGS-LEVEL**
Bestimmen Sie hier, welche akzeptierten beziehungsweise nicht akzeptierten Pakete in Logdateien protokolliert werden sollen. Die Voreinstellung **NUR KRITISCHE PROTOKOLLIEREN** ist für gewöhnlich in Ordnung. **ALLES PROTOKOLLIEREN** kann Ihnen beim Debuggen selbst programmierter Netzwerkanwendungen oder neu konfigurierter Serverdienste helfen.

► BENUTZERDEFINIERTER REGELN

In diesem Bereich können Sie selbst beliebige TCP- oder UDP-Ports angeben, über die die Netzwerkkommunikation als Ausnahme erlaubt werden soll.

13.7 Zusammenfassung

In diesem Kapitel haben Sie sehr ausführlich erfahren, wie Sie Ihr openSUSE -System für den Einsatz in verschiedenen lokalen Netzwerken und im Internet konfigurieren können. Zunächst wurden die wichtigsten Grundlagen des TCP/IP-Netzworkings erläutert, weil es für die Netzwerkkonfiguration wichtig zu verstehen ist, was Sie tun. Das hier gebotene Wissen über das TCP/IP-Schichtenmodell, IP-Adressierung und Transportprotokolle genügt, um ein Heimnetzwerk zu konfigurieren. Wenn Sie für ein Firmennetzwerk verantwortlich sind, benötigen Sie allerdings auf jeden Fall zusätzliche Kenntnisse. Dazu empfehle ich Ihnen beispielsweise das Buch »TCP/IP-Netzwerkadministration«, 3. Auflage, von Craig Hunt (Köln 2003, O'Reilly Verlag).

Nach den Grundlagen haben Sie die Inbetriebnahme der beiden wichtigsten Arten lokaler Netzwerkschnittstellen kennengelernt: Ethernet-Karten und Wireless LAN. Da heute fast alle LANs mit diesen beiden Schnittstellensorten betrieben werden, dürfte dies für die Praxis ausreichen. Sollte in Ihrem Netzwerk etwa Exotisches zum Einsatz kommen, funktioniert die Konfiguration allerdings sehr ähnlich – vorausgesetzt, openSUSE unterstützt die entsprechende Technik.

Mindestens genauso wichtig sind Wählverbindungen ins Internet. Dazu wird immer häufiger DSL eingesetzt, aber openSUSE kann selbstverständlich auch mit Modems und ISDN umgehen. Damit steht auch der Nutzung sämtlicher Internetdienste nichts im Wege.

Ein professionelles Netzwerk benötigt eine zentrale Möglichkeit, die Rechnernamen in IP-Adressen aufzulösen. Dazu werden Nameserver verwendet, die das Domain Name System (DNS) bilden. Die verbreitetste Nameserver-Software, die auch in openSUSE enthalten ist, ist BIND. Im entsprechenden Abschnitt haben Sie die wichtigsten Grundlagen seiner Konfiguration kennengelernt.

Als Nächstes haben wurden einige praktische Tools zur Problembewegung in TCP/IP-Netzwerken vorgestellt. Wenn Sie für ein Netzwerk verantwortlich sind, werden Sie Tools wie `ping` oder `dig` wahrscheinlich täglich einsetzen.

Zu guter Letzt ging es noch um das wichtigste Thema im Bezug auf Netzwerke: Grundlagen der Sicherheit. Nach einem Überblick über Gefahren und Gegenmaßnahmen wurde die Konfiguration der openSUSE-Firewall erläutert –

zunächst manuell über das Kommandozeilentool `iptables`, anschließend auch mit YaST.

Mit Netzwerkthemen geht es auch in den nachfolgenden Kapiteln weiter: Im nächsten Kapitel, »Datei- und Verzeichnisdienstserver«, lernen Sie die wichtigsten Serverdienste zur Freigabe von Dateisystemen und für die zentrale Anmeldung kennen. In Kapitel 15, »LAMP«, erfahren Sie, wie Sie mit Apache, PHP und MySQL eine leistungsfähige Plattform für Webanwendungen einrichten können, und Kapitel 16, »Weitere Internet-Serverdienste«, beschreibt FTP, E-Mail & Co.

*Ein guter Diener muss stark sein.
– Jean-Jacques Rousseau*

14 Datei- und Verzeichnisdienstserver

Eines der wichtigsten Anwendungsgebiete für Netzwerke ist die Bereitstellung von Dateisystemen für Benutzer anderer Rechner. Dafür wurden im Lauf der Jahre viele verschiedene Lösungen entwickelt. Nicht nur in diesem Zusammenhang sind zentrale Anmeldedienste interessant, das heißt, die Verwaltung von Benutzern und Passwörtern auf einem zentralen Rechner im Netzwerk. Für beide Arten von Netzwerkdiensten lernen Sie in diesem Kapitel einige wichtige Beispiele kennen; im letzten Abschnitt geht es um weitere Serverarten.

14.1 Dateiserver

Ein Dateiserver oder File-Server macht Dateisysteme über ein Netzwerk verfügbar. Diese können dann von den Benutzern anderer Rechner genauso gemountet werden wie lokale Partitionen, sowohl manuell per `mount`-Kommando als auch automatisiert durch einen `/etc/fstab`-Eintrag. In diesem Abschnitt lernen Sie zwei Dateiserver-Dienste kennen: NFS für die Datenfreigabe an Linux und andere UNIX-Systeme sowie Samba als freien Server für Windows-Netzwerke.

14.1.1 NFS

Die klassische, unter allen gängigen UNIX-Systemen standardisierte Form der Dateifreigabe ist das Network File System (*NFS*). Es gibt zwar zahlreiche modernere, leistungsfähigere Netzwerkdateisysteme, aber noch immer bildet NFS eine Art kleinsten gemeinsamen Nenner. Es wurde von Sun Microsystems für die haus-eigenen UNIX-Derivate (SunOS, Solaris) entwickelt, steht aber frei zur Verfügung (Internetstandard in RFC 1094, RFC 1813 und weitere).

Wenn Sie auf Ihrem Rechner Dateien über NFS freigeben möchten, müssen Sie zunächst den NFS-Server starten, dessen eigentlicher Name `nfsd` (Network File System Daemon) lautet. Als grundlegender Kommunikationsdienst wird *RPC* (Remote Procedure Call) verwendet; er ermöglicht, wie der Name schon sagt, die

Steuerung von Diensten auf entfernten Rechnern und wurde ursprünglich eigens für NFS entwickelt. Sein wichtigster Bestandteil ist der *Portmap*-Server, der auf dem TCP- und UDP-Port 111 lauscht und die Verbindungen vermittelt.

Auf einem openSUSE-System können Sie den NFS-Server bequem über YaST konfigurieren und für den automatischen Start vorbereiten. Dies wird weiter unten erläutert, nach einem Blick auf die NFS-Konfigurationsdatei.

NFS-Freigaben

Welche Verzeichnisse Sie für den Netzwerkzugriff freigeben möchten, regelt die Konfigurationsdatei */etc/exports*. Ein Eintrag zur Freigabe eines Verzeichnisses besitzt das folgende Format:

```
Pfad Netzwerk[(Option[,Option...])] [Netzwerk...]
```

Wenn Sie also beispielsweise das Verzeichnis */var/test* ohne besondere Einstellungen an alle Rechner im Netzwerk 192.168.0.0/24 freigeben möchten, lautet die Zeile in der Konfigurationsdatei:

```
/var/test 192.168.0.0/24
```

Als Netzwerk können Sie Werte in folgenden Formaten angeben:

- ▶ Eine einzelne IP-Adresse, um einem bestimmten Host den Zugriff zu erlauben. Beispiel: 192.168.0.7
- ▶ Eine Netzwerkadresse in CIDR-Schreibweise, wahlweise auch mit vollständiger Teilnetzmaske. Beispiele: 192.168.0.0/24 beziehungsweise 192.168.0.0/255.255.255.0
- ▶ Einen vollständigen Host- und Domainnamen (FQDN) wie etwa `nfscient.test.local`
- ▶ Einen Domainnamen mit Platzhaltern (* für beliebig viele Zeichen, ? für genau ein beliebiges Zeichen), um definierten Teilen einer (Sub-)Domain den Zugriff zu gestatten. Ein einzelnes * steht für alle Rechner, die den Host überhaupt erreichen können. Beispiel: `*.test.local`
- ▶ Eine NIS-Gruppe mit vorangestelltem @-Zeichen. Beispiel: `@netusers`

Einige der wichtigsten Optionen, die durch Komma getrennt in Klammern angegeben werden können, sind:

- ▶ `rw|ro`
Lese- und Schreibzugriffe erlaubt (`rw`) oder nur Lesezugriffe erlaubt (`ro`; dies ist Standard).

- ▶ `sync|async`
`sync` liefert Daten erst aus, nachdem sie auf den lokalen Datenträger geschrieben wurden; dies ist langsamer, aber sicherer. `async` liefert sie dagegen stets sofort bei Anfrage, was potenziell unsicher, aber schneller ist.
- ▶ `root_squash|no_root_squash`
 Wenn der Remote-Benutzer `root` ist, werden seine Zugriffe auf den anonymen Benutzer `nobody` umgelenkt (`root_squash`). Netzwerkübergreifende `root`-Zugriffe ermöglicht dagegen `no_root_squash`; dies sollte aus Sicherheitsgründen nur in Ausnahmefällen erfolgen.

Das folgende Beispiel gibt den Verzeichnisbaum `/home` synchron mit Lese- und Schreibrechten frei:

```
/home 192.168.0.0/24(rw, sync)
```

Eine verbreitete Konvention besteht darin, für NFS-Freigaben ein eigenes Verzeichnis unterhalb des Hauptverzeichnisbaums zu erstellen; meistens wird dieses Verzeichnis `/exports` oder `/export` genannt. In der Regel wird es nicht als Ganzes freigegeben, sondern enthält einzelne Unterverzeichnisse, die separat exportiert werden, sodass Sie die passenden Verzeichniszugriffsrechte jeweils individuell anpassen können.

Wenn Sie auf ein freigegebenes NFS-Verzeichnis eines anderen Rechners zugreifen möchten, müssen Sie es mithilfe des `mount`-Befehls in Ihren Verzeichnisbaum importieren. Das funktioniert schematisch folgendermaßen:

```
mount -t nfs Host:/Freigabepfad /lokaler/Pfad
```

Die Option `-t nfs` gibt den Dateisystemtyp NFS an. Das zu mountende Verzeichnis beginnt mit dem Namen des gewünschten Hosts, der durch einen Doppelpunkt vom entfernten Pfad getrennt wird; die letzte Angabe ist wie gehabt der Mountpoint, unter dem der importierte Pfad eingebunden werden soll.

Ein konkretes Beispiel sieht etwa so aus:

```
# mount -t nfs nfserver:/exports/homes /remotehome
```

Anstelle des temporären Mountings per Kommandozeilenbefehl können Sie NFS-Freigaben auch in die `/etc/fstab` aufnehmen. Dies sollten Sie allerdings nur tun, wenn diese auf ständig verfügbaren Server-Hosts liegen. Hier ein entsprechender Beispieleintrag:

```
nfserver:/exports/public /pub nfs defaults 0 0
```

NFS-Server und -Client mit YaST einrichten

Wählen Sie in YaST NETZWERKDIENTSTE • NFS-SERVER, um den NFS-Server auf Ihrem Rechner zu konfigurieren. Auf der ersten Seite geht es um das Startverhalten des Dienstes und einige Grundeinstellungen (siehe Abbildung 14.1):

- ▶ **STARTEN**
Aktiviert den Server sofort und startet ihn in Zukunft automatisch beim Booten, während NICHT STARTEN ihn beendet, auf Dauer deaktiviert und den Dialog verlässt.
- ▶ **FIREWALL-PORT ÖFFNEN**
Sorgt dafür, dass die für NFS notwendigen TCP/UDP-Ports als Ausnahme in den Firewall-Regeln betrachtet werden. Unter FIREWALL-DETAILS können Sie gegebenenfalls festlegen, für welche Netzwerkschnittstellen dies gelten soll.



Abbildung 14.1 YaST-Optionen für den Start des NFS-Servers

- ▶ **NFSV4 AKTIVIEREN**
Diese Option aktiviert eine neuere NFS-Version, die mehr Stabilität und bessere Performance bietet. Sie ist standardmäßig aktiviert, da sie inzwischen von den meisten NFS-Clients unterstützt wird. Sie brauchen die Option nur zu deaktivieren, falls Sie wissen, dass sich in Ihrem Netzwerk inkompatible Clients befinden. Falls Sie NFSv4 verwenden, müssen Sie zusätzlich einen Domänennamen eingeben.

► GSS-SICHERHEIT AKTIVIEREN

Schalten Sie diese Option ein, wenn Sie möchten, dass der Server und die Clients sich über die GSS-Bibliothek authentifizieren müssen. Dies erhöht die NFS-Sicherheit, erfordert aber, dass Kerberos und `gssapi` (`nfs-utils` in einer neueren Version als 1.0.7) installiert werden.

Sobald Sie auf WEITER klicken (steht nur zur Verfügung, wenn Sie eingestellt haben, dass der NFS-Server grundsätzlich gestartet werden soll), gelangen Sie in den Dialog ZU EXPORTIERENDE VERZEICHNISSE. Dieser Dialog wird in Abbildung 14.2 gezeigt. Hier können Sie gemäß der oben erläuterten Exportlogik diejenigen Verzeichnisse auswählen, die freigegeben werden sollen.



Abbildung 14.2 Verzeichnisse in YaST für den NFS-Export auswählen

Klicken Sie auf VERZEICHNIS HINZUFÜGEN, um ein lokales Verzeichnis zur Freigabe auszuwählen. Geben Sie den absoluten Pfad des Verzeichnisses ein, beispielsweise `exports/public`, oder klicken Sie DURCHSUCHEN an, um einen Auswahl-dialog aufzurufen. Sobald Sie das gewählte Verzeichnis mit OK bestätigen, folgt ein Dialogfeld zum Festlegen der Rechner beziehungsweise Netze, die auf die Freigabe zugreifen dürfen. Geben Sie unter RECHNER-WILDCARD einen Wert aus den oben erläuterten Gruppen ein; die Voreinstellung ist * (beliebige Rechner). Die voreingestellten Optionen sind `ro,root_squash,sync`; geben Sie eventuell andere Werte ein.

Sobald Sie OK betätigt haben, erscheint die neue Freigabe im oberen Feld des Hauptdialogs. Unten werden die Berechtigungen für den jeweils ausgewählten Export angezeigt. Sie können jeweils BEARBEITEN wählen, um die entsprechenden Einstellungen zu ändern. Wenn Sie dieselbe Freigabe für weitere Rechner oder Netze bereitstellen möchten, klicken Sie auf HOST HINZUFÜGEN.

Auch für die Konfiguration des NFS-Clients auf Ihrem Rechner besitzt YaST eine Option; sie dient dazu, automatisch */etc/fstab*-Einträge für entfernte Freigaben hinzuzufügen. Wählen Sie NFS-CLIENT in der Kategorie NETZWERKDIENTSTE.

Mit HINZUFÜGEN können Sie sich einen NFS-Export von einem anderen Rechner aussuchen. Geben Sie unter HOSTNAME DES NFS-SERVERS den gewünschten Serverrechner ein, oder versuchen Sie, ihn mittels WÄHLEN aus einer Liste auszuwählen (dies gelingt nicht immer, weil möglicherweise nicht alle im Netzwerk verfügbaren NFS-Server aufgeführt werden). Unter ENTFERNTES DATEISYSTEM wird der gewünschte Pfad angegeben; auch hier können Sie die Option WÄHLEN ausprobieren. Den MOUNTPUNKT (LOKAL) können Sie natürlich immer mittels DURCHSUCHEN festlegen; innerhalb des Auswahldialogs besteht auch die Möglichkeit, per Klick auf eine Schaltfläche ein neues Verzeichnis zu erstellen. Unter Optionen können Sie beliebige *mount*- beziehungsweise */etc/fstab*-Optionen eingeben; eine Liste finden Sie in Kapitel 9, »Datenträger und Dateisysteme«. Abbildung 14.3 zeigt den fertig ausgefüllten HINZUFÜGEN-Dialog.



Abbildung 14.3 Der fertig ausgefüllte YaST-Dialog zum Hinzufügen eines NFS-Eintrags in »*/etc/fstab*«

Auch für den Client ist die oben beschriebene Option NFSV4 SHARE vorausgewählt, und mithilfe der Option FIREWALL-PORT ÖFFNEN können Sie wie gehabt dafür sorgen, dass die Firewall die entsprechende Kommunikation zulässt.

14.1.2 Samba

Der Dateiserver *Samba* ist ein seit vielen Jahren erfolgreiches und weitverbreitetes Open-Source-Projekt; inzwischen steht Version 4 mit zahlreichen neuen Möglichkeiten kurz vor der Fertigstellung. Samba ermöglicht die Freigabe von Verzeichnissen und Druckern an Windows-Clients. Der Name leitet sich von *SMB* (Server Message Blocks) ab, dem File Sharing-Protokoll der Windows-Betriebssysteme.

Wenn Sie Samba benutzen möchten, muss der Samba-Server `smbd` laufen. Die Konfiguration des Servers ist in der Datei `/etc/smb.conf` festgelegt. Sie verfügt über Dutzende möglicher Konfigurationsbefehle, die auf der Manpage `man smb.conf` dokumentiert sind. Die Datei ähnelt eher einer Windows-*ini*-Datei als einer typischen UNIX-Konfigurationsdatei: Unter verschiedenen Abschnitten, deren Namen in eckigen Klammern stehen, werden die Konfigurationsbefehle zeilenweise als `Name=Wert`-Paare gesetzt. Hier wird zunächst kurz die Syntax der `smb.conf`-Datei anhand von Beispielen erläutert; anschließend wird die Samba-Konfiguration mit YaST vorgestellt.

smb.conf

Die folgende kurze `smb.conf`-Datei ermöglicht beliebigen Windows-Benutzern den anonymen Vollzugriff auf das Verzeichnis `/exports/public` und auf einen Drucker vom Typ HP LaserJet:

```
[global]
netbios name = Server
workgroup    = Workgroup
hosts allow  = 192.168.0.
log file     = /var/log/samba/log.%m
printing    = cups
guest account = nobody
security     = share

[public]
comment      = Serververzeichnis für alle
path         = /exports/public
public       = yes
writable     = yes
printable    = yes
browseable   = yes
create mask  = 0775

[hp1j]
comment      = HP LaserJet
```

```
printer name = hplj
writable     = yes
public      = yes
printable   = yes
browseable  = yes
```

Die Gruppe `[global]` enthält zunächst die allgemeine Konfiguration:

```
netbios name = Server
```

Der `netbios name` ist der Rechnername, der in der Windows-Netzwerkumgebung angezeigt werden soll. Wenn Sie ihn weglassen, wird stattdessen der TCP/IP-Hostname exportiert.

```
workgroup    = Workgroup
```

Die `workgroup` gibt an, in welcher Windows-Netzwerk-Arbeitsgruppe der Samba-Server auftauchen soll. Alternativ zur `Workgroup` können Sie auch eine `domain`-Direktive einfügen, sodass der Samba-Server Bestandteil einer Windows-Domäne, eventuell sogar Domain-Controller (zentraler Anmeldeserver) wird. Solche Konfigurationsoptionen gehen allerdings über den Rahmen dieses Kapitels hinaus.

```
hosts allow  = 192.168.0.
```

`hosts allow` enthält eine Liste von IP-Adress-Schemata, die auf den Server zugreifen dürfen – in diesem Beispiel das LAN 192.168.0.0/24.

```
log file     = /var/log/samba/log.%m
```

Über alle Samba-Zugriffe soll eine Logdatei geführt werden. Die Angabe `%m` hinter dem Namen des `log file` gibt an, dass pro Monat eine neue Logdatei angelegt werden soll, die mit der Nummer des laufenden Monats endet; neben `%m` sind auch alle anderen `date`-Kürzel erlaubt (siehe Kapitel 4, »Mit der Shell arbeiten«).

```
printing     = cups
```

Für Druckdienste wird CUPS verwendet (siehe Kapitel 6, »Wichtige Hardware konfigurieren«) – etwas anderes kommt heutzutage praktisch nicht mehr infrage.

```
guest account = nobody
```

Wenn Gäste zugreifen, werden ihre Aktionen unter dem lokalen Benutzerkonto `nobody` ausgeführt.

```
security     = share
```

`security` gibt die Windows-Freigabeebene an: `share` steht für Share Level Security (alle Benutzer dürfen mit einem gemeinsamen, freigabespezifischen Passwort zugreifen), während `user` für User Level Security (UNIX-ähnlicher benutzer-spezifischer Zugriff mit Benutzername und Passwort) verwendet wird.

Unter dem selbst gewählten Namen `[public]` wird das Verzeichnis `/exports/public` freigegeben. Das Setzen sämtlicher Optionen auf den Wert `yes` bedeutet, dass der vollständige Zugriff auf das Verzeichnis möglich ist. Die `create mask` gibt an, mit welchen Benutzerrechten neu angelegte Dateien gespeichert werden sollen. Ähnliche Einstellungen gelten auch für den Drucker.

Die Samba-Distribution enthält neben dem Server übrigens auch noch einen SMB-Client, mit dessen Hilfe Sie umgekehrt auf Windows-Freigaben (oder natürlich auch andere Samba-Freigaben) im Netzwerk zugreifen können. Er wird über den Konsolenbefehl `smbclient` aufgerufen und bildet eine Art eigene Shell. Zunächst können Sie testen, welche Freigaben ein bestimmter Windows-Host überhaupt zur Verfügung stellt:

```
$ smbclient -L winbox
Sharename      Type          Comment
-----
IPC$           IPC          Remote-IPC
post          Disk
```

Sie erhalten eine Liste aller freigegebenen Ordner auf dem angegebenen Host, in diesem Fall `winbox`. Auf die in der Liste gezeigte Freigabe `post` vom Typ `Disk` (Festplatte; eigentlich Verzeichnis) können Sie folgendermaßen zugreifen:

```
$ smbclient \\\winbox\post
```

Der NetBIOS-Freigabename des Verzeichnisses lautet eigentlich `\\winbox\pub`. Da die UNIX-Shell den Backslash aber für Escape-Sequenzen einsetzt, müssen Sie je einen durch zwei von ihnen darstellen. Anschließend können Sie eine Reihe verschiedener Befehle eingeben, um Dateioperationen an den Inhalten der Freigabe vorzunehmen – viele von ihnen entsprechen Windows-Konsolenbefehlen wie `dir` (Verzeichnisinhalt, äquivalent zu `ls`) oder `del` (Löschen, wie `rm`). Der Befehl `h` listet alle verfügbaren Befehle auf. Mittels `quit`, `exit` oder `[Strg]+[D]` können Sie den `smbclient` beenden.

Noch komfortabler ist es allerdings, eine entfernte SMB-Freigabe einfach zu mounten und wie ein lokales Verzeichnis zu verwenden. Dazu steht der Dateisystemtyp `smbfs` zur Verfügung. Um das oben angegebene Verzeichnis `post` auf dem Rechner `winbox` zu mounten, können Sie beispielsweise Folgendes eingeben:

```
# mount -t smbfs //winbox/post /smbpost
```

Samba-Konfiguration mit YaST

Wenn Sie Samba komfortabel über YaST einrichten möchten, wählen Sie **SAMBA-SERVER** im Bereich **NETZWERKDIENTSTE**. Auf der ersten Seite des Dialogs werden Sie aufgefordert, den Namen der Windows-Arbeitsgruppe oder -Domain einzugeben, in der der Samba-Server aktiv werden soll. Auf der nächsten Seite wird ausgewählt, ob Samba als Primary Domain Controller (PDC), also als zentraler Windows-Anmeldeserver, arbeiten soll oder nicht. Wenn es nur um einfache Dateifreigaben geht, sollten Sie **KEIN DOMÄNEN-CONTROLLER** wählen.

Nach diesen beiden Einleitungsseiten folgt die eigentliche Samba-Konfiguration. Sie verfügt über die drei Registerkarten **START**, **FREIGABEN** und **IDENTITÄT**. Im Einzelnen werden auf ihnen folgende Optionen eingestellt:

▶ **START**

Hier können Sie wie bei anderen Netzwerkdiensten regeln, ob Samba automatisch oder manuell gestartet werden soll; auch die Firewall-Einstellungen können wie üblich vorgenommen werden.

▶ **FREIGABEN**

Auf dieser Registerkarte wird detailliert geregelt, welche Verzeichnisse und Dienste der Samba-Server veröffentlichen soll. Jeder Eintrag in der Freigabetabelle besteht aus fünf Spalten:

- ▶ **STATUS** (**AKTIVIERT** oder **DEAKTIVIERT**) bestimmt, ob die Freigabe zurzeit aktiv ist oder nicht. Klicken Sie auf **STATUS WECHSELN**, um dies zu ändern.
- ▶ **NUR-LESEN** (**JA** oder **NEIN**) legt fest, ob die Freigabe nur gelesen oder auch verändert werden darf. Den Wert in dieser Spalte können Sie nur ändern, indem Sie auf **BEARBEITEN** klicken und die Option `read only` modifizieren.
- ▶ **NAME** bestimmt den Freigabenamen, unter dem die jeweilige Freigabe im Netzwerk zur Verfügung steht. Namen, die mit einem Dollarzeichen enden, sind keine Verzeichnisfreigaben, sondern Systemdienste. `print$` stellt beispielsweise Druckertreiber für den Zugriff auf Netzwerkdrucker bereit.
- ▶ **PFAD** ist der lokale Verzeichnispfad der Freigabe.
- ▶ Die **ANMERKUNG** wird als zusätzliche Beschreibung exportiert und erscheint auf Windows-Rechnern in der Detailansicht der Freigaben.

Klicken Sie auf **HINZUFÜGEN**, um eine neue Freigabe zu erstellen, oder auf **BEARBEITEN**, um eine bestehende zu ändern. Geben Sie bei neuen Freigaben einen Freigabenamen und eine Beschreibung ein. Wählen Sie das **VERZEICHNIS** oder den **DRUCKER** zum Freigeben aus. Auch die Option **NUR-LESEN** können Sie in diesem Dialog festlegen. Abbildung 14.4 zeigt die Erstellung der Freigabe für `/exports/public`.

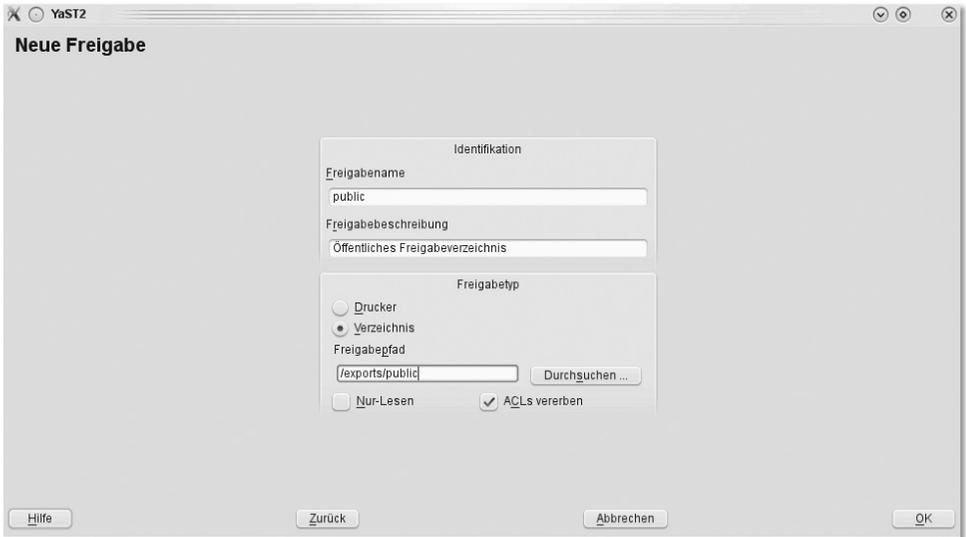


Abbildung 14.4 Erstellen einer neuen Samba-Freigabe

Im Dialog FREIGABE BEARBEITEN (siehe Abbildung 14.5) können Sie mittels HINZUFÜGEN beliebig viele weitere Samba-Konfigurationsdirektiven setzen. Einen ersten Anhaltspunkt bieten die */etc/smb.conf*-Beispiele weiter oben; Details entnehmen Sie bitte der *smb.conf*-Manpage.

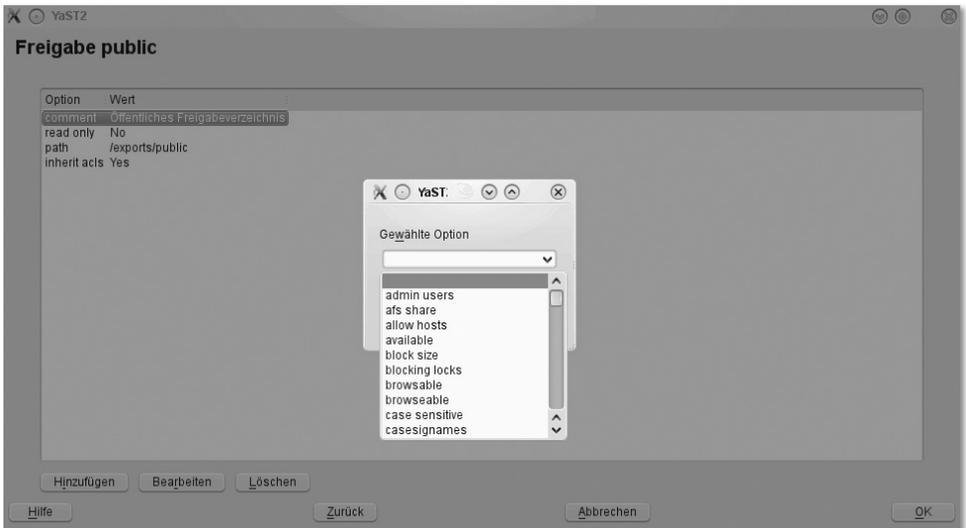


Abbildung 14.5 Die Optionen einer Samba-Freigabe bearbeiten; auch der mittels »Hinzufügen« geöffnete Dialog zum Auswählen einer Konfigurationsdirektive wird gezeigt.

- ▶ IDENTITÄT: Hier können Sie die bei der Erstkonfiguration des Samba-Servers (siehe oben) vorgenommenen Einstellungen für ARBEITSGRUPPE/DOMAIN und DOMAIN-CONTROLLER-Eigenschaft ändern.

14.2 Verzeichnisdienste

Heutzutage verwenden immer mehr Firmen eine Lösung für ihre unternehmensweite Authentifizierungsverwaltung, die man als *Verzeichnisdienst* bezeichnet. Dabei handelt es sich um einen Serverdienst, der die Daten sämtlicher Netzwerke, Rechner, Benutzer und Gruppen des Unternehmens speichert und bei jeder Netzwerkanmeldung überprüft, ob die aktuelle Rechner-Benutzer-Kombination die erforderliche Berechtigung besitzt. Ein Verzeichnisdienst ist praktisch, weil er einmal zentral eingerichtet wird, und bietet mehr Leistungsfähigkeit und Sicherheit als ein Gestrüpp aus lokalen Berechtigungen für einzelne Rechner.

Konkrete Verzeichnisdienste gibt es wie Sand am Meer: NIS beziehungsweise NIS+ von Sun (früher yp oder Yellow Pages genannt), Novell Directory Services (NDS) beziehungsweise eDirectory, OpenLDAP und Microsoft Active Directory sind nur einige von ihnen. Bis auf die Sun-Eigenentwicklung NIS(+) basieren all diese Verzeichnisdienste auf der abgespeckten Variante der X.500-Spezifikation, auf die mithilfe von LDAP (Lightweight Directory Access Protocol) zugegriffen werden kann.

Im Übrigen gehört auch das im vorigen Kapitel besprochene Domain Name System zu den Verzeichnisdiensten, es wird aber nicht für Anmelde- und Verwaltungsdienste aller Art, sondern nur für die Namens- und Adressauflösung verwendet.

In diesem Abschnitt werden die beiden mit openSUSE gelieferten Verzeichnisdienstserver NIS und OpenLDAP besprochen.

14.2.1 NIS

NIS wurde wie das verwandte, oben besprochene NFS von Sun entwickelt und basiert ebenfalls auf RPC. Die Datenbanken eines NIS-Servers sind in der Lage, lokale Benutzer- und Gruppenverwaltungsdateien wie */etc/passwd*, */etc/group* oder */etc/hosts* im Netzwerk zu veröffentlichen und auf diese Weise als zentralen Anmelde- und Konfigurationsdienst bereitzustellen.

Zu diesem Zweck legt der NIS-Server sogenannte Maps an, die diese Daten leseoptimiert bereitstellen. Leseoptimiert bedeutet, dass für dieselbe Datei manchmal mehrere Maps eingerichtet werden, um die Suche nach unterschiedlichen

Aspekten zu beschleunigen. Für */etc/hosts* werden beispielsweise zwei Maps erstellt: *hosts.byname* für die Suche nach der IP-Adresse zu einem gegebenen Hostnamen und *hosts.byaddr* für den umgekehrten Fall. Entsprechend gibt es für Userdaten (*/etc/passwd*) die beiden Maps *passwd.byname* (nach Benutzernamen) und *passwd.byuid* (nach numerischen User-IDs).

Der NIS-Server heißt intern *ypserv*, der Client *ybind*. Konfiguriert werden sie in den Dateien */etc/ypserv.conf* beziehungsweise */etc/yp.conf*, aber am einfachsten lassen sie sich über YaST einrichten. Dies wird im Folgenden erläutert, mitsamt gelegentlichen Verweisen auf manuelle Konfigurationsmöglichkeiten.

Einen NIS-Server konfigurieren

Wenn Sie Ihren Rechner als NIS-Server konfigurieren möchten, wählen Sie NETZWERKDIENTE • NIS-SERVER. Zuerst werden Sie aufgefordert, sich zwischen einem NIS MASTER SERVER und einem NIS SLAVE SERVER zu entscheiden. Wie bei anderen Serverdiensten, etwa dem im vorigen Kapitel vorgestellten DNS, besteht der Unterschied darin, dass der Master-Server selbst Daten veröffentlicht, während der Slave sie aus Sicherheits- und Performance-Gründen spiegelt (repliziert). Einen Slave-Server können Sie sinnvollerweise natürlich nur dann einrichten, wenn Sie bereits einen Master betreiben. Deshalb konzentriert sich die nachfolgende Beschreibung auf einen NIS Master Server.

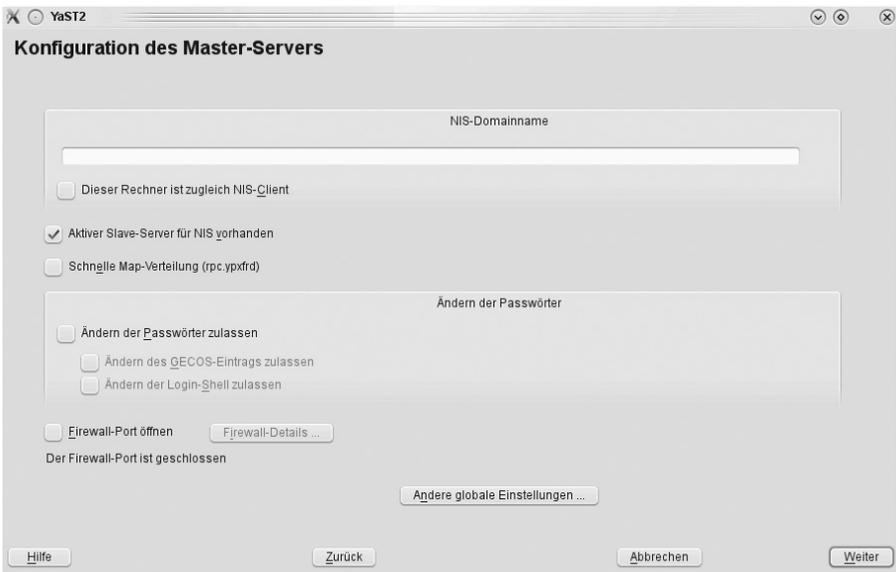


Abbildung 14.6 Einrichten eines NIS Master Servers mit YaST

Auf der ersten Seite, KONFIGURATION DES MASTER-SERVERS, die in Abbildung 14.6 zu sehen ist, werden folgende Einstellungen vorgenommen:

► NIS DOMAINNAME

Alle Rechner, die Anmeldedaten von demselben NIS-Server abfragen, verwenden zu diesem Zweck einen gemeinsamen Namen, den NIS-Domainnamen. Er ist völlig unabhängig vom DNS-Domainnamen, kann aber natürlich aus Gründen der Übersichtlichkeit identisch benannt werden. Neben der Konfiguration in YaST können Sie ihn auf NIS-Servern wie -Clients mithilfe des Kommandos `domainname` steuern. Ohne Argumente zeigt dieser Befehl den aktuellen NIS-Domainnamen an; um einen neuen zu setzen, müssen Sie den gewünschten Domainnamen dahinterschreiben. Das folgende Beispiel setzt den NIS-Domainnamen `test.local`:

```
# domainname test.local
```

► DIESER RECHNER IST ZUGLEICH NIS-CLIENT

Kreuzen Sie diese Option an, wenn der Host selbst auch als Client auf seinen eigenen NIS-Serverdienst zugreifen soll. Um eine funktionierende zentrale Anmeldeverwaltung mit NIS zu gewährleisten, ist dies erforderlich.

► AKTIVER SLAVE-SERVER FÜR NIS VORHANDEN

Aktivieren Sie dieses Kontrollkästchen, wenn im lokalen Netzwerk, genauer gesagt innerhalb derselben NIS-Domain, auch ein NIS-Slave-Server betrieben wird. In kleinen bis mittleren Netzwerken ist dies zumeist nicht erforderlich.

► SCHNELLE MAP-VERTEILUNG (RPC.YPXFDR)

Diese Option startet einen zusätzlichen Dienst, der die Maps schneller auf die Slaves überträgt. Wenn keine Slaves vorhanden sind, ist diese Einstellung uninteressant.

► ÄNDERN DER PASSWÖRTER ZULASSEN

Wenn Sie dies ankreuzen, dürfen Benutzer auf den Rechnern, die die Authentifizierung per NIS-Client durchführen, wie bei lokalen Benutzerkonten ihre Passwörter ändern.

► ÄNDERN DES GECOS-EINTRAGS ZULASSEN

Diese Option funktioniert nur, wenn auch die Passwortänderung erlaubt wurde. Sie sorgt dafür, dass Benutzer auch das Info-Feld ihres `/etc/passwd`-Eintrags ändern dürfen, in dem meist der vollständige Name steht (siehe die Beschreibung der `/etc/passwd`-Einträge in Kapitel 3, »Systemrundgang und Grundkonfiguration«).

- ▶ **ÄNDERN DER LOGIN-SHELL ZULASSEN**
Auch die Login-Shell eines Users wird standardmäßig durch einen */etc/passwd*-Eintrag festgelegt, sodass sie bei einem NIS-Login geändert werden kann (siehe ebenfalls Kapitel 3).
- ▶ **FIREWALL-PORT ÖFFNEN**
Damit der NIS-Server in Ihrem Netzwerk verfügbar ist, müssen Sie diese Option wie üblich aktivieren, um die Firewall entsprechend freizuschalten.
- ▶ **ANDERE GLOBALE EINSTELLUNGEN**
An dieser Stelle werden einige Detailinstellungen vorgenommen. Unter YP QUELLVERZEICHNIS können Sie bei Bedarf ein anderes Verzeichnis als den Standard */etc* auswählen; da die zu veröffentlichenden Authentifizierungs- und Konfigurationsdateien sich in aller Regel dort befinden, ist dies meist unnötig. Wenn Sie unter PASSWÖRTER ZUSAMMENFÜHREN die Option JA wählen, werden die Daten aus */etc/passwd* und */etc/shadow* zu einer gemeinsamen Map kombiniert (Standard); Gruppen zusammenführen erledigt dasselbe für */etc/group* und */etc/gshadow* (falls *gshadow* existiert).

Die KLEINSTE UID und die KLEINSTE GID legen die jeweils niedrigsten ID-Nummern fest, die für neue Benutzer beziehungsweise Gruppen über NIS vergeben werden. Der Vorgabewert 500 sorgt dafür, dass sie den lokalen Benutzerkonten (ab 100) für gewöhnlich nicht in die Quere kommen.

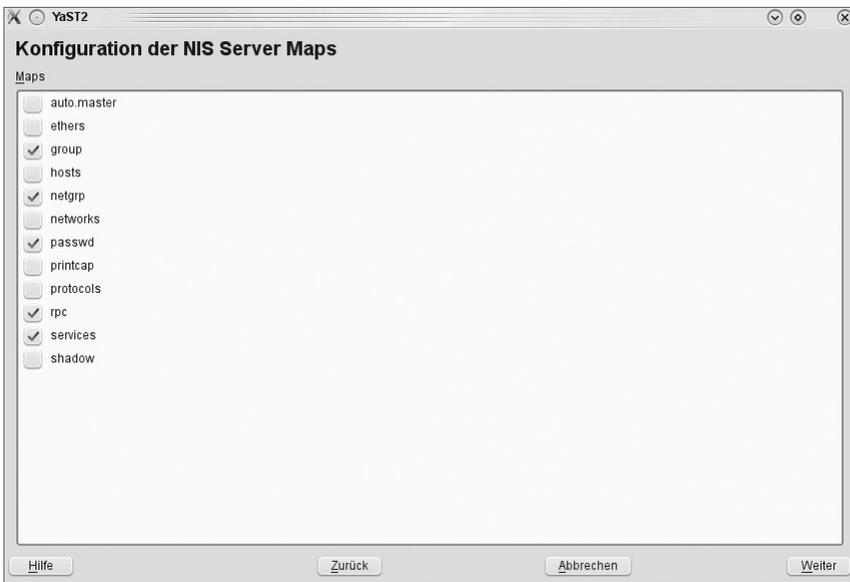


Abbildung 14.7 Auswahl der zu erstellenden NIS-Maps

Auf der nachfolgenden Seite, KONFIGURATION DER NIS SERVER MAPS (siehe Abbildung 14.7), legen Sie fest, welche Maps Sie überhaupt veröffentlichen möchten. Die wichtigsten sind:

- ▶ **ETHERS**
Die Datei */etc/ethers* enthält Zuordnungen zwischen den MAC-Adressen von Ethernet-Karten und IP-Adressen.
- ▶ **GROUP**
Veröffentlicht */etc/group* zum Nachschlagen der Gruppenzugehörigkeiten.
- ▶ **HOSTS**
Veröffentlicht */etc/hosts* mit den Zuordnungen von Hostnamen und IP-Adressen. Diese Maps brauchen Sie nur zu erstellen, wenn in Ihrem Netz keine Nameserver (siehe voriges Kapitel) aktiv sind.
- ▶ **PASSWD**
Diese Option erstellt die wahrscheinlich wichtigsten Maps, nämlich die Freigabe von */etc/passwd* mit den Benutzeranmeldedaten.
- ▶ **PROTOCOLS**
Erstellt eine Map aus den Transportprotokoll-Nummern in */etc/protocols*.
- ▶ **SERVICES**
Stellt die TCP-Port- beziehungsweise UDP-Dienst-Definitionen aus */etc/services* im Netzwerk zur Verfügung. Da die Standard-Portnummern normalerweise auf jedem Rechner gespeichert sind, benötigen Sie diese Map nur, wenn Sie spezielle Netzwerkanwendungen mit eigenen Portnummern betreiben.

Kreuzen Sie die gewünschten Maps an, und klicken Sie auf WEITER.

Die letzte Seite besitzt den Titel KONFIGURATION DES NIS SERVER QUERY HOSTS. Hier wird eingestellt, welche Netzwerke beziehungsweise Hosts auf den NIS-Server zugreifen dürfen. Klicken Sie auf HINZUFÜGEN, um einen neuen Adressbereich einzugeben, bestehend aus Teilnetzmaske und IP-Adresse, jeweils in dezimaler Punktschreibweise. Um etwa dem Netzwerk 192.168.0.0/24 den Zugriff zu gestatten, geben Sie die Netzmaske 255.255.255.0 und das Netzwerk 192.168.0.0 ein.

Standardmäßig sind zwei Einträge vorhanden: Netzmaske 255.0.0.0 und Netzwerk 127.0.0.0 sind für Loopback-Anfragen reserviert und werden zwingend benötigt. Der Eintrag mit 0.0.0.0 in beiden Feldern erlaubt jedem beliebigen Host den Zugriff; normalerweise sollten Sie ihn aus Sicherheitsgründen löschen und stattdessen etwas eingeben, was Ihrer Netzwerkinfrastruktur entspricht.

Wenn Sie diesen Dialog durch BEENDEN bestätigen, wird Ihr neuer NIS Master Server gestartet.

Wichtig: Egal wie viele lokale User-Accounts Sie in Zukunft nicht mehr lokal auf den Client-Rechnern, sondern auf dem NIS-Server verwalten – Sie sollten niemals auf die Idee kommen, das `root`-Konto auszulagern. Schließlich benötigen Sie eine Möglichkeit, sich anzumelden, wenn es Netzwerkprobleme gibt.

Den NIS-Client konfigurieren

Auf den Rechnern, die künftig Anmeldedaten vom NIS-Server beziehen sollen, muss als Nächstes der zugehörige LDAP-Client-Dienst konfiguriert werden. Dazu dient der YaST-Dialog **NETZWERKDIENTE • LDAP-CLIENT**. Hier brauchen Sie nicht sonderlich viel einzustellen; der Dialog besteht nur aus einer Seite und wird in Abbildung 14.8 gezeigt.

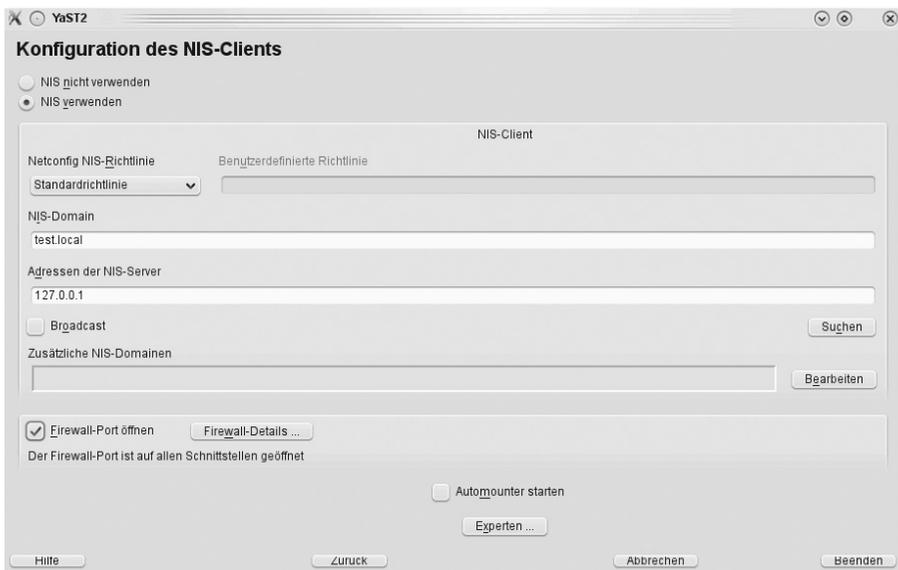


Abbildung 14.8 Den NIS-Client mit YaST einrichten

Im Einzelnen stehen hier folgende Einstellmöglichkeiten zur Verfügung:

- ▶ **NIS NICHT VERWENDEN | NIS VERWENDEN**
Damit legen Sie grundsätzlich fest, ob der Rechner als NIS-Client fungieren soll.
- ▶ **AUTOMATISCHES EINRICHTEN (MIT DHCP) | STATISCHE ADRESSEN EINRICHTEN**
Bestimmt, ob der Client die Informationen über die NIS-Server per DHCP beziehen soll oder ob Sie sie manuell eingeben möchten. DHCP können Sie nur dann wählen, wenn der Host bereits als DHCP-Client eingerichtet wurde und wenn der zuständige DHCP-Server auch Informationen über NIS-Server

verteilt. Die Konfiguration eines DHCP-Servers wird weiter unten in diesem Kapitel besprochen. Falls Sie sich für DHCP entscheiden, werden die restlichen Optionen im Rahmen NIS-CLIENT inaktiv.

► **NIS-DOMAIN**

Geben Sie in dieses Feld den auf dem NIS-Server konfigurierten NIS-Domainnamen ein; im durchgehenden Beispiel dieses Buches also `test.local`. (Falls Sie sich erst jetzt zugeschaltet haben: Der NIS-Domainname ist unabhängig vom DNS-Domainnamen; es schadet aber nichts, beiden aus praktischen Gründen denselben Wert zuzuweisen.)

► **ADRESSEN DER NIS-SERVER**

Geben Sie die (diesmal DNS-)Domainnamen oder IP-Adressen der verfügbaren NIS-Server (Master und Slaves) durch Leerzeichen getrennt ein. Wenn Sie ihre Adressen nicht kennen, können Sie auf **SUCHEN** klicken. Schneller ist die Suche allerdings, wenn Sie ein Konsolenfenster öffnen und Folgendes eingeben:

```
# ypwhich
192.168.0.7
```

Die so ermittelten Adressen können Sie dann in das Feld eintragen – meist ist es auch möglich, sie mit der linken Maustaste zu markieren und mit der mittleren einzufügen.

► **BROADCAST**

Wenn Sie dieses Kontrollkästchen ankreuzen, werden die NIS-Anfragen per IP-Broadcasting in das gesamte Netzwerk versandt. Dies ist sicherheitstechnisch kaum zu verantworten, sodass Sie nur in Ausnahmefällen davon Gebrauch machen sollten.

► **ZUSÄTZLICHE NIS-DOMAINS**

Falls Sie in Ihrem Netzwerk mehrere NIS-Domains konfiguriert haben, können Sie sie angeben, um auch ihre Anmelde- und Konfigurationsinformationen zu erhalten.

► **AUTOMOUNTER STARTEN**

Aktivieren Sie diese Option, wenn die Home-Verzeichnisse der Benutzer sich nicht auf dem lokalen Rechner, sondern auf einem NFS-Server im Netzwerk befinden. Auf diese Weise wird das passende Verzeichnis bei der Anmeldung automatisch gemountet, sodass ein Benutzer auf jedem beteiligten Rechner dieselbe Arbeitsumgebung vorfindet.

► **EXPERTEN**

Hier können Sie einige Detailoptionen für `ypbind` vornehmen, wobei die Standardeinstellungen in der Regel in Ordnung sind. Wichtig ist dagegen wie üblich der Punkt **FIREWALL-PORT ÖFFNEN**.

Wenn Sie in der YaST-Benutzerverwaltung (siehe Kapitel 3, »Systemrundgang und Grundkonfiguration«) unter `OPTIONEN FÜR EXPERTEN • AUTHENTIFIZIERUNG UND BENUTZERQUELLEN` den Link NIS anklicken, gelangen Sie übrigens in denselben Dialog. Die Benutzeranmeldung per NIS wird mit anderen Worten gleich in einem Aufwasch erledigt.

Wenn die Client-Konfiguration abgeschlossen ist, können Sie überprüfen, ob alles funktioniert, indem Sie in einem Terminalfenster `yptest` eingeben. Sie sollten eine längere Liste bestandener Tests erhalten, unter anderem die NIS-Server sowie die verfügbaren Maps.

Zudem enthalten die lokalen Ausgaben der Dateien `/etc/passwd`, `/etc/group` und so weiter jeweils eine spezielle Zeile, die das Nachschlagen derjenigen Benutzer, die nicht lokal vorhanden sind, auf den NIS-Servern veranlasst. In `/etc/passwd` sieht diese Zeile beispielsweise folgendermaßen aus:

```
+:::~:::
```

Die Variante in `/etc/group` besitzt entsprechend weniger Doppelpunkte, weil diese Datei weniger Informationskategorien enthält.

14.2.2 OpenLDAP

Der OpenLDAP-Server ist die freie Implementierung eines Verzeichnisdienstes, der auf dem LDAP-Protokoll basiert. LDAP-Server sind leseoptimiert, da in der Regel mehr Informationen ausgelesen als geschrieben werden müssen. Diese Server können viel mehr leisten, als nur Anmeldeinformationen bereitzustellen, aber es würde an dieser Stelle zu weit führen, auf alle ihre Aspekte einzugehen. Dieser Abschnitt beschränkt sich daher im Wesentlichen auf den Einsatz von LDAP zur Benutzerverwaltung.

LDAP-Grundlagen

Ein LDAP-Verzeichnis enthält hierarchisch organisierte Informationen, den sogenannten LDAP Directory Information Tree (DIT). Sie werden in einem textbasierten Format namens LDIF (LDAP Interchange Format) eingetragen und dargestellt. Der Baum im LDIF-Format enthält verschachtelte Einträge im Format `Name=Wert`, wobei die Namen standardisierte Objekte sind. Für bestimmte Knoten innerhalb des Baums wie Benutzer, Geräte oder Netzwerke werden Attribute in der Schreibweise `Attributname: Wert` gespeichert, wobei die Attributnamen ebenfalls standardisiert sind.

Die zulässigen Elemente und Attribute in LDAP-Verzeichnissen werden durch das sogenannte *LDAP-Schema* festgelegt. Es enthält die Namen aller Objektklassen sowie ihre internen Nummern.

Die Wurzel eines solchen Informationsbaums besteht aus *dc*-Einträgen (Domain Component). Für das lokale Beispielnetzwerk in diesem Buch wäre der passende Wurzeleintrag `dc=test,dc=local`. Eine solche Kombination von Einträgen, die einen Knoten des Baums eindeutig kennzeichnet, wird als DN (Distinguished Name) des Eintrags bezeichnet. `dc=test,dc=local` ist somit der DN des Wurzeleintrags.

Die nächste Stufe unterhalb der Wurzel bilden optional sogenannte Organisationseinheiten (Organizational Units, daher *ou*-Einträge). Wichtige Beispiele sind `ou=devices` für Rechner und andere Geräte oder `ou=people` für Benutzer.

Innerhalb der Organisationseinheit `ou=people`, auf die sich der Rest dieses Abschnitts konzentriert, können Sie bei Bedarf untergeordnete Organisationseinheiten bilden, um beispielsweise Abteilungen darzustellen. In einem Softwareunternehmen könnte es zum Beispiel die Abteilungen `ou=developers`, `ou=sales`, `ou=finance` und `ou=management` geben. Bedingung ist diese Unterteilung allerdings nicht; in kleineren Firmen oder Netzwerken ist sie in der Regel unnötig.

Die einzelnen Personen werden durch einen *cn*-Knoten (Common Name) gekennzeichnet, dessen Wert in der Regel Vor- und Nachnamen abbildet. Der DN einer Person im `test.local`-Netzwerk wäre also beispielsweise:

```
cn=Sascha Kersken,ou=people,dc=test,dc=local
```

Für Personen (Objektklasse *person*) stehen zahlreiche Attribute zur Verfügung. Ein modernes, firmenweites Personenverzeichnis verwendet allerdings besser die von *person* abgeleitete Objektklasse *inetOrgPerson*, die unter anderem das Attribut *mail* für E-Mail-Adressen bereitstellt. Ein nicht besonders langer Personeneintrag könnte zum Beispiel wie folgt aussehen:

```
dn: cn=Sascha Kersken,ou=people,dc=test,dc=local
objectClass: inetOrgPerson
cn: Sascha Kersken
sn: Kersken
givenName: Sascha
l: Köln
o: Lingoworld IT Services
telephoneNumber: 0221-1234567
mail: sk@lingoworld.de
userPassword: {MD5}GzIxZVzrt6H3g+3fJ9JUyg==
```

Die einzelnen Attribute bedeuten Folgendes:

- ▶ `dn` (Distinguished Name)
Dieses Attribut bestimmt den bereits erwähnten, vollständigen Pfad des Eintrags im LDAP-Verzeichnisbaum.
- ▶ `objectClass`
Legt die Kategorie des Eintrags und damit die vorgeschriebenen sowie erlaubten Felder fest.
- ▶ `cn`
Dieses Attribut ist der Common Name des Knotens; bei Personen meist Vor- und Nachname.
- ▶ `sn` (Surname)
Gibt den Nachnamen an.
- ▶ `givenName`
Bezeichnet den Vornamen.
- ▶ `l` (Location)
Dieses Attribut enthält den Wohnort.
- ▶ `o` (Organization)
Enthält den Firmen- oder Institutionsnamen.
- ▶ `telephoneNumber`
Ist die Telefonnummer der Person (natürlich ist dies nicht meine echte Telefonnummer).
- ▶ `mail`
Bezeichnet die E-Mail-Adresse (diese ist allerdings korrekt).
- ▶ `userPassword`
Gibt in geschweiften Klammern den Verschlüsselungsalgorithmus und dahinter das verschlüsselte Passwort an.

Eine weitere nützliche Objektklasse sind User-Accounts (`posixAccount`); genau diese werden zur Authentifizierung verwendet. Wichtige Einträge sind hier beispielsweise `uid` (der Login-Name), `uidNumber` (die numerische User-ID), `homeDirectory` (das Home-Verzeichnis) oder `loginShell` (die Shell, die diesem User bei der Anmeldung präsentiert wird).

Bei Bedarf besteht die Möglichkeit, einem Knoten auch mehrere `objectClass`-Attribute zuzuweisen, um unterschiedliche Informationen dafür speichern zu können. Zudem kann jeder Knoten untergeordnete Knoten enthalten. Der DN wird dazu um den `cn` des übergeordneten Objekts ergänzt. Angenommen, Sie möchten den detaillierten Personeneintrag als Unterknoten des User-Accounts erstellen, dann lautet dessen DN beispielsweise folgendermaßen:

```
dn: cn=Sascha Kersken,uid=sascha,ou=people,dc=test,dc=local
```

Den OpenLDAP-Server in Betrieb nehmen

Die openSUSE-Distribution wird mit der neuesten Version des OpenLDAP-Servers geliefert, und in Version 11 gibt es auch endlich ein YaST-Modul, um diesen einzusetzen. Dennoch lohnt es sich, auch die manuelle Konfiguration zu kennen – schon allein, weil Sie bei einem gemieteten Root-Server selten die neueste openSUSE-Version enthalten. Deshalb wird hier die Handarbeit beschrieben; entsprechende grafische Einstellungen finden Sie unter NETZWERKDIENTSTE • LDAP-SERVER.

Der OpenLDAP-Serverdienst selbst heißt `slapd`; die Hauptkonfigurationsdatei ist `/etc/openldap/slapd.conf`.

Überprüfen Sie in YaST zunächst, ob LDAP-Server und -Client installiert sind. Das betreffende Schema heißt VERZEICHNISSESERVER (LDAP). `openldap2` ist der Server, während `yast2-ldap-server` die diversen Kommandozeilentools enthält. Sehr wichtig ist außerdem `pam_ldap`; dieses PAM-Modul ermöglicht die Benutzer- und Passwortverwaltung sowie den Log-in über den LDAP-Server.

Bearbeiten Sie nun die Konfigurationsdatei `slapd.conf` mit einem beliebigen Editor. Die gesamte Syntax wird auf der Manpage `slapd.conf(5)` erläutert. Hier geht es nur darum, den OpenLDAP-Server überhaupt zum Laufen zu bringen und ein grundlegendes Verzeichnis für die Benutzeranmeldung im `test.local`-Netzwerk anzulegen.

Die Datei verwaltet vor allem Zugriffsrechte für das LDAP-Verzeichnis. Ein spezielles Objekt namens `rootdn` besitzt eine universelle Berechtigung. Im Wesentlichen brauchen Sie nur zwei Konfigurationseinstellungen anzupassen: `suffix`, die Wurzel Ihres LDAP-Baums, und den besagten `rootdn`. Für die Domain `test.local` sehen die beiden Einträge beispielsweise folgendermaßen aus:

```
suffix      "dc=test,dc=local"
rootdn      "cn=Manager,dc=test,dc=local"
```

Das Suffix muss natürlich mit den Domain-Komponenten von `rootdn` übereinstimmen. Der Common Name `Manager` für den `rootdn` ist keine Vorschrift, aber eine weitverbreitete Konvention, an die Sie sich halten sollten.

Nach dieser Anpassung können Sie OpenLDAP starten:

```
# rcldap start
```

Nun geht es ans Anlegen von Verzeichniseinträgen. Zur OpenLDAP-Distribution gehören zahlreiche Kommandozeilentools, mit denen Sie dies tun können.

Am einfachsten ist es, Sie schreiben die Einträge für Ihre LDAP-Wurzel in eine Textdatei. Das folgende Beispiel legt die Domain `test.local` und die Organisationseinheit `people` an:

```
dn: dc=test,dc=local
dc: test
objectClass: dcObject
objectClass: organizationalUnit
ou: test.local
```

```
dn: ou=people,dc=test,dc=local
ou: people
objectClass: organizationalUnit
```

Geben Sie nun Folgendes ein, um den Inhalt dieser Datei (ein passender Dateiname wäre zum Beispiel `top.ldif`) in das LDAP-Verzeichnis einzutragen:

```
# slapadd -v -l top.ldif
```

Mit den Einträgen für die Benutzerkonten (`objectClass: posixAccount`) können Sie ebenso verfahren; oder machen Sie sich per Manpage mit Tools wie `ldapadd`, `ldapmodify` und `ldapsearch` vertraut. Das folgende Beispiel gibt mittels `ldapsearch` alle Datensätze Ihres LDAP-Verzeichnisses (bisher nur zwei, wenn Sie dieser Anleitung gefolgt sind) aus:

```
# ldapsearch -x -b "dc=test,dc=local" "(objectclass=*)"
```

`(objectclass=*)` ist in diesem Zusammenhang ein LDAP-Filter. Da alle LDAP-Einträge ein `objectclass`-Attribut besitzen, trifft dieser Filter auf jeden Knoten des Verzeichnisses zu.

Bevor Sie Benutzerdaten speichern, müssen Sie die Passwörter mit externen Tools wie `md5sum` manuell verschlüsseln, bevor Sie sie eintragen können.

Viel komfortabler als die Konsolen-LDAP-Tools sind natürlich grafische Clients. Einer der besten ist *phpLDAPadmin*; diese Software können Sie unter <http://phpldapadmin.sourceforge.net> herunterladen. Sie wird – ähnlich wie das in Kapitel 15, »LAMP«, vorgestellte MySQL-Administrations-Tool *phpMyAdmin* – als PHP-Anwendung auf einem Webserver installiert. Im Prinzip brauchen Sie sie nur in Ihr Apache-Server-Verzeichnis zu kopieren, die Konfigurationsdatei anzupassen und die betreffende URL im Browser zu öffnen.

Beachten Sie aber bitte, dass *phpLDAPadmin* nicht mit der PHP-Version funktioniert, die in openSUSE enthalten ist. Sie benötigen nämlich die nicht mitgelieferte PHP-Erweiterung `gettext` dafür. Bevor Sie *phpLDAPadmin* in Betrieb neh-

men, müssen Sie deshalb eine neue PHP-Version installieren – wie das geht, wird ebenfalls im nächsten Kapitel beschrieben.

Die Anmeldung über LDAP einrichten

Wenn Sie die Benutzerkonten auf Ihrem OpenLDAP-Server angelegt haben, können Sie die Rechner in Ihrem Netzwerk als LDAP-Clients konfigurieren. Wählen Sie dazu den Punkt LDAP-CLIENT im YaST-Bereich NETZWERKDIENTSTE.

In diesem Dialog können Sie folgende Einstellungen vornehmen:

- ▶ **BENUTZERAUTHENTIFIZIERUNG**
Wählen Sie aus, ob der LDAP-Client aktiviert werden soll. LDAP NICHT VERWENDEN startet ihn nicht, LDAP VERWENDEN benutzt LDAP unter anderem für die Benutzerverwaltung. LDAP VERWENDEN, JEDOCH ANMELDUNGEN DEAKTIVIEREN ermöglicht dagegen die Nutzung von LDAP für andere Dienste – etwa E-Mail- und Telefonverzeichnisse –, aber lässt keine LDAP-basierte Benutzeranmeldung zu.
- ▶ **ADRESSEN VON LDAP-SERVERN**
Geben Sie die IP-Adressen der LDAP-Server in Ihrem Netzwerk ein.
- ▶ **LDAP BASE DN**
Hier wird der DN Ihres LDAP-Wurzeleintrags eingegeben, im Fall des vorliegenden Gesamtbeispiels also `dc=test,dc=local`.
- ▶ **LDAP TLS/SSL**
Diese Option aktiviert verschlüsselte LDAP-Abfragen. Der Einsatz von TLS/SSL-Zertifikaten und -Verschlüsselungen wird im nächsten Kapitel anhand von HTTPS-Verbindungen mit Apache erläutert.
- ▶ **LDAP VERSION 2**
Kreuzen Sie dieses Feld an, um die Abwärtskompatibilität mit einem älteren LDAP-Server herzustellen – aktuelle OpenLDAP-Versionen verwenden LDAPv3.
- ▶ **AUTOMOUNTER STARTEN**
Falls sich die Home-Verzeichnisse der Benutzer auf einem Server im Netzwerk befinden, sorgt diese Option dafür, dass sie bei der Anmeldung automatisch gemountet werden.
- ▶ **ERWEITERTE KONFIGURATION**
Geben Sie hier LDAP-spezifische Besonderheiten ein. Wenn Sie den Server nach den obigen Informationen grundlegend eingerichtet haben, ist dies nicht erforderlich.

14.3 Weitere Dienste

Ihr openSUSE-System ist in der Lage, einige weitere Netzwerk-Serverdienste zu versehen, die zwar keine Datei- oder Verzeichnisdienst-Server sind, aber doch so elementar, dass sie hier besprochen werden sollen und nicht erst in den beiden nachfolgenden Internet-Kapiteln. Im Einzelnen handelt es sich um folgende Dienste:

- ▶ DHCP-Server
- ▶ Proxyserver
- ▶ SSH-Server
- ▶ Remote-Verwaltung (VNC-Server)

14.3.1 DHCP-Server

Mit DHCP (Domain Host Configuration Protocol) können Sie Rechnern in Ihrem Netzwerk automatisch IP-Adressen und andere TCP/IP-Konfigurationsoptionen zuweisen. Den DHCP-Client haben Sie bereits im vorigen Kapitel kennengelernt – sobald Sie für eine Netzwerkkarte ankreuzen, dass sie über DHCP konfiguriert werden soll, brauchen Sie so gut wie keine weiteren Einstellungen mehr vorzunehmen. Das Gegenstück ist der DHCP-Server (`dhcpd`), der ebenfalls zum Lieferumfang von openSUSE gehört.

Sie können den DHCP-Server bequem über YaST konfigurieren. Wählen Sie dazu den Eintrag `DHCP-SERVER` unter `NETZWERKDIENTE`. Bei der Erstkonfiguration startet YaST den `DHCP-SERVER-WIZARD`, der Sie Schritt für Schritt durch die Konfiguration leitet.

Auf der ersten Seite werden mittels `HINZUFÜGEN` die Netzwerkschnittstellen ausgewählt, über die der DHCP-Server ansprechbar sein soll. In aller Regel werden Sie hier nur lokale, statische Schnittstellen (Ethernet-Karten, WLAN oder Ähnliches) auswählen und keine Wählverbindungen. Aktivieren Sie zusätzlich `FIREWALL FÜR GEWÄHLTE SCHNITTSTELLEN ÖFFNEN`, damit die Kommunikation überhaupt möglich ist.

Der zweite Teil des Dialogs trägt den Titel `Globale Einstellungen` – hier werden alle statischen Informationen eingetragen, die der DHCP-Server an die Clients im Netz weiterreichen soll. Im Einzelnen sind folgende Felder definiert (hier in der Reihenfolge aufgelistet, in der Sie sie mit der -Taste durchblättern können):

- ▶ `DOMAINNAME`
Falls ein fester Domainname verteilt werden soll, geben Sie ihn hier ein. Das Beispiel für dieses Buch lautet natürlich `test.local`.

▶ IP DES PRIMÄREN NAMESERVERS

Wenn Sie in Ihrem Netzwerk eigene Nameserver betreiben, können Sie hier die IP-Adresse des ersten von ihnen (meistens des Primary Master-Nameservers) eingeben. Nameserver, die in fremden Netzwerken (beispielsweise bei Ihrem Provider) liegen, können Sie nur im Zusammenhang mit einem Standardgateway (siehe unten) angeben; andernfalls sind die fraglichen Nameserver nicht erreichbar. Ein Beispiel: 192.168.0.2.

▶ IP DES SEKUNDÄREN NAMESERVERS

Tragen Sie hier gegebenenfalls die IP-Adresse eines zweiten Nameservers ein. Beispiel: 192.168.0.3.

▶ STANDARDGATEWAY (ROUTER)

Wenn der vorliegende Rechner, ein anderer Computer in Ihrem Netzwerk oder auch eine kompakte DSL-Router-Box¹ als Router ins Internet fungiert, geben Sie hier dessen IP-Adresse ein. Damit erhalten die DHCP-Clients automatisch Routing-Informationen und können den zentralen Internetzugang nutzen. Typisches Beispiel: 192.168.0.1.

▶ NTP-ZEITSERVER

Das Network Time Protocol (NTP) ermöglicht die automatische Synchronisation der Uhrzeit mit einem Zeitserver im Netz. Falls Sie einen Zeitserver in Ihrem Netzwerk betreiben oder über einen Router einen solchen im Internet erreichen können, geben Sie hier seine Adresse ein.

▶ DRUCKSERVER

Wenn Sie einen zentralen CUPS-Server oder einen Netzwerkdrucker mit eingebauter Serverfunktion betreiben, können Sie hier seine Adresse eingeben, um auch diese Information zu verbreiten.

▶ WINS-SERVER

WINS ist der Windows Internet Naming Service, das heißt der Namensdienst für Windows-Rechner. Falls zu Ihren DHCP-Clients Windows-Maschinen gehören, können Sie sie durch dieses Feld über den WINS-Server informieren.

▶ STANDARD-LEASING-ZEIT

DHCP teilt den Clients die dynamischen IP-Adressen durch ein Leasing-Verfahren zu. Nach dem Ende der maximalen Leasing-Dauer muss jeder Client erneut eine IP-Adresse anfordern; auf diese Weise werden die Adressen inaktiver Clients nach Gebrauch automatisch wieder frei. Den voreingestellten Wert von vier Stunden können Sie in einem kleinen, überschaubaren Netzwerk erhöhen – geben Sie beispielsweise »1« ein, und wählen Sie die Zeiteinheit TAGE.

1 Diese kleinen, leistungsfähigen Geräte versehen allerdings in der Regel auch den DHCP-Dienst selbst, sodass Sie auf keinem anderen Rechner einen DHCP-Server einzurichten brauchen.

Der dritte Konfigurationsschritt wird als *Dynamisches DHCP* bezeichnet. Hier wird die sogenannte DHCP-Range, das heißt der für die dynamische Verteilung verfügbare Adressbereich, eingestellt. Aktuelles Netzwerk und aktuelle Netzmaske können Sie nur wechseln, wenn Sie DHCP für mehrere Schnittstellen konfigurieren.

Wichtig sind dagegen die Felder ERSTE IP-ADRESSE und LETZTE IP-ADRESSE: Diese bestimmen den gesamten Pool von IP-Adressen, die Sie den Clients zuweisen können. Gehen Sie nicht zu großzügig mit den Adressressourcen Ihres Netzwerks um; denken Sie daran, dass Router, Nameserver und am besten auch andere Server-Rechner statische IP-Adressen benötigen, die Sie hier einplanen müssen. Geben Sie für das Netz 192.168.0.0/24 beispielsweise die erste Adresse 192.168.0.51 und die letzte Adresse 192.168.0.150 ein – somit können maximal 100 Clients mit IP-Adressen versorgt werden; die Bereiche von 192.168.0.1 bis 192.168.0.50 sowie 192.168.0.151 bis 192.168.0.254 bleiben für andere Zwecke frei.

Im gleichen Dialog können Sie die LEASING-ZEIT noch einmal detaillierter festlegen: STANDARD ist die normale Dauer einer Lease, während MAXIMUM die Zeitspanne bestimmt, nach der ein Client *gezwungen* wird, die IP-Adresse zurückzugeben und eine neue anzufordern.

Auf der letzten Seite schließlich wird nur noch eingestellt, ob der DHCP-Server automatisch oder manuell gestartet werden soll. Der automatische Start empfiehlt sich natürlich, zudem müssen Sie darauf achten, dass Sie den DHCP-Dienst auf einem Rechner betreiben, der normalerweise nicht ausgeschaltet wird – das heißt eher auf einem explizit dafür bestimmten Server-Rechner als nebenbei auf einem Arbeits-PC. Notfalls können Sie auch zwei oder mehr DHCP-Server einrichten, allerdings mit DHCP-Ranges, die einander nicht überschneiden.

Damit ist die Konfiguration abgeschlossen. Aus den oben genannten Konfigurationsbeispielen ergibt sich die folgende Original-DHCP-Konfigurationsdatei (*/etc/dhcpd.conf*):

```
option domain-name "test.local";
option domain-name-servers 192.168.0.2, 192.168.0.3;
option routers 192.168.0.1;
ddns-update-style none;
default-lease-time 86400;
subnet 192.168.0.0 netmask 255.255.255.0 {
    range 192.168.0.51 192.168.0.150;
    default-lease-time 86400;
    max-lease-time 345600;
}
```

Wenn Sie diese Informationen mit den weiter oben beschriebenen YaST-Angaben vergleichen, müssten Sie alles so weit verstehen – beachten Sie nur, dass Zeitangaben hier grundsätzlich in Sekunden vorgenommen werden.

14.3.2 Proxyserver

Wenn Sie für Ihr lokales Netzwerk einen privaten IP-Adressbereich wie 192.168.0.0/24 verwenden, ist ein direkter Internetzugang nicht ohne Weiteres möglich. Eine beliebte Lösung – neben dem im vorigen Kapitel im Rahmen der `iptables`-Konfiguration angesprochenen NAT-Router – ist ein *Proxyserver*. Proxies sind Stellvertreter, die im Namen von Clients Netzwerkkommunikation betreiben – sie geben die Client-Anfragen gegenüber den Servern als ihre eigenen aus und leiten die Antworten wiederum an die Clients weiter. Aus diesem Grund werden Proxyserver auch als *Application-Level Gateways* bezeichnet, weil ihre Dienste auf der Anwendungsschicht des TCP/IP-Referenzmodells stattfinden. Der Proxy-Client bekommt davon nichts mit, wenn er erst einmal konfiguriert wird.

Proxies können für die verschiedensten Netzwerkdienste eingerichtet werden – beispielsweise für HTTP und HTTPS (Web), FTP oder gar E-Mail. Webproxies erbringen neben der Bereitstellung einer indirekten Verbindung eine zusätzliche Dienstleistung: Sie können häufig angeforderte Webseiten zwischenspeichern (cachen), sodass nachfolgende Anfragen für dieselben Inhalte schneller vonstattengehen. Daher kann sich der Einsatz eines Webproxies sogar dann lohnen, wenn eigentlich ein direkter Internetzugang möglich wäre.

squid konfigurieren

Der beliebteste und verbreitetste Webproxy, der auch Bestandteil von openSUSE ist, heißt `squid`. Konfigurieren müssen Sie ihn manuell, indem Sie seine Konfigurationsdatei `/etc/squid/squid.conf` editieren – sie hilft Ihnen durch ausführliche Kommentare weiter, die wie üblich mit `#` beginnen. Grundsätzlich läuft er auch mit Standardeinstellungen, was Sie feststellen können, indem Sie als `root` das Kommando `ncsquid start` eingeben – der Proxy startet klaglos. Allerdings nützt das nichts, weil standardmäßig zum Beispiel keinerlei Zugriffe erlaubt sind.

Deshalb müssen Sie sich einige Konfigurationseinstellungen genauer ansehen:

- ▶ `http_port`
Legt den Port fest, über den der Proxy zu erreichen ist. Diesen Wert müssen Sie zusammen mit der IP-Adresse auf den Proxy-Clients eintragen. Die Voreinstellung ist 3128, was normalerweise so bleiben sollte – es sei denn, `squid` ersetzt einen Proxyserver mit einem anderen Port.

- ▶ `acl`
Diese Einstellung definiert Zugriffskontrolllisten (Access Control Lists), die bestimmen, wer überhaupt auf den Proxy zugreifen darf. Das Schema ist relativ komplex; hier nur ein Beispiel, das eine ACL namens `default` für beliebige Clients (`src`) aus dem kompletten Netzwerk `192.168.0.0/24` definiert:

```
acl default src 192.168.0.0/24
```

- ▶ `http_access`
Erlaubt oder verwehrt den Zugriff auf den HTTP-Proxy-Dienst auf der Basis von ACLs. Das folgende Beispiel gewährt der soeben definierten ACL `default` die Nutzung des HTTP-Proxys:

```
http_access allow default
```

Achtung: Schreiben Sie *niemals* in irgendeine Regel `allow all`! Dies gewährt Benutzern aus beliebigen Netzen den Zugriff auf Ihren HTTP-Proxy, der daraufhin für anonyme Zugriffe auf illegale Inhalte oder gar für Angriffe genutzt werden kann.

Für die anderen Dienste, für die `squid` als Proxy fungieren kann, gibt es adäquate Einstellungen.

- ▶ `cache_peer`
Kommuniziert mit anderen Proxy-Servern. Auf diese Weise können Sie beispielsweise einen mehrfach indirekten Internet-Zugang realisieren, indem Sie sämtlichen Traffic an einen Upstream-Proxy (`parent`) weiterreichen. Hier ein entsprechendes Konfigurationsbeispiel, das auf den Proxy `192.168.0.2` zugreift (`3128` ist der HTTP-Proxy-Port des Peers, während `3130` als Port für die interne Kommunikation zwischen den Proxies dient):

```
cache_peer 192.168.0.2 parent 3128 3130
```

- ▶ `ftp_passive`
Diese Option bestimmt, ob `squid` als Proxy für aktives oder passives FTP verwendet wird. Aktiv ist die Standardvariante; hier wird die ursprüngliche Anforderung vom Client, die Datenverbindung aber vom Server initiiert. Da einige Firewalls mit Letzterem Schwierigkeiten haben (unter anderem auch die `openSUSE-Firewall` bei Standardeinstellungen), sollten Sie `ftp_passive` auf `on` setzen – bei passivem FTP werden beide Verbindungen durch den Client eingeleitet.

Nachdem Sie die Konfigurationsdatei geändert und gespeichert haben, können Sie `squid` (neu) starten. Geben Sie für einen Neustart Folgendes ein (die Anweisung für den normalen Start wurde oben bereits erwähnt):

```
# rcsquid restart
```

Wenn `squid` automatisch beim Booten gestartet werden soll, können Sie entweder den Runlevel-Editor verwenden oder einfach folgendes Kommando eingeben:

```
# chkconfig -a squid
```

Anstatt `squid` zu benutzen, können Sie auch den Webserver Apache (siehe nächstes Kapitel) als Proxy einsetzen. Eine grundlegende Anleitung finden Sie beispielsweise unter der Adresse http://buecher.lingoworld.de/apache2/mod_proxy.html auf den Seiten zu meinem Buch »Apache 2«, (3. Auflage; Bonn 2009, Galileo Press).

Den Proxy-Client einrichten

Nachdem auf einem Rechner in Ihrem Netzwerk ein Proxy läuft, können Sie beliebig viele Clients einrichten, die darauf zugreifen. Die grundlegendste Möglichkeit besteht darin, die Umgebungsvariablen `http_proxy`, `ftp_proxy` und so weiter zu setzen. Beispiel (gilt allerdings nur in der aktuellen Shell):

```
$ http_proxy=http://192.168.0.2:3128
$ export http_proxy
```

Dies ist beispielsweise nützlich, wenn Sie innerhalb desselben Terminals anschließend mit dem Textbrowser `lynx` surfen möchten.

Um diese Variablen dauerhaft und systemweit zu setzen, können Sie stattdessen das YaST-Modul `PROXYSERVER` verwenden, das Sie unter `NETZWERKDIENTSTE` finden. Geben Sie hier folgende Einstellungen ein:

- ▶ **HTTP-PROXY-URL**
Die URL für HTTP-Proxy-Zugriffe im Format `http://proxy-host:Port`. Beispiel: `http://192.168.0.2:3128`.
- ▶ **HTTPS-PROXY-URL**
URL für den HTTPS-Proxy (SSL-gesicherte HTTP-Verbindungen). Ein vernünftig konfigurierter `squid` erledigt dies unter derselben URL wie den HTTP-Proxy-Dienst.
- ▶ **FTP PROXY URL**
Die URL des FTP-Proxys; auch sie entspricht oft dem HTTP-Proxy.
- ▶ **KEINE PROXY DOMAINS**
Für Server im lokalen Netzwerk soll der Proxy oft umgangen werden, sodass Sie hier neben `localhost` und der zugehörigen IP-Adresse `127.0.0.1` (Voreinstellung) auch das Domain-Suffix oder den gemeinsamen Teil der IP-Adressen Ihres LANs eingeben sollten, also beispielsweise `test.local` oder `192.168.0.`

► PROXY-BENUTZERNAME

Proxies wie `squid` können optional so konfiguriert werden, dass eine Authentifizierung erforderlich ist. Geben Sie hier in diesem Fall den passenden Benutzernamen ein.

► PROXY-PASSWORT

Falls der Proxy eine Anmeldung verlangt, müssen Sie hier auch das zugehörige Passwort eingeben.

Nebenher können Sie die Proxy-Einstellungen auch direkt in Webbrowsern eingeben. In Firefox finden Sie den entsprechenden Dialog unter BEARBEITEN • EINSTELLUNGEN im Bereich ALLGEMEIN unter VERBINDUNGS-EINSTELLUNGEN. Wählen Sie hier MANUELLE PROXY-KONFIGURATION, und geben Sie Adressen und Ports der gewünschten Proxy-Dienste ein. Oft genügen die Eingabe des HTTP-Proxys und das Ankreuzen der Option FÜR ALLE PROTOKOLLE DIESEN PROXYSERVER VERWENDEN.

14.3.3 SSH

Die Secure Shell (SSH) ermöglicht die Anmeldung auf einem entfernten Rechner und das Arbeiten in einer normalen Shell-Umgebung; der gesamte Vorgang findet verschlüsselt statt, sodass SSH-Verbindungen auch über das unsichere Internet genutzt werden können.

Ihr openSUSE-Rechner wird bei der Installation automatisch als SSH-Server eingerichtet. Wie in Kapitel 2, »openSUSE installieren«, beschrieben, werden Sie auch gleich danach gefragt, ob Sie den zuständigen TCP-Port in der Firewall öffnen möchten (es handelt sich um Port 22). Falls Sie dies bei der Installation versäumt haben, müssen Sie es gemäß der Anleitung im vorigen Kapitel nachholen. Ob der SSH-Server (`sshd`) tatsächlich läuft, können Sie mithilfe der folgenden Anweisung testen:

```
# ps aux |grep sshd
root      2925  0.0  0.1  4952  368 ?        Ss   May08   0:00 /
usr/sbin/sshd -o PidFile=/var/run/sshd.init.pid
root      14391  0.0  0.2   2816   756 pts/4    S+   11:47   0:00
grep sshd
```

Der erste Prozess in der Ausgabe ist hier der gewünschte SSH-Server; der zweite gibt Ihren Suchfilter (`grep sshd`) wieder. Falls der Server nicht aktiv sein sollte, können Sie ihn mithilfe der üblichen Anweisung starten:

```
# rcsshd start
```

Zusätzlich können Sie per `chkconfig` oder im Runlevel-Editor sicherstellen, dass der `sshd` in Zukunft automatisch gestartet wird. Details zur Konfiguration erfahren Sie aus den Manpages `sshd(8)` und `sshd_config(5)`.

Für SSH-Client-Verbindungen steht auf allen wichtigen UNIX-Systemen der Kommandozeilenclient zur Verfügung. Schematisch gesehen verwenden Sie ihn wie folgt:

```
ssh -l Username Hostname
```

Die Option `-l Username` können Sie weglassen, falls der Benutzername auf dem Remote-System mit Ihrem aktuellen Log-in-Namen identisch ist.

Hier ein Beispiel für einen Zugriff auf einen Rechner im lokalen Netz:

```
$ ssh -l sascha tux2.test.local
```

Eine Alternative zu dieser Schreibweise ist:

```
$ ssh Username@Hostname
```

Also beispielsweise:

```
$ ssh sascha@tux2.test.local
```

Beim ersten Zugriff auf einen neuen SSH-Server erhalten Sie eine Meldung wie diese, weil seine Identität noch nicht bekannt ist:

```
The authenticity of host tux2.test.local (192.168.0.4)
can't be established.
RSA key fingerprint is
7e:74:8a:6a:3e:2f:c0:89:15:1f:ea:ff:87:26:f2:a9.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'tux2.test.local' (RSA) to the list of
known hosts.
```

Beantworten Sie die Sicherheitsfrage wie gezeigt mit `yes`. Als Nächstes werden Sie nach Ihrem Passwort (auf dem Fremdrechner!) gefragt. Geben Sie es ein, und Sie gelangen in die Shell auf dem entfernten System. Hier können Sie ganz normal arbeiten (über das Internet möglicherweise etwas langsamer als lokal). Wenn Sie fertig sind, geben Sie einfach `exit` ein.

Falls Sie mit einem Windows-Rechner auf einen SSH-Server zugreifen möchten, empfehle ich Ihnen den freien Windows-SSH-Client PuTTY. Sie können ihn beispielsweise auf der Original-Projekt-Homepage <http://www.chiark.greenend.org.uk/~sgtatham/putty/> herunterladen.

Mithilfe des Befehls `scp` (Secure CoPy) können Sie über eine eigens dafür aufgebaute SSH-Verbindung Dateien von einem oder auf einen entfernten Host kopieren. Die grundlegende Syntax sieht wie folgt aus:

```
scp [Optionen] [[User@]Host:]Quellpfad [[User@]Host:]Zielpfad
```

Möchten Sie beispielsweise die Datei `info.txt` aus dem aktuellen Arbeitsverzeichnis nach `/home/user/texte` auf dem Host `tux2.local` kopieren, dann lautet der Befehl folgendermaßen:

```
$ scp info.txt user@tux2.local:/home/user/texte
```

Das Präfix »User@« können Sie beim Remote-Rechner weglassen, wenn Sie dort unter demselben Usernamen arbeiten wie lokal. In jedem Fall werden Sie beim ersten Zugriff auf einen bestimmten entfernten Computer in der aktuellen Sitzung nach Ihrem dortigen Passwort gefragt, bevor der Kopiervorgang durchgeführt wird. Während des Kopierens zeigt `scp` einen Fortschrittsbalken im Textformat an.

Der Pfad auf dem Remote-Host wird üblicherweise absolut, das heißt mit führendem `/`, angegeben. Falls Sie dennoch einen relativen Pfad angeben, funktioniert dies nur, wenn der zugreifende User ein Home-Verzeichnis auf diesem hat; in einem solchen Fall wird die Verzeichnisangabe relativ zum Home-Verzeichnis interpretiert.

Die wichtigste Option für `scp` ist `-R` (recursive): Alle Dateien im angegebenen Quellverzeichnis und dem gesamten Unterverzeichnisbaum, die dem Namensschema entsprechen, werden kopiert. Das folgende Beispiel kopiert das Verzeichnis `/home/user/dokumente` und alle enthaltenen Dateien und Unterverzeichnisse vom Host `tux2.local` in das aktuelle Verzeichnis auf dem lokalen Rechner:

```
$ scp -R user@tux2.local:/home/user/dokumente .
```

14.3.4 Remote-Verwaltung (VNC)

Virtual Network Computing (VNC) ermöglicht den grafischen Vollzugriff auf einen entfernten Linux- oder sonstigen UNIX-Rechner (unter Windows steht ein ähnlicher Dienst namens Remote Desktop zur Verfügung).

Wenn Ihr Rechner als VNC-Server eingerichtet werden soll, sodass Sie ihn von einem anderen Computer aus auch über die grafische Oberfläche steuern können, wählen Sie `ADMINISTRATION VON EINEM ENTFERNTEN RECHNER` im YaST-Bereich `NETZWERKDIENTE`. Hier brauchen Sie nicht viel einzustellen. Aktivieren Sie einfach `VERWALTUNG VIA ENTFERNTEM RECHNER (REMOTE) ERLAUBEN` sowie `FIREWALL-PORT ÖFFNEN`, und alles ist bereit.

Für die Steuerung Ihres Computers von einem anderen aus benötigen Sie nun einen VNC-Client. Eine praktische Lösung für Windows, Linux und viele andere Systeme ist RealVNC. Eine abgespeckte Free Edition dieser Client- und Server-Kombination (freie GPL-Software) können Sie unter <http://www.realvnc.com> herunterladen. Führen Sie die Installation gemäß Anleitung durch, und starten Sie dann den enthaltenen VNC-Viewer. Als VNC-Passwort müssen Sie das `root`-Passwort des Server-Rechners eingeben, danach wird der gewohnte grafische Log-in (zum Beispiel KDM) in einem Fenster angezeigt, und Sie können sich als gewöhnlicher Benutzer einloggen.

Eine andere praktische Methode besteht darin, einen Java-fähigen Webbrowser als VNC-Client einzusetzen. Geben Sie dazu eine URL nach dem Schema `http://remote-host:5801` ein. Beispiel:

`http://192.168.0.4:5801`

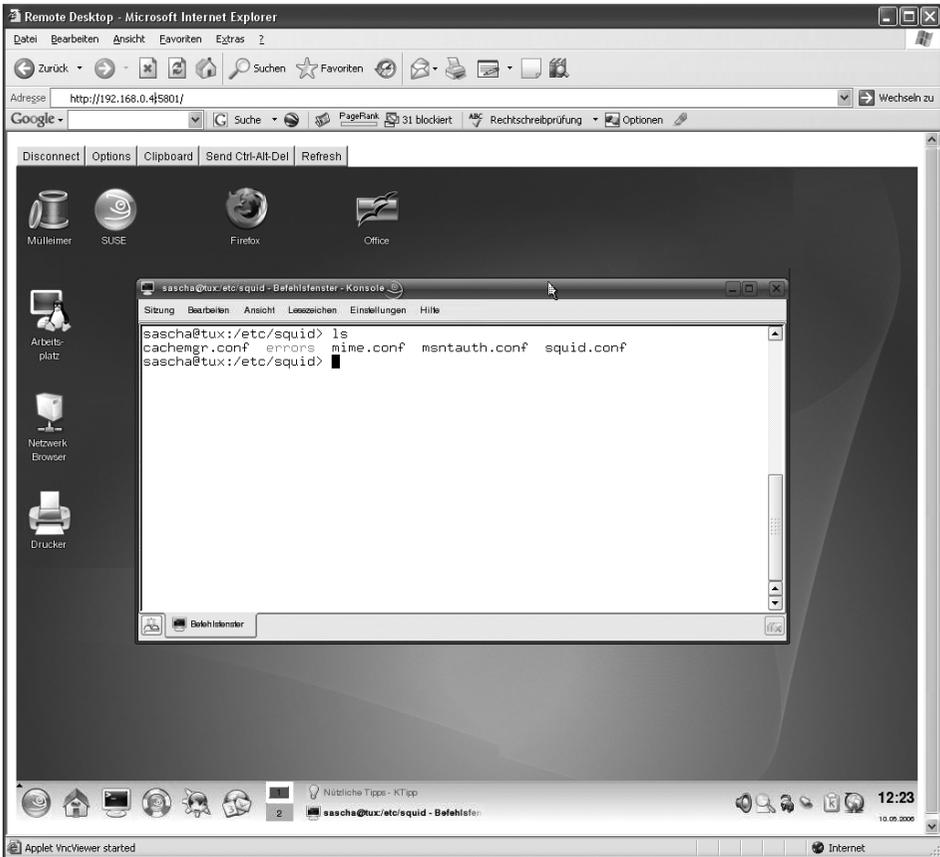


Abbildung 14.9 Der KDE-3-Desktop eines openSUSE-Rechners im Microsoft Internet Explorer als VNC-Client

Daraufhin wird zunächst der eigentliche VNC-Client als Java-Applet geladen (das kann ein wenig dauern). Anschließend müssen Sie auch hier das `root`-Passwort eingeben und können sich danach grafisch anmelden. Die Arbeit geht etwas langsamer vonstatten als auf dem »normalen« Desktop, aber für den Notfall ist das Ganze einigermäßen praktisch. Abbildung 14.9 zeigt eine entsprechende KDE-Sitzung im Microsoft Internet Explorer unter Windows XP.

Bitte beachten Sie, dass VNC-Sitzungen unsicherer sind als SSH. Im Alltag ist es vernünftiger, eine normale SSH-Sitzung zu starten und die gewünschten grafischen Programme vom entfernten Rechner dann als X-Clients auf Ihrem lokalen X-Server auszuführen. Das folgende Beispiel startet innerhalb einer `root`-SSH-Sitzung ein `xterm` auf Ihrem eigenen Rechner (den Sie aus der SSH-Perspektive natürlich nicht als `localhost` bezeichnen dürfen):

```
otherhost:/ # xterm -display tux.test.local:0.0
```

14.4 Zusammenfassung

UNIX-Systeme sind aus Gründen ihrer Geschichte die idealen Netzwerkserver. openSUSE ist da keine Ausnahme, sondern eher besonders üppig mit Serversoftware ausgestattet. In diesem Kapitel haben Sie alle wichtigen Serverdienste für Ihr lokales Netzwerk kennengelernt; im nächsten und übernächsten Kapitel geht es mit diversen Internet-Serverdiensten weiter.

Sehr nützlich für die Zusammenarbeit in lokalen Netzen sind die diversen Arten von Dateiservern. Sie ermöglichen die Freigabe von Dateisystemen für den Zugriff durch andere Rechner und damit die Zusammenarbeit an gemeinsamen Dateibeständen. Neben dem UNIX-Klassiker NFS haben Sie auch einen der wichtigsten Server für heterogene Netzwerke kennengelernt, nämlich den besonders für Windows-Clients konzipierten Samba.

Eine weitere praktische Klasse von Serverdiensten bilden die *Verzeichnisdienstserver*. In diesem Kapitel wurden zwei Exemplare behandelt: NIS, eine ideale Ergänzung zu NFS (beide basieren auf RPC und stammen von Sun), sowie der erheblich mächtigere, aber auch komplexere OpenLDAP-Server. Beide können unter anderem eingesetzt werden, um die Benutzerauthentifizierung zu zentralisieren.

Zu guter Letzt haben Sie noch die Bekanntschaft einiger weiterer Tools gemacht, die in einem lokalen Netzwerk eine Rolle spielen. Der DHCP-Server erleichtert die Konfiguration von TCP/IP-Netzwerken entscheidend, da Sie nicht mehr auf jedem einzelnen Rechner allerlei Detailsinstellungen vornehmen müssen. Der

Proxyserver `squid` gibt als Stellvertreter den Zugriff auf Web, FTP und andere Internetdienste frei, die aufgrund von Firewall-Regeln eigentlich gesperrt sind, und dient zudem als beschleunigender Webcache. Mit SSH können Sie auf entfernten Rechnern (oder von fern auf Ihrem eigenen Rechner) arbeiten, als säßen Sie direkt davor. VNC schließlich ermöglicht auch den Fernzugriff auf die grafische Benutzeroberfläche.

Mein Vater musste nach dem Studium seinen Zivildienst in einem Jugendhaus ableisten. Er zeigte mir die handxerographierten Programmzettel des Jahres 1975: mit Nadel und Faden im Webkurs, glasierte Ziegel in der Emailwerkstatt und (damals noch ein echtes Novum, wie er mir versicherte) Surfen für Anfänger auf Husum. Nachdenklich geworden klickte ich mich spät in der Nacht zur Homepage des Jugendhauses durch: Es existierte noch, und auch das Angebot hatte sich nicht verändert:

Webkurs, Emailworkshop, Surfen für Anfänger, alles noch da.

– Lino Wirag in »Titanic« 4/2006

15 LAMP (Linux, Apache, MySQL und PHP)

Nachdem Sie im vorigen Kapitel Dateiserver und andere lokale, native Netzwerkdienste kennengelernt haben, geht es hier um die Einrichtung und den Betrieb eines Serversystems für Webanwendungen. Dazu wird hier das bekannte LAMP-System – Linux, Apache, MySQL und PHP – vorgestellt, da es die beliebteste Lösung für Webanwendungen ist. Zuvor erhalten Sie einen kurzen Überblick über das Web-Protokoll HTTP, weil es die Grundlage für das Verständnis eines entsprechenden Servers bildet.

15.1 HTTP-Grundlagen

Das TCP/IP-Anwendungsprotokoll HTTP (Hypertext Transfer Protocol) bildet die Basis der Kommunikation zwischen Webserver und Browser. Es handelt sich um ein klartextbasiertes Protokoll mit wenigen Client-Befehlen und wohldefinierten Server-Antworten, das nach dem Vorbild von Protokollen wie dem viel älteren FTP (File Transfer Protocol) gestaltet wurde. Die aktuelle Version ist HTTP/1.1; ihre Spezifikation steht in RFC 2616.

15.1.1 Die HTTP-Kommunikation

HTTP ist ein typisches *Request/Response-Verfahren* (Anfrage und Antwort). Der Ablauf einer solchen HTTP-Anfrage und -Antwort ist immer derselbe:

Der Browser ermittelt aus der URL den Hostnamen des angesprochenen Servers und stellt eine TCP-Verbindung zu diesem her. Standardmäßig wird der TCP-Server-Port 80 gewählt, ansonsten die Portnummer, die in der URL steht.

1. Angenommen, die angeforderte URL ist `http://www.test.local/seiten/info.html`. Dann ist der Server `www.test.local`. Der Browser ermittelt über DNS die IP-Adresse dieses Hosts und stellt eine Verbindung zu dessen Port 80 her.
2. Der Browser übermittelt eine Anfrage an den Server. Die erste Zeile enthält den HTTP-Befehl, den Pfad des angesprochenen Dokuments und die Protokollversion. In weiteren Zeilen folgen optionale Header-Felder nach dem Schema `Feldname: Wert`, in denen der Browser beispielsweise seine eigene Versionsangabe, den Hostnamen des anfordernden Rechners oder die URL des Dokuments, in dem sich der angeklickte Hyperlink befand (den sogenannten Referrer), mitteilt. Bestimmte HTTP-Anfragen enthalten nach einer Leerzeile auch noch einen Body, in dem zusätzliche Daten geschickt werden.

Der einfachste HTTP-Befehl (und der einzige, der in der ursprünglichen Protokollversion HTTP/0.9 enthalten war!) ist `GET`. Er bedeutet ganz einfach, dass ein bestimmtes Dokument geliefert werden soll.

Gemäß dem Beispiel aus Punkt 1 wird hier der folgende `GET`-Befehl betrachtet:
`GET /seiten/info.html HTTP/1.1.`

3. Der Server nimmt die `GET`-Anfrage entgegen und muss je nach Dateityp des angeforderten Dokuments unterschiedlich verfahren: Wenn es sich um ein statisches Dokument handelt (was für dieses Beispiel der Einfachheit halber angenommen werden soll), kann er es einfach ausliefern. Handelt es sich dagegen um die Adresse eines serverseitigen Skripts, muss er zunächst das zuständige Programm aufrufen, dessen Ausgabe entgegennehmen und diese genau wie ein statisches Dokument an den Client liefern.

Bevor der Server das Dokument sendet, schickt er eine Statusmeldung und HTTP-Header mit diversen Angaben, gefolgt von einer Leerzeile. Die Statusmeldung in der ersten Zeile der ausgegebenen Daten besteht aus der HTTP-Version, einer standardisierten Statuscodenummer und deren Textauflösung. Die beiden häufigsten Statusmeldungen dürften `200 OK` (Dokument ist vorhanden und wird mitgeliefert) und `404 Not Found` (Dokument existiert nicht) sein. In diesem Beispiel soll gelten, dass das Dokument gefunden wurde. Die erste Zeile der Antwort lautet also:

```
HTTP/1.1 200 OK
```

In der nächsten Zeile beginnen die HTTP-Header. Die Mindestanforderung für den Header einer HTTP-Antwort ist die Angabe des MIME-Typs des geliefer-

ten Dokuments. Im Fall eines HTML-Dokuments sieht die entsprechende Header-Zeile so aus:

```
Content-Type: text/html
```

Weitere häufig verwendete Header-Felder enthalten beispielsweise das Änderungs- und das Verfallsdatum des Dokuments, die Versionsangabe des Servers oder Cookies. Weiter unten erhalten Sie eine kurze Übersicht.

Der Body der HTTP-Antwort, im vorliegenden Beispiel also das eigentliche Dokument, muss durch eine Leerzeile vom Header getrennt werden.

4. Der Client nimmt die HTTP-Antwort des Servers entgegen. Er interpretiert zunächst die empfangenen Header-Daten und anschließend das eigentliche Dokument. Bei einem HTML-Dokument hat dies meist zur Folge, dass er zahlreiche weitere HTTP-Anfragen stellen muss, um die in das Dokument eingebetteten Bilder und Multimedia-Dateien anzufordern.

Formal sind sowohl die HTTP-Anfrage als auch die Server-Antwort RFC-822-kompatible Nachrichten. Dieses Format (aktuelle Version: RFC 2822) wurde ursprünglich für E-Mails entwickelt. Es besteht aus beliebig vielen Header-Zeilen mit dem Format `Headername: Wert`; der Body (Nachrichteninhalt) wird durch eine Leerzeile von den Headern getrennt.

Das RFC-822-Format wurde später durch MIME erweitert, um E-Mails in anderen Formaten als reinem Text und mit Dateianhängen versenden zu können. Eine MIME-Nachricht benötigt das zusätzliche Header-Feld `Content-Type`, in dem der Datentyp des Body-Inhalts angegeben wird. Auch für HTTP spielt MIME eine wichtige Rolle, weil der Content-Type dem Browser mitteilt, wie er mit der fraglichen Datei (zum Beispiel HTML-Dokument, JPEG-Bild und so weiter) umgehen soll.

Eine vollständige Client-Anfrage mit allen ihren Headern könnte beispielsweise so aussehen:

```
GET /seiten/info.html HTTP/1.1
Accept: */*
Accept-Language: de, en-US
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/5.0 (X11; U; Linux i686; de; rv: 1.9b5) Gecko/
2008032600 SUSE/2.9.95-25.1 Firefox/3.0b5
Host: www.test.local
Connection: Keep-Alive
```

Kurz gesagt enthalten die Header-Felder der Anfrage in diesem Beispiel folgende Informationen:

- ▶ `Accept: */*`
Der Browser akzeptiert Dokumente mit beliebigem MIME-Type (`/*/*` bedeutet: beliebiger Typ/beliebiger Untertyp).
- ▶ `Accept-Language: de, en-US`
Der Browser bevorzugt Dokumente in den Sprachen Deutsch und US-Englisch (in dieser Reihenfolge).
- ▶ `Accept-Encoding: gzip, deflate`
Neben unkomprimierten Daten unterstützt der Browser auch Dateien, die mit `gzip` beziehungsweise `ZIP (deflate)` komprimiert sind.
- ▶ `User-Agent: Mozilla/5.0 (X11; U; Linux i686; de; rv: 1.9b5) Gecko/2008032600 SUSE/2.9.95-25.1 Firefox/3.0b5`
Der Browser übergibt seine eigene Versionsangabe. Dies dient dem Erstellen der bekannten Browser-Statistiken und ermöglicht es Webentwicklern, an verschiedene Browser angepasste Versionen von Dokumenten anzubieten. In diesem Beispiel handelt es sich um Firefox 3.0 unter openSUSE 11.0.
- ▶ `Host: www.test.local`
Diese Angabe ist zur Unterscheidung virtueller Hosts wichtig, die sich dieselbe IP-Adresse teilen. Da die IP-Datagramme nur die IP-Adresse, aber nicht den Hostnamen enthalten, weiß der Server nur durch dieses Header-Feld, welche konkrete Website gemeint ist.
- ▶ `Connection: Keep-Alive`
Die HTTP-Verbindung soll nach der Beantwortung der Anfrage offen gehalten werden (nur HTTP/1.1). Der Server kann allerdings nicht dazu gezwungen werden, die Verbindung beizubehalten; es handelt sich nur um eine Art Empfehlung.

Wenn die Anfrage erfolgreich beantwortet wird, könnte die Antwort des Servers beispielsweise so aussehen:

```
HTTP/1.1 200 OK
Date: Tue, 29 Jan 2008 11:29:26 GMT
Server: Apache/2.2.8 (Unix)
Last-Modified: Mon, 21 Jan 2008 01:45:53 GMT
ETag: "2c986a-b20-fc93242e"
Accept-Ranges: bytes
Content-Length: 1936
Connection: close
Content-Type: text/html

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//Transitional//EN">
<html>
```

```
<head>
[...]
```

Die Header bedeuten Folgendes:

- ▶ **Date:** Tue, 29 Jan 2008 11:29:26 GMT
Dieses Feld gibt Datum und Uhrzeit auf dem Server inklusive Zeitzone an.
- ▶ **Server:** Apache/2.2.8 (Unix)
Dies ist die Versionsangabe des Servers. Sie erfahren weiter unten in diesem Kapitel, wie sich der ausgegebene Wert anpassen lässt.
- ▶ **Last-Modified:** Mon, 21 Jan 2008 01:45:53 GMT
Datum und Uhrzeit der letzten Änderung. Diese Angabe ermöglicht es dem Browser beispielsweise zu entscheiden, ob die Kopie der Seite in seinem Cache noch aktuell ist.
- ▶ **ETag:** "2c986a-b20-fc93242e"
Das ETag (Entity-Tag) wird vom Server erzeugt, wenn er entsprechend konfiguriert ist. Es bestätigt die Identität eines Dokuments über die URL beziehungsweise den Pfad hinaus. Dies ist nützlich für Caching- und Proxy-Zwecke.
- ▶ **Accept-Ranges:** bytes
Durch diesen Header gibt der Server an, dass er Bereichsanfragen akzeptiert: Der Client kann einen Range-Header übermitteln, der angibt, dass er nur einen bestimmten Byte-Bereich des Bodys anfordern möchte.
- ▶ **Content-Length:** 1936
Dies ist die Länge des Bodys in Bytes. Der Browser kann anhand dieser Angabe überprüfen, ob die Lieferung vollständig ist.
- ▶ **Connection:** close
Der Server schließt die Verbindung nach Abschluss dieser Anfrage.
- ▶ **Content-Type:** text/html
Der MIME-Type des Dokuments ist HTML: Haupttyp Text, Untertyp HTML.

Neben dem bereits erwähnten GET gibt es noch einige andere HTTP-Befehle oder Client-Methoden. Die meisten von ihnen wurden mit HTTP/1.0 eingeführt, einige wenige dagegen erst mit HTTP/1.1. In Tabelle 15.1 sehen Sie zunächst eine Übersicht über alle HTTP-Methoden.

HTTP-Methode	Seit Version	Bedeutung
GET	0.9	Dokument anfordern
HEAD	1.0	nur Header anfordern

Tabelle 15.1 Übersicht über die HTTP-Methoden

HTTP-Methode	Seit Version	Bedeutung
POST	1.0	umfangreiche Formulare Daten senden
PUT	1.0	Dokument auf dem Server ablegen
DELETE	1.0	Dokument vom Server löschen
LINK	1.0	Verknüpfung erstellen
UNLINK	1.0	Verknüpfung löschen
TRACE	1.1	Proxies im Header anzeigen
CONNECT	1.1	Proxy-Zugriff auf gesicherte Server
OPTIONS	1.1	Liste von Optionen anfordern

Tabelle 15.1 Übersicht über die HTTP-Methoden (Forts.)

15.1.2 HTTP-Statuscodes

Wie Sie bereits gesehen haben, enthalten HTTP-Server-Antworten einen Statuscode – eine dreistellige Zahl, die die Grundbedeutung der Antwort angibt. HTTP verwendet fünf verschiedene Grundtypen von Statuscodes. Der Typ ist an der ersten Stelle des Statuscodes erkennbar. Tabelle 15.2 zeigt eine Übersicht über die Statuscode-Typen.¹

Nummernbereich	Bedeutung
1xx	Information
2xx	Erfolgsmeldungen
3xx	Umleitungsmeldungen
4xx	Client-Fehler
5xx	Server-Fehler

Tabelle 15.2 Übersicht über die HTTP-Statuscode-Typen

1xx-Statuscodes (Informationen)

In Tabelle 15.3 sehen Sie die beiden einzigen 1xx-Statuscodes. In der Praxis hat dieser Nummernbereich keine besonders große Bedeutung; nur selten antworten Server mit diesen Meldungen.

¹ LINK und UNLINK wurden kaum jemals implementiert. Selbst RFC 1945, das die HTTP/1.0-Spezifikation enthält, schweigt sich bis auf je einen lapidaren Satz über ihren Sinn aus. In der HTTP/1.1-Spezifikation kommen sie gar nicht mehr vor.

Statuscode	Bezeichnung	Erläuterung
100	Continue	Der Server erwartet eine Fortsetzung der Anfrage.
101	Switching Protocols	Wechsel auf die im Upgrade-Header angegebene HTTP-Version

Tabelle 15.3 1xx-Statuscodes

100 `Continue` gibt der Server zurück, wenn die Anfrage unvollständig ist und er deshalb eine Fortsetzung erwartet. 101 `Switching Protocols` besagt, dass der Server mit einer anderen Version des HTTP-Protokolls antwortet als derjenigen, die in der Anfrage verwendet wurde. Die eingesetzte Version des Protokolls wird im `Upgrade-Header` angegeben.

2xx-Statuscodes (Erfolgsmeldungen)

Statuscodes zwischen 200 und 299 zeigen unterschiedliche Arten erfolgreicher Verarbeitung durch den Server an. Tabelle 15.4 zeigt eine entsprechende Übersicht.

Statuscode	Bezeichnung	Erläuterung
200	OK	Anfrage erfolgreich verarbeitet; angeforderte Ressource im Body enthalten.
201	Created	Durch PUT-Anfrage übertragene Datei wurde gespeichert.
202	Accepted	Anfrage angenommen; keine weiteren Folgen.
203	Non-Authoritative Information	Antwort wurde nicht vom eigentlichen Server, sondern z. B. von einem Proxy erzeugt.
204	No Content	Anfrage in Ordnung; Antwort besitzt keinen Body.
205	Reset Content	Fordert den Client auf, das Formular zurückzusetzen, das die Anfrage erzeugt hat.
206	Partial Content	Antwort auf eine Anfrage mit <code>Range-Header</code> ; sie enthält nur den gewünschten Bereich der Ressource und einen entsprechenden <code>Content-Range-Header</code> .

Tabelle 15.4 2xx-Statuscodes

Die Mehrheit aller erfolgreich verarbeiteten HTTP-Anfragen liefert eine Antwort mit der Meldung 200 OK, insbesondere bei GET, POST, HEAD und DELETE. Bei einer

erfolgreich abgeschlossenen PUT-Anfrage wird dagegen die Meldung 201 Created zurückgegeben. Alle anderen 2xx-Meldungen sind wesentlich exotischer.

3xx-Statuscodes (Umleitungsmeldungen)

In Tabelle 15.5 finden Sie sämtliche 3xx-Statuscodes, die Auskunft über diverse Arten der Um- und Weiterleitung geben.

Statuscode	Bezeichnung	Erläuterung
300	Multiple Choices	Die Pfadangabe in der Anfrage passt auf mehrere Ressourcen. Der Body enthält in der Regel eine Liste all dieser Dateien.
301	Moved Permanently	Die Ressource wurde dauerhaft verschoben und ist unter der im Location-Header angegebenen URL zu finden.
302	Found	Vorübergehend verschoben; wurde durch 307 ersetzt.
303	See Other	Die Ressource ist unter der im Location-Header angegebenen URL zu finden.
304	Not Modified	Antwort auf spezielle Anfragen mit Headern wie If-Modified-Since; ohne Body
305	Use Proxy	Die Anfrage muss über den im Location-Header angegebenen Proxy erfolgen.
307	Temporary Redirect	Die Ressource wurde vorübergehend verschoben und ist zurzeit unter der im Location-Header angegebenen URL zu finden.

Tabelle 15.5 3xx-Statuscodes

In dieser Gruppe ist beispielsweise der Statuscode 301 Moved Permanently wichtig, der zurückgegeben wird, wenn sich ein angefordertes Dokument nun unter einem neuen (in der Antwort angegebenen) Pfad befindet. 307 Temporary Redirect (in alten HTTP-Versionen auch 302 Found) wird dagegen zurückgegeben, wenn eine Ressource vorübergehend nicht erreichbar ist, beispielsweise wegen Server-Wartungsarbeiten. Die dritte Art der Umleitung ist 303 See Other; in diesem Fall handelt es sich um eine absichtliche Weiterleitung auf die angegebene URL. Sie wird etwa dann verwendet, wenn ein serverseitiges Skript keine eigene Ausgabe besitzt, sondern nach getaner Arbeit eine Weiterleitung auf eine andere Seite durchführt. In all diesen Fällen enthält die Antwort eine Header-Zeile nach dem Schema Location: Neue_URL.

Ein anderer interessanter Statuscode ist 304 Not Modified. Es handelt sich um die body-lose Antwort auf eine sogenannte bedingte Anfrage: Spezielle Header-Felder wie If-Modified-Since ermöglichen es dem Client oder einem Proxy, Dokumente nur unter bestimmten Voraussetzungen zu laden – im genannten Fall beispielsweise nur dann, wenn auf dem Server eine neuere Version des Dokuments vorliegt als diejenige, die der Client oder der Proxy noch in seinem Cache gespeichert hat.

4xx-Statuscodes (Client-Fehler)

Statuscodes zwischen 400 und 499 stehen für verschiedene Arten von Fehlern oder Mängeln in der Client-Anfrage. Tabelle 15.6 zeigt zunächst eine Übersicht aller 4xx-Meldungen.

Statuscode	Bezeichnung	Erläuterung
400	Bad Request	syntaktisch fehlerhafte Client-Anfrage
401	Unauthorized	Authentifizierung erforderlich
402	Payment Required	Zahlung (Micropayment) erforderlich – vorgesehen, aber noch nicht implementiert.
403	Forbidden	Zugriff verweigert; auch Authentifizierung hilft nichts.
404	Not Found	Die angeforderte Ressource ist nicht vorhanden.
405	Method Not Allowed	Die verwendete Client-Zugriffsmethode ist für die angeforderte Ressource nicht zulässig; die erlaubten Methoden stehen im Allow-Header.
406	Not Acceptable	Dateityp der Ressource widerspricht einem Accept-...-Header der Anfrage.
407	Proxy Authentication Requested	Benutzeranmeldung am Proxy erforderlich
408	Request Timeout	Der Server hat innerhalb einer bestimmten Zeit nach Verbindungsaufbau keine Anfrage erhalten.
409	Conflict	Anfrage kann aufgrund eines Konflikts nicht ausgeführt werden. Beispiel: PUT-Anfrage versucht, veraltete Version zu speichern, obwohl WebDAV-Versionskontrolle aktiviert ist.

Tabelle 15.6 4xx-Statuscodes

Statuscode	Bezeichnung	Erläuterung
410	Gone	Ressource nicht mehr vorhanden. Sollte bei absichtlicher Entfernung zeitlich begrenzter Angebote statt 404 Not Found zurückgegeben werden.
411	Length Required	Server weist die Anfrage (mit Body) zurück, weil sie keinen Content-Length-Header enthält.
412	Precondition Failed	Im Header der Anfrage stand eine Bedingung, die nicht zutrifft.
413	Request Entity Too Long	Body der Anfrage zu groß
414	Request-URI Too Long	Die Bezeichnung der angeforderten Ressource ist zu lang – kann sehr selten bei GET-Anfragen mit CGI-Daten passieren.
415	Unsupported Media Type	Body der Anfrage hat ein unzulässiges Format.
416	Request Range Not Satisfiable	Der im Range-Header der Anfrage angeforderte Bereich existiert nicht.
417	Expectation Failed	Die im Expect-Header der Anfrage angegebenen Erwartungen konnten vom Server nicht erfüllt werden.

Tabelle 15.6 4xx-Statuscodes (Forts.)

Wie Sie in der Tabelle sehen können, gibt es weit mehr unterschiedliche 4xx-Statuscodes als Meldungen der anderen Kategorien. Da die »Schuld« für das Scheitern der Anfrage bei dem Client liegt, der sie formuliert hat, ist es nützlich, dass ganz genau mitgeteilt wird, worin das Problem besteht. Auf diese Weise kann der Benutzer (oder manchmal sogar der Browser selbst) den Fehler einfacher beheben.

Der bekannteste Statuscode aus dieser Gruppe dürfte 404 Not Found sein – er wird zurückgegeben, wenn das (meist per GET-Anfrage) angeforderte Dokument nicht existiert –, beispielsweise weil Sie die Adresse im Browser falsch geschrieben haben. Auch 401 Unauthorized oder gar 403 Forbidden kommen recht häufig vor.

Viele andere 4xx-Statuscodes interessieren eher Programmierer als Benutzer: Wenn Sie Software programmieren, die HTTP-Anfragen an entfernte Webserver sendet, können Sie an den entsprechenden Meldungen sehen, dass Ihre HTTP-

Header unvollständig oder fehlerhaft sind. Sollten Sie dagegen eine solche Meldung erhalten, wenn Sie einen normalen Browser benutzen, hat entweder Ihr Browser einen Bug oder der angesprochene Server ist nicht standardkonform. In beiden Fällen können Sie nicht viel daran ändern.

5xx-Statuscodes (Server-Fehler)

Statuscodes ab 500 weisen darauf hin, dass der Server trotz korrekter Client-Anfrage nicht antworten konnte. In Tabelle 15.7 finden Sie eine Übersicht über diese Codes.

Statuscode	Bezeichnung	Erläuterung
500	Internal Server Error	fehlerhafte Client-Anfrage
501	Not Implemented	Der Server unterstützt die angeforderte Methode oder Funktionalität nicht.
502	Bad Gateway	fehlerhafte Antwort von einem übergeordneten Server oder Proxy
503	Service Unavailable	Dienst zurzeit nicht erreichbar
504	Gateway Timeout	Timeout in einer per Proxy weitergeleiteten Anfrage
505	HTTP Version Not Supported	Die in der Anfrage verwendete HTTP-Version wird vom Server nicht unterstützt.

Tabelle 15.7 5xx-Statuscodes

Die Fehlermeldung `500 Internal Server Error` bekommen Sie wahrscheinlich dann häufig zu Gesicht, wenn Sie eigene serverseitige Programme mit CGI, PHP oder ähnlichen Sprachen schreiben: Dieser Statuscode weist fast immer auf einen schwerwiegenden Fehler in einem Webserver-Skript hin.

15.1.3 HTTP-Header

Weiter oben haben Sie bereits einige HTTP-Header kennengelernt. Jeder Header steht in einer eigenen Zeile und hat das folgende, grundlegende Format:

```
Header-Name: Wert1[, Wert2, Wert3 ...]
```

Bei den Namen der Header-Felder spielt Groß- und Kleinschreibung keine Rolle. Falls für einen Header mehrere Werte existieren, werden diese durch Kommas getrennt.

In Tabelle 15.8 finden Sie eine Übersicht über sämtliche Header mit kurzer Erläuterung ihrer jeweiligen Aufgabe. In der Tabelle gibt es drei Felder mit den Kontext-Informationen »Anfr.« (Anfrage), »Antw.« (Antwort) und »Ent.« (Entity). Sie bedeuten Folgendes:

- ▶ *Anfrage*
Wenn dieses Feld angekreuzt ist, kann der Header in der HTTP-Anfrage vorkommen.
- ▶ *Antwort*
Ist dieses Feld angekreuzt, dann kann der Header in einer HTTP-Antwort stehen.
- ▶ *Entity*
Ein Header, bei dem dieses Feld angekreuzt ist, enthält Informationen über den Body der HTTP-Meldung – bekannte Beispiele sind Content-Type oder Content-Length.

Header	seit Version	Anfr.	Antw.	Ent.	Bedeutung
Accept	1.0	X			MIME-Types, die der Client akzeptiert
Accept-Charset	1.0	X			Zeichensätze, die der Client akzeptiert
Accept-Encoding	1.0	X			Komprimierungsformate, die der Client akzeptiert
Accept-Language	1.0	X			Sprachen, die der Client akzeptiert
Accept-Ranges	1.1		X		Server kann Anfragen nach Dokumentteilen beantworten.
Age	1.1		X		Zeitspanne seit der letzten Änderung
Allow	1.0		X	X	erlaubte HTTP-Methoden (mit Status 405)
Authorization	1.0	X			Anmeldedaten für geschützte Bereiche (Antwort auf WWW-Authenticate)
Cache-Control	1.1	X	X		Einstellungen für Caching der Ressource
Connection	1.0	X	X		Verbindung geöffnet halten (keep-alive) oder schließen (close).

Tabelle 15.8 Übersicht über sämtliche HTTP-Header

Header	seit Version	Anfr.	Antw.	Ent.	Bedeutung
Content-Encoding	1.0	X	X	X	Komprimierungsformat des Body-Inhalts
Content-Language	1.0	X	X	X	Sprache des Body-Inhalts
Content-Length	1.0	X	X	X	Länge des Body-Inhalts in Byte
Content-Location	1.1		X	X	alternative URL der angeforderten Ressource
Content-MD5	1.1	X	X	X	MD5-Hash des Body-Inhalts zur Konsistenzprüfung
Content-Range	1.1	X	X	X	Start- und Endbyte des gelieferten Bereiches bei Teillieferungen
Content-Type	1.0	X	X	X	MIME-Type der gelieferten Ressource
Cookie	1.0	X			Lieferung eines Cookies, das der Browser zuvor von der angeforderten URL empfangen hatte
Date	1.0	X	X		Serverdatum und -uhrzeit bei Auslieferung
ETag	1.1		X		eine aus diversen Meta-Informationen berechnete ID, z. B. zur Aktualitätsprüfung
Expect	1.1	X			Ankündigung des Clients, dass vor einer Anfrage mit Body 100 Continue erwartet wird.
Expires	1.0		X	X	»Verfallsdatum« der gelieferten Ressource (für Caching)
From	1.0	X			E-Mail-Adresse des Anfragenden (spielt in der Praxis keine Rolle)
Host	1.1	X			Hostname, an den die Anfrage gerichtet ist (Pflicht-Header wegen namensbasierten virtuellen Hosts, siehe unten)
If-Match	1.1	X			Fordert eine Ressource unter der Bedingung an, dass das ETag (siehe oben) einem bestimmten Wert entspricht.
If-Modified-Since	1.0	X			Fordert die Ressource nur an, falls sie seit dem angegebenen Datum geändert wurde.

Tabelle 15.8 Übersicht über sämtliche HTTP-Header (Forts.)

Header	seit Version	Anfr.	Antw.	Ent.	Bedeutung
If-None-Match	1.1	X			Fordert die Ressource an, falls das ETag nicht dem angegebenen Wert entspricht.
If-Range	1.1	X			Fordert einen bestimmten Bereich unter einer Bedingung (ETag oder Änderungsdatum) an.
If-Unmodified-Since	1.1	X			Fordert die Ressource an, falls sie seit dem angegebenen Datum nicht geändert wurde.
Last-Modified	1.0		X	X	Gibt Datum und Uhrzeit der letzten Änderung an.
Location	1.0		X		geänderte URL, unter der die Ressource erreichbar ist (Weiterleitung)
Max-Forwards	1.1	X			maximal erlaubte Anzahl von Proxy-Weiterleitungen (vor allem für TRACE-Anfragen)
Negotiate	1.1	X			Fordert eine Liste von Alternativdokumenten (300 Multiple Choices) an, die verschiedenen Sprach-, Zeichensatz- und Dateitypwünschen genügen.
Pragma	1.0	X	X		Veraltete Einstellung für Caching-Verbot (Wert: no-cache); wurde durch Cache-Control ersetzt.
Proxy-Authenticate	1.1		X		Proxy-Authentifizierungsanforderung
Proxy-Authorization	1.0	X			Proxy-Anmeldedaten (Antwort auf Proxy-Authenticate)
Range	1.1	X			Anforderung eines Teilbereichs einer Ressource (Start- und Endbyte)
Referer ²	1.0	X			URL des Dokuments, das auf das aktuelle verwies (in der Regel Hyperlink)
Retry-After	1.0		X		Information bei Ausfällen, wann der Server oder die gewünschte Ressource wieder verfügbar sein wird.

Tabelle 15.8 Übersicht über sämtliche HTTP-Header (Forts.)

² Wenngleich das englische Wort »referrer« heißt, wird der Name dieses Headers nur mit einem »r« geschrieben.

Header	seit Version	Anfr.	Antw.	Ent.	Bedeutung
Server	1.0		X		Selbstidentifikation der Serversoftware – z. B. Apache/2.2.2 (Unix)
Set-Cookie	1.0		X		Cookie, das der Browser im Namen des Servers speichern soll
TE	1.1	X			Ein Client kann mittels TE: trailers melden, dass er mehrteilige Antworten mit jeweils eigenen Headern akzeptiert.
Trailer	1.1	X	X		Liste der Header, die in einem späteren Block folgen
Transfer-Encoding	1.1	X	X		Der einzige übliche Wert, chunked, bedeutet, dass die Antwort mehrteilig ist.
Upgrade	1.1	X	X		Angabe der unterstützten HTTP-Version (1.1); der Server kann dies per 101 Switching Protocols akzeptieren.
User-Agent	1.0	X			Selbstidentifikation der Client-Software
Vary	1.1		X		Gibt an, welche Aspekte (MIME-Type, Zeichensatz usw.) an Client-Vorlieben angepasst wurden (Content-Negotiation).
Via	1.1	X	X		Liste der Proxies, über die die Anfrage oder Antwort weitergeleitet wurde
Warning	1.1	X	X		Warnungen über den Body-Inhalt, die das Caching betreffen (veraltet usw.)
WWW-Authenticate	1.0		X		Authentifizierungsanforderung für geschützte Verzeichnisse

Tabelle 15.8 Übersicht über sämtliche HTTP-Header (Forts.)

Mithilfe des Apache-Webservers können Sie die meisten Header gezielt modifizieren. Dafür sind die Konfigurationsdirektiven `Header` (für Antwort-Header) beziehungsweise `RequestHeader` (für die Manipulation von Anfrage-Headern vor der Verarbeitung der Anfrage) zuständig, deren Behandlung im Rahmen dieses Buches allerdings zu weit führt. Lesen Sie bei Bedarf die Beschreibungen dieser Direktiven unter <http://buecher.lingoworld.de/apache2/showdir.php?id=557> (Header) beziehungsweise <http://buecher.lingoworld.de/apache2/showdir.php?id=666> (`RequestHeader`); beide auf der Website zum Buch »Apache 2«.

15.2 Der Webserver Apache 2

Zu den verbreitetsten Aufgaben von Linux-Systemen gehören die Veröffentlichung von Websites und der Betrieb von Webanwendungen, in der Regel auf der Basis des Open-Source-Webserver Apache. openSUSE enthält die passenden Zutaten in relativ aktuellen Versionen: Version 11.0 wird mit Apache 2.2, PHP 5.2 und MySQL 5.0 geliefert. Falls Sie einige der neuen Funktionen von MySQL 5.1 benötigen, müssen Sie diese Version manuell installieren; wie das geht, wurde bereits in Kapitel 12, »Der Datenbankserver MySQL«, beschrieben.

Apache ist mit etwa 70 % aller Websites der mit Abstand verbreitetste Webserver oder HTTP-Server. Er wird seit 1995 von der Apache Group entwickelt und entstand aus einer Weiterentwicklung des klassischen NCSA-Webserver – der Name gilt daher als Anspielung auf »a patchy web server«. Inzwischen ist die Apache Group ein Bestandteil der aus ihr hervorgegangenen Apache Software Foundation, die noch viele andere Open-Source-Projekte betreut. Dazu gehören etwa der Servlet-Container Tomcat, zahlreiche Java- und XML-Frameworks sowie SpamAssassin.

Der Apache-Webserver zeichnet sich durch hervorragende Stabilität und Performance, seinen modularen Aufbau und extreme Anpassungsfähigkeit aus. Mehr über dieses bemerkenswerte Open-Source-Produkt erfahren Sie auf dessen Website (<http://d.apache.org>) sowie in meinem Buch »Apache 2« (Galileo Press).

Zurzeit werden drei verschiedene Versionszweige von Apache parallel angeboten:

► *Apache 1.3.x*

Diese Version ist noch auf vielen Servern verbreitet, besonders bei Webhostern. Für eine Neuinstallation sollten Sie diese Version nicht mehr in Betracht ziehen, da sie nicht mehr weiterentwickelt, sondern nur noch durch Bugfixes aktualisiert wird. Einige bestehende Sites sind dagegen noch immer darauf angewiesen, weil einige Zusatzmodule und vor allem selbstgemachte Erweiterungen noch nicht auf Version 2 portiert wurden.

► *Apache 2.0.x*

Der ältere Zweig der aktuellen Hauptversion 2. Gegenüber dem Versionszweig 1.3 wurden zahlreiche technische Neuerungen eingeführt (siehe nächster Unterabschnitt).

► *Apache 2.2.x*

Dies ist der aktuelle Versionszweig; die ersten stabilen Releases sind 2006 erschienen. Gegenüber 2.0 gibt es nur wenige Neuentwicklungen; unter ande-

rem wurde die Authentifizierung völlig neu strukturiert. Zudem gibt es eine eingebaute API für Datenbankzugriffe, was aber eher Modulentwickler als Endanwender interessieren dürfte.

15.2.1 Über Apache 2

Von den drei genannten Codebäumen enthält openSUSE 11.2 die neueste Version 2.2 in der fast aktuellen Release 2.2.13. Für eine manuelle Neuinstallation eines Produktions-Webrowsers sollten Sie auf jeden Fall *nur* eine 2.2er-Version in Betracht ziehen. Apache 1.3 ist zwar stabil, sicher und relativ schnell, aber in vielen seiner Eigenschaften nicht mehr zeitgemäß.

Zu den wichtigsten Eigenschaften von Apache 2 gehören folgende:

- ▶ *Dynamisch ladbare Module*
Schon von Anfang an war Apache durch Zusatzmodule erweiterbar. Bis Version 1.3 musste er allerdings neu kompiliert werden, um ein Modul hinzuzufügen. Die aktuelle Version unterstützt dagegen Module im DSO-Format (Dynamic Shared Objects), die sich per Konfigurationsdirektive hinzufügen oder entfernen lassen.
- ▶ *Verschiedene Laufzeitmodelle*
Bis Apache 1.3 konnte der Webserver ausschließlich als Preforking-Server betrieben werden: Client-Anfragen wurden jeweils durch eigene Prozesse verarbeitet, die auf Vorrat erzeugt wurden. In der neuen Hauptversion ermöglichen die sogenannten Multiprocessing-Module (MPM) auch diverse Alternativen, die auf Threads oder hybriden Thread/Prozess-Modellen basieren. openSUSE liefert zum einen das MPM `prefork` mit, dessen Laufzeitverhalten der Version 1.3 entspricht, und zum anderen das threadbasierte MPM `worker`. Für den Einsatz mit PHP werden allerdings keine thread-basierten MPMs empfohlen, da diverse PHP-Zusatzbibliotheken nicht thread-sicher sind.
- ▶ *Plattformabstraktion*
Ebenfalls neu in Version 2 ist die Abstraktionsbibliothek APR (Apache Portable Runtime), die zahlreiche plattformabhängige Systemfunktionen wie Speicher-, Dateisystem- und Netzwerkzugriffe vereinheitlicht. Da dies gut gelingt, greifen inzwischen auch andere Projekte der Apache Software Foundation sowie einige kommerzielle Softwareprodukte auf die APR zurück.
- ▶ *Integrierte SSL-Funktionalität*
Apache 2 bietet bereits in der Grundausstattung gesicherte HTTPS-Verbindungen über SSL beziehungsweise TLS. In der Vorgängerversion musste die SSL-Unterstützung nachträglich hinzugefügt werden.

15.2.2 Installation mit YaST

Die benötigte Software befindet sich zum größten Teil im YaST-Schema WEB-UND LAMP-SERVER unter SOFTWARE • SOFTWARE INSTALLIEREN ODER LÖSCHEN. In Tabelle 15.9 sehen Sie eine Übersicht über die entsprechenden Installationspakete und ihre Bedeutung.

RPM-Paket	Funktionalität
apache2	Grundpaket Apache-Webserver
apache2-doc	Apache-Dokumentation
apache2-example-pages	Apache-Beispielseite in verschiedenen Sprachen
apache2-mod_perl	Perl-Interpreter als Apache-2-Modul
apache2-mod_php5	PHP als Apache-2-Modul
apache2-mod_python	Python als Apache-2-Modul
apache2-prefork	Multiprocessing-Modul prefork
mysql	Der Datenbankserver MySQL (siehe gleichnamiges Kapitel 12; dort werden noch weitere zugehörige Pakete beschrieben)
php5-ctype	PHP-Erweiterung zur Überprüfung internationaler Zeichen
php5-dom	PHP-Erweiterung zur Verarbeitung von HTML- und XML-Bäumen
php5-gd	PHP-Erweiterung mit Bindung für die Grafikbibliothek gd
php5-iconv	PHP-Erweiterung zur Zeichenkonvertierung
php5-mbstring	PHP-Erweiterung für Mehr-Byte-Zeichensätze (UTF-8 etc.)
php5-zip	PHP-Erweiterung zum Packen und Entpacken von ZIP-Dateien
php5-zlib	PHP-Erweiterung zum Packen und Entpacken von gzip-Dateien
postgresql	Clients und Hilfsprogramme für PostgreSQL
postgresql-contrib	Zusatzmodule und Erweiterungen für PostgreSQL
postgresql-server	der eigentliche PostgreSQL-Datenbankserver
yast2-http-server	YaST-Verwaltungs-Applet für Apache 2 (wird weiter unten beschrieben)
apache2-worker	Multiprocessing-Modul worker

Tabelle 15.9 Die Pakete des YaST-Schemas »Web- und LAMP-Server«

Für einen Webserver zur Auslieferung statischer Webseiten brauchen Sie im Grunde nur die Pakete `apache2`, eines der beiden Multiprocessing-Module `apache2-prefork` oder `apache2-worker`³ sowie die Abstraktionsbibliothek `libapr1` (Letztere ist nicht im Schema enthalten, sondern wird automatisch über die Paketabhängigkeiten installiert). Anschließend können Sie die fertige Website in das Verzeichnis `/srv/www/htdocs` kopieren, und es sollte alles funktionieren.

Heutzutage werden Webserver immer häufiger auch zur Bereitstellung webbasierter Anwendungen verwendet. Die beliebteste Lösung dafür ist die hier vorgestellte Kombination aus Apache, der Skriptsprache PHP und der Open-Source-Datenbank MySQL.

Im YaST-Schema WEB- UND LAMP-SERVER ist eine recht willkürliche Auswahl an PHP-Erweiterungen enthalten. Verwenden Sie besser die Software-Suchfunktion, und geben Sie »php« ein. Wählen Sie mindestens `apache2-mod_php5` (das PHP-Modul für Apache) sowie `php5` (die Sprache selbst) aus. Aus den restlichen Paketen, den PHP Extension Modules, können Sie nach Bedarf auswählen. Für den weiter unten beschriebenen Zugriff auf MySQL-Datenbanken benötigen Sie beispielsweise `php5-mysql`; für viele Anwendungen werden auch `php5-mbstring` (Unterstützung für Mehr-Byte-Zeichensätze), XML-Erweiterungen wie `php5-dom` und `php5-xml` sowie die Kompressionsfunktionen aus `php5-zlib` und `php5-zlib` benötigt.

15.2.3 Apache-Konfigurationsmöglichkeiten in openSUSE

Hier wird zunächst beschrieben, wo Sie in Ihrem System die diversen Apache-Konfigurationsdateien finden und wie Sie YaST verwenden können, um Apache per GUI zu konfigurieren. Die Bedeutung der wichtigsten unter den zahlreichen Apache-Konfigurationsdirektiven wird dagegen weiter unten erläutert.

Apache-Konfigurationsdateien in openSUSE

Die Konfiguration für die mit openSUSE 11 gelieferten Apache-Pakete befindet sich unter `/etc/apache2` beziehungsweise in diversen Unterverzeichnissen. Neben der Hauptkonfigurationsdatei `httpd.conf` wird eine Reihe zusätzlicher, per `Include`-Direktive eingebundener Dateien verwendet; in der dort angegebenen Reihenfolge sind es im Einzelnen folgende:

³ Nicht im Schema »Web- und LAMP-Server« enthalten; muss gegebenenfalls per Suche hinzugefügt werden

- ▶ *uid.conf*
Benutzer- und Gruppen-ID für Worker-Prozesse (Direktiven `User` und `Group`)
- ▶ *server-tuning.conf*
MPM-spezifische Prozess- und Performance-Einstellungen
- ▶ *sysconfig.d/loadmodule.conf*
Zu ladende Module. Änderungen an dieser Datei werden allerdings nicht akzeptiert; verwenden Sie YaST, oder tragen Sie die gewünschten Modulkürzel (ohne `mod_`) in die weiter unten beschriebene Datei `/etc/sysconfig/apache2` ein.
- ▶ *listen.conf*
IP-Adressen und Ports, an denen Apache lauschen soll (Direktive `Listen`)
- ▶ *mod_log_config.conf*
Definition der Log-Formate und -Dateien
- ▶ *sysconfig.d/global.conf*
Server-Grundeinstellungen (auch diese Konfiguration wird durch `/etc/sysconfig/apache2` bestimmt).
- ▶ *mod_status.conf*
Statusinformationen
- ▶ *mod_info.conf*
Server-Informationen
- ▶ *mod_usertrack.conf*
Cookie-basiertes Usertracking
- ▶ *mod_autoindex-defaults.conf*
Automatisch generierte Verzeichnis-Indizes
- ▶ *mod_mime-defaults.conf*
MIME-Einstellungen
- ▶ *errors.conf*
Angepasste Fehlermeldungen
- ▶ *ssl-global.conf*
SSL-Konfiguration
- ▶ *default-server.conf*
Optionen des Standard-Servers (beantwortet Anfragen, für die kein virtueller Host zuständig ist).
- ▶ *mod_userdir.conf*
Benutzerspezifische Web-Verzeichnisse; wird von *default-server.conf* eingebunden.

- ▶ *conf.d/apache2-manual.conf*
Online-Dokumentation; dieses (ebenfalls in *default-server.conf* eingebundene) `Include` wird durch den Platzhalter `? optional`.
- ▶ *sysconfig.d/include.conf*
Includes für Ihre eigenen Erweiterungsdateien (wird durch `/etc/sysconfig/apache2` festgelegt).
- ▶ *vhosts.d/*.conf*
Einstellungen für virtuelle Hosts; alle **.conf*-Dateien in diesem Verzeichnis werden automatisch geladen. Die enthaltenen Beispieldateien *vhost.template* und *vhost-ssl.template* zeigen, wie ein normaler beziehungsweise SSL-verschlüsselter virtueller Host konfiguriert wird.

Das YaST-Modul zur Apache-Konfiguration

Wenn Sie die mit openSUSE gelieferte Apache-Version verwenden, können Sie die meisten Einstellungen in YaST vornehmen. Das zuständige YaST-Modul ist ein `HTTP-SERVER` in der Kategorie `NETZWERKDIENSTE`. Beachten Sie, dass der größte Teil dieses YaST-Moduls für Sie nutzlos ist, wenn Sie nicht über die manuelle Apache-Konfiguration Bescheid wissen, denn die meisten Einträge sind nichts weiter als wörtlich wiedergegebene Zeilen aus den Konfigurationsdateien. Das Wichtigste über die Konfiguration erfahren Sie im Unterabschnitt »Apache-Konfiguration«.

Beim ersten Start werden Sie durch einen Konfigurations-Wizard mit fünf Einzelschritten geführt:

1. **NETZWERKGERÄTEAUSWAHL:** Wählen Sie hier den Port (voreingestellt ist der HTTP-Standard-Port 80) sowie die Netzwerkschnittstellen, an denen Apache lauschen soll. Später im Verlauf der Konfiguration können Sie bei Bedarf weitere Ports hinzufügen. Wählen Sie hier auch `FIREWALL-PORT ÖFFNEN`, wenn Ihr Webserver von außen erreichbar sein soll, oder nehmen Sie unter `FIREWALL-DETAILS` genauere Einstellungen vor.
2. **MODULE:** Wählen Sie hier, welche Skriptsprachen in Ihrem Webserver zur dynamischen Erzeugung von Inhalten verfügbar sein sollen. Zur Auswahl stehen PHP5, Perl, Python und Ruby. Wie weiter oben bei der Paketauswahl zu sehen war, fehlt `mod_ruby` allerdings im openSUSE-11.2-Lieferumfang. Falls Sie dennoch Ruby-basierte Webanwendungen⁴ ausführen möchten, müssen

⁴ Gemeint ist hier tatsächlich reines Ruby, nicht etwa das Web-Framework Ruby on Rails. Für Letzteres benötigen Sie entweder den speziellen Webserver Mongrel (wird mit openSUSE geliefert) oder das neue Apache-Modul Passenger (auch `mod_rails` genannt), das noch nicht zum Lieferumfang der Apache-Standard-Distribution gehört.

Sie sich `mod_ruby` separat besorgen oder aber den Ruby-Interpreter über die CGI-Schnittstelle extern ansprechen.

3. STANDARD-HOST: Hier werden die Voreinstellungen für den Haupt-Server vorgenommen. Die Vorgehensweise entspricht der weiter unten beschriebenen Registerkarte HAUPTHOST im Standard-Dialog.
4. VIRTUELLE HOSTS: An dieser Stelle können Sie beliebig viele virtuelle Hosts einrichten. Dieser Punkt entspricht der in Kürze beschriebenen Registerkarte HOSTS des Hauptdialogs.
5. ZUSAMMENFASSUNG: Die wichtigste Einstellung in diesem Schritt ist die Auswahl zwischen APACHE2-SERVER BEIM SYSTEMSTART STARTEN oder APACHE2-SERVER MANUELL STARTEN (Standard). Für den automatischen Start können Sie natürlich auch `chkconfig` oder den Runlevel-Editor verwenden. Außerdem finden Sie hier noch einmal alle Einstellungen der vorherigen Punkte auf einen Blick. Wählen Sie ZURÜCK, wenn Sie noch etwas ändern möchten, VERWERFEN, um den Dialog zu beenden, ohne die neuen Einstellungen zu übernehmen, oder BEENDEN, um die Änderungen zu speichern. Für zusätzliche Einstellungen können Sie auch auf EXPERTENKONFIGURATION klicken; es erscheint der im Folgenden beschriebene Dialog, der anschließend bei jedem weiteren Start des Apache-Konfigurations-Applets angezeigt wird.

Der Hauptdialog besitzt vier Registerkarten:

- ▶ Unter LAUSCHEN AUF PORTS UND ADRESSEN wird zunächst ausgewählt, ob Apache automatisch aktiv sein soll (AKTIVIERT) oder nur auf Anfrage gestartet werden soll (DEAKTIVIERT). Aber vor allem werden hier die TCP-Ports angegeben, unter denen der Apache verfügbar sein soll. Ein Standard-HTTP-Server lauscht am Port 80, sodass dieser vorkonfiguriert ist. Wenn Sie virtuelle Hosts für andere Ports einrichten möchten, müssen Sie diese mithilfe der Schaltfläche HINZUFÜGEN angeben. Auch wenn Apache für zusätzliche Protokolle konfiguriert werden soll, müssen Sie die entsprechenden Ports hier angeben, beispielsweise 443 für verschlüsseltes HTTPS über das Apache-Modul `mod_ssl`. Für jeden Port lässt sich optional die IP-Adresse einer Netzwerkschnittstelle auswählen; in diesem Fall steht der Server nur über diese Schnittstelle zur Verfügung. Standardmäßig werden dagegen ALLE ADRESSEN verwendet.
- ▶ SERVER-MODULE enthält eine Tabelle aller mit Apache installierten Zusatzmodule (Näheres dazu erfahren Sie im nächsten Unterabschnitt). Die Spalte NAME zeigt den Namen des jeweiligen Moduls (ohne das häufig, aber nicht immer verwendete Präfix `mod_`) an. STATUS gibt darüber Auskunft, ob das Modul zurzeit AKTIVIERT oder DEAKTIVIERT ist. Unter BESCHREIBUNG erhalten Sie schließlich eine kurze Information darüber, was das Modul tut. Um ein noch nicht aktives Modul zu aktivieren oder umgekehrt ein zurzeit aktives

auszuschalten, markieren Sie die entsprechende Zeile in der Tabelle, und klicken Sie auf **STATUS WECHSELN**. Mithilfe der Schaltfläche **MODUL HINZUFÜGEN** können Sie weitere Module auswählen, die sich im Module-Verzeichnis, aber noch nicht in der Tabelle befinden.

- ▶ Die Registerkarte **HAUPTHOST** enthält die Konfigurationseinstellungen des Haupt-Servers. Unter diesem Begriff versteht man die Standard-Website, die HTTP-Anfragen an diesen Host beantwortet. Der Gegenbegriff dazu sind virtuelle Hosts, das heißt mehrere Sites innerhalb desselben Apache-Webservers. Virtuelle Hosts werden von Webhostern verwendet, um die Sites verschiedener Kunden auf demselben Serverrechner abzulegen; sie werden aber auch benötigt, wenn Sie mehrere Protokolle verwenden möchten, etwa SSL oder Apache als Proxy-Server.

Jede Zeile in der Liste entspricht einer einzelnen Konfigurationsdirektive. Klicken Sie auf **BEARBEITEN** oder führen Sie einen Doppelklick aus, um den Wert der entsprechenden Einstellung zu verändern. Daraufhin wird ein zusätzliches Dialogfeld eingeblendet, in dem Sie den Wert der Direktive anpassen können. Mit **OK** führen Sie die gewünschte Änderung durch, während **VERWERFEN** alles beim Alten lässt.

Bei manchen Einträgen wie `Directory` oder `Location` handelt es sich um sogenannte Container-Direktiven, die ihrerseits mehrere Direktiven für ein Verzeichnis beziehungsweise einen URL-Pfad umschließen. Wenn Sie für solche Einträge auf **BEARBEITEN** klicken, öffnet sich kein zusätzliches Dialogfeld, sondern das YaST-Fenster zeigt die enthaltenen Konfigurationseinstellungen an. Wenn Sie mit deren Bearbeitung fertig sind, klicken Sie auf **OK**, um die Änderungen anzunehmen, oder auf **ZURÜCK**, um sie zu verwerfen; in beiden Fällen kehren Sie zum übergeordneten Konfigurationskontext zurück.

Über die Schaltfläche **HINZUFÜGEN** können Sie im jeweiligen Kontext eine weitere Direktive einfügen. Wählen Sie zunächst den Namen der gewünschten Direktive aus dem Pulldown-Menü aus, klicken Sie auf **OK**, geben Sie einen Wert ein, und verwenden Sie erneut **OK**. Um den Dialog unverrichteter Dinge wieder zu verlassen, können Sie jederzeit **VERWERFEN** drücken.

LÖSCHEN entfernt die ausgewählte Direktive nach einer Sicherheitsnachfrage. Beachten Sie, dass sich einige zentrale Direktiven wie `DOCUMENT ROOT` (Stammverzeichnis der Website) oder `E-MAIL DES SERVERADMINISTRATORS` (entspricht der Direktive `ServerAdmin`) nicht löschen lassen, weil sie für den ordnungsgemäßen Betrieb des Servers unerlässlich sind.

- ▶ Unter **Hosts** können Sie schließlich die Konfiguration der einzelnen virtuellen Hosts vornehmen. Auch dieses Verfahren verstehen Sie besser, wenn Sie sich mit der weiter unten beschriebenen manuellen Konfiguration auskennen.

Wenn Sie einen virtuellen Host erstellen möchten, klicken Sie auf HINZUFÜGEN. Sie gelangen in einen mehrseitigen Dialog, in dem Sie die Grundeinstellungen dieses Hosts eingeben müssen. Zu Referenzzwecken werden hier die Namen der zugehörigen Originaldirektiven angegeben, die Sie weiter unten im Abschnitt über die manuelle Konfiguration mit genaueren Beschreibungen wiederfinden.

Geben Sie auf der ersten Seite im Block SERVER-IDENTIFIKATION zunächst unter SERVERNAME (Direktive `ServerName`) den Hostnamen für den virtuellen Host ein, zum Beispiel `web2.test.local`. Dies garantiert nicht etwa die Auflösung dieses Namens im Netzwerk (dafür ist wie immer `/etc/hosts` oder DNS zuständig), sondern wird von Apache zur Identifikation des virtuellen Hosts benötigt. Unter ÜBERGEORDNETES VERZEICHNIS DER SERVER-INHALTE (Direktive `DocumentRoot`) können Sie mittels DURCHSUCHEN ein Verzeichnis wählen, das die entsprechende Website enthält. Beachten Sie, dass dieses Verzeichnis für das Webserver-Userkonto lesbar sein muss. Die ADMINISTRATOR-E-MAIL (Direktive `ServerAdmin`) dient dem Komfort Ihrer Besucher, aber auch Ihrer eigenen Sicherheit: Sie wird auf automatisch generierten Fehlermeldungsseiten angezeigt und kann von aufmerksamen Benutzern verwendet werden, um Ihnen per E-Mail eine entsprechende Mitteilung zu schicken. Für Internet-Websites wird in der Regel eine Adresse nach dem Schema `webmaster@Ihre-Domain` verwendet, während im Intranet Ihre gewöhnliche, interne E-Mail-Adresse ausreichen dürfte.

Besonders wichtig sind die Einstellungen im Block SERVER-AUFLÖSUNG: Sie bestimmen, wie Apache Anfragen an den fraglichen virtuellen Host von anderen HTTP-Anfragen unterscheidet. Dazu gibt es zwei verschiedene Methoden: ANFRAGE-SERVER DURCH HTTP-HEADER BESTIMMEN (Direktive `NameVirtualHost`) ergibt einen *namensbasierten virtuellen Host*, der durch einen Vergleich des eingestellten Servernamens mit dem Wert des HTTP-Anfrage-Headers `Host` (siehe oben im Abschnitt *HTTP-Grundlagen*) erkannt wird. Wenn Sie dagegen ANFRAGE-SERVER DURCH SERVER-IP-ADRESSE BESTIMMEN wählen, wird ein *IP-basierter virtueller Host* erzeugt. Dessen Zuständigkeit wird durch IP-Adresse und/oder TCP-Port ermittelt; in der Praxis kommt häufiger der TCP-Port zum Einsatz. Klicken Sie auf VIRTUALHOST-ID ÄNDERN, und geben Sie einen der folgenden Werte ein:

- ▶ Eine IP-ADRESSE, wenn Sie Anfragen tatsächlich nach den unterschiedlichen IP-Adressen Ihres Hosts unterscheiden möchten (selten). Alle konfigurierten Schnittstellenadressen Ihres Rechners können aus dem Pull-down-Menü gewählt werden.

- ▶ IP-ADRESSE:PORTNUMMER, wenn Sie ein bestimmtes Protokoll oder einen Zweitserver an einem anderen TCP-Port unter einer bestimmten IP-Adresse betreiben möchten (ebenfalls selten).
- ▶ *:PORTNUMMER wird für den recht häufigen Fall verwendet, dass ein weiterer TCP-Port an *allen* Schnittstellen bereitgestellt werden soll. Häufig sind beispielsweise *:443 für SSL, *:3128 für Apache als Proxy oder *:8080 für einen Sekundär- oder Intranet-Webserver.
- ▶ * (ohne eine Portnummer) – die Auswahl ALLE ADRESSEN – kommt schließlich bei einem normalen namensbasierten virtuellen Host zum Einsatz.

Beachten Sie, dass Sie jeden hier angegebenen TCP-Port (und gegebenenfalls auch die IP-Adressen) auf der Registerkarte LAUSCHEN AUF PORTS UND ADRESSEN eintragen müssen.

Auf der zweiten Seite des Dialogs werden einige Einstellungen vorgenommen, die das Verhalten des virtuellen Hosts bestimmen. Unter CGI-OPTIONEN wird festgelegt, ob der virtuelle Host CGI (Common Gateway Interface, klassische Webanwendungen über externe Skripte) unterstützen soll. Wenn Sie die Option CGI FÜR DIESEN VIRTUELLEN HOST AKTIVIEREN ankreuzen, müssen Sie unter CGI-VERZEICHNISPFAD ein CGI-Verzeichnis auswählen, das aus Sicherheitsgründen außerhalb des Website-Verzeichnisses selbst liegen sollte. Beide Einstellungen zusammen erzeugen eine `ScriptAlias`-Direktive, die CGI für das entsprechende Verzeichnis einschaltet und es unter dem URL-Pfad `/cgi-bin` in die Website einbindet.

Unter *SSL-Unterstützung* können Sie festlegen, dass der Server gesicherte HTTPS-Verbindungen über SSL/TLS unterstützen soll. Dies sollte in der Regel nur für einen virtuellen Host auf Port 443 geschehen. Dazu müssen Sie den Pfad zur Zertifikatdatei angeben; wie Sie diese mit `openssl` erstellen können, wird ausführlich auf meiner Website unter http://buecher.lingoworld.de/apache2/mod_ssl.html beschrieben.

Der VERZEICHNISINDEX (`DirectoryIndex`) gibt an, welche Datei(en) Apache als Startseite ausliefern soll, wenn ein Besucher ein Verzeichnis ohne Dateinamen anfordert. Sie können mehrere Dateien durch Leerzeichen getrennt angeben; Apache sucht in der angegebenen Reihenfolge nach einer Datei mit diesen Namen. Der Standardwert ist `index.html`; je nach Ihren Serversprachen sollten Sie beispielsweise auch `index.php` hinzufügen.

Wenn Sie schließlich die Option PUBLIC HTML AKTIVIEREN ankreuzen, kann jeder Benutzer in seinem Home-Verzeichnis eine persönliche Website betreiben, die unter `http://Hostname/~Username` veröffentlicht wird.

Klicken Sie zum Schluss auf **BEENDEN**, um Ihre Änderungen zu speichern und zu aktivieren, oder auf **ABBRECHEN**, um sie zu ignorieren.

Die Datei `/etc/sysconfig/apache2`

Bestimmte Aspekte der Apache-Konfiguration, die sich in der Regel auch über den soeben beschriebenen YaST-Dialog einstellen lassen, stehen unter openSUSE in der Datei `/etc/sysconfig/apache2`. Ihre Syntax entspricht den anderen in Kapitel 5, »Grafische Oberflächen«, angesprochenen Systemkonfigurationsdateien; sie ist nicht kompatibel zu den weiter unten behandelten Apache-eigenen Konfigurationsdateien.

Jeder Eintrag besitzt die Schreibweise `VARIABLE=WERT`; die durch `#` eingeleiteten Kommentare innerhalb der Datei erläutern jeweils Funktion, Datentyp und Standardwert. Hier folgt eine kurze Aufzählung der verschiedenen Variablen und ihrer Bedeutung; die meisten von ihnen entsprechen echten Apache-Konfigurationsdirektiven, die im Unterabschnitt »Apache-Konfiguration« besprochen werden.

- ▶ `APACHE_CONF_INCLUDE_FILES="Datei [Datei ...]"`
Hier können Sie eine jeweils durch Leerzeichen getrennte Liste zusätzlicher Apache-Konfigurationsdateien angeben, die durch `httpd.conf` importiert werden sollen.
- ▶ `APACHE_CONF_INCLUDE_DIRS="Verzeichnis [Verzeichnis ...]"`
Mit diesem Eintrag geben Sie ebenfalls eine durch Leerzeichen getrennte Liste von Verzeichnissen an, aus denen aber jeweils alle Konfigurationsdateien importiert werden sollen.
- ▶ `APACHE_MODULES="Modul [Modul ...]"`
Hier bestimmen Sie die Liste der Apache-Module, die dynamisch geladen werden.
- ▶ `APACHE_SERVER_FLAGS="Flag1 [Flag2 ...]"`
Hier können Sie eine Liste beliebiger Definitionsnamen einfügen. Diese werden verwendet, um optionale Abschnitte vom Typ `<IfDefine Flag> ... </IfDefine>` in Ihren Konfigurationsdateien zu aktivieren.
- ▶ `APACHE_HTTPD_CONF="Dateipfad"`
Wählen Sie den Dateipfad der zu verwendenden Konfigurationsdatei (standardmäßig `/etc/apache2/httpd.conf`).
- ▶ `APACHE_MPM="prefork|worker"`
Diese Option bestimmt das zu verwendende Multiprocessing-Modul (MPM); entweder das klassische, prozessbasierte `prefork` oder das thread-basierte `MPM worker`.

- ▶ `APACHE_SERVERADMIN="E-Mail-Adresse"`
Bestimmen Sie die E-Mail-Adresse des Administrators für Fehlermitteilungen.
- ▶ `APACHE_SERVERNAME="Hostname"`
Dieser Eintrag gibt den Hostnamen des Haupt-Servers an.
- ▶ `APACHE_START_TIMEOUT="Sekunden"`
Hier setzen Sie die Anzahl der Sekunden, bis das Startskript mit einer Fehlermeldung abbricht (dies entspricht keiner Apache-Direktive, sondern ist distributionsspezifisch).
- ▶ `APACHE_SERVERSIGNATURE="on|off|email"`
Die Fußzeile automatisch generierter Fehlermeldungsseiten (Server-Version, keine, Server-Version mit Link auf die E-Mail-Adresse des Administrators) wird mit dieser Option festgelegt.
- ▶ `APACHE_LOGLEVEL="Syslog-Level"`
Hier regeln Sie die Stufe, ab der Apache-Meldungen in die Error-Logdatei geschrieben werden.
- ▶ `APACHE_ACCESS_LOG="Dateipfad Dateischema"`
Mit diesem Eintrag können Sie den Pfad und das Eintragungsschema der Access-Logdatei bestimmen, die alle Serverzugriffe protokolliert.
- ▶ `APACHE_USE_CANONICAL_NAME="on|off|DNS"`
Hier legen Sie fest, wie Apache den Domainnamen für selbstreferenzierte URLs (etwa für Weiterleitungen und Ähnliches) generiert: aus der Server-Name-Einstellung (`on`), aus dem Host-Header der Anfrage (`off`) oder per Reverse-DNS-Lookup (`DNS`).
- ▶ `APACHE_SERVERTOKENS="Major, Minor, ..."`
Mit diesem Eintrag nehmen Sie Einfluss auf die Ausführlichkeit der Server-Identifikation im Server-Header und in der `ServerSignature` (die Liste der möglichen Werte und ihre Bedeutung finden Sie weiter unten in der Beschreibung der Originaldirektive `ServerTokens`).
- ▶ `APACHE_EXTENDED_STATUS="on|off"`
Bestimmen Sie hier, ob das Modul `mod_status` erweiterte oder nur einfache Server-Statusmeldungen liefern soll.
- ▶ `APACHE_BUFFERED_LOGS="on|off"`
Legen Sie fest, ob Logdateien im Arbeitsspeicher gepuffert (schneller) oder direkt auf die Datenträger geschrieben (sicherer) werden sollen.
- ▶ `APACHE_TIMEOUT="Sekunden"`
Dieser Eintrag zeigt die Gesamtwartezeit an, bis Apache eine Verbindung als gescheitert ansieht und schließt.

► `DOC_SERVER="yes|no"`

Soll der Webserver die SUSE-Dokumentation im Netzwerk veröffentlichen oder nicht? Wenn Sie hier "yes" eintragen, müssen Sie weitere Einstellungen in `/etc/sysconfig/susehelp` vornehmen.

15.2.4 Apache manuell installieren

Wenn Sie eine neuere Apache-Version benötigen als die in Ihrer Distribution mitgelieferte, können Sie die aktuelle Release auch selbst kompilieren. Besuchen Sie dazu die Website des Apache HTTP Server-Projekts (httpd.apache.org), und laden Sie sich die neueste Version herunter. Zurzeit (November 2009) ist dies 2.2.14.

Sollten Sie sich für Apache 2.0 statt 2.2 entscheiden (wofür es keinen vernünftigen Grund gibt), wählen Sie ebenfalls die neueste Release aus diesem Versionszweig; im Moment ist es 2.0.63.

Eine Neuinstallation von Apache 1.3 ist heutzutage eigentlich nicht mehr zu rechtfertigen, es sei denn, Sie sind auf ein sehr exotisches Drittanbietermodul angewiesen, das für die 2er-Versionen nicht verfügbar ist. Wählen Sie in einem solchen seltenen Fall die neueste 1.3er-Release, zurzeit 1.3.41.

Da Sie automatisch auf einen Mirror in Ihrer Nähe umgeleitet werden, sollten Sie die Integrität des Downloads sicherstellen, indem Sie seine MD5- oder PGP-Signatur mit den Angaben auf der Apache-Website vergleichen.

Zunächst gilt es, das `.tar.gz`- oder `.tar.bz2`-Paket zu entpacken, dies geschieht wie üblich mittels

```
# tar xzvf httpd-2.2.14.tar.gz
```

beziehungsweise

```
# tar xjvf httpd-2.2.14.tar.bz2
```

Wechseln Sie in das neue Verzeichnis, und rufen Sie das `configure`-Skript auf, wobei einige Optionen zu beachten sind. Eine vollständige Liste erhalten Sie über die Eingabe:

```
# ./configure --help |less
```

Folgende Einstellmöglichkeiten sind dabei besonders wichtig:

► `--prefix=/Verzeichnispfad`

Diese Einstellung bestimmt das übergeordnete Verzeichnis für die Apache-Installation. Das Standardverzeichnis ist `/usr/local/apache2`.

- ▶ `--enable-layout=Layoutname`
Diese Option erlaubt die Auswahl eines *Installationslayouts*; es legt die Verzeichnisse für die einzelnen Komponenten fest. Die Layout-Definitionen stehen in der Datei *layout.config*. Nach dem leicht verständlichen Schema in dieser Datei können Sie auch Ihre eigenen Layouts hinzufügen.
- ▶ `--with-mpm=MPM-Modul`
Mit dieser Einstellung wird das gewünschte Multi-Processing-Modul (Laufzeitverhalten) ausgewählt. Die Voreinstellung ist `prefork`. Solange Sie thread-sichere Module verwenden, kann `--with-mpm=worker` die bessere Wahl sein; dieses thread-basierte MPM ist speicherschonender. Wenn Sie PHP einsetzen, wird allerdings dringend `prefork` empfohlen, da diverse PHP-Module nicht threadsicher sind.
- ▶ `--enable-so`
Eine fast lebenswichtige Einstellung: Diese Option aktiviert die Unterstützung für Dynamic Shared Objects (DSOs), damit Sie Module dynamisch hinzufügen können. Glücklicherweise ist die Einstellung inzwischen Standard.
- ▶ `--enable-modules="modul1 modul2 ..."`
Diese Option kompiliert die angegebenen Module statisch ein. Die Modulnamen müssen hier ohne das Kürzel `mod_` angegeben werden – beispielsweise `"rewrite imap"` für `mod_rewrite` und `mod_imap`. Für einige Standardmodule gibt es auch die Kurzfassung `enable-modulname`. Eine Auflistung aller mit Apache gelieferten Module finden Sie in Tabelle 15.10.
- ▶ `--enable-mods-shared="modul1 modul2 ..."`
Diese Option kompiliert die angegebenen Module als DSOs. Dadurch können sie jederzeit wieder entfernt werden. Für wichtige Module existiert die alternative Schreibweise `enable-modulname=shared`.
- ▶ `--enable-modules=most` oder `enable-mods-shared=most`
Hier werden fast alle Module statisch beziehungsweise als DSOs kompiliert. Sie sollten stets `most` statt `all` wählen; dies lässt vor allem diejenigen Module weg, mit denen es beim Kompilieren Schwierigkeiten gäbe.
- ▶ `--disable-modules="modul1 modul2 ..."`
Mit dieser Option können Sie die angegebenen Module explizit ausschließen.

Mit dem folgenden Beispiel wird Apache 2 zur Kompilierung fast aller Module als DSOs und zur Installation im Standardverzeichnis `/usr/local/apache2` vorbereitet:

```
# ./configure --enable-layout=Apache --enable-so \
--enable-mods-shared=most
```

Nach dem Durchlauf von `configure`, der etliche Minuten dauern kann, müssen Sie (als `root!`) die beiden Befehle zur Kompilierung und Installation aufrufen:

```
# make
# make install
```

Modul	Automatisch aktiviert	Beschreibung
<code>mod_authz_host</code> (bis 2.0 <code>mod_access</code>)	ja	Durch dieses Modul können Sie Zugriffe auf bestimmte Verzeichnisse einer Website anhand von Hostname und IP-Adresse des anfragenden Clients steuern.
<code>mod_actions</code>	ja	Dieses Modul ermöglicht die automatische Auslösung von CGI-Handlern anhand der Dateiendung beziehungsweise des MIME-Types von Dateien.
<code>mod_alias</code>	ja	<code>mod_alias</code> ist das klassische Modul für die einfache Umwandlung von URLs. Erheblich mehr Möglichkeiten bietet das neue Modul <code>mod_rewrite</code> .
<code>mod_asis</code>	ja	Mithilfe dieses Moduls können Sie eine Datei an den Client senden, »wie sie ist«. Das bedeutet, dass Apache für eine solche Datei keine automatischen HTTP-Header generiert; diese können Sie flexibel selbst bereitstellen.
<code>mod_auth_basic</code>	ja	Mithilfe der durch dieses Modul gebotenen Basic-Authentifizierung können Sie Verzeichnisse einer Website schützen: Benutzer müssen sich durch Benutzername und Passwort anmelden, um Zugriff zu erhalten.
<code>mod_authn_anon</code>	nein	Bietet anonymen Zugriff mit dem Benutzernamen <code>anonymous</code> und der eigenen E-Mail-Adresse als Passwort. Diese Funktionalität wurde dem bekannten Anonymous FTP nachempfunden, hat aber auf einem Webserver in den meisten Fällen keinen Sinn.
<code>mod_authn_dbd</code>	nein	Liest das Vergleichspasswort für einen gegebenen Benutzernamen aus einer mittels <code>mod_dbd</code> hergestellten SQL-Datenbankverbindung.

Tabelle 15.10 Die mitgelieferten Module für Apache 2.2

Modul	Automatisch aktiviert	Beschreibung
mod_authn_dbm	nein	Diese Variante steigert die Effizienz, wenn Ihre Site viele Verzeichnisse mit unterschiedlichen Benutzername-Passwort-Paaren enthält: Die Anmeldedaten werden nicht in einer einfachen Passwortdatei gespeichert, sondern in datenbankähnlichen DBM-Dateien.
mod_auth_digest	nein	Das Modul verbessert die Sicherheit der Authentifizierung, indem es statt der BASIC-Authentifizierung mit base64-kodierten Klartextdaten MD5-Digests gemäß RFC 2617 verwendet.
mod_authnz_ldap	nein	Dieses Modul ermöglicht die Benutzer-Authentifizierung anhand existierender Daten aus LDAP-Verzeichnissen.
mod_authn_alias	nein	Bietet die Möglichkeit, verschiedene Authentifizierungs-Provider zu komplexeren Modellen zu verknüpfen.
mod_authn_default	nein	Steuert die Weitervermittlung der Authentifizierung an untergeordnete Module.
mod_authn_file	ja	Liest die Anmeldedaten aus einfachen Textdateien (bis Version 2.0 in mod_auth enthalten).
mod_authz_dbm	nein	Gruppenberechtigungen anhand von DBM-Dateien
mod_authz_default	nein	Steuert die Weitervermittlung der Zugriffskontrolle an untergeordnete Module.
mod_authz_groupfile	ja	Liest Gruppeninformationen aus einfachen Textdateien (bis Version 2.0 in mod_auth enthalten).
mod_authz_owner	nein	Zugriffsberechtigung auf der Basis von Dateieigentumsrechten
mod_authz_user	nein	Zugriffsberechtigungen anhand von Benutzernamen (bis Version 2.0 in mod_auth enthalten).

Tabelle 15.10 Die mitgelieferten Module für Apache 2.2 (Forts.)

Modul	Automatisch aktiviert	Beschreibung
<code>mod_autoindex</code>	ja	Das Modul stellt den Inhalt des angeforderten Verzeichnisses auf (in Grenzen) konfigurierbare Weise dar, wenn dieses Verzeichnis keine Indexseite enthält. Ideal für umfangreiche Download-Verzeichnisse, aber ein gewisses Sicherheitsrisiko, sofern es unkontrolliert aktiviert wird: Wenn ein Website-Autor den Namen der Startseite falsch schreibt oder in einem Unterverzeichnis keine Startseite verwendet, wird die Liste aller Dateien angezeigt – bei wenig sicherheitsbewussten Webmastern können dies Dateien sein, die einen externen Besucher nichts angehen!
<code>mod_bucketeer</code>	nein	Dieses Modul ist ein Beispiel für einen Ausgabefilter. Er führt bei Auftreten spezieller Sonderzeichen Buckets (interne Apache-Funktionen) aus.
<code>mod_cache</code>	nein	Dieses Modul konfiguriert Apache als Web-Cache – dies ist neben der Funktion als Anwendungs-Gateway (die durch <code>mod_proxy</code> zur Verfügung gestellt wird) die wichtigste Aufgabe eines Proxy-Servers. Provider-Module (zurzeit <code>mod_disk_cache</code> und <code>mod_mem_cache</code>) stellen den eigentlichen Zwischenspeicher bereit.
<code>mod_case_filter_in</code>	nein	Dieses Modul ist ein Beispiel für einen Eingabefilter: Es wandelt die Eingabe, das heißt den Body der Client-Anfrage (zum Beispiel Formulardaten), in Großbuchstaben um.
<code>mod_case_filter</code>	nein	Dies ist ein Beispiel für einen Ausgabefilter: Das Modul wandelt den Body der Server-Antwort vor dem Versand an den Client in Großbuchstaben um.
<code>mod_cern_meta</code>	nein	Dieses Modul implementiert META-Files des CERN HTTPd (nur aus Kompatibilitätsgründen). Besser <code>mod_headers</code> verwenden!

Tabelle 15.10 Die mitgelieferten Module für Apache 2.2 (Forts.)

Modul	Automatisch aktiviert	Beschreibung
mod_cgi	ja	Das Modul <code>mod_cgi</code> bietet die klassische Unterstützung für CGI-Skripte: Das entsprechende Programm wird als externer Prozess gestartet. Es enthält Formularaten als Eingabe; seine Ausgabe wird an den Client weitergeleitet.
mod_cgid	ja	Wenn ein thread-basiertes MPM-Modul verwendet wird, kann <code>mod_cgid</code> die Performance von CGI-Skripten verbessern, da sie von einem separaten Prozess als leichtgewichtige Threads ausgeführt werden.
mod_charset_lite	ja	Das Modul bietet Unterstützung für diverse nichtlateinische Zeichensätze und kann die entsprechende Konvertierung in bestimmten Fällen automatisch vornehmen.
mod_dav	nein	<code>mod_dav</code> richtet Apache 2 als WebDAV-Server ein: Mehrere Autoren können auf den Server ähnlich wie auf ein CVS-Repository zugreifen und ihre Änderungen mit Versionsinformationen versehen.
mod_dav_fs	nein	Bei installiertem <code>mod_dav</code> ermöglicht dieses Zusatzmodul den WebDAV-Zugriff auf das lokale Dateisystem.
mod_dav_lock	nein	Stellt Locking-Funktionalität für <code>mod_dav</code> zur Verfügung.
mod_dbd	nein	Eingebaute Unterstützung für SQL-Datenbankverbindungen; erste eingebaute Praxisanwendung ist <code>mod_auth_dbd</code> (siehe oben), ansonsten auch interessant für die Entwicklung eigener Module.
mod_deflate	nein	Dieses Modul ermöglicht die Auslieferung deflate-komprimierter Daten gemäß RFC, was bei modernen Browsern, die dies unterstützen, Netzwerkbandbreite spart.

Tabelle 15.10 Die mitgelieferten Module für Apache 2.2 (Forts.)

Modul	Automatisch aktiviert	Beschreibung
mod_dir	ja	Dieses Modul erledigt zwei wichtige Aufgaben: Es definiert den Namen der Startseite und ermöglicht die automatische Verarbeitung von Verzeichnis-URLs, die nicht mit einem / enden.
mod_disk_cache	nein	mod_cache benötigt ein Provider-Modul, das entscheidet, wo der Zwischenspeicher zur Verfügung gestellt wird. Dieses Modul realisiert ihn auf der Festplatte.
mod_dumpio	nein	Ermöglicht die Sammlung der vollständigen Ein- und/oder Ausgabedaten in einer Logdatei.
mod_echo	nein	Dieses Modul ist als Demonstration der Multiprotokoll-Fähigkeiten von Apache 2 gedacht und sollte normalerweise nicht auf einem Produktions-Server verwendet werden. Es konfiguriert Apache zusätzlich zu HTTP als Server für das ECHO-Protokoll, das die übergebenen Daten zurücksendet.
mod_env	ja	Das Modul ermöglicht die Manipulation von Server-Umgebungsvariablen für bestimmte Anfragen und Verzeichnisse.
mod_example	nein	Das Modul mod_example bietet keine praktische Funktionalität, sondern demonstriert allgemein, wie Apache-Module überhaupt funktionieren. Es sollte niemals auf Produktions-Servern eingesetzt werden!
mod_expires	nein	Das Modul ermöglicht das Setzen des HTTP/1.1-Headers Expires, der die Gültigkeitsdauer von Dokumenten für Browser und vor allem Proxies angibt.
mod_ext_filter	nein	Über dieses Modul können Sie fast beliebige externe Programme als Ausgabefilter definieren. Dies ermöglicht beispielsweise Zeichensatz- oder gar Inhaltskonvertierungen.

Tabelle 15.10 Die mitgelieferten Module für Apache 2.2 (Forts.)

Modul	Automatisch aktiviert	Beschreibung
mod_filter		Ermöglicht eine genauere Kontrolle des Filterprozesses – vor allem die Steuerung der Filterabfolge (Filter Chain).
mod_file_cache	nein	Mithilfe dieses Moduls kann Apache 2 bestimmte Dateien beim Start ins RAM kopieren. Dies ermöglicht den schnelleren Zugriff auf häufig verwendete Daten.
mod_headers	nein	Über <code>mod_headers</code> können Sie fast alle HTTP-Header setzen, ändern oder entfernen. Nur einige wenige Header sind davon ausgenommen, weil sie erst nach der Verarbeitung der Direktiven dieses Moduls automatisch gesetzt werden.
mod_ident		Abfrage der RFC1413-Identität des Client-Hosts (bis Version 2.0 teilweise im Core)
mod_imagemap (bis 2.0 mod_imap)	ja	Dieses Modul bietet interne Unterstützung für die (selten gewordenen) serverseitigen Image Maps.
mod_include	ja	Mithilfe von <code>mod_include</code> wird die grundsätzliche Unterstützung für Server Side Includes bereitgestellt. Diese relativ alte Technologie ermöglicht das schnelle Einfügen dynamischer Daten wie Dateien, Variablen oder Datum und Uhrzeit.
mod_info	nein	Dieses Modul ermöglicht die Einrichtung einer virtuellen URL, über die sich ausführliche Konfigurationsinformationen des Servers abfragen lassen.
mod_isapi	nein	Unterstützung für die vom Microsoft Internet Information Server bekannte ISAPI-Schnittstelle (nur Windows)

Tabelle 15.10 Die mitgelieferten Module für Apache 2.2 (Forts.)

Modul	Automatisch aktiviert	Beschreibung
mod_ldap	nein	Das Lightweight Directory Access Protocol hat sich in den letzten Jahren zum Standard für Verzeichnisdienste entwickelt; sowohl UNIX-Implementierungen wie OpenLDAP als auch Microsoft ActiveDirectory basieren darauf. Das Modul verbessert die grundsätzliche Zusammenarbeit von Apache mit LDAP-Diensten und stellt so beispielsweise Caching- und Connection-Pooling-Dienste für LDAP bereit, insbesondere für mod_authnz_ldap.
mod_log_config	ja	Bereits ab Werk liefert Apache 2 recht aussagekräftige Logdateien. Mithilfe von mod_log_config können Sie noch allerlei Erweiterungen daran vornehmen und die Dateien durch externe Programme sortieren oder weiterverarbeiten lassen.
mod_log_forensic	nein	Stellt forensische Logdateien bereit, die vollständige Client-Anfragen und deren korrekte Verarbeitung protokollieren – dies vereinfacht die Suche nach Fehlern oder Angriffen.
mod_logio	nein	Dieses Modul protokolliert zusätzlich zu den Standardinformationen die Anzahl der tatsächlich gesendeten beziehungsweise empfangenen Bytes in Logdateien.
mod_mem_cache	nein	Ähnlich wie bei mod_disk_cache können Sie mithilfe dieses Moduls den Zwischenspeicher für mod_cache vorkonfigurieren. Hier wird er im RAM eingerichtet.
mod_mime	ja	Das Modul setzt den wichtigen Content-Type-Header der Server-Antwort automatisch anhand der Dateinamenerweiterung.
mod_mime_magic	nein	mod_mime_magic untersucht einige Bytes des Datei-Inhalts, um den MIME-Type für die Antwort bei Bedarf noch sorgfältiger zu ermitteln als mod_mime.

Tabelle 15.10 Die mitgelieferten Module für Apache 2.2 (Forts.)

Modul	Automatisch aktiviert	Beschreibung
mod_negotiation	ja	Dieses Modul ist für die verschiedenen Arten der Content-Negotiation – die automatische Lieferung unterschiedlicher Inhalte als Reaktion auf Accept*-Anfrage-Header – zuständig.
mod_nw_ssl	nein	gesicherte SSL-Verbindungen für die Apache-Version unter Novell Netware
mod_proxy	nein	Dieses Modul stellt die Grundfunktion für Apache als Proxy-Server zur Verfügung. Einige weitere Module richten spezielle Proxy-Dienste wie HTTP- oder FTP-Proxy ein.
mod_proxy_ajp	nein	In Zusammenarbeit mit mod_proxy stellt dieses Modul einen Proxy-Dienst für das Apache JServ Protocol (AJP) Version 1.3 bereit, das unter anderem von Tomcat verwendet wird.
mod_proxy_balancer	nein	Erweitert mod_proxy um Load-Balancing-Funktionalität für HTTP, FTP und AJP.
mod_proxy_connect	nein	Zusammen mit mod_proxy stellt dieses Modul eine Implementierung der HTTP-Methode CONNECT zur Verfügung: Der Proxy-Server überträgt die Daten gesicherter Verbindungen (SSL), ohne ihren Inhalt – den er ohnehin nicht verstehen kann – zu modifizieren.
mod_proxy_ftp	nein	Das Modul stellt einen FTP-Proxy-Server bereit, der zusammen mit mod_proxy funktioniert.
mod_proxy_http	nein	Wenn mod_proxy installiert ist, bietet dieses Modul die wichtigste Komponente eines Web-Proxy-Servers, nämlich einen einfachen HTTP-Proxy.
mod_rewrite	nein	Dieses Modul ermöglicht die fast beliebige Umwandlung von URLs aus HTTP-Anfragen mithilfe regulärer Ausdrücke.
mod_setenvif	ja	Das Modul ermöglicht die Manipulation von Server-Umgebungsvariablen als Reaktion auf bestimmte HTTP-Anfrage-Header.

Tabelle 15.10 Die mitgelieferten Module für Apache 2.2 (Forts.)

Modul	Automatisch aktiviert	Beschreibung
mod_spelling	nein	Dieses Modul kümmert sich um die automatische Korrektur von Schreibfehlern in Anfrage-URLs, vor allem bezüglich Groß- und Kleinschreibung. Das Modul heißt tatsächlich <code>mod_spelling</code> , mit einem <code>l</code> (ist eben ein Rechtschreibfehler).
mod_ssl	nein	<code>mod_ssl</code> stellt die grundsätzliche Unterstützung für gesicherte HTTP-Verbindungen über SSL beziehungsweise TLS zur Verfügung.
mod_status	ja	Das Modul liefert über eine virtuelle URL oder mithilfe des Textbrowsers Lynx und einer Kommandozeilenoption für das Programm <code>httpd</code> Statusinformationen über den laufenden Server.
mod_suexec	nein	<code>suexec</code> ermöglicht es, CGI-Skripte unter einer bestimmten User- und Group-ID auszuführen, was bei sorgfältiger Konfiguration die Sicherheit verbessert.
mod_unique_id	nein	Dieses Modul ermöglicht es, jeder einzelnen Client-Anfrage eine eindeutige ID zuzuweisen. Dies kann für bestimmte Aspekte der serverseitigen Programmierung und für Logging-Zwecke hilfreich sein.
mod_userdir	ja	Das Modul bildet die Home-Verzeichnisse der Benutzer (beziehungsweise ein konfigurierbares Unterverzeichnis davon) nach dem Schema <code>http://servername/~user</code> automatisch auf den Webserver ab.
mod_usertrack	nein	Dieses Modul ermöglicht das Session-Tracking durch Cookies. Sie müssen die Genehmigung der Besucher einholen, um es zu verwenden. Außerdem ist der Nutzen fraglich, da etliche Benutzer die Annahme von Cookies verweigern.
mod_version	ja	Bietet die Möglichkeit, Konfigurationsanweisungen versionsabhängig zu setzen.

Tabelle 15.10 Die mitgelieferten Module für Apache 2.2 (Forts.)

Modul	Automatisch aktiviert	Beschreibung
mod_vhost_alias	nein	Dies ist das wichtigste Modul für Hosters: Es ermöglicht die automatische Erzeugung von Unmengen virtueller Hosts anhand von Datei- und Verzeichnismustern.

Tabelle 15.10 Die mitgelieferten Module für Apache 2.2 (Forts.)

15.2.5 Apache steuern

Wenn die Installation abgeschlossen ist, lässt sich Apache am einfachsten über das Skript `apachectl` im *bin*-Verzeichnis Ihrer Installation steuern. Folgende Optionen sind besonders interessant:

- ▶ `apachectl start`
Diese Option startet den Server.
- ▶ `apachectl stop`
Mit dieser Anweisung wird Apache beendet.
- ▶ `apachectl graceful-stop` (nur 2.2)
Diese Option stoppt den Server ebenfalls; es wird aber zunächst abgewartet, bis alle aktuellen Client-Anfragen verarbeitet wurden.
- ▶ `apachectl restart`
Diese Option sorgt für einen Neustart – wichtig nach jeder Konfigurationsänderung.
- ▶ `apachectl graceful`
Hiermit bewirken Sie ebenfalls einen Neustart; Apache wartet aber auf den Abschluss der aktuellen Transaktionen. Beachten Sie, dass manche Konfigurationsänderungen einen richtigen Neustart benötigen.

Da `apachectl` die Parameter `start` und `stop` kennt, besitzt es das richtige Format für ein Startskript. Daher können Sie es nach `/etc/init.d` linken:⁵

```
# ln -s /usr/local/apache2/bin/apachectl /etc/init.d/apache2
```

⁵ Falls das Apache-Paket der Distribution installiert ist, heißt dessen Startskript bereits »apache2«. Nennen Sie den neuen Symlink in diesem Fall einfach anders, zum Beispiel »apache22«, und vergessen Sie nicht, den alten Starteintrag mittels `chkconfig -d` oder im Runlevel-Editor zu deaktivieren.

Anschließend können Sie den Runlevel-Editor verwenden oder folgende Anweisung eingeben, um Apache künftig beim Systemstart automatisch mitzustarten:

```
# chkconfig -a apache2
```

15.2.6 Apache-Konfiguration

Die Konfiguration des Apache-Webservers erfolgt, wie bereits erwähnt, durch zahlreiche verschiedene Konfigurationsdirektiven. Diese werden je nach Version und Distribution in verschiedene Konfigurationsdateien geschrieben:

- ▶ Apache 2.0 benutzt bei einer Standardinstallation nur eine einzige Konfigurationsdatei für alle Einstellungen. Sie heißt *httpd.conf* und liegt im *conf*-Verzeichnis der Apache-Installation.
- ▶ In Apache 2.2 wurde ein wenig umgeräumt: Die Hauptkonfigurationsdatei *httpd.conf* bindet hier zahlreiche, nach Themen beziehungsweise Modulen sortierte Unterkonfigurationsdateien ein, die im Unterverzeichnis *extra* liegen. Im Einzelnen handelt es sich um folgende Dateien:
 - ▶ *httpd-mpm.conf* – MPM-spezifische Einstellungen, die vor allem die Performance betreffen
 - ▶ *httpd-multilang-errordoc.conf* – mehrsprachige Fehlermeldungsseiten
 - ▶ *httpd-autoindex.conf* – Optionen für automatisch generierte Verzeichnis-Indizes
 - ▶ *httpd-languages.conf* – Einstellungen für die Zuordnung zwischen zusätzlichen Dateiendungen und Dokumentsprachen
 - ▶ *httpd-userdir.conf* – Veröffentlichung benutzerspezifischer Websites unter *~/user*-URL-Pfaden
 - ▶ *httpd-info.conf* – Informationen und Statusmeldungen des Servers unter speziellen URLs veröffentlichen (*mod_info* und *mod_status*)
 - ▶ *httpd-vhosts.conf* – Einstellungen für virtuelle Hosts
 - ▶ *httpd-manual.conf* – Einbindung der Apache-Dokumentation unter dem URL-Pfad */manual*
 - ▶ *httpd-dav.conf* – WebDAV (Web-based Distributed Authoring And Versioning, eine Art Web-Repository, das bei manchen Hostern bequemerweise FTP ersetzt und auf dem unter anderem auch Subversion aufsetzt)
 - ▶ *httpd-default.conf* – einige zusätzliche Einstellungen für den Haupt-Server
 - ▶ *httpd-ssl.conf* – SSL-Konfiguration
- ▶ Die Anordnung der Apache-Konfigurationsdateien in openSUSE wurde bereits weiter oben beschrieben.

Allgemeines über Konfigurationsdirektiven

Die Apache-Direktiven besitzen jeweils folgendes Format:

```
Direktive Wert [Wert ...]
```

Daneben gibt es zusätzlich sogenannte Container-Bereiche wie `<Directory> ... </Directory>`, die die Wirkung der umschlossenen Direktiven etwa auf ein einzelnes Verzeichnis, einige Dateien oder bestimmte HTTP-Methoden beschränken.

Im Folgenden werden nur kurz die allerwichtigsten Direktiven beschrieben. Zu jeder Direktive wird das Modul angegeben, in dem sie definiert ist (`core` ist die immer verfügbare Grundausstattung). Darüber hinaus werden die folgenden Buchstaben verwendet, um die Kontexte anzuzeigen, in denen eine Direktive stehen darf:

- ▶ **S** – Server-Kontext
Die Direktive steht nicht in einem Container und gilt serverweit.
- ▶ **V** – virtueller Host
Die Direktive kann in einem `<VirtualHost>`-Container stehen.
- ▶ **D** – Verzeichnis-Kontext
Die Direktive gilt für einen Verzeichnisbereich und darf sich in einem `<Directory>`-, `<Location>`- oder `<Files>`-Container befinden.
- ▶ **H** – `.htaccess`-Kontext
die Direktive darf in einer `.htaccess`-Datei stehen, über die Sie Konfigurationsdaten in die einzelnen Verzeichnisse der Website selbst auslagern können.
.htaccess-Dateien sind eigentlich nur erforderlich, wenn Sie Webspace ohne eigenen Server gemietet haben beziehungsweise umgekehrt virtuelle Hosts für andere User bereitstellen – auf Ihrem eigenen Server gehören in der Regel alle Direktiven in die `httpd.conf` beziehungsweise die verknüpften Konfigurationsdateien.

Weitere Informationen finden Sie in der Apache-Dokumentation (<http://httpd.apache.org/docs-2.2>) sowie in der Direktivenliste auf der Site zu meinem Apache-Buch (<http://buecher.lingoworld.de/apache2/dirs.php>).

Wichtige Direktiven im Überblick

- ▶ `Alias` URL-Pfad Verzeichnispfad
Kontext: SV; Modul: `mod_alias`
Bindet einen Verzeichnispfad, der eigentlich außerhalb der Website liegt, unter dem URL-Pfad ein. Beispiel:
`Alias /test /var/daten/test`

Nun kann mittels `http://Server/test/...` auf die Inhalte des Verzeichnisses `/var/daten/test` zugegriffen werden.

- ▶ `Allow from all` | IP-Adresse | Hostname

Kontext: DH; Modul: `mod_authz_host`

Rechner mit den genannten Adressen dürfen auf den Verzeichnisbereich zugreifen, in dem die Direktive steht. `Allow` arbeitet in Kombination mit `Order` und `Deny` (siehe unten). Das folgende Beispiel erlaubt allen Rechnern den Zugriff auf den URL-Pfad `/info`:

```
<Location /info>
    Order allow,deny
    Allow from all
</Location>
```

- ▶ `AllowOverride None` | `All` | Direktiventyp [Direktiventyp ...]

Kontext: D; Modul: `core`

Bestimmt, welche Direktiven innerhalb eines Verzeichnisbereiches in `.htaccess`-Dateien überschrieben werden dürfen. `All` steht für alle Arten von Direktiven; wichtige Einzeltypen sind etwa `AuthConfig` (Authentifizierungseinstellungen) oder `FileInfo` (Datei-Informationen wie etwa MIME-Einstellungen). Wenn Sie keine `.htaccess`-Dateien brauchen, empfiehlt es sich, sie aus Sicherheits- und Performance-Gründen über den Wert `None` ganz abzuschalten.

- ▶ `AuthBasicProvider provider`

Kontext: DH; Modul: `mod_auth_basic`

Gibt an, welches Provider-Modul (`mod_authn_*`) die Vergleichsdaten für die klartextbasierte Authentifizierung bereitstellen soll. Das folgende Beispiel wählt einfache Textdateien (Modul `mod_authn_file`):

```
AuthBasicProvider file
```

- ▶ `AuthName Bereichsname`

Kontext: DH; Modul: `core`

Legt den Namen eines Authentifizierungsbereiches (»Realm«) fest. Browser zeigen diesen Bereichsnamen an, damit Benutzer wissen, welche Anmeldedaten sie eingeben müssen.

- ▶ `AuthType Basic` | `Digest`

Kontext: DH; Modul: `core`

Bestimmt, wie Browser und Server Anmeldedaten austauschen: `Basic` bedeutet im Klartext, `Digest` verschlüsselt. Für Letzteres ist das Modul `mod_auth_digest` zuständig.

- ▶ `AuthUserFile` Dateipfad

Kontext: DH; **Modul:** `mod_authn_file`

Verweist auf eine Textdatei, in der Benutzerdaten zur Überprüfung bei der Anmeldung gespeichert sind. Eine solche Datei lässt sich über das mit Apache gelieferte Kommandozeilenprogramm `htpasswd` erzeugen. Beispiel:

```
# htpasswd [-c] .htuser username
```

Dieses Kommando fragt zweimal nach dem neuen Passwort für `username` und speichert es verschlüsselt in der Datei `.htuser` im aktuellen Verzeichnis. Die Option `-c` (`create`) dient dazu, die entsprechende Datei neu anzulegen; dies ist beim ersten User erforderlich.

Der Dateipfad gilt als relativ zur `ServerRoot` (siehe unten), wenn Sie keinen absoluten Pfad angeben. Das folgende Beispiel wählt die Datei `.htuser` aus dem Verzeichnis `credentials` im Apache-Installationsverzeichnis als Datenquelle:

```
AuthUserFile credentials/.htuser
```

Aus Sicherheitsgründen darf sich die angegebene Datei auf keinen Fall innerhalb des Website-Verzeichnisses selbst befinden!

- ▶ `Deny from all` | IP-Adresse | Hostname

Kontext: DH; **Modul:** `mod_authz_host`

Verbietet den angegebenen Rechnern den Zugriff auf das Verzeichnis, in dem die Direktive steht. Das folgende Beispiel schließt alle Rechner vom URL-Pfad `/geheim` aus:

```
<Location /geheim>
    Order deny,allow
    Deny from all
</Location>
```

- ▶ `<Directory Verzeichnis> ... </Directory>`

Kontext: SV; **Modul:** `core`

Umschließt Einstellungen für ein bestimmtes Verzeichnis der Website. Das folgende Beispiel verbietet `.htaccess`-Dateien auf dem gesamten Server (kann für einzelne Verzeichnisse wieder überschrieben werden):

```
<Directory />
    AllowOverride None
</Directory>
```

- ▶ `DirectoryIndex` Dateiname [`Dateiname ...`]

Kontext: SVDH; **Modul:** `mod_dir`

Gibt Namen von Dokumenten an, nach denen Apache in der angegebenen Reihenfolge suchen soll, wenn nur ein Verzeichnis angefordert wurde (Startseiten). Die Voreinstellung ist `index.html`; das folgende Beispiel erlaubt zusätzlich `index.htm` und `index.php`:

```
DirectoryIndex index.html index.htm index.php
```

- ▶ `DocumentRoot` Verzeichnis

Kontext: SV; **Modul:** core

Wurzelverzeichnis der Website, dem der URL-Pfad / zugeordnet ist. Die Voreinstellung ist `/htdocs` im Apache-Verzeichnis. Beispiel:

```
DocumentRoot /usr/local/apache2/htdocs
```

- ▶ `Listen` [IP-Adresse:]Port

Kontext: S; **Modul:** MPM-Module

Gibt den TCP-Port an, an dem Apache lauscht. Wenn Sie IP-Adressen hinzufügen, wartet er nur an einzelnen Netzwerkschnittstellen auf Verbindungen. Standard:

```
Listen 80
```

Für andere Verwendungszwecke benötigen Sie mehrere `Listen`-Direktiven, etwa Port 443 für verschlüsselte SSL-Verbindungen über `mod_ssl`.

- ▶ `LoadModule` Modulname Dateipfad

Kontext: S; **Modul:** `mod_so`

Wenn Module als Dynamic Shared Objects (DSOs) kompiliert wurden, müssen sie mithilfe dieser Direktive geladen und dadurch aktiviert werden. Der Modulname ist in der Regel der Grundname mit angehängtem `_module` (bei `mod_autoindex` also zum Beispiel `autoindex_module`); der Dateipfad verweist in der Regel auf das Verzeichnis `modules` relativ zur `ServerRoot` (siehe unten). Beispiel:

```
LoadModule dir_module modules/mod_dir.so
```

- ▶ `<Location URL-Pfad> ... </Location>`

Kontext: SV; **Modul:** core

Enthält Einstellungen für einen bestimmten URL-Pfad. Das folgende Beispiel stellt unterhalb von `http://Server/test` die Startseite `start.html` ein:

```
<Location /test>
    DirectoryIndex start.html
</Location>
```

- ▶ `NameVirtualHost IP-Adresse[:Port]`

Kontext: S; Modul: core

Konfiguriert eine Adresse (und optional einen TCP-Port) für den Einsatz namensbasierter virtueller Hosts. Als Adresse sollten Sie bevorzugt eine IP-Adresse angeben; wenn Sie einen Hostnamen verwenden, kann es zu Problemen kommen, sobald ein DNS-Server nicht erreichbar ist. Wenn alle Netzwerkschnittstellen angesprochen werden sollen, können Sie statt einer konkreten Adresse `*` benutzen.

Beachten Sie, dass es für die angegebene IP-Adresse keinen Haupt-Server mehr gibt – Sie müssen die bisherige Haupt-Server-Konfiguration in einen `<VirtualHost>`-Container (siehe unten) verlagern. Beispiele:

```
NameVirtualHost 195.23.17.42
NameVirtualHost *:8080
```

- ▶ `Options None | All | Optionstyp [Optionstyp ...]`

Kontext: SVDH; Modul: core

Setzt Verzeichnisoptionen. `None` verbietet alle Optionen, `All` erlaubt sie alle.⁶ Wichtige Einzeloptionen sind etwa `Indexes` (automatische Erzeugung einer Indexseite durch `mod_autoindex`) oder `ExecCGI` (Ausführung von CGI-Skripten).

In untergeordneten Kontexten können Sie übrigens die Schreibweise `+Optionstyp` oder `-Optionstyp` verwenden, um gezielt Optionen zu den Einstellungen des übergeordneten Kontextes hinzuzufügen beziehungsweise daraus zu entfernen.

- ▶ `Order allow,deny | deny,allow`

Kontext: DH; Modul: `mod_authz_host`

Bestimmt die Reihenfolge, in der `Allow` und `Deny` ausgewertet werden. Betrachtet den vorderen Wert als Voreinstellung (meist mit dem Wert `All`) und den hinteren als Ausnahme. Beachten Sie, dass in `allow,deny` beziehungsweise `deny,allow` kein Leerzeichen hinter dem Komma stehen darf. Das folgende Beispiel erlaubt nur Rechnern aus dem Netzwerk `192.168.0.0/24` den Zugriff auf den URL-Pfad `/intranet`:

```
<Location /intranet>
    Order deny,allow
    Deny from all
```

⁶ Die Option `MultiViews`, die anhand von Dateierendungen automatisch mehrere Sprach-, Zeichensatz- oder Dateitypversionen für Content-Negotiation anbietet, ist nicht in `All` enthalten und muss bei Bedarf immer explizit angegeben werden.

```
    Allow from 192.168.0
</Location>
```

- ▶ `Redirect [Status] URL-Pfad URL`
Kontext: SVDH; Modul: `mod_alias`

Anfragen für den angegebenen URL-Pfad werden an die (externe) URL weitergeleitet. Das folgende Beispiel leitet alle Anfragen für `http://Server/hier` an entsprechende Ressourcen unter `http://woanders.com` weiter:

```
Redirect /hier http://woanders.com
```

Beachten Sie, dass der Server die Weiterleitung nicht stillschweigend selbst durchführt, sondern den anfragenden Browser per `3xx-Header` und `Location-Header` (beides wurde weiter oben in diesem Kapitel besprochen) darüber informiert.

- ▶ `Require user Username | group Groupname | valid-user`
Kontext: DH; Modul: `core`

Gibt bei einer Authentifizierung an, welche Benutzer (`user`) oder Gruppen (`group`) sich anmelden dürfen. `valid-user` steht für alle Benutzer aus der aktuellen Datenquelle, zum Beispiel dem `AuthUserFile` (siehe oben).

- ▶ `Satisfy Any | All`
Kontext: DH; Modul: `core`

Wenn Sie Authentifizierung und Zugriffskontrolle per IP-Adresse für das gleiche Verzeichnis verwenden möchten, gibt `Satisfy` an, ob beide Kriterien erfüllt sein müssen (`All`) oder ob eines von ihnen genügt (`Any`). Die Voreinstellung ist `All`.

- ▶ `ScriptAlias URL-Pfad Verzeichnispfad`
Kontext: SV; Modul: `mod_alias`

Genau wie bei `Alias` wird der externe Verzeichnispfad unter dem angegebenen URL-Pfad eingebunden. Zusätzlich werden die Dateien im Verzeichnis als CGI-Skripte betrachtet. Das heißt, sie werden ausgeführt, und ihre Ausgabe wird als dynamisches Dokument an den Client gesendet. Die übliche Voreinstellung bindet ein Verzeichnis namens `cgi-bin` ein:

```
ScriptAlias /cgi-bin /usr/local/apache2/cgi-bin
```

- ▶ `ServerAdmin E-Mail`
Kontext: SV; Modul: `core`

Die E-Mail-Adresse des Server-Administrators, damit Benutzer Fehlermeldungen direkt an den Webmaster senden können. Beispiel:

```
ServerAdmin webmaster@test.local
```

- ▶ `ServerName Domainname`

Kontext: SV; Modul: core

Gibt den Domainnamen des Servers oder des virtuellen Hosts an. Beachten Sie, dass der Server in der Öffentlichkeit nur dann tatsächlich unter diesem Namen verfügbar ist, wenn entsprechende Nameserver-Einträge existieren. Die nötigen Einzelheiten zur BIND-Konfiguration wurden bereits in Kapitel 13, »Netzwerkgrundlagen«, erläutert. Beispiel:

```
ServerName www.test.local
```

- ▶ `ServerRoot Verzeichnispfad`

Kontext: S; Modul: core

Das Dachverzeichnis, unter dem Apache installiert ist. Die Verzeichnisangaben in vielen Direktiven können relativ zu dieser Einstellung sein. Hier ein Beispiel:

```
ServerRoot /usr/local/apache2
```

- ▶ `ServerSignature On|Off|E`

Kontext: SVDH; Modul: core

Legt fest, ob Apache in automatisch generierte Dokumente wie Index- oder Fehlermeldungsseiten eine Fußzeile einfügen soll, die Informationen über die Server-Software, den Hostnamen und eventuell die E-Mail-Adresse des Administrators enthält. Es gibt drei mögliche Werte: `Off` – keine Informationsfußzeile; `On` – Fußzeile ohne E-Mail-Adresse (Beispiel: Apache/2.2.9 (Unix) Server at www.test.local Port 80); `EMail` – wie `On`, aber als Link auf die `ServerAdmin`-E-Mail-Adresse.

- ▶ `ServerTokens Major|Minor|Minimal|ProductOnly|OS|Full`

Kontext: S; Modul: core

Bestimmt, wie ausführlich die Server-Software im Server-Header und auf automatisch generierten Seiten genannt wird. Angenommen, es handelt sich um Apache 2.2.9 auf einem UNIX-System, dann lautet `ProductOnly: Apache, Major: Apache/2, Minor: Apache/2.2, Minimal: Apache/2.2.9` und `OS: Apache/2.2.9 (Unix)`. `Full` hängt noch die Selbstbeschreibungen diverser Module an, zum Beispiel `Apache/2.2.9 (Unix) Dav/2 PHP/5.3.0`.

- ▶ `<VirtualHost Host[:Port]> ... </VirtualHost>`

Kontext: S; Modul: core

Enthält Einstellungen für einen virtuellen Host, angegeben durch IP-Adresse, Hostname und/oder TCP-Port. Beachten Sie, dass die zugehörigen Listen-Direktiven (siehe oben) im Server-Kontext stehen müssen. Näheres über dieses komplexe Thema erfahren Sie in der Beschreibung dieser Direktive auf der

Site zu meinem Buch »Apache 2«: <http://buecher.lingoworld.de/apache2/show-dir.php?id=757>.

Tipps zur Grundkonfiguration

Hier noch einige Hinweise zu guten Voreinstellungen für die Verzeichnisse eines Produktions-Servers. Das Ziel ist ein vernünftiges Maß an Sicherheit, ohne die Benutzbarkeit allzu sehr einzuschränken.

Zunächst sollten Sie sehr restriktive Voreinstellungen für alle Verzeichnisse des Server-Hosts vornehmen:

```
<Directory />
  # Alle Optionen ausschalten
  Options None
  # .htaccess ausschalten (für Sicherheit & Performance)
  AllowOverride None
  # ALLE Zugriffe verbieten
  Order deny,allow
  Deny from all
</Directory>
```

Anschließend werden die Einstellungen für die Website selbst, also die DocumentRoot, gesetzt. Das folgende Beispiel erlaubt alle Zugriffe und einige Optionen, lässt aber die Einstellung für *.htaccess*-Dateien durch einfache Nicht-erwähnung bestehen:

```
<Directory /usr/local/apache2/htdocs>
  # Einige Optionen aktivieren
  Options Indexes FollowSymLinks
  # Alle Zugriffe gestatten
  Order deny,allow
  Allow from all
</Directory>
```

Zu guter Letzt noch ein Beispiel für ein geschütztes Verzeichnis, dessen Inhalte Benutzer nur nach Anmeldung mit Benutzername und Passwort aufrufen dürfen. Es geht davon aus, dass sich die gewünschten Benutzerdaten in der Datei *.htuser* im *Apache-conf*-Verzeichnis befinden:

```
<Location /privat>
  AuthType Basic
  AuthBasicProvide file
  AuthName "Privater Bereich"
  AuthUserFile conf/.htuser
  Require valid-user
</Location>
```

Die Zeile `AuthBasicProvider file` müssen Sie bei Apache 2.0 weglassen; in 2.2 wurde die Flexibilität der Authentifizierungsmodule durch ein Provider-Modell verbessert: Die Client-Kommunikation erledigen `mod_auth_basic` beziehungsweise `mod_auth_digest`. Die diversen `mod_authn_*`-Module dienen als Provider für die Vergleichsdaten, die beispielsweise in Textdateien (`mod_authn_file`), DBM-Dateien (`mod_authn_dbm`) oder Datenbanken (`mod_authn_dbd`) enthalten sind. Die `mod_authz_*`-Module kümmern sich dagegen um die eigentliche Zugangskontrolle (darf der korrekt angemeldete User die gewünschte Anfrage durchführen?). In Apache 2.0 dagegen musste das einzelne Authentifizierungsmodul (zum Beispiel `mod_auth` oder `mod_auth_dbm`) die gesamte Logik allein implementieren.

Dies akzeptiert alle Benutzer/Passwort-Kombinationen aus der Datei `.htuser`. Wenn ein User einen URL-Pfad aus diesem Verzeichnis anfordert, erhält sein Browser den Status `401 Unauthorized` als Antwort und zeigt einen Anmelde-dialog an. Bei weiteren Zugriffen auf denselben Bereich sendet der Browser – zumindest innerhalb derselben Sitzung – die Anmeldedaten automatisch.

15.2.7 PHP manuell installieren

Ein Linux-System haben Sie bereits, Apache haben Sie nun entweder mit der Distribution oder von Hand installiert, und die Installation von MySQL wurde in Kapitel 12, »Der Datenbankserver MySQL«, beschrieben. Zu einem LAMP-System fehlt also noch PHP. Wenn Sie die Apache-Version aus der Distribution verwenden, können Sie PHP einfach über YaST hinzufügen. Bei einer selbst installierten Apache-Version müssen Sie es dagegen von Hand nachinstallieren.

Laden Sie sich zunächst unter <http://www.php.net> das neueste Quellcodearchiv herunter (zurzeit PHP 5.3.0 oder 5.2.11). Vergleichen Sie den MD5-Hash Ihres Downloads aus Sicherheitsgründen mit dem auf der Website angegebenen Wert. Entpacken Sie das Paket danach wie folgt:

```
# tar xzvf php-5.3.0.tar.gz
```

beziehungsweise

```
# tar xjvf php-5.3.0.tar.bz2
```

Wechseln Sie anschließend in das neu erstellte Verzeichnis, in dem sich der entpackte Quellcode-Baum befindet:

```
# cd php-5.3.0
```

Nun wird das `configure`-Skript aufgerufen. Die allgemeinen Optionen sind `Autoconf`-Standard; andere Kommandozeilenargumente, insbesondere die zahlrei-

chen Schalter im Format `--enable-feature` oder `--with-option`, sind PHP-spezifisch. Geben Sie folgenden Befehl ein, wenn Sie alle Optionen in Ruhe lesen möchten:

```
# ./configure --help |less
```

Sie sollten auf jeden Fall die Option `--with-mysql=Verzeichnis` verwenden, um die wichtige MySQL-Unterstützung zu aktivieren. Wenn Sie eine MySQL-Version ab 4.1.2 verwenden, sollten Sie (zusätzlich oder ausschließlich) die `mysqli`-Extension aktivieren. Diese bietet eine moderne, objektorientierte Schnittstelle zu MySQL-Datenbanken. Die Installations-Option lautet hier `--with-mysqli=/Pfad/zur/Datei/mysqli_config`. Weiter unten wird die Programmierung mit beiden Varianten erläutert.

In PHP 5 kommt eine dritte Datenbankschnittstelle (nicht nur für MySQL) hinzu: PHP Data Objects (PDO) abstrahiert die Datenbankverbindung mithilfe verschiedener Treiber. Fügen Sie `--with-pdo_mysql=/Pfad/zur/Datei/mysqli_config` hinzu, um den PHP-Treiber zu aktivieren; PDO selbst wird automatisch installiert.

Für den (zu empfehlenden) Betrieb von PHP als Apache-Modul ist außerdem die Option `--with-apxs2[=/Pfad/zu/apxs]` erforderlich. Das »APache eXtension tool« befindet sich im *bin*-Verzeichnis Ihrer Apache-Installation.

Hier sehen Sie einen `configure`-Aufruf, der die Makefiles zur Installation als Apache-2-Modul mit `mysql`- und `mysqli`-Unterstützung in das Basisverzeichnis `/usr/local/php5` erstellt, wenn sich `apxs` unter `/usr/local/apache2/bin` und MySQL unter `/usr/local/mysql` befindet:

```
# ./configure --prefix=/usr/local/php5 \
--with-mysql=/usr/local/mysql \
--with-mysqli=/usr/local/mysql/bin/mysqli_config \
--with-apxs2=/usr/local/apache2/bin/apxs
```

Nun wird die übliche `Checking...`-Parade gestartet, die je nach Systemgeschwindigkeit und gewählten Optionen ziemlich lange dauern kann. Wenn `configure` beendet wurde, erfolgen der eigentliche Build-Vorgang und die anschließende Installation über die beiden folgenden bekannten Befehle:

```
# make
# make install
```

Wenn Sie PHP mit der Option `--with-apxs2` konfiguriert haben, sollte Ihre `httpd.conf`-Datei automatisch folgende Zusatzzeile erhalten haben:

```
LoadModule php5_module modules/libphp5.so
```

Allerdings müssen Sie selbst dafür sorgen, dass Apache 2 Dateien mit der Endung *.php* (oder, wenn Sie möchten, auch mit den veralteten Endungen *.php3*, *.php4* oder *.html*) als PHP-Skripte betrachtet. Dazu wird im gewünschten Kontext eine der folgenden Zeilen eingesetzt:

```
AddHandler php5-script .php
# Alternative zur Verknüpfung weiterer Endungen:
AddHandler php5-script .php .php3 .php4 .html
```

Nach dem Beenden und erneuten Start Ihres Apache-Servers (ein normaler Neustart reicht für die PHP-Installation leider nicht) können Sie bereits die PHP-Funktionalität testen. Speichern Sie dazu folgende kleine Datei unter dem Namen *info.php* in einem Kontext Ihrer Server-Verzeichnisse, in dem die soeben eingestellte Typzuordnung gilt:

```
<?php
phpinfo();
?>
```

15.2.8 Die PHP-Konfigurationsdatei *php.ini*

PHP besitzt eine eigene Konfigurationsdatei, *php.ini*. Wie der Dateiname vermuten lässt, wurde ihre Syntax (auch unter UNIX) den klassischen Windows-INI-Dateien nachempfunden:

Sie enthält verschiedene Abschnitte, die durch Schlüsselwörter in eckigen Klammern gekennzeichnet sind. Beispiel:

```
[PHP]
```

Jede Einstellung steht in einer eigenen Zeile und hat das Format *Name=Wert*. Beispiel:

```
doc_root = /usr/local/apache2/htdocs
```

Ein Semikolon leitet einen Kommentar ein, der bis zum Ende der jeweiligen Zeile reicht. Beispiel:

```
; doc_root: Die DocumentRoot der Website
doc_root = "/srv/www/htdocs" ; SUSE-Standard
```

In Tabelle 15.11 finden Sie die wichtigsten Optionen für die Datei. Beachten Sie, dass die Syntax in der Regel toleranter ist, als es in der Tabelle den Anschein erweckt. Die Boolean-Alternativen *On|Off*, *"On"|"Off"* und *"1"|"0"* sind beispielsweise in der Regel synonym – die Angaben stammen aber aus der Originaldokumentation, sodass Sie auf der sicheren Seite sind, wenn Sie sich daran halten.

Einstellung	Standardwert	Beispiel
<code>short_open_tag = On Off</code>	On	Bestimmt, ob <code><? ... ?></code> statt <code><?php ... ?></code> erlaubt ist.
<code>asp_tags = On Off</code>	Off	Bestimmt, ob die ASP-Syntax <code><% ... %></code> statt <code><?php ... ?></code> zulässig ist.
<code>max_execution_time</code>	30	maximale Ausführungsdauer von PHP-Skripten in Sekunden, bevor sie abgebrochen werden
<code>precision = "INT"</code>	"14"	Genauigkeit von Fließkommazahlen als Ziffernanzahl
<code>y2k_compliance = On Off</code>	Off	Jahr-2000-Kompatibilität (funktioniert nur mit neueren Browsern)
<code>expose_php = On Off</code>	On	Bestimmt, ob die PHP-Existenz z. B. in den ServerTokens veröffentlicht wird.
<code>memory_limit = "...M K B"</code>	"8M" (8 Megabyte)	Maximale Speichergröße, die ein Skript verbrauchen darf. Existiert nur, wenn PHP mit <code>--enable-memory-limit</code> kompiliert wurde.
<code>track_vars = "On" "Off"</code>	"On"	Anfrage- und Server-Daten werden in <code>\$_ENV</code> , <code>\$_GET</code> , <code>\$_POST</code> , <code>\$_COOKIE</code> und <code>\$_SERVER</code> zur Verfügung gestellt.
<code>arg_separator.output = "STRING"</code>	"&"	Trennzeichen in Query-Strings, die von PHP erzeugt wurden
<code>arg_separator.input = "STRING"</code>	"&"	Trennzeichen für die Auswertung von Query-Strings (jedes Zeichen des Wertes wird einzeln geprüft, z. B. "&;")
<code>variables_order = "STRING"</code>	"EGPCS"	Reihenfolge, in der Variablenarten für <code>register_globals</code> ausgewertet werden: E = Umgebungsvariablen, G = GET-Felder, P = POST-Felder, C = Cookies, S = Server-Variablen

Tabelle 15.11 Die wichtigsten Optionen der Datei »php.ini«

Einstellung	Standardwert	Beispiel
<code>register_globals = "On" "Off"</code>	"Off"	Stellt Formularfelder, Umgebungsvariablen, Server-Variablen und Cookies automatisch als globale Variablen mit ihrem Feldnamen bereit. Praktisch, aber sicherheitstechnisch riskant; daher seit 4.2 standardmäßig "Off"
<code>register_argc_argv = "On" "Off"</code>	"On"	Stellt GET-Variablen und Kommandozeilenargumente als C-ähnliche <code>argc / argv</code> bereit.
<code>register_long_arrays = "On" "Off"</code>	"On"	Definiert <code>\$HTTP_POST_VARS</code> und <code>\$HTTP_GET_VARS</code> zusätzlich zu <code>\$_POST</code> und <code>\$_GET</code> (seit 5.0 konfigurierbar; davor immer eingeschaltet).
<code>post_max_size = "...M K B"</code>	"8M"	maximale Größe von POST-Daten
<code>auto_prepend_file = "DATEIPFAD"</code>	""	Include-Datei, die automatisch am Kopf jeder Datei importiert wird
<code>auto_append_file = "DATEIPFAD"</code>	""	Include-Datei, die automatisch am Fuß jeder Datei importiert wird
<code>default_mimetype = "TYPE/SUBTYPE"</code>	"text/html"	Standardtyp für den Header Content-Type
<code>default_charset = "ZEICHENSATZ"</code>	"iso-8859-1"	Standardzeichensatz für den Header Content-Type
<code>always_populate_raw_post_data = "0" "1"</code>	"0"	Bestimmt, ob <code>\$HTTP_RAW_POST_DATA</code> (POST-Daten ohne Variablenaufteilung) immer erzeugt wird.
<code>allow_webdav_methods = "0" "1"</code>	"0"	Bestimmt, ob WebDAV-Methoden zulässig sind.
<code>include_path = "PFAD1[:PFAD2:...]"</code>	Umgebungsvar. <code>PHP_INCLUDE_PATH</code>	Verzeichnisse, in denen nach Include-Dateien gesucht wird

Tabelle 15.11 Die wichtigsten Optionen der Datei »php.ini« (Forts.)

Einstellung	Standardwert	Beispiel
doc_root = "PFAD1[:PFAD2:...]"	PHP_INCLUDE_PATH	Übergeordnetes Verzeichnis für PHP-Skripte (z.B. Apache-DocumentRoot)
user_dir = "VERZEICHNIS"	NULL	Name des Benutzerverzeichnisses mit PHP-Dateien (entspricht dem Apache-mod_userdir-Konzept)
extension_dir = "PFAD"	Umgebungsvar. PHP_EXTENSION_DIR	Verzeichnis mit Extension-Dateien, die beim PHP-Start geladen werden
cgi.rfc2616_headers = "0" "1"	"0"	Ausgeschaltet Status:-Header; eingeschaltet RFC-2616-Header
file_uploads = "0" "1"	"1"	Datei-Uploads zulässig
upload_tmp_dir = "PFAD"	NULL (Systemvorgabe)	Temp-Verzeichnis für Uploads
upload_max_filesize = "...M K B"	"2M"	maximale Größe für Upload-Datei
sql.safe_mode = "0" "1"	"0"	sicheres SQL; Schutz vor SQL-Injection (aber normale PHP-MySQL-Beispiele funktionieren nicht mehr)

Tabelle 15.11 Die wichtigsten Optionen der Datei »php.ini« (Forts.)

15.2.9 phpMyAdmin installieren

Ein sehr praktisches Tool für die Verwaltung von MySQL-Datenbanken ist die Software phpMyAdmin. Sie wird in einem PHP-fähigen Webserver ausgeführt und kann über jeden beliebigen Browser benutzt werden. Hier erfahren Sie in Kürze, wie Sie diese nützliche Anwendung installieren können.

Laden Sie sich zunächst die aktuelle Version von der Website des Projekts (<http://www.phpmyadmin.net>) herunter. Dies ist zurzeit 3.2.3. Entpacken Sie die *.tar.gz*- oder *.tar.bz2*-Datei:

```
# tar xzvf phpMyAdmin-3.2.3-all-languages.tar.gz
```

oder

```
# tar xjvf phpMyAdmin-3.2.3-all-languages.tar.bz2
```

Verschieben Sie das neue Verzeichnis an eine Stelle oberhalb Ihrer Apache-DocumentRoot. Beispiel:

```
# mv phpMyAdmin-3.2.3-all-languages /usr/local/apache2
```

Fügen Sie in Ihrer Apache-Hauptkonfigurationsdatei *httpd.conf* folgende Zeilen ein (suchen Sie am besten eine andere Zeile, die mit `Alias` beginnt, und schreiben Sie diese Zeilen darunter):

```
Alias /phpmyadmin /usr/local/apache2/phpMyAdmin-3.2.3-all-languages
<Directory /usr/local/apache2/phpMyAdmin-3.2.3-all-languages>
    Order deny,allow
    Deny from all
    Allow from 127.0.0.1
</Directory>
```

Wenn Sie auch von anderen Rechnern in Ihrem LAN aus zugreifen möchten, müssen Sie `Allow from` um die entsprechende IP-Adresse ergänzen, beispielsweise folgendermaßen, wenn Sie das Netz 192.168.0.0/24 verwenden:

```
Allow from 127.0.0.1 192.168.0
```

Stellen Sie außerdem sicher, dass Ihr `DirectoryIndex` (Startseite) unter anderem den Eintrag `index.php` enthält.

Jetzt können Sie Apache neu starten und dann die Adresse `http://localhost/phpmyadmin` in die Adressleiste Ihres Browsers eingeben. Daraufhin wird der typische HTTP-Anmeldedialog eingeblendet. Geben Sie hier für maximalen Nutzen `root` und das entsprechende Passwort für den MySQL-Administrator ein. Daraufhin befinden Sie sich auf dem phpMyAdmin-Startbildschirm. Von hier aus müssen Sie zunächst im linken Frame eine Datenbank wählen oder in der Mitte eine neue erstellen. Danach funktioniert die Anwendung überaus intuitiv.

15.3 LAMP-Webanwendungen

Dieser Abschnitt ist weder eine umfassende Einführung noch eine ausführliche Referenz zur Programmiersprache PHP. Beides würde den Rahmen dieses Buches sprengen. Bei Galileo Computing sind aber eine ganze Reihe von PHP-5-Büchern mit unterschiedlichen Schwerpunkten erschienen (siehe Anhang B). Im Übrigen ist auch die Online-Dokumentation von PHP unter <http://www.php.net/> sehr ausführlich und recht gut strukturiert.

Wenn Sie PHP programmieren, brauchen Sie nicht – wie bei der CGI-Programmierung in Perl und anderen Sprachen – den HTML-Code durch Ausgabebefehle

zu erzeugen. Stattdessen können Sie an einer beliebigen Stelle eines HTML-Dokuments einen PHP-Block einfügen, in dem der PHP-Interpreter nach PHP-Anweisungen sucht und diese ausführt.⁷ Das impliziert, dass jedes HTML-Dokument – abgesehen von der Dateiendung – ein gültiges PHP-Skript ist, auch wenn der PHP-Interpreter darin noch nichts zu tun hat. Ein PHP-Block sieht folgendermaßen aus:

```
<?php

ANWEISUNG1;
[ANWEISUNG2;
 ...]

?>
```

Hier ein konkretes Beispiel, das die berühmten Worte »Hallo Welt!« auf die Webseite ausgibt:

```
<?php

echo ("Hallo Welt!");

?>
```

Der PHP-Block kann übrigens an jeder beliebigen Stelle des HTML-Dokuments stehen – wenn Sie möchten, sogar innerhalb der Anführungszeichen des Wertes eines HTML-Attributs. Angenommen, Sie haben irgendwo zu Beginn Ihres Skripts eine Variable namens `$farbe` definiert (zum Beispiel aus einem Cookie ausgelesen). Sie können sie folgendermaßen als Seitenhintergrundfarbe setzen:⁸

```
<body bgcolor="<?php echo($farbe); ?>">
```

Wie Sie sehen, sind die Zeilenumbrüche im PHP-Block optional – der Browser erhält ohnehin nur die Ausgabe des `echo()`-Befehls, also beispielsweise Folgendes:

```
<body bgcolor="#00FFFF">
```

⁷ Dieser vermeintliche Vorteil wird allerdings zum Nachteil, wenn Sie codelastige Webanwendungen schreiben – sie werden schnell sehr unübersichtlich. Hier kann die Auslagerung in `include`-Dateien (siehe unten) helfen.

⁸ Natürlich werden Farben und andere Design-Einstellungen auf modernen Websites ausschließlich per CSS gesetzt. Dies ist nur ein Beispiel.

Sprachformalitäten von PHP – die Definition von Variablen, Funktionen, Klassen, Methoden und Objekten – können hier nicht explizit behandelt werden. Sie werden in den Beispielen einfach ohne weitere Erläuterung verwendet. Wenn Sie noch nie in PHP programmiert haben, sollten Sie eine der erwähnten Informationsquellen konsultieren.

15.3.1 Formulardaten auslesen

In PHP ist der Zugriff auf Formulardaten sehr einfach: Das globale Array `$_GET` stellt die mithilfe einer GET-Anfrage versandten Inhalte eines Query-Strings zur Verfügung, während POST-Daten in dem Array `$_POST` bereitstehen. In PHP gibt es, anders als in Perl, keinen Unterschied zwischen Array und Hash (die Indizes von Arrays können Zahlen oder Strings sein).

Angenommen, ein Formular enthielt die Felder `kunde` und `anschrift` und wurde per POST-Anfrage an das aktuelle Skript geschickt. Sie können folgendermaßen auf die beiden Felder zugreifen:

```
$kunde = $_POST['kunde'];
$anschrift = $_POST['anschrift'];
```

Ein Problem ergibt sich allerdings, wenn ein Benutzer eines der Felder nicht ausgefüllt hat. Ein vollständiger Test für das Feld `kunde`, der zu früheren und aktuellen PHP-Versionen kompatibel ist und der Variablen `$kunde` in jedem Fall einen Wert zuweist, sieht beispielsweise so aus:

```
$kunde = 'NO_NAME'; // oder anderen Standardwert einfügen
if (isset($_POST['kunde']) && $_POST['kunde'] != '') {
    $kunde = $_POST['kunde'];
}
```

Die Funktion `isset()` überprüft, ob eine Variable überhaupt definiert wurde. In älteren PHP-Versionen liefert `isset()` aber unter Umständen für jedes Formularfeld `true`, weil das Array selbst existiert. Deshalb wird zusätzlich verlangt, dass das Feld nicht leer sein darf.

Ein solcher Test ist für alle Formularfelder zu empfehlen. Das bedeutet bei sehr vielen Feldern natürlich eine Menge Schreiarbeit. Deshalb empfiehlt es sich, zum Einlesen von Formularfeldern folgende Funktion zu benutzen, die versucht, den Wert des Feldes mit der Bezeichnung `$field` auszulesen, und, falls das Feld leer ist, den Standardwert `$default` verwendet:

```
function cgi_param ($field, $default='') {
    // Standardwert setzen
    $value = $default;
```

```

    if (isset($_GET[$field]) && $_GET[$field] != '') {
        $value = $_GET[$field];
    } elseif (isset($_POST[$field]) && $_POST[$field] != '') {
        $value = $_POST[$field];
    }
    return $value;
}

```

Mit `cgi_param()` reduziert sich der Aufwand zum Auslesen der beiden Felder `kunde` und `anschrift` auf folgende zwei Zeilen:

```

$kunde = cgi_param('kunde', 'NO_NAME');
$anschrift = cgi_param('anschrift', 'NO_PLACE');

```

Wenn Sie einen leeren String als Vorgabewert haben möchten, können Sie den zweiten Parameter sogar weglassen. Beispiel:

```

$bemerkung = cgi_param('anschrift');

```

Es ist davon auszugehen, dass Sie diese Funktion ziemlich häufig benötigen werden. Deshalb lohnt es sich, sie in einer externen Importdatei abzulegen. Auch in einer solchen Datei müssen PHP-Anweisungen in einem `<?php ... ?>`-Block stehen. Die Datei erhält üblicherweise die doppelte Dateiendung `.inc.php` und wird folgendermaßen eingebettet (der Pfad ist relativ zum Verzeichnis des aktuellen PHP-Skripts):

```

include("php_tools.inc.php");

```

15.3.2 Datei-Uploads

HTML bietet ein spezielles Formularfeld, das Benutzern den Versand einer lokalen Datei mit dem Formular ermöglicht. Es handelt sich um ein `<input>`-Element mit dem Attribut `type="file"`. Damit der Upload funktionieren kann, muss das Formular mit der HTTP-Methode `POST` und der speziellen MIME-Codierung `multipart/form-data` versandt werden. PHP verarbeitet optional ein zusätzliches Formularfeld mit der Bezeichnung `MAX_FILE_SIZE`, das die maximal zulässige Dateigröße in Byte enthält und meist als Hidden-Formularfeld gesetzt wird.

Das folgende Beispiel definiert ein Formular für den Versand einer GIF-Bilddatei (dies wird erst auf dem Server geklärt!) von maximal 400 Kilobyte Größe:

```

<form action="bildupload.php" method="post"
  enctype="multipart/form-data">
  <input type="hidden" name="MAX_FILE_SIZE" value="409600" />
  Bitte eine GIF-Bilddatei w&auml;hlen:<br />
  <input type="file" name="bild" />

```

```

<br />
<input type="submit" value="Abschicken" />
</form>

```

Die Datei lässt sich in PHP über das globale Array `$_FILES` auslesen; der Index ist der Feldname. Das entsprechende Array-Element ist wiederum ein Array mit folgenden Elementen:

- ▶ `tmp_name`
Der temporäre Dateiname, unter dem der Server die hochgeladene Datei gespeichert hat
- ▶ `type`
Der MIME-Type der Datei (aus der Dateiendung oder aus dem Content-Type-Header des Formularabschnitts ermittelt)
- ▶ `size`
Gibt die Größe der Datei oder 0 bei einem Fehler an.
- ▶ `name`
Ursprünglicher Pfad und Dateiname der Datei auf dem Client-Rechner
- ▶ `error`
Fehlercode bei einem Upload-Fehler. Seit PHP 4.3.0 sind die angegebenen symbolischen Konstanten definiert, in früheren Versionen nur die zugehörigen numerischen Werte:
 - ▶ `UPLOAD_ERR_OK (0)`: kein Fehler
 - ▶ `UPLOAD_ERR_INI_SIZE (1)`: Datei ist größer als die in *php.ini* angegebene Höchstgrenze `upload_max_filesize`.
 - ▶ `UPLOAD_ERR_FORM_SIZE (2)`: Datei ist größer als die mit dem Formular übertragene Höchstgrenze `MAX_FILE_SIZE`.
 - ▶ `UPLOAD_ERR_PARTIAL (3)`: Die Datei wurde nur zum Teil hochgeladen.
 - ▶ `UPLOAD_ERR_NO_FILE (4)`: Es wurde gar keine Datei hochgeladen.

Das folgende PHP-Fragment kopiert die hochgeladene Datei unter dem Namen *bild.gif* in das Verzeichnis des Skripts selbst, falls es sich um eine Bilddatei vom Typ GIF handelt:

```

if (is_uploaded_file($_FILES['bild']['tmp_name'])) {
    if($_FILES['bild']['type'] == 'image/gif') {
        move_uploaded_file($_FILES['bild']['tmp_name'],
            "bild.gif");
    } else {
        echo("Falscher Dateityp!<br />");
    }
}

```

```

} else {
    echo("Upload-Fehler (Datei zu groß&szlig;)? <br />");
}

```

15.3.3 Sessions

Eines der interessantesten Web-Features von PHP ist das *Session-Tracking*. Wie Sie wissen, ist HTTP ein zustandsloses Protokoll – jede Client-Anfrage steht für sich allein. Für größere Webanwendungen, die sich über viele Seiten erstrecken, wird deshalb ein Verfahren zur Datenweitergabe an nachfolgende Anfragen benötigt. PHP stellt sehr praktische Funktionen zur Verwaltung von Session-Daten bereit.

Beachten Sie, dass Sie Session-Befehle in einen PHP-Block zu Beginn Ihres Skripts setzen müssen: Session-Daten werden als Cookies übertragen, wenn der Client dies zulässt, ansonsten über eine Session-ID im Query-String. Beides erfordert, dass das eigentliche geparste Dokument noch nicht begonnen hat, sodass mit dem ersten Zeichen Ihres Skripts die Sequenz `<?php` einsetzen muss.

Jedes Skript, das auf Session-Daten zugreifen soll, muss zunächst folgende Anweisung enthalten:

```
session_start();
```

Anschließend können Sie das spezielle Array `$_SESSION` mit Name-Wert-Paaren bestücken oder Werte von einer zuvor besuchten Seite daraus lesen. Aus dem soeben erwähnten Grund muss das Lesen vor dem Schreibzugriff erfolgen:

```

// Stückzahl auslesen
$stueckzahl = $_SESSION['stueck'];
// GET-Formularfeld zahlungsart auslesen
$zahlungsart = $_GET['zahlungsart'];
// Zur zukünftigen Verwendung in Session speichern*
$_SESSION['zahlungsart'] = $zahlungsart;

```

15.3.4 Cookies

Für die meisten Anwendungen ist die Verwendung von Session-Daten gegenüber *Cookies* vorzuziehen. Der einzige Nachteil von Sessions besteht darin, dass die Daten beim nächsten Besuch eines Benutzers nicht mehr zur Verfügung stehen. Dies lässt sich oft durch die Speicherung der Daten in einer Datenbank und eine persönliche Anmeldung beheben.

Dennoch gibt es Fälle, in denen Cookies praktischer sind – es kann für einen Besucher unter Umständen nützlich sein, Einstellungen, die er auf einer Seite vorgenommen hat, beim nächsten Besuch wieder vorzufinden. Sie sollten allerdings niemals eine Webanwendung programmieren, die von Cookies abhängt: Aufgrund des Missbrauchs, der häufig zu Werbe- und Usertracking-Zwecken mit Cookies getrieben wird, schalten viele Benutzer Cookies generell ab.

In PHP wird ein Cookie mithilfe der folgenden Funktion gesetzt:

```
setcookie($name, $wert[, $verfallsdatum[, $pfad
    [, $domain[, $secure]]]])
```

Hier eine Übersicht über die Parameter:

- ▶ `$name`
Der Name des Cookies, über den es später wieder abgefragt werden kann
- ▶ `$wert`
Der Wert, den Sie für den entsprechenden Namen festlegen möchten
- ▶ `$verfallsdatum`
Der Verfallszeitpunkt für das Cookie in Sekunden seit EPOCH. Sie können einfach `time() + $sekundenzahl` verwenden, da die Funktion `time()` den aktuellen Zeitpunkt in diesem Format liefert. Wenn Sie diesen Wert weglassen, erzeugen Sie automatisch ein Session-Cookie, das nur während der aktuellen Client-Sitzung gilt.
- ▶ `$pfad`
Der URL-Pfad, unter dem das Cookie zur Verfügung steht
- ▶ `$domain`
Die Domain, unter der das Cookie gültig ist
- ▶ `$secure`
Wenn Sie diesen Wert auf 1 setzen, wird das Cookie nur über eine gesicherte HTTPS-Verbindung übertragen.

Wie Sie wissen, wird ein Cookie als HTTP-Header gesetzt. Deshalb muss dieser Befehl – genau wie die Session-Anweisungen – vor jeglichem HTML-Code in der PHP-Datei stehen. Die folgende Anweisung setzt ein sieben Tage gültiges Cookie namens `lastvisit`, dessen Wert die aktuelle Uhrzeit ist:

```
setcookie("lastvisit", time(), time() + 7 * 24 * 60 * 60);
```

Die Cookies, die der Client bei der Anfrage mitgeschickt hat, können Sie aus dem globalen Array `$_COOKIE` lesen. Dieses Beispiel liest das Cookie `lastvisit` und speichert seinen Wert in einer gleichnamigen Variablen:

```
$lastvisit = $_COOKIE['lastvisit'];
```

Diese Variable lässt sich im weiteren Verlauf des Skripts etwa so verwenden:

```
$lastvisitformat = date("d.m.Y, H:i", $lastvisit);
echo("Ihr letzter Besuch: $lastvisitformat <br />");
```

Die Funktion `date($format, $sek_seit_epoch)` erzeugt formatierte Datumsangaben. Die Buchstaben in `$format` entsprechen zum Teil dem Systembefehl `date`, allerdings ohne Prozentzeichen (die wichtigste Ausnahme bildet `i` statt `M` für die Minuten). Einen Überblick über die `date`-Formatangaben finden Sie übrigens in Kapitel 4, »Mit der Shell arbeiten«.

15.3.5 Zugriff auf MySQL-Datenbanken

Ernsthafte Webanwendungen greifen so gut wie immer auf Datenbanken zu, um Informationen zu laden oder zu speichern. Der in Kapitel 12 ausführlich vorgestellte Datenbankserver MySQL eignet sich vorzüglich für die Zusammenarbeit mit PHP. Hier wird kurz beschrieben, wie es funktioniert.

Eine Testdatenbank

Die einfachen MySQL-Beispiele in diesem Abschnitt beziehen sich alle auf eine kleine Datenbank namens `buecher`, in der eine einzige gleichnamige Tabelle enthalten ist. Die Felder dieser Datenbanktabelle sehen Sie in Tabelle 15.12.

Damit die Beispiele nicht allzu umfangreich werden, ist die Tabelle sehr einfach gehalten: Das relationale Datenbankmodell würde eigentlich nicht erlauben, dass der Autor in derselben Tabelle steht, weil ein Buch mehrere Autoren besitzen kann und weil ein und derselbe Autor mehrere Bücher geschrieben haben kann. Ähnliches gilt auch für den Verlag. In den Beispielinhalten erscheint deshalb jeweils nur der Nachname des Autors, und es werden nur Bücher von Einzelautoren aufgenommen.⁹

9 Die drei Bindestriche befinden sich bei einer ISBN immer an derselben Stelle – eine redundante Information, die wohl niemand in einer Datenbank speichern würde. Bei Amazon wird die ISBN sogar ohne Bindestriche angezeigt.

Feldbezeichnung	Datentyp
isbn	char(10), Primärschlüssel
autor	varchar(30)
titel	varchar(40)
verlag	varchar(30)

Tabelle 15.12 Inhalte der Beispieldatenbanktabelle »buecher«

Sie können die Datenbank und die Tabelle erstellen, indem Sie im Kommandozeilen-Client `mysql` (siehe Kapitel 12) nacheinanderfolgende SQL-Befehle eintippen (die Antworten des MySQL-Clients wurden hier weggelassen):

```
mysql> CREATE DATABASE buecher;
mysql> use buecher
mysql> CREATE TABLE buecher (
    ->     isbn CHAR(10),
    ->     autor varchar(30),
    ->     titel varchar(40),
    ->     verlag varchar(30),
    ->     PRIMARY KEY (isbn)
    -> );
```

Als Nächstes benötigt die Datenbank einige Beispieldaten. Sie können entweder die Werte aus Tabelle 15.12 übernehmen oder Ihre eigenen benutzen.

ISBN	Autor	Titel	Verlag
3898423611	Bongers	XSLT 2.0	Galileo Press
3897213494	Friedl	Reguläre Ausdrücke	O'Reilly
3898425576	Lubkowitz	Webseiten programmieren und gestalten	Galileo Press
3898424642	Rettig	Tomcat 5	Galileo Press
0596007248	Ristic	Apache Security	O'Reilly
3897214059	Schroder	Linux Kochbuch	O'Reilly

Tabelle 15.13 Beispieldatensätze für die Datenbanktabelle »buecher«

Die Werte können Sie ebenfalls mithilfe des Clients `mysql` eintippen:

```
mysql> INSERT INTO buecher VALUES
    -> ('3898423611', 'Bongers', 'XSLT 2', 'Galileo Press'),
```

```
-> ('3897213494', 'Friedl', 'Reguläre Ausdrücke', 'O\'Reilly'),
-> ...
-> ('3897214059', 'Schroder', 'Linux Kochbuch', 'O\'Reilly');
```

Beachten Sie bitte das Semikolon hinter dem letzten Datensatz (im Gegensatz zu den Kommas bei allen anderen) sowie den Backslash vor dem Apostroph bei 'O\'Reilly' – alternativ können Sie die Felder durch doppelte Anführungszeichen umschließen.

Für die PHP-Beispiele weiter unten sollten Sie nun noch einen User erstellen, der auf diese (und nur auf diese) Datenbank zugreifen darf:

```
mysql> CREATE USER buchuser@localhost
-> IDENTIFIED BY 'geheim';
mysql> GRANT ALL PRIVILEGES ON buecher.* TO buchuser@localhost;
```

Für MySQL 4.x müssen Sie die Zeile `CREATE USER ...` wieder durch

```
mysql> GRANT USAGE ON *.* TO buchuser@localhost
-> IDENTIFIED BY 'geheim';
```

ersetzen.

Natürlich sollten Sie sich in jedem Fall bemühen, ein besseres Passwort als »geheim« zu finden. Wenn Sie vorhaben, mit alten Clients (zum Beispiel PHP-Versionen vor 4.4 oder 5.0.3) zusammenzuarbeiten, müssen Sie den Bereich `IDENTIFIED BY 'geheim'` weglassen und stattdessen die folgende zusätzliche Zeile eingeben:

```
mysql> SET PASSWORD FOR buchuser@localhost =
-> OLD_PASSWORD('geheim');
```

MySQL-Zugriff – klassisch

Wie bereits erwähnt, gibt es zwei verschiedene PHP-Schnittstellen für den Zugriff auf MySQL-Datenbanken: die klassische `mysql`-Schnittstelle und die Neuentwicklung `mysqli`. Letztere existiert nur in PHP 5 und arbeitet nur mit MySQL ab Version 4.1 zusammen; sie ermöglicht die objektorientierte Programmierung von Datenbank Anwendungen.

Hier wird zunächst einmal die Vorgehensweise für die klassische Schnittstelle vorgestellt; weiter unten lernen Sie dieselben Anweisungen für `mysqli` kennen.

Als Erstes müssen Sie eine Verbindung zum MySQL-Server herstellen. Dies funktioniert mithilfe der klassischen Schnittstelle so:

```
$connID = mysql_connect($host[, $user[, $password]]);
```

Angenommen, der Datenbank-Server läuft auf demselben Host wie Apache, der Benutzername ist `buchuser` und das Passwort `geheim`. Dann wird der Befehl folgendermaßen aufgerufen:

```
$connID = mysql_connect("localhost", "buchuser", "geheim");
```

Sie sollten es bei einem Produktivsystem dringend vermeiden, anonyme Zugriffe auf den Datenbank-Server zuzulassen. Es sollte also keine Webanwendung geben, bei der ein Zugriff auf MySQL ohne Benutzername und Passwort möglich ist.

Die Verbindungskennung `$connID` ist übrigens im Erfolgsfall ein positiver Integer, andernfalls erhält die Variable den Wert 0. Deshalb könnte die konkrete Verwendung folgendermaßen aussehen:

```
$connID = mysql_connect("localhost", "buchuser", "geheim");
if ($connID) {
    // Datenbankverarbeitung ...
} else {
    echo("Fehler: Kein Datenbankzugriff möglich!<br />");
}
```

Als Nächstes müssen Sie die Datenbank auswählen, auf die zugegriffen wird. Dies geschieht mithilfe des Befehls `mysql_select_db()`. Der erste Parameter ist der Name der gewünschten Datenbank. Optional können Sie die Verbindungskennung angeben – dies ist aber nur erforderlich, wenn mehrere Datenbankverbindungen bestehen. Beispiel:

```
mysql_select_db("buecher");
// Falls mehrere Datenbankverbindungen geöffnet sind:
mysql_select_db("buecher", $connID);
```

Nun können Sie der Datenbank SQL-Abfragen senden. Dies geschieht mithilfe der Funktion `mysql_query($abfrage[, $connID])`. Die Verbindungskennung braucht auch hier wieder nur angegeben zu werden, wenn Sie mehrere Datenbankverbindungen gleichzeitig verwenden. Der Rückgabewert ist eine Ergebniskennung, deren Inhalt Sie mit weiteren Funktionen auslesen können, oder `false`, wenn die Abfrage fehlschlägt. Im letzteren Fall liefert die Funktion `mysql_error()` eine Fehlerbeschreibung.

Hier ein Beispiel für eine Auswahlabfrage:

```
$result = mysql_query(
    "SELECT * FROM buecher WHERE verlag LIKE 'Galileo%'");
```

Die einzelnen Zeilen des Ergebnisses lassen sich mithilfe von `mysql_fetch_row($result)`, `mysql_fetch_assoc($result)` oder `mysql_fetch_array($result)` auslesen. Trotz der unterschiedlichen Bezeichnungen liefern beide ein Array. Der Unterschied ist, dass die Indizes bei `mysql_fetch_row()` numerisch und bei `mysql_fetch_assoc()` die eigentlichen Feldnamen sind; `mysql_fetch_array()` schließlich stellt alle Werte jeweils unter beiden Schlüsseln zur Verfügung. Sie können alle drei Funktionen in der Bedingung einer `while()`-Schleife aufrufen, weil sie `false` zurückgeben, falls kein weiterer Datensatz mehr vorhanden ist. Hier ein Beispiel für `mysql_fetch_row()`:

```
echo("<table border=\"2\"><tr>");
echo("<th>Autor</th><th>Titel</th><th>Verlag</th>");
echo("</tr>");
while ($record = mysql_fetch_row($result)) {
    $autor = $record[0];
    $titel = $record[1];
    $verlag = $record[2];
    echo("<tr>");
    echo("<td>$autor</td>");
    echo("<td>$titel</td>");
    echo("<td>$verlag</td>");
    echo("</tr>");
}
echo("</table>");
```

Mit `mysql_fetch_assoc()` sieht der relevante Ausschnitt der `while()`-Schleife dagegen so aus:

```
while ($record = mysql_fetch_assoc($result)) {
    $autor = $record['autor'];
    $titel = $record['titel'];
    $verlag = $record['verlag'];
    // ... wie gehabt ...
}
```

Bei den unterschiedlichen Arten von Änderungsabfragen (`UPDATE`, `INSERT`, `DELETE` und `REPLACE`) gibt es natürlich kein Ergebnis in Form zurückgegebener Datensätze. Stattdessen können Sie über die Funktion `mysql_affected_rows()` erfahren, wie viele Datensätze geändert (beziehungsweise hinzugefügt oder gelöscht) wurden. Beispiel:

```
// Neues Buch hinzufügen
$result = mysql_query(
    "INSERT INTO buecher VALUES ('3827322537',
    'Kofler', 'MySQL 5', 'Addison-Wesley')");
```

```
// Hat es funktioniert?
if (mysql_affected_rows() == 1) {
    echo("Buch erfolgreich hinzugef&uuml;gt.<br />");
} else {
    echo("Buch konnte nicht hinzugef&uuml;gt werden.<br />");
}
```

Wenn Sie eine Abfrage durchführen, die besonders viele Datensätze liefert, sollten Sie anschließend den Speicher leeren, den das Ergebnis belegt hat:

```
mysql_free_result($result);
```

Zum Schluss sollten Sie die Datenbankverbindung ordnungsgemäß schließen:

```
mysql_close($connID); // $connID entfällt, wenn eindeutig
```

Die mysqli-Schnittstelle

Die neuen `mysqli_*`-Anweisungen können sowohl prozedural als auch objektorientiert verwendet werden. Wenn Sie die prozedurale Schreibweise verwenden, unterscheiden sie sich nicht sehr stark von den gerade vorgestellten klassischen `mysql_*`-Funktionen. Die wichtigste Besonderheit ist, dass Sie die Verbindungskennung nicht mehr weglassen können. Deshalb ist der objektorientierte Zugriff zu bevorzugen – hier wird für jede Verbindung ein eigenes MySQL-Objekt erzeugt.

Hier sehen Sie zunächst sämtliche Anweisungen aus dem `mysql`-Beispiel unter Verwendung der prozeduralen Form der `mysqli`-Schnittstelle:

```
$connID = mysqli_connect("localhost", "buchuser",
    "geheim", "buecher");
$result = mysqli_query($connID,
    "SELECT * FROM buecher WHERE verlag LIKE "Galileo%");
echo("<table border=\"2\"><tr>");
echo("<th>Autor</th><th>Titel</th><th>Verlag</th>");
echo("</tr>");
while ($record = mysqli_fetch_row($result)) {
    $autor = $record[0];
    $titel = $record[1];
    $verlag = $record[2];
    echo("<tr>");
    echo("<td>$autor</td>");
    echo("<td>$titel</td>");
    echo("<td>$verlag</td>");
    echo("</tr>");
}
```

```

echo ("</table>");
mysqli_free_result($result);
mysqli_close($connID);

```

Die objektorientierte Variante von `mysqli` verursacht weniger Schreibarbeit und ist erheblich praktischer. Da die Verbindung und das Ergebnis Objekte sind, entfällt beim Aufruf der verschiedenen Methoden das Präfix `mysqli_` und auch die Angabe der Verbindungskennung. Hier sehen Sie das gesamte Beispiel noch einmal in objektorientierter Syntax:

```

$conn = new mysqli("localhost", "buchuser", "geheim",
    "buecher");
$result = $conn->query(
    "SELECT * FROM buecher WHERE verlag LIKE 'Galileo%'");
echo("<table border=\"2\"><tr>");
echo("<th>Autor</th><th>Titel</th><th>Verlag</th>");
echo("</tr>");
while ($record = $result->fetch_row()) {
    $autor = $record[0];
    $titel = $record[1];
    $verlag = $record[2];
    echo("<tr>");
    echo("<td>$autor</td>");
    echo("<td>$titel</td>");
    echo("<td>$verlag</td>");
    echo("</tr>");
}
echo("</table>");
$result->free();
$conn->close();

```

Bitte beachten Sie bei Änderungsabfragen mithilfe der objektorientierten Schreibweise von `mysqli`, dass `affected_rows` hier eine Eigenschaft und keine Methode ist – hinter dem Namen dürfen keine leeren Klammern stehen. Beispiel:

```

// Buch einfügen, mysqli
$result = $conn->query(
    "INSERT INTO buecher VALUES ('3827322537',
    'Kofler', 'MySQL 5', 'Addison-Wesley')");
// Hat es geklappt?
if ($conn->affected_rows == 1) {
    echo("Buch erfolgreich hinzugef&uuml;gt.<br />");
} else {
    echo("Buch konnte nicht hinzugef&uuml;gt werden.<br />");
}

```

Einen von mehreren MySQL-Servern zufällig wählen

Wenn Sie die in Kapitel 12, »Der Datenbankserver MySQL«, besprochene Replikation durchführen, kann es sich günstig auf die Performance Ihrer Webanwendungen auswirken, Daten abwechselnd von verschiedenen Replikations-Slaves zu lesen (schreiben müssen Sie allerdings weiterhin auf den Primary Master). Die hier kurz skizzierte Lösung bietet kein echtes Load-Balancing, da keine Gewichtung der jeweils gelesenen Datenmenge stattfindet. Aber auch eine solche einfache Zufallsauswahl ist in vielen Fällen gut genug.

Zuerst benötigen Sie ein Array mit den Domainnamen oder IP-Adressen der möglichen Slaves für die Lesezugriffe. Hier ein Beispiel:

```
$readhosts = array
    ("192.168.0.20", "192.168.0.21", "192.168.0.22");
```

Als Nächstes wird einer dieser Hosts zufällig ausgewählt; dabei wird die praktische PHP-Funktion `array_rand()` verwendet, um einen zufälligen Index auszuwählen.

```
$readhost = $readhosts[array_rand($readhosts)];
```

Nun kann wie gewohnt die Verbindung hergestellt werden (in diesem Beispiel mit `mysqli`):

```
$conn = new mysqli ($readhost, $user, $pass, $db);
```

Für die Schreibzugriffe auf den Master-Server benötigen Sie in diesem Fall eine weitere Verbindung.

15.4 Zusammenfassung

LAMP ist die allgegenwärtige Bausubstanz des Webs; keine andere Kombination von Technologien arbeitet auf so vielen Websites wie Apache, MySQL und PHP auf Linux-Systemen. Ihr openSUSE-System ist die ideale Basis zum Betrieb eines Servers für leistungsfähige Webanwendungen mit sehr guter Verfügbarkeit.

Um erfolgreich einen Webserver zu betreiben, sollten Sie zunächst die grundlegende Funktionsweise des HTTP-Protokolls verstehen. Einen kurzen Überblick darüber erhalten Sie im ersten Abschnitt dieses Kapitels. Durch die Analyse der Kommunikation zwischen Webserver und Browser können Sie die Leistungsfähigkeit Ihrer Webanwendungen steigern und Fehlern auf die Spur kommen.

openSUSE 11.2 wird mit einem Apache-Webserver der allerneuesten Generation – Version 2.2 – geliefert. Wie Sie ihn jeweils mit YaST installieren und konfigurieren können, wurde zu Beginn des zweiten Abschnitts ausführlich beschrieben. Wenn Sie eine noch neuere Version installieren oder Apache noch stärker an Ihre

Bedürfnisse anpassen möchten, können Sie sich auch an die Anleitungen zur manuellen Installation und Konfiguration in den späteren Abschnitten dieses Kapitels halten.

Als Nächstes sollten Sie PHP installieren und testen; auch dafür wurde sowohl der Umgang mit den mitgelieferten Paketen als auch die manuelle Kompilierung beschrieben. Wenn PHP läuft, können Sie zudem das überaus nützliche Tool phpMyAdmin installieren, das die komfortable Verwaltung von MySQL-Datenbanken ermöglicht.

Um dieses Kapitel abzurunden, wurden zum Schluss noch einige Grundlagen der PHP-Programmierung erläutert. Besondere Erwähnung finden hier einige spezielle Web-Features wie Uploads und Session-Handling. Vielleicht noch wichtiger ist schließlich der Zugriff auf MySQL-Datenbanken über eine der beiden Schnittstellen `mysql` oder `mysqli`.

*Regel: Von allen Briefen Kopien machen und ordentlich aufbewahren.
– Leo Tolstoi*

16 Weitere Internet-Serverdienste

Nachdem Sie im vorigen Kapitel das wichtigste Serversystem für Webanwendungen kennengelernt haben, werden in diesem Kapitel einige weitere Server behandelt, die auf die Internet-Protokollfamilie aufsetzen. Im Einzelnen geht es um folgende Dienste:

► *xinetd*

Der »Super Daemon« `xinetd`, ein moderner Nachfolger des Klassikers `inetd`, ist in der Lage, bestimmte Serverdienste nur bei Bedarf zu starten, das heißt, wenn auf den betreffenden TCP-Ports Anfragen eintreffen. Dies ist nützlich für eher selten genutzte Dienste.

► *FTP-Server*

Einer der ältesten Internetdienste und nach wie vor die klassische Methode zur Bereitstellung umfangreicher Download-Archive sowie für den Datei-Upload bei Webhostern. Zum Lieferumfang von openSUSE gehört der hier behandelte FTP-Server `vsftpd`.

► *E-Mail-Server*

Neben dem Web der wichtigste aller Internet-Dienste. Hier liegen die Dinge etwas komplizierter, da für Versand, Weiterleitung und Empfang von E-Mails unterschiedliche Protokolle und Server eingesetzt werden. Zudem ist die Sicherheitskonfiguration von entscheidender Bedeutung, da Ihre E-Mail-Server sonst beispielsweise leicht als Spam-Relays missbraucht werden können. Im Mail-Abschnitt geht es um die lokale Mail-Konfiguration mit dem Versands-server Postfix und dem Verteilungsprogramm Fetchmail.

16.1 `xinetd`

Viele Internet-Serverdienste wie etwa die im vorigen Kapitel vorgestellten Webserver oder auch E-Mail-Server haben in der Regel genug zu tun, sodass es sich lohnt, sie permanent als Daemons laufen und selbstständig auf Verbindungsan-

forderungen lauschen zu lassen. Andere dagegen kommen in viel selteneren Fällen zum Einsatz, sodass es nicht zu rechtfertigen ist, sie dauerhaft im Speicher zu halten. Daher bietet sich für sie die Möglichkeit an, einen externen Dienst mit der Überwachung ihres TCP-Ports zu beauftragen und sie erst dann zu starten, wenn dort eine Anfrage eintrifft.

16.1.1 Der klassische inetd

Der herkömmliche UNIX-Dienst zur Lösung der beschriebenen Aufgabe wird `inetd` (Internet Server Daemon) genannt. Auch unter Linux wurde diese Variante jahrelang verwendet. In openSUSE (wie bereits in SUSE Linux-Distribution seit einigen Jahren) ist der Original-`inetd` gar nicht mehr enthalten. Seine Funktion wurde von `xinetd` (Extended `inetd`) übernommen, der erheblich leistungsfähiger ist. Dennoch sollten Sie sich kurz einige Beispieleinträge aus der Konfigurationsdatei `inetd.conf` ansehen, da dieser Daemon auf manchen Systemen noch zum Einsatz kommt und entsprechende Konfigurationsbeispiele in der Dokumentation vieler Serverdienste stehen:

```
echo stream tcp nowait root internal
echo dgram udp wait root internal
ftp stream tcp nowait root /usr/sbin/tcpd in.ftpd -l -a
```

Von links nach rechts finden Sie hier folgende Einträge:

- ▶ *Dienstname*
Das Alias des Dienstes oder Ports, wie er in `/etc/services` steht; beispielsweise `echo`.
- ▶ *Socket-Typ*
Ein Socket ist einer der beiden Endpunkte einer Netzwerkverbindung (die Socket-Implementierung und -Programmierung wird in Kapitel 18, »Hinter den Kulissen«, angesprochen). Die beiden wichtigen Typen sind Streaming-Sockets (`stream`), die eine zuverlässige Verbindung erstellen, sowie Datagramm-Sockets (`dgram`) für den Versand einfacher Nachrichten. Wie zu erwarten ist, korrespondieren `stream`-Sockets mit TCP und `dgram`-Sockets mit UDP.
- ▶ *Protokoll*
Hier wird das Transportprotokoll festgelegt, das heißt `tcp` oder `udp` gemäß den Erläuterungen in Kapitel 13, »Netzwerkgrundlagen«. Wie Sie in den Beispieleinträgen sehen, stehen manche Dienste über TCP und UDP zur Verfügung, hier beispielsweise `echo` (einfache Rücksendung der empfangenen Daten zum Testen der Netzwerkintegrität, das heißt eine Art `ping` über komplexere Transportprotokolle).

► *Nebenläufigkeit*

Der Eintrag in dieser Spalte beantwortet die Frage, ob der betreffende Dienst in der Lage ist, mehrere Anfragen gleichzeitig zu verarbeiten (`nowait`) oder nur linear nacheinander (`wait`). Da UDP Einzelnachrichten und keine Datenströme verarbeitet, ist es nicht weiter verwunderlich, dass für UDP-Dienste `wait` genügt. TCP-Server benötigen dagegen in aller Regel irgendeine Art von Multiplexing-Verfahren (siehe etwa die Diskussion über die Apache-MPM-Module im vorigen Kapitel); wenige Ausnahmen, die die Verbindung nach der Beantwortung einer einzelnen Anfrage selbst beenden, können ebenfalls `wait` verwenden.

► *UID*

Hier wird eingetragen, unter welcher User-ID der betreffende Dienst gestartet werden soll. Da nur `root` die privilegierten Ports 0 bis 1023 öffnen darf, steht hier oft dieser Name.

► *Befehl*

Der letzte Eintrag bestimmt, was geschehen soll, wenn eine TCP-Verbindungsanfrage beziehungsweise ein UDP-Datagramm für den betreffenden Dienst eintrifft. `internal` bedeutet, dass der `xinetd` die Anfrage selbstständig behandelt (einige grundlegende Dienste wie `echo` oder `daytime` sind eingebaut). Andernfalls wird der Name eines zu startenden Serverprogramms angegeben. `tcpd` ist der TCP-Wrapper, ein weiterer Vermittler, der dann seinerseits den eigentlichen Server startet.

16.1.2 xinetd-Konfigurationsdateien

Der modernere `xinetd` verwendet ein etwas komplexeres Format für seine Konfigurationsdateien; dies ist nötig, weil er erheblich flexibler ist. Die Hauptkonfigurationsdatei `/etc/xinetd.conf` enthält zunächst einige allgemeine Voreinstellungen in einem Block namens `defaults { ... }`:

```
defaults
{
    log_type           = FILE /var/log/xinetd.log
    log_on_success     = HOST EXIT DURATION
    log_on_failure     = HOST ATTEMPT
#   only_from         = localhost
    instances         = 30
    cps                = 50 10

#   interface        = 127.0.0.1
}
```

Wie in UNIX-Konfigurationsdateien üblich, leiten #-Zeichen einen Kommentar ein. Die Zugriffsbeschränkungen `only_from` (zulässige Client-Hosts) und `interface` (bediente Netzwerkschnittstellen) sind standardmäßig auskommentiert; der `xinetd` bedient also standardmäßig »alle Welt« – einmal abgesehen von Ihren Firewall-Einstellungen.

Einige weitere Grundeinstellungen betreffen das Logging: Als `log_type` wird eine Datei festgelegt (die Alternative wäre Syslog – siehe Kapitel 17, »System-Automatisierung«); daneben steht der Name der Protokolldatei. `log_on_success` und `log_on_failure` bestimmen, welche erfolgreichen beziehungsweise gescheiterten Ereignisse protokolliert werden. `instances` beschränkt die Anzahl der Server-Instanzen, während die beiden Werte von `cps` (clients per second) festlegen, wie viele neue Verbindungen pro Sekunde hergestellt werden dürfen – standardmäßig 50 – und wie viele Sekunden gewartet werden soll, falls dieser Wert überschritten wird (normalerweise 10).

Auf den `default`-Block folgt diese entscheidende Zeile:

```
includedir /etc/xinetd.d
```

Das bedeutet, dass alle Dateien aus dem Verzeichnis `/etc/xinetd.d` eingebunden werden. Dort befinden sich die Konfigurationsdateien für die einzelnen Dienste. Betrachten Sie als Beispiel die Datei `/etc/xinetd.d/daytime`. Dieser klassische Internet-Standarddienst (RFC 867) beantwortet jede TCP- oder UDP-Anfrage auf dem Standard-Port 13 mit Datum und Uhrzeit und beendet anschließend sofort die Verbindung.¹

Für die UDP-Version existiert übrigens eine weitere Konfigurationsdatei namens `/etc/xinetd.d/daytime-udp`. Hier der vollständige Inhalt der Datei `daytime` für die TCP-Variante (ohne einige Kommentarzeilen):

```
service daytime
{
    type            = INTERNAL
    id              = daytime-stream
    socket_type     = stream
    protocol       = tcp
    user           = root
    wait           = no
```

1 Unter http://buecher.lingoworld.de/apache2/mod_daytime.html finden Sie das Apache-Modul »mod_daytime«, das diesen Dienst innerhalb des Apache-Webservers bereitstellt – ich schrieb es als Demonstration des Multi-Protokoll-Supports für das ebenfalls bei Galileo Computing erschienene Handbuch »Apache 2«.

```

    disable      = yes
    FLAGS        = IPv6 IPv4
}

```

Einige der Konfigurationsdirektiven entsprechen den weiter oben beschriebenen Feldern der klassischen `inetd`-Konfiguration; einige andere wurden hinzugefügt. Im Einzelnen bedeuten sie Folgendes:

- ▶ `service daytime`
Dieser Block konfiguriert den `daytime`-Dienst; der Name stammt aus `/etc/services`.
- ▶ `type = INTERNAL: xinetd`
Dieser Dienst erledigt diese Aufgabe selbst; weiter unten finden Sie eine Liste der eingebauten Dienste. Andere Standardtypen sind `RPC` (siehe NFS und NIS in Kapitel 14, »Datei- und Verzeichnisdienstserver«), `TCPMUX` (TCP-Multiplexing gemäß RFC 1078) und `UNLISTED` (benutzerdefinierter Dienst, dessen Portnummer nicht in `/etc/services` steht).
- ▶ `id = daytime-stream`
Eindeutige ID des Dienstes. Normalerweise entspricht sie der Bezeichnung hinter dem Schlüsselwort `service`. Bei Diensten wie `daytime`, die sowohl über TCP als auch über UDP funktionieren, muss allerdings ein Unterscheidungsmerkmal hinzugefügt werden – hier folgerichtig `daytime-stream` im Gegensatz zu `daytime-dgram` bei der UDP-Version.
- ▶ `socket_type = stream`
Die Art des Netzwerk-Sockets ist `stream` für eine verlässliche, kontinuierliche Verbindung (passend für TCP); bei UDP-Diensten wird `socket_type = dgram` verwendet.
- ▶ `protocol = tcp`
Das Transportprotokoll ist TCP; als Alternative ist `udp` möglich.
- ▶ `user = root`
Der Dienst wird unter der User-ID `root` ausgeführt; da `daytime` den privilegierten Port 13 verwendet, ist dies erforderlich.
- ▶ `wait = no`
Der TCP-basierte `daytime`-Dienst ist in der Lage, mehrere Verbindungsanfragen gleichzeitig anzunehmen; bei der UDP-Version lautet die Zeile entsprechend `wait = yes`, weil die Anfragen auf jeden Fall nacheinander verarbeitet werden.

Der `daytime`-Dienst ist allerdings insofern etwas Besonderes, als er die Verbindung auch in der TCP-Variante sofort nach der Übermittlung von Datum und Uhrzeit wieder abbricht. Er benötigt daher keine Nebenläufigkeit zur Bedie-

nung mehrerer Clients, sondern kann nachfolgende Anfragen bequem in der eingebauten Warteschlange des lauschenden TCP-Sockets zwischenspeichern. Die Größe dieser Queue wird durch die Systemkonstante `SOMAXCONN` bestimmt (siehe Kapitel 18, »Hinter den Kulissen«).

- ▶ `disable = yes`
Der `daytime`-Dienst ist standardmäßig deaktiviert. Wenn Sie ihn ausprobieren möchten, können Sie diese Zeile einfach durch ein vorangestelltes `#` auskommentieren.
- ▶ `FLAGS = IPv6 IPv4`
Eine der entscheidenden Neuerungen gegenüber `inetd` besteht darin, dass `xinetd` auch mit IPv6 umgehen kann. Dieser Eintrag bestimmt, dass beide Varianten des IP-Protokolls zulässig sind.

Wenn Sie, wie in der Liste angesprochen, den `disable`-Eintrag auskommentiert und die Datei gespeichert haben, müssen Sie den `xinetd` dazu bringen, seine Konfigurationsdateien neu einzulesen. Das geht am schnellsten so:

```
# rcxinetd reload
```

Danach können Sie den `daytime`-Dienst testen, beispielsweise über `telnet`:

```
$ telnet localhost daytime
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
13 SEP 2009 21:54:37 CEST
Connection closed by foreign host.
```

Neben `daytime` unterstützt der `xinetd` noch die folgenden Dienste intern (jeweils TCP und UDP):

- ▶ `echo` (Port 7) gibt jede an den Server gesendete Nachricht wörtlich wieder zurück.
- ▶ `time` (Port 37) ist ein weiterer Zeit-Server. Im Gegensatz zu `daytime` sendet er die Zeitinformation nicht als String, sondern numerisch, sodass Sie dieses Protokoll nicht mit `telnet` testen können – die Zeichen-Umsetzung des gelieferten 32-Bit-Integers ist nicht im Klartext lesbar.
- ▶ `chargen` (Port 19) erzeugt einen endlosen Strom von Zeichen; normalerweise wird der sichtbare Bereich des ASCII-Zeichensatzes in einer Endlosschleife ausgegeben. Auf diese Weise können Sie beispielsweise die Zuverlässigkeit einer Netzwerkverbindung testen.

- ▶ `discard` (Port 9) ist das Netzwerk-Pendant zu `/dev/null`: Alles, was Sie an diesen Dienst senden, wird verworfen.

Um externe Server-Programme über `xinetd` zu registrieren, benötigen ihre Konfigurationsdateien vor allem die zusätzliche Direktive `server = Pfad [Optionen]`, die das zu startende Programm bestimmt. Zudem kann es notwendig sein, `port = Portnummer` anzugeben, wenn es sich nicht um einen in `/etc/services` verzeichneten Standardserver handelt.

16.2 FTP

Das File Transfer Protocol (FTP) ist ein altherwürdiger Internet-Dienst, dessen erste Definition bereits in RFC 172 von 1971 steht. Die (immer noch!) aktuelle Spezifikation befindet sich in RFC 959. In diesem Abschnitt wird zunächst der FTP-Kommandozeilenclient vorgestellt, damit Sie ein Gefühl für das Protokoll bekommen; anschließend wird die Installation und Konfiguration des FTP-Servers `vsftpd` beschrieben.

Der *TFTP-Server* (Trivial FTP), den Sie unter den YaST-Netzwerkdiensten finden, ist übrigens kein vollwertiger FTP-Server, sondern ein Dienst, der für das Booten eines Betriebssystems über das Netzwerk eingesetzt wird.

16.2.1 FTP-Clients

Als Client für FTP-Downloads können Sie jeden beliebigen Webbrowser verwenden; in der Praxis bietet sich insbesondere Firefox an (siehe Kapitel 8, »Desktop-Anwendungen«). Geben Sie dazu das Schema `ftp://` und die Adresse des FTP-Servers ein, zum Beispiel `ftp://ftp.uni-koeln.de/`. Nun können Sie sich genau wie bei einem automatisch generierten Website-Index (siehe `mod_autoindex` im vorigen Kapitel) durch die Unterverzeichnisse und Dateien klicken; sobald Sie eine Datei anklicken, können Sie entscheiden, wohin Sie sie herunterladen möchten.

In der Praxis werden jedoch überwiegend separate, grafisch orientierte FTP-Clients verwendet, die dem lokalen Dateinavigator Ihrer grafischen Oberfläche ähneln, da diese Programme nicht nur Downloads, sondern auch Uploads beherrschen. Die häufigste Anwendung für diese Programme dürfte die Pflege einer eigenen Website sein. Dabei bearbeiten Sie die Daten in der Regel auf Ihrem eigenen Rechner und laden sie anschließend mithilfe eines solchen FTP-Programms auf den Server Ihres Hosting-Providers hoch, um sie zu veröffentlichen.

In openSUSE ist kein solcher Client enthalten, aber es gibt Hunderte von ihnen, die Sie herunterladen und installieren können. Einer von ihnen ist beispielsweise *gFTP*. Laden Sie ihn auf der Projektwebsite <http://gftp.seul.org> herunter, und installieren Sie ihn mithilfe des bekannten »Automake-Dreikampfs« (siehe Kapitel 3, »Systemrundgang und Grundkonfiguration«):

```
# tar xjvf gftp-2.0.19.tar.bz2
# cd gftp-2.0.19
# ./configure
# make
# make install
```

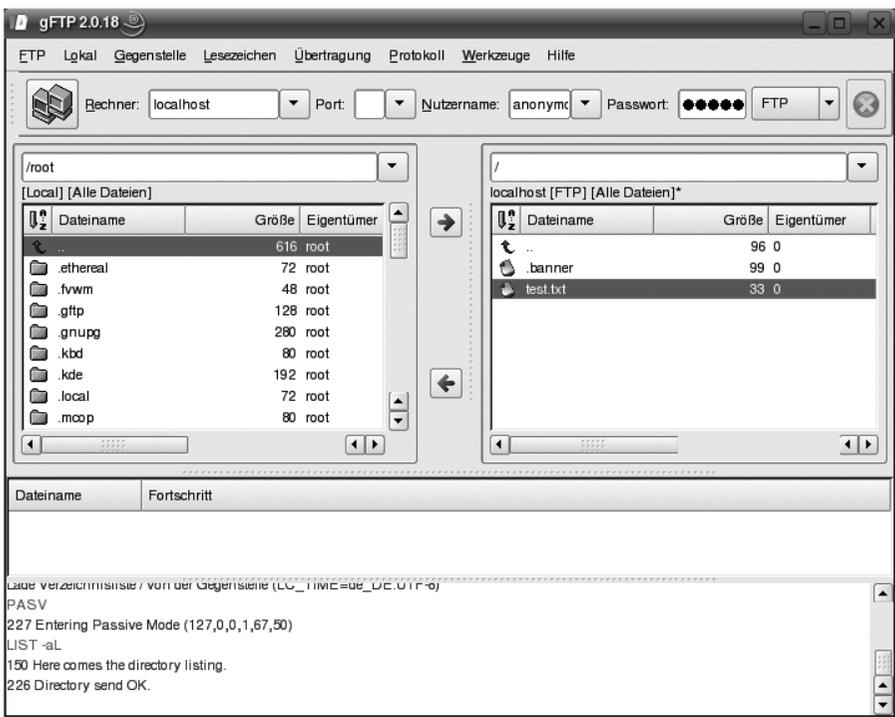


Abbildung 16.1 gFTP, ein grafischer FTP-Client

Das Executable *gftp* wird standardmäßig nach */usr/local/bin* installiert; da dieses Verzeichnis für gewöhnlich im *PATH* liegt, können Sie den Client einfach wie folgt ausführen:

```
$ gftp &
```

In Abbildung 16.1 sehen Sie das Programm im Einsatz. Die Bedienung ist sehr leicht: Geben Sie oben unter RECHNER den Hostnamen oder die IP-Adresse des gewünschten FTP-Servers ein; zusätzlich werden NUTZERNAME und PASSWORT verlangt. Bei Anonymous-FTP-Servern für Massen-Downloads, die beispielsweise oft bei Universitäten zu finden sind, lautet der Benutzername `anonymous`; als PASSWORT wird Ihre E-Mail-Adresse erwartet.

Nach dem Login erscheint im linken Fensterbereich ein Dateibrowser für Ihren lokalen Rechner, während Sie rechts die Dateien und Verzeichnisse des FTP-Servers durchsuchen können. Mithilfe der beiden Pfeil-Schaltflächen dazwischen werden Up- beziehungsweise Downloads durchgeführt. Im unteren Bereich können Sie die hinter den Kulissen stattfindende, tatsächliche FTP-Kommunikation mitlesen.

Falls Sie genau sehen möchten, wie der FTP-Client (auf Ihrem eigenen Rechner) und der FTP-Server (auf dem entfernten Rechner) miteinander kommunizieren, können Sie auch das Konsolenprogramm `ftp` verwenden. Die Befehle, die Sie auf der Client-Seite eingeben, sind jeweils einzeilig und bestehen aus einem Schlüsselwort mit eventuellen Parametern, gefolgt von einem Zeilenumbruch. Die Antwort des Servers ist wie bei HTTP zunächst eine Statusmeldung, die aus einer dreistelligen dezimalen Codenummer und einem Meldungstext besteht; häufig folgen auf die Statusmeldung zusätzliche Datenzeilen. Um dem Client das Ende einer solchen Datensequenz zu signalisieren, beginnt die letzte Zeile der Serverantwort wieder mit derselben Codenummer wie die erste.

Die folgenden Zeilen zeigen den Mitschnitt einer FTP-Sitzung mit dem Host `www.lingoworld.de` (der Name `www` besagt natürlich, dass es sich eigentlich um einen Webserver handelt; es ist durchaus üblich, dass Hosting-Provider den Webserver unmittelbar per FTP zugänglich machen, damit Kunden die eigene Website hochladen können:

```
# ftp www.lingoworld.de
Verbunden zu www.lingoworld.de.
220 FTP Server ready.
Benutzer (www.lingoworld.de:(none)): XXXXX
331 Password required for XXXXX.
Kennwort:
230 User XXXXX logged in.
Ftp> pwd
257 "/" is current directory.
Ftp> cd extra
250 CWD command successful.
Ftp> ls
```

```

200 PORT command successful.
150 Opening ASCII mode data connection for file list.
test.txt
info.txt
226-Transfer complete.
226 Quotas off
21 Bytes empfangen in 0,01 Sekunden (2,10 KB/s)
Ftp> get test.txt
200 PORT command successful.
150 Opening ASCII mode data connection for test.txt (2589 bytes).
226 Transfer complete.
2718 Bytes empfangen in 0,04 Sekunden (67,95 KB/s)
Ftp> quit
221 Goodbye.

```

In dieser Sitzung wird zunächst die Anmeldung durchgeführt (Benutzername wird angezeigt, allerdings habe ich ihn hier geändert; die Passwordeingabe hat kein grafisches Feedback), anschließend werden die folgenden Befehle eingesetzt:

- ▶ `pwd`: »Print working directory« – aktuelles Arbeitsverzeichnis ausgeben
- ▶ `cd`: »Change directory« – Verzeichnis auf dem entfernten Server wechseln
- ▶ `ls`: »list« – Verzeichnisinhalt anzeigen
- ▶ `get`: die angegebene Datei in das aktuelle lokale Verzeichnis herunterladen
- ▶ `quit`: die Sitzung und das FTP-Programm beenden

Weitere wichtige Befehle sind folgende:

- ▶ `put`: die angegebene Datei in das aktuelle entfernte Verzeichnis hochladen
- ▶ `binary`: umschalten in den Binärmodus
- ▶ `ascii`: umschalten in den ASCII-Modus
- ▶ `help`: eine Liste aller verfügbaren Befehle anzeigen

Es ist wichtig, dass Sie den Unterschied zwischen dem ASCII- und dem Binärmodus verstehen. Das ganze Problem hat damit zu tun, dass die verschiedenen Betriebssystementwickler sich nicht auf einen gemeinsamen Standard für Zeilenumbrüche in Textdateien einigen konnten. UNIX verwendet das ASCII-Zeichen mit dem Code 10 (LF), das klassische Mac OS das ASCII-Zeichen 13 (CR), und Windows und die meisten Netzwerk-Anwendungsprotokolle benutzen beide Zeichen hintereinander.

Im ASCII-Modus werden die Zeilenumbrüche innerhalb einer Datei bei der Übertragung jeweils umgewandelt, sodass beispielsweise die auf Ihrem Windows-Rechner gespeicherten Textdateien mit CR/LF auf dem entfernten UNIX-Server mit dem für dessen Verhältnisse korrekten Nur-LF ankommen und umgekehrt. Sie sollten jedoch begreifen, dass dieses bei Textdateien recht segensreiche Feature bei Binärdateien wie Bildern oder Programmen den sicheren Tod zur Folge hat. Wird jedes Vorkommen des Byte-Wertes 10 durch die beiden Bytes 13 und 10 ersetzt oder umgekehrt, so werden die Bytes in einer solchen Datei verändert und planlos verschoben! Natürlich ist eine auf diese Weise behandelte Bild-, Sound- oder Programmdatei unbrauchbar.

Die meisten grafisch orientierten FTP-Programme entscheiden je nach Dateityp sinnvoll selbst, ob sie ASCII- oder Binär-Übertragung verwenden sollen. Bei dem Konsolen-FTP-Programm müssen Sie für jede einzelne Datei selbst in den richtigen Modus umschalten. Das ist ein – aber nicht der einzige – Grund dafür, dass die Arbeit mit der Konsolenversion von FTP absolut unzumutbar ist.

16.2.2 Den FTP-Server vsftpd einrichten und verwenden

Mit openSUSE wird der FTP-Server `vsftpd` (Very Secure File Transfer Protocol Daemon) geliefert. Es gibt kein YaST-Modul, um ihn zu konfigurieren; dies müssen Sie manuell über seine Konfigurationsdatei erledigen. Völlig unmodifiziert läuft er als Anonymous-FTP-Server, von dem Dateien nur heruntergeladen werden können. Das Wurzelverzeichnis für Anonymous FTP ist `/srv/ftp`.

Für die Konfigurationsdatei `/etc/vsftpd.conf` sind zahlreiche Konfigurationsdirektiven zulässig. Die wichtigsten werden in den Kommentaren der Datei selbst erläutert; eine ausführlichere Beschreibung aller Optionen enthält die Manpage `vsftpd.conf(5)`. Hier nur einige wenige Einstellungsmöglichkeiten im Überblick (in der Reihenfolge, in der sie – zunächst auskommentiert – in der Standarddatei stehen):

- ▶ `write_enable=YES|NO`
Bestimmt, ob Schreibzugriffe zulässig sind. Gilt standardmäßig erst einmal nur für lokale User.
- ▶ `ftpd_banner=Text`
Eine Begrüßungsnachricht, die den Benutzern bei der Anmeldung präsentiert werden soll. Enthält oft auch Angaben zu Einstellungen und zur »Geschäftspolitik« des FTP-Servers. Beispiel:

```
ftpd_banner=Welcome to The Domain of The King FTP Server.
```

- ▶ `local_enable=YES|NO`
Regelt, ob sich neben den standardmäßig aktivierten anonymen Log-ins auch lokale User (aus `/etc/passwd` beziehungsweise mit PAM-Unterstützung) anmelden dürfen.
- ▶ `chroot_local_user=YES|NO`
Wenn Sie diese Option aktivieren, wird der jeweils angemeldete User mittels `chroot(1)` in einen »Dateisystemkäfig« gesperrt, sodass sein Home-Verzeichnis die absolute Wurzel bildet. Solange kein gemeinsamer Zugriff auf FTP-Verzeichnisse erfolgen muss, ist diese Vorgehensweise zu empfehlen.
- ▶ `anonymous_enable=YES|NO`
Bestimmt, ob anonyme Log-ins zulässig sind oder nicht. Der Standardwert ist YES.

Nachdem Sie die passenden Änderungen an der Konfigurationsdatei vorgenommen haben, können Sie den FTP-Server in Betrieb nehmen:

```
# rcvsftpd start
```

Nach späteren Konfigurationsänderungen muss er dazu gebracht werden, seine Konfigurationsdatei neu einzulesen:

```
# rcvsftpd reload
```

Falls `vsftpd` beim Booten automatisch starten soll, können Sie ihn wie folgt aktivieren:

```
# chkconfig -a vsftpd
```

16.3 E-Mail

Die E-Mail, die sich unter dem Dach eines Mail-Clients wie Mozilla Thunderbird so einheitlich präsentiert, bedarf in Wirklichkeit der Zusammenarbeit mit mindestens zwei verschiedenen Servern. Der eine ist für den Versand von E-Mails zuständig und führt zu diesem Zweck das Protokoll SMTP (Simple Mail Transport Protocol) aus. Ein anderer enthält das E-Mail-Postfach, in dem an Sie adressierte Nachrichten ankommen; dieser Dienst wird entweder von dem weitverbreiteten POP3-Protokoll (Post Office Protocol Version 3) oder dem komfortableren, aber bisher eher selten verwendeten IMAP (Internet Message Access Protocol) versehen.

In diesem Abschnitt wird nach einer theoretischen Einführung der Mail-Server Postfix vorgestellt; zusätzlich erfahren Sie etwas über die lokale Mail-Verteilung mit `fetchmail`.

Warnung

Bitte beachten Sie, dass der Betrieb eines Mail-Servers mit erheblichen Risiken verbunden ist. Es ist sehr leicht, einen fehlerhaft konfigurierten oder unzureichend abgesicherten Mail-Server als Spam-Relay zu missbrauchen. Abgesehen von dem überaus schlechten Ruf, der Ihnen dadurch in der Internet-Welt droht, können auch erhebliche Kosten auf Sie zukommen: Falls Sie einen dedizierten oder virtuellen Webserver mieten und darauf einen Mail-Server betreiben, der für Spam missbraucht wird, müssen Sie für sämtlichen über den Rahmen Ihres Vertrags hinausgehenden Traffic zahlen. Bei einigen Millionen Spam-Mails am Tag kann der betreffende Betrag Sie leicht innerhalb eines Monats ruinieren.

Wenn Sie also den Betrieb eines öffentlichen Mail-Servers erwägen, der über die hier beschriebenen lokalen Einstellungen hinausgeht, konsultieren Sie dringend einen Experten, oder besorgen Sie sich zusätzliche Literatur (siehe Anhang B), die Sie gründlich studieren sollten.

16.3.1 Theoretische Grundlagen

Eine E-Mail wird beim Versenden stets an einen SMTP-Server übermittelt. Dieser kümmert sich um die Weiterleitung der Nachricht an den Empfänger. Das in RFC 2821 (aktuelle Neufassung von 821) definierte SMTP ist ein textbasiertes Protokoll, das ähnlich wie HTTP oder FTP mit wenigen, einfachen Befehlen auskommt. SMTP-Server lauschen standardmäßig am TCP-Port 25.

Traditionell beinhaltet SMTP keine Kontrolle der Identität eines Absenders. Bis vor wenigen Jahren haben die Provider auch nicht versucht, anderweitig Abhilfe zu schaffen – mit der problematischen Folge, dass der Versand von Spam über offene Server enorm erleichtert wurde, sogar unter Vortäuschung falscher Identitäten. Glücklicherweise unterstützen neuere SMTP-Versionen Authentifizierungsverfahren, und auch die Provider haben das Problem inzwischen erkannt und verwenden in der Regel eines der drei folgenden Verfahren:

- ▶ Prüfung der IP-Adresse: Manche SMTP-Server lassen nur Mails zu, die von bestimmten IP-Adress-Bereichen aus abgeschickt wurden. Dies ist besonders praktisch für Internetprovider, denn so können nur die eigenen Kunden auf den SMTP-Dienst zugreifen. Nachteil: Durch IP-Spoofing kann die Absender-IP-Adresse gefälscht werden; gerade für Spam, der keine direkte TCP-Antwort benötigt, ist das ein allzu verlockender Gedanke.
- ▶ »SMTP after POP«: Manche Provider verlangen, dass ein E-Mail-Client sich zunächst an einem Mail-Empfangs-Server anmeldet, bevor er senden darf. Nachteil: Nicht jeder E-Mail-Client kann so konfiguriert werden, dass er dies automatisch durchführt, sodass der Benutzer vor jedem Versand manuell auf »Empfangen«, »Aktualisieren« oder dergleichen klicken muss.

- ▶ **Persönliche Anmeldung:** Die Anmeldung mit Benutzernamen und Passwort ist das bei Weitem sicherste Verfahren. Glücklicherweise setzt es sich allmählich bei immer mehr Providern durch.

Wenn Sie genau wissen möchten, wie die Kommunikation mit einem SMTP-Server funktioniert, können Sie sie durch direkte Texteingabe in `telnet` ausprobieren. Hier ein Beispiel, wie eine solche Sitzung aussieht (Namens- und Adressdaten wurden anonymisiert):

```
# telnet smtp.myprovider.de smtp
220 smtp.myprovider.de ESMTP Mon, 14 Sep 2009 09:32:57 +0100
HELO
250 smtp.myprovider.de Hello [203.51.81.17]
MAIL From: sender@myprovider.de
250 <sender@myprovider.de> is syntactically correct
RCPT To: recipient@otherprovider.com
250 <recipient@otherprovider.com> verified
DATA
354 Enter message, ending with "." on a line by itself
FROM: Absender <sender@myprovider.de>
To: Empfänger <recipient@otherprovider.com>
Subject: SMTP-Test

Hallo Empfänger,
dies ist nur ein kleiner SMTP-Test.
Kommt die Mail an?
Gruss, Absender
.
250 OK id=18QdIY-00048Y-00
QUIT
221 smtp.myprovider.de closing connection.
```

In dieser kurzen Konversation werden die folgenden SMTP-Befehle verwendet:

- ▶ **HELO**
Mit diesem Befehl meldet sich der Client beim Server an; eventuell findet in diesem Zusammenhang die oben beschriebene Überprüfung der Client-IP-Adresse statt.
- ▶ **MAIL**
Dieser Befehl leitet die Erzeugung einer neuen Nachricht ein; der Empfänger muss im Format `From: E-Mail-Adresse` oder `From: Name <E-Mail-Adresse>` angegeben werden.

- ▶ RCPT
Angabe eines Empfängers im Format `To: E-Mail-Adresse` oder `To: Name <E-Mail-Adresse>`.
- ▶ DATA
Alle folgenden Zeilen des Clients werden als Teil der eigentlichen E-Mail-Nachricht aufgefasst, bis eine Zeile folgt, die nur einen Punkt (.) enthält.
- ▶ QUIT
Die Sitzung wird hiermit beendet; alle bis zu diesem Zeitpunkt erzeugten Mail-Nachrichten werden versandt.

Die E-Mail-Nachricht selbst (zwischen DATA und der Abschlusszeile mit dem Punkt) ist eine klassische Textnachricht, deren Aufbau in RFC 822 (aktualisiert in RFC 2822) beschrieben wird. Prinzipiell besteht sie aus mehreren Header-Zeilen im Format `Feldname: Wert`, gefolgt von einer Leerzeile und dem eigentlichen Text. Der minimale Header enthält den Absender (`From`), den Empfänger (`To`) und einen Betreff (`Subject`). Absender und Empfänger dürfen wie bei den SMTP-Befehlen MAIL und RCPT diverse Formate besitzen. Weitere häufige Header-Felder sind die Kopien-Empfänger (`Cc` für »Carbon Copy«) sowie die unsichtbaren Kopien-Empfänger (`Bcc` für »Blind Carbon Copy«). Die normalen Kopien-Empfänger werden in der Nachricht selbst angezeigt, die unsichtbaren nicht.

Ein alternatives Format für E-Mails, das heutzutage bereits häufiger verwendet wird als RFC 822, ist das *MIME*-Format. Die verschiedenen Aspekte von MIME werden in RFC 2045 bis 2049 dargelegt. Die Abkürzung MIME steht für *Multi-purpose Internet Mail Extensions*. Es handelt sich um ein Format, das für den Versand beliebiger Text- und Binärdaten geeignet ist, sogar von verschiedenen Datentypen innerhalb derselben Nachricht.

Der MIME-Header ist eine Erweiterung des RFC-822-Headers. Die wichtigsten neuen Felder sind `Content-type`, das den Datentyp angibt, und `Content-Transfer-Encoding`, mit dessen Hilfe das Datenformat festgelegt wird. Ersteres beschreibt also den Inhalt der Nachricht, Letzteres die Form, in der sie versandt wird. Der Inhaltstyp (`Content-Type`), meist MIME-Type genannt, besteht aus zwei Bestandteilen, die durch einen Slash (/) voneinander getrennt werden: der Haupttyp und der genauere Untertyp. Haupttypen sind beispielsweise `text` (ASCII-Text), `image` (Bilddaten), `audio` (Sounddaten), `video` (Digitalvideo) oder `application` (proprietäres Datenformat eines bestimmten Anwendungsprogramms). Tabelle 16.1 listet einige gängige MIME-Typen auf. Die vollständigen Listen aller registrierten Typen finden Sie online unter <http://www.iana.org/assignments/media-types/index.html>.

Typ	Beschreibung
text/plain	reiner Text ohne Formatierungsbefehle
text/html	HTML-Code
text/xml	XML-Code
image/gif	Bild vom Dateityp GIF
image/jpeg	Bild vom Dateityp JPEG
image/png	Bild vom Dateityp PNG
audio/wav	Sounddatei vom Typ Microsoft Wave
audio/aiff	Sounddatei vom Typ Apple AIFF
audio/mpeg	komprimierte Sounddatei vom Typ MP3
video/avi	Digitalvideo vom Typ Microsoft Video for Windows
video/mov	Digitalvideo vom Typ Apple QuickTime
video/mpeg	Digitalvideo vom Typ MPEG
application/ x-shockwave-flash	komprimierter Macromedia-Flash-Film (Dateiendung .swf)
multipart/mixed	»Umschlag« für mehrere MIME-Unterabschnitte
multipart/alternative	»Umschlag« für denselben Inhalt in mehreren Alternativformaten

Tabelle 16.1 Einige gängige MIME-Datentypen

Die beiden letzten Typen in der Tabelle machen MIME besonders interessant: Ein MIME-Dokument vom Typ `multipart/mixed` kann beliebig viele Teile enthalten, die jeweils einen vollständigen MIME-Header besitzen und wiederum beliebige MIME-Types aufweisen können. Mithilfe dieser Technik werden der Mail in modernen E-Mail-Programmen Attachments (Datei-Anhänge) hinzugefügt. Dagegen wird ein Abschnitt vom Typ `multipart/alternative` eingesetzt, um denselben Inhalt in verschiedenen alternativen Darstellungsformen zu umschließen, beispielsweise ein Bild im GIF- und im PNG-Format oder (wahrscheinlich die häufigste Anwendung) den Text einer E-Mail-Nachricht einmal im Text- und einmal im HTML-Format.

Der Content-Transfer-Encoding-Header gibt dem Empfänger-Client einen Hinweis, auf welche Weise die ankommenden Daten zu interpretieren sind. Häufig verwendete Werte sind etwa folgende:

- ▶ 7bit
Keine Codierung; nur geeignet für 7-Bit-ASCII (englischer Text). Automatischer Zeilenumbruch nach spätestens 1.000 Zeichen.
- ▶ 8bit
Keine Codierung; geeignet für 8-Bit-Text (internationaler Text). Ebenfalls automatischer Zeilenumbruch nach spätestens 1.000 Zeichen.
- ▶ Binary
Keine Codierung, kein automatischer Zeilenumbruch
- ▶ quoted-printable
Spezielle Codierung von Sonderzeichen, die über 7-Bit-ASCII hinausgehen. Beispiel: »größer« wird zu »gr=FC=DFer« (die Codierung besteht aus einem Gleichheitszeichen, gefolgt vom hexadezimalen Zeichencode).
- ▶ base64
Bevorzugte Codierung für Binärdateien. Ein spezieller Algorithmus packt die Daten 7-Bit-kompatibel um.

Die Codierungsformen `quoted-printable` und `base64` besitzen den Vorteil, dass die Mail-Nachricht formal RFC-822-kompatibel bleibt und entsprechend auch über alte Mail-Server versandt und empfangen werden kann.

Der E-Mail-Empfang über einen POP3-Server erfolgt auf textbasierte Art und Weise, ähnlich wie bei SMTP. Der Server kommuniziert über den TCP-Port 110. Die Beschreibung von POP3 steht in RFC 1939. Zur Verdeutlichung hier wiederum eine Telnet-basierte Konversation mit einem (unkennlich gemachten) POP3-Server:

```
# telnet pop.myprovider.de pop3
+OK POP3 server ready
USER absender
+OK
PASS XXXXX
+OK
LIST
1 898
.
RETR 1
+OK 953 octets
Return-path: <recipient@otherprovider.com>
Envelope-to: sender@myprovider.de
Delivery-date: Mon, 14 Sep 2009 09:35:48 +0100
Received: from [207.18.31.76] (helo=smtp.otherprovider.com)
        by mxng13.myprovider.de with esmtp (Exim 3.35 #1)
```

```
id 18QeUU-000270-00
for sender@myprovider.de; Mon, 14 Sep 2009
09:35:48 +0100
Received: from box (xds1-202-21-109-17.otherprovider.com
[202.21.109.17])
by smtp.otherprovider.com (Postfix) with SMTP id CA500866C1
for <sender@myprovider.de>; Mon, 14 Sep 2009 09:32:41 +0100
(MET)
Message-ID: <001901c2aaf2$31ce81e0$0200a8c0@box>
From: "Empfänger" <recipient@otherprovider.com>
To: "Absender" <sender@provider.de>
Subject: E-Mail-Test
Date: Mon, 14 Sep 2009 09:32:41 +0100
MIME-Version: 1.0
Content-Type: text/plain;
charset="iso-8859-1"
Content-Transfer-Encoding: quoted-printable
```

```
Hallo!
Dies ist noch eine Test-E-Mail.
Kommt sie im POP3-Konto an?
Ciao.
```

```
.
DELE 1
+OK
QUIT
+OK
```

In dieser Sitzung kommen die folgenden POP3-Befehle zum Einsatz:

- ▶ USER
Angabe des Benutzernamen für die Anmeldung
- ▶ PASS
Angabe des Passworts für die Anmeldung
- ▶ LIST
Nummerierte Liste der verfügbaren E-Mails mit der jeweiligen Länge in Byte
- ▶ RETR
E-Mail mit der angegebenen Nummer empfangen.
- ▶ DELE
E-Mail mit der angegebenen Nummer vom Server löschen.
- ▶ QUIT
Sitzung beenden.

Die meisten E-Mail-Programme führen `RETR` und `DELE` standardmäßig unmittelbar nacheinander durch, die Nachrichten verbleiben also in der Regel nicht auf dem Server. Bei IMAP-Servern ist es dagegen meist anders: Der besondere Vorteil des IMAP-Protokolls besteht darin, dass auf dem Mail-Server selbst verschiedene Ordner eingerichtet werden können, um Mails dort zu verwalten. Auf diese Weise erleichtert IMAP die E-Mail-Verwaltung für mobile Benutzer. Die aktuelle Version von IMAP ist das in RFC 2060 dargestellte IMAP4. Ein IMAP-Server funktioniert ähnlich wie ein POP3-Server, verwendet allerdings den TCP-Port 142.

Eine weitere beliebte Form der E-Mail-Nutzung sind webbasierte Free-Mail-Dienste wie GMX oder GoogleMail. Dabei handelt es sich um gewöhnliche POP/SMTP-Kombinationen, die über eine Website mit persönlicher Anmeldung zugänglich gemacht werden. Das Programm, das mit den Mail-Servern kommuniziert, läuft auf dem Webserver und wird dem Kunden per Browser zugänglich gemacht.

16.3.2 Lokale Mail-Konfiguration mit Postfix

Postfix ist ein sogenannter Mail Transfer Agent (MTA), den der niederländische Internet-Veteran *Wietse Venema* als Alternative zu dem Klassiker Sendmail unter besonderer Berücksichtigung von Sicherheitsaspekten völlig neu geschrieben hat. Ein MTA ist in der Lage, E-Mail zu versenden und weiterzuleiten.

Postfix wird automatisch mit openSUSE installiert; weitere Pakete wie den Mail-Abholer `fetchmail` finden Sie in der Selektion `NETZWERK UND SERVER`.

Auf der Konsole können Sie Postfix über das Kommando `sendmail` benutzen – wie der Name schon vermuten lässt, bietet es Sendmail-kompatible Befehle, da unzählige Anwendungsprogramme sowie Programmiersprachen-Bibliotheken auf Sendmail angewiesen sind. Verwenden Sie einen Aufruf wie diesen, um mit Postfix eine E-Mail zu versenden:

```
$ sendmail -t Empfänger
```

Anschließend können Sie die Nachricht eingeben; genau wie bei der oben gezeigten direkten Kommunikation mit einem SMTP-Server kann diese aus Headern und dem durch eine Leerzeile davon getrennten Body bestehen. Zum Abschluss müssen Sie wieder einen Punkt allein in einer Zeile eingeben.

Um Postfix grundlegend zu konfigurieren, können Sie das YaST-Modul *Mail Transfer Agent* unter `NETZWERKDIENTE` verwenden. Auf dem ersten Bildschirm müssen Sie die Art der Internetverbindung einstellen: Wählen Sie `PERMANENT`, wenn Sie – beispielsweise über einen Router in Ihrem lokalen Netzwerk – per

Standleitung mit dem Internet verbunden sind. Wenn Sie einen Internetzugang über DSL, ISDN oder Modem verwenden, können Sie EINWAHL ankreuzen. Die Option KEINE VERBINDUNG schließlich dient dazu, Postfix nur innerhalb des lokalen Netzwerks einzusetzen. Die zusätzliche Option VIRUSÜBERPRÜFUNG (AMAViS) aktiviert Antivirensoftware und überprüft damit automatisch die übertragenen E-Mail-Nachrichten.

Auf dem nachfolgenden Bildschirm, AUSGEHENDE MAIL, wird der Domainname eines SMTP-Servers verlangt. Dies kann entweder ein SMTP-Server in Ihrem lokalen Netzwerk oder aber ein Server Ihres Providers sein, beispielsweise `smtp.meinprovider.de`. Natürlich können Sie Postfix auch auf dem lokalen Rechner so einrichten, dass er als vollwertiger SMTP-Server arbeitet – dies geht allerdings über den Rahmen dieses Abschnitts hinaus; besorgen Sie sich dazu ein gutes Postfix-Buch (siehe Anhang B).

Die Schaltfläche MASQUERADING öffnet einen Dialog, der Ihnen das »Verkleiden« Ihrer E-Mail-Nachrichten und Benutzernamen ermöglicht. In der Regel sollen E-Mails nämlich nicht die Absenderadresse *Benutzer@Rechner.Domain* tragen, sondern ein einfaches *Benutzer@Domain*. Dies können Sie unter DOMAIN FÜR DEN HEADER 'VON' eintragen. Im Übrigen können Sie die Schaltfläche HINZUFÜGEN ganz unten betätigen, wenn Sie einzelnen Benutzern eine besondere Absenderangabe zuweisen möchten – etwa *p_schmitz@Domain* statt dem lokalen Username `peter`.

Unter AUTHENTIFIZIERUNG müssen Sie gegebenenfalls den Benutzernamen und das Passwort für den SMTP-Server eingeben, falls dieser zu den weiter oben angesprochenen modernen, sicheren Servern gehört, die eine Anmeldung verlangen.

Die nächste Seite trägt den Titel EINGEHENDE MAIL. Hier können Sie festlegen, wie Sie E-Mails empfangen möchten – eine solche Konfiguration ist beispielsweise nützlich, wenn nicht alle Rechner in Ihrem Netzwerk einen direkten Internetzugang besitzen und ihre Mails somit nicht selbst von den POP- oder IMAP-Servern des Providers abholen können.

Wenn Sie EXTERNE SMTP-VERBINDUNGEN AKZEPTIEREN ankreuzen, können SMTP-Server E-Mails direkt an den lokalen Rechner senden. Die Mail wird mit anderen Worten nicht von einem externen Mail-Account abgerufen, sondern unmittelbar vom Absender oder einer Vermittlungsstelle empfangen und dann lokal verteilt. Diese Option sollten Sie nur wählen, wenn Sie genau wissen, was Sie tun – bei unzureichender Sicherheitskonfiguration ist dies ein erhebliches Risiko.

Viel üblicher und ungefährlicher sind die Einstellungen im Feld HERUNTERLADEN. Hier können Sie einstellen, dass E-Mails von einem entfernten Postfach-Server abgeholt und lokal verteilt werden sollen. Dazu wird der Dienst `fetchmail` verwendet. Geben Sie in das Feld SERVER den Namen des entsprechenden Postfach-Servers ein, etwa `pop3.meinprovider.de`. Daneben können Sie das PROTOKOLL wählen, über das `fetchmail` mit dem entfernten Server kommunizieren soll. Die Voreinstellung AUTO versucht, das Protokoll selbstständig zu ermitteln. Dies gelingt zwar fast immer, kostet aber nutzlose Zeit – falls Sie das Protokoll also kennen, schadet es nichts, es einzutragen. In aller Regel wird es sich entweder um POP3 oder um IMAP handeln; alles andere ist ziemlich exotisch.

ENTFERNTER BENUTZERNAME und PASSWORT sind die Anmeldedaten für den entfernten Mail-Server, die Sie normalerweise von Ihrem Provider erfahren. Im Pull-down-Menü LOKALER BENUTZER können Sie wählen, an welchen Benutzer die abgeholte Mail weitergeleitet werden soll. Dieser kann sie mithilfe des Lokal-Mail-Kommandos `mail` lesen (siehe nächstes Kapitel). Die Schaltfläche DETAILS ermöglicht die Konfiguration mehrerer E-Mail-Weiterleitungen für verschiedene Benutzer.

Die Option MAIL VON ROOT WEITERLEITEN AN betrifft lokale, administrative Mails, die von Systemprozessen und anderen Daemons versendet werden. Wie Sie wissen, sollten Sie im Alltag nicht als `root` arbeiten; diese Einstellung, die Sie normalerweise bereits während der Installation vorgenommen haben (siehe Kapitel 2), sorgt dafür, dass Sie dennoch asynchron über wichtige Ereignisse informiert werden.

Der AUSLIEFERUNGSMODUS schließlich bestimmt, wie die abgeholten E-Mails den Benutzern zugestellt werden sollen:

- ▶ DIREKT leitet die Mails an die System-Mailboxen der Benutzer weiter; sie erhalten dann beim Log-in die Meldung »You have new mail«.
- ▶ ÜBER PROCMAIL verwendet den gleichnamigen Mail-Vermittler; Sie können die Weiterbehandlung der E-Mails dann für jeden Einzelfall separat einstellen.
- ▶ AN CYRUS IMAP-SERVER schließlich verwendet Cyrus als lokalen Postfach-Server; dies ist nützlich, damit auch lokale Benutzer ohne direkten Internetzugang Standard-E-Mail-Clients wie Thunderbird verwenden können.

Die Schaltfläche ALIASE ermöglicht das Festlegen von Alternativnamen für lokale Benutzer, während VIRTUELLE DOMAINS für die Definition zusätzlicher E-Mail-Domainnamen zuständig ist.

16.4 Zusammenfassung

Nach dem im vorigen Kapitel relativ ausführlich vorgestellten Server für Webanwendungen haben Sie in diesem Kapitel einen Überblick über den Einsatz einiger weiterer Internet-Serverdienste erhalten.

Der »Super Daemon« `xinetd` kann für alle Netzwerkdienste verwendet werden, die nur gelegentlich eingesetzt werden: Er lauscht stellvertretend für diese auf eingehende Verbindungsanfragen und startet sie bei Bedarf. Einige traditionelle Protokolle wie `echo` oder `daytime` sind eingebaut, andere werden durch den Start externer Serverprogramme bereitgestellt.

Das File Transfer Protocol (FTP) ist einer der ältesten Internetdienste, wird aber nach wie vor sehr häufig eingesetzt: zum einen für umfangreiche, anonyme Download-Archive, zum anderen für die Bereitstellung der eigenen Website in gemietetem Webespace. In diesem Kapitel haben Sie zunächst erfahren, wie Sie den grafischen FTP-Client `gFTP` installieren und nutzen können. Anschließend wurden einige Grundlagen der Konfiguration des FTP-Servers `vsftpd` erläutert, der zum Lieferumfang von openSUSE gehört.

Die E-Mail-Server schließlich gehören nach wie vor zu den populärsten Internet-Diensten. Ihre Konfiguration ist recht komplex, weil mehrere Serverdienste beteiligt sind. In diesem Kapitel wurden lediglich die Grundeinstellungen erläutert, die für die Weiterleitung versendeter und die Verteilung empfangener E-Mails im lokalen Netzwerk ausreichen. Für weitergehende Aufgaben benötigen Sie ein Buch über die betreffende Mailserver-Software, vorzugsweise Postfix.

Arbeit ist das, was man tut, um es einmal nicht mehr tun zu müssen.
– Alfred Polgar

17 System-Automatisierung

Linux wäre nicht Linux (und auch nicht UNIX, was das angeht), wenn sich die zahlreichen Einzelbausteine, die die Systemprogramme darstellen, nicht auf vielfältige Art und Weise kombinieren ließen, um Arbeitsvorgänge zu automatisieren und zu vereinfachen. In diesem Kapitel lernen Sie die wichtigsten Möglichkeiten dafür kennen: Aliasse, Cronjobs und Logdateien, Shell-Skripte sowie Perl als »bessere Shell-Alternative«.

17.1 Nützliche Helfer

Bevor im nächsten Abschnitt die Shell-Skripte vorgestellt werden, sollten Sie die hier gezeigten Hilfsmittel kennen. Für manche Aufgaben brauchen Sie kein Shell-Skript, sondern können ein einfaches Alias verwenden. Die ebenfalls in diesem Abschnitt vorgestellten Cronjobs ermöglichen die regelmäßige, automatische Ausführung von Aufgaben.

17.1.1 Aliasse

Das Kommando `alias` ist ein Shell-Built-in, mit dessen Hilfe Sie Kürzel für häufig genutzte Befehle verfassen können. Die allgemeine Syntax lautet:

```
alias Name='Kommando [Parameter]'
```

Das folgende Beispiel definiert ein Alias namens `la`:

```
$ alias la='ls -lA |sort'
```

Achten Sie darauf, dass um das Gleichheitszeichen herum keine Leerzeichen gesetzt werden dürfen. Das Kommando selbst, `ls -lA |sort`, gibt den Verzeichnisinhalt in der Langform aus (`-l`), zeigt dabei fast alle Inhalte an (bis auf die Standardeinträge `.` und `..`; dafür ist die Option `-A` zuständig) und sortiert die Ausgabe schließlich, was zu einer Sortierung nach Dateitypen (erstes Zeichen des Rechte-Strings wie `d`, `l`, `-` und so weiter) führt.

Wenn Sie dieses Kommando verwenden möchten, können Sie einfach seinen Namen eingeben:

```
$ la
```

Es ist auch kein Problem, an ein bestehendes Alias weitere Kommandozeilenparameter anzuhängen, sofern diese kompatibel sind. Im nachfolgenden Beispiel wird der Parameter `-F` hinzugefügt, der die Dateien durch spezielle Sonderzeichen hinter dem Dateinamen klassifiziert:

```
$ la -F
```

Eine Alias-Definition gilt nur für die aktuelle Sitzung. Wenn Sie sie dauerhaft verwenden möchten, müssen Sie sie in eine Shell-Konfigurationsdatei schreiben. Die möglichen Dateien werden in Kapitel 4, »Mit der Shell arbeiten«, vorgestellt.

Wenn Sie sich die Definition eines einzelnen Alias anschauen möchten, geben Sie einfach `alias Name` ein. Beispiel:

```
$ alias la
alias la='ls -lA |sort'
```

`alias` ohne Parameter zeigt dagegen alle definierten Aliasse an (dies ist die Standardliste, die in openSUSE voreingestellt ist):

```
$ alias
alias += 'pushd .'
alias -= 'popd'
alias ..='cd ..'
alias ...='cd ../..'
alias beep='echo -en "\007"'
alias cd..='cd ..'
alias dir='ls -l'
alias l='ls -lF'
alias la='ls -lA'
alias ll='ls -l'
alias ls='/bin/ls $LS_OPTIONS'
alias ls-l='ls -l'
alias md='mkdir -p'
alias o='less'
alias rd='rmdir'
alias rehash='hash -r'
alias umount='echo "Error: Try the command: umount" 1>&2; false'
alias you='test "$EUID" = 0 && /sbin/yast2 online_update ||
su - -c "/sbin/yast2 online_update"'
```

Das Shell-Built-in `unalias` ist das Gegenstück zu `alias` – es entfernt eine Alias-Definition. Beispiel:

```
$ unalias la
```

Die Option `-a` löscht sogar alle Alias-Definitionen:

```
$ alias -a
```

Keine Sorge, wenn Sie dies versehentlich getan haben: Da die oben dargestellte Gruppe von Aliassen aus Shell-Konfigurationsdateien stammt, ist sie beim nächsten Start einer Shell wieder verfügbar.

17.1.2 Cronjobs

Manche Aufgaben in einem Betriebssystem müssen regelmäßig erledigt werden, insbesondere Administrationsaufgaben. Bekannte Beispiele sind der regelmäßige Wechsel von Logdateien (`logrotate`), das Aufräumen temporärer Verzeichnisse oder automatische E-Mail-Benachrichtigungen bei Fehlerzuständen. Für all diese Aufgaben bietet Linux eine Dienstleistung, die als *Cronjobs* bezeichnet wird. Im Einzelnen stehen sogar zwei verschiedene Lösungen zur Verfügung:

- ▶ Die einfachere Lösung: Erstellen Sie in den Verzeichnissen `/etc/cron.hourly`, `/etc/cron.daily`, `/etc/cron.weekly` oder `/etc/cron.monthly` Symlinks auf beliebige Skripte oder Anweisungen, um diese stündlich, täglich, wöchentlich beziehungsweise monatlich auszuführen.
- ▶ Die flexible Lösung: Editieren Sie die *crontab* eines Benutzers – diese Einträge besitzen eine eigenständige Syntax, aber lassen sich erheblich flexibler steuern.

Die `/etc/cron.*`-Dateien

Dazu gibt es eigentlich nicht besonders viel zu sagen – im Konfigurationsverzeichnis `/etc` finden Sie die folgenden vier Unterverzeichnisse:

- ▶ `/etc/cron.hourly` (stündliche Ausführung)
- ▶ `/etc/cron.daily` (tägliche Ausführung)
- ▶ `/etc/cron.weekly` (wöchentliche Ausführung)
- ▶ `/etc/cron.monthly` (monatliche Ausführung)

Wenn Sie in einem dieser Verzeichnisse einen Symlink auf ein beliebiges Shell-Skript oder ein anderes ausführbares Programm/Skript erstellen, wird es im angegebenen Rhythmus ausgeführt.

Angenommen, Sie möchten ein Skript namens *dailyjob.sh* täglich ausführen. Erstellen Sie zu diesem Zweck einen Symlink auf dieses Skript in */etc/cron.daily*:

```
# ln -s /usr/bin/myscripts/dailyjob.sh \
> /etc/cron.daily/dailyjob.sh
```

Die crontab

Eine *crontab* ist eine spezielle Datei, die die Cronjobs eines einzelnen Users auflistet. Jeder Eintrag besitzt schematisch gesehen das folgende Format:

```
Min Std Tag Mon Wochentag Kommando
```

Im Einzelnen können die Felder folgende Formate besitzen:

- ▶ Minute: 0–59
- ▶ Stunde: 0–23
- ▶ Tag des Monats: 1–31
- ▶ Monat: 1–12
- ▶ Wochentag: 0–6 (0 = Sonntag, 1 = Montag ... 6 = Samstag)

Für jedes dieser Felder können Sie außerdem ein *** als Platzhalter angeben – der Cronjob gilt damit für beliebige Werte dieses Feldes. Das folgende Beispiel führt ein Skript namens *test.sh* jeden Mittwoch um 19:30 Uhr aus:

```
30 19 * * 3 /usr/bin/myscripts/test.sh
```

Hier ein weiteres Beispiel – es ruft einmal stündlich das Skript *hourly.sh* auf:

```
0 * * * * /usr/bin/myscripts/hourly.sh
```

Bei Bedarf können Sie für ein bestimmtes Feld auch einen Bereich im Format Startwert-Endwert angeben; etwa die Wochentage 1-5 für Montag bis Freitag. Hier sehen Sie ein Beispiel, in dem das Skript *worktime.sh* montags bis freitags von 9 bis 17 Uhr jede Stunde aufgerufen wird:

```
0 9-17 * * 1-5 /usr/bin/myscripts/worktime.sh
```

Hinter ***- oder Von-bis-Einträgen können Sie zusätzlich mit einem */* ein Intervall anfügen, sodass ein Job beispielsweise nicht mehr jede Minute, sondern alle zehn Minuten ausgeführt wird. Das folgende Beispiel führt alle zehn Minuten *ps aux* aus, erstellt also eine Liste aller laufenden Prozesse:

```
*/10 * * * * ps aux
```

Schließlich besteht noch die Möglichkeit, eine feste Liste bestimmter Werte durch Komma (aber ohne Leerzeichen) getrennt anzugeben. Das folgende Beispiel führt das Skript *backup.sh* um 02:00 und 04:00 Uhr nachts aus:

```
0 2,4 * * * backup.sh
```

Wenn Skripte oder Kommandos, die per Cronjob ausgeführt werden, eine Ausgabe besitzen, wird diese per System-Mail (Kommando `mail`) an den betreffenden User geschickt. Im Zweifelsfall sollten Sie vor dem ersten richtigen Cronjob-Eintrag die Umgebungsvariable `MAILTO` setzen, damit der richtige User benachrichtigt wird. Beispiel:

```
MAILTO=sascha
```

Um solche Nachrichten zu empfangen, können Sie einfach `mail` eingeben. Tippen Sie anschließend die Nummer einer Nachricht ein, um diese zu lesen; die einzelne Nachricht und das Programm `mail` können Sie jeweils durch die Eingabe von `q` beenden.

Um die *crontab* zu editieren, wird das gleichnamige Kommando mit der Option `-e` verwendet. Dies öffnet den Editor `vi` (siehe Kapitel 10, »Klassische Texteditoren«). Hier die Kurzanleitung, was zu tun ist:

1. Drücken Sie `I`, um in den Eingabemodus zu wechseln.
2. Geben Sie gemäß der obigen Beschreibungen beliebig viele Cronjobs ein,
3. Verlassen Sie den Eingabemodus mit `Esc`.
4. Geben Sie `:` `W` `Q` ein, um die Datei zu speichern und das Programm *crontab* zu verlassen.

Wenn Sie eine Liste aller Ihrer Cronjobs sehen möchten, können Sie Folgendes eingeben:

```
$ crontab -l
# DO NOT EDIT THIS FILE - edit the master and reinstall.
# (/tmp/crontab.XXXX8kbph2 installed on Mon May 12 14:25:29 2008)
# (Cron version V5.0 -- $Id: crontab.c,v 1.12 2004/01/23 18:56:42
vixie Exp $)
MAILTO=sascha
*/10 * * * * ps aux
```

Um die *crontab* ganz zu entfernen, geben Sie Folgendes ein:

```
$ crontab -r
```

Im Übrigen besteht die Möglichkeit, mithilfe der Option `-u Benutzername` auch fremde *crontab*-Dateien zu bearbeiten; aufgrund mangelnder Zugriffsrechte ist dies allerdings fast immer `root` vorbehalten.

17.1.3 Logdateien und syslog

Das Betriebssystem Linux selbst und zahlreiche Programme – besonders Daemons, die standardmäßig keine Ausgabe vornehmen – führen Logdateien oder zu Deutsch Protokolldateien. Darin wird über Operationen und insbesondere über Fehler Buch geführt.

Auf UNIX-Systemen existiert traditionell eine Standardschnittstelle für Logging-Aufgaben: der Syslog-Daemon (*syslogd*). Dieser Daemon empfängt Log-Meldungen von verschiedenen Systembestandteilen und Programmen; er entscheidet aufgrund diverser Kriterien, in welche Logdatei eine Meldung eingetragen werden soll; optional können Sie ihn auch so konfigurieren, dass er besonders wichtige Meldungen per Mail verschickt.

Jede Syslog-Meldung besteht aus drei Komponenten:

- ▶ Fehlerquelle (facility)
- ▶ Dringlichkeitsstufe (priority)
- ▶ Warnmeldung (message)

Die Fehlerquelle (facility) beschreibt die Art des Programms oder Prozesses, von dem die Meldung stammt. Die folgenden Quellen sind in *syslog* definiert:

- ▶ `auth`: Authentifizierung
- ▶ `authpriv`: Authentifizierung privilegierter Benutzer
- ▶ `cron`: cron-Daemon
- ▶ `daemon`: sonstige Daemons
- ▶ `ftp`: FTP-Server
- ▶ `kern`: Kernel
- ▶ `local0` bis `local7`: zur freien Verwendung durch beliebige Programme
- ▶ `lpr`: Drucker-Subsystem
- ▶ `mail`: Mail-Subsystem
- ▶ `news`: News-Subsystem
- ▶ `syslog`: interne Meldungen von *syslog* selbst
- ▶ `uucp`: UUCP-Subsystem
- ▶ `user`: Anwendungsprogramme

Die Dringlichkeitsstufe legt fest, wie erheblich eine bestimmte Meldung ist:

- ▶ `emerg`: Notfall; das System ist unbrauchbar
- ▶ `alert`: Alarm – sofortiger Eingriff erforderlich
- ▶ `crit`: kritischer Fehler
- ▶ `error`: normaler Fehler
- ▶ `warn`: Warnung
- ▶ `notice`: Hinweis
- ▶ `info`: normale Information
- ▶ `debug`: Debugging-Information

Die Warnmeldung schließlich ist ein beliebiger Text, den die jeweilige Systemkomponente oder Anwendung zur Information des Benutzers hinzufügt.

Von Shellskripten aus können Sie das Kommando `logger` verwenden, um selbst Syslog-Meldungen zu verfassen. Die allgemeine Syntax lautet wie folgt:

```
logger [-p [facility:]priority] message
```

Geben Sie zum Testen etwa Folgendes ein:

```
$ logger -p notice Hallo Welt
```

Da keine spezielle Quelle angegeben wurde, können Sie die betreffende Nachricht in `/var/log/messages` nachlesen:

```
$ tail -1 /var/log/messages
Apr 07 14:46:39 tux sascha: Hallo Welt
```

`/var/log/messages` ist überhaupt die Haupt-Logdatei des Betriebssystems, in der beispielsweise die Meldungen des Kernels (Facility `kern`) gesammelt werden. Wenn irgendetwas mit dem System schiefgeht, ist die Chance groß, dass Sie hier eine entsprechende Nachricht finden. Auch andere wichtige Logdateien befinden sich in `/var/log`, viele davon in Unterverzeichnissen.

17.1.4 `sudo` – Einzelne Kommandos als `root` ausführen

Bereits in Kapitel 4, »Mit der Shell arbeiten«, haben Sie das Kommando `su` kennengelernt, mit dessen Hilfe Sie eine Shell als anderer Benutzer (vorzugsweise als `root`) starten können. Wenn es hingegen darum geht, einzelne Befehle als `root` auszuführen, können Sie auf `sudo` zurückgreifen – dies lässt sich auch in Shellskripten nutzen.

Wer genau welche Befehle ausführen darf, wird in der Konfigurationsdatei */etc/sudoers* geregelt. Diese darf nicht direkt editiert werden; verwenden Sie stattdessen das Kommando `visudo`, das diese Datei in `vi` (siehe Kapitel 10, »Klassische Texteditoren«) öffnet. Einzelheiten zur Syntax erhalten Sie auf der Manpage `sudoers(5)`.

Wenn Sie noch keine spezielle `sudoers`-Konfiguration durchgeführt haben, können Sie `sudo` explizit verwenden, um einzelne Befehle als `root` auszuführen. Geben Sie dazu beispielsweise Folgendes ein, um den Rechner sofort neu zu starten:

```
$ sudo shutdown -r now
```

Falls Sie `sudo` zum ersten Mal aufrufen, wird folgende »Belehrung« angezeigt:

```
We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:
    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.1
```

Anschließend werden Sie – wie bei jedem weiteren `sudo`-Aufruf – nach dem `root`-Passwort gefragt.

17.2 Shell-Skripte

Die grundlegende Arbeit mit der Shell haben Sie bereits sehr früh in diesem Buch, nämlich in Kapitel 4, kennengelernt. Dies verdeutlicht die enorme Bedeutung, die die Kommandozeile auf UNIX-Systemen noch heute besitzt. Eine besonders komfortable Eigenschaft der Shell wurde bisher allerdings nur erwähnt, aber noch nicht ausgeführt: Sie können beliebige Abfolgen von Shell-Kommandos und Systembefehlen zu Skripten kombinieren, angereichert durch diverse Steuerungsmöglichkeiten, wie sie auch in Programmiersprachen zu finden sind.

17.2.1 Eigenschaften von Shell-Skripten

Ein Shell-Skript ist im Grunde eine einfache Textdatei, die eine Abfolge von Shell- und Systembefehlen enthält. Diese Befehle werden bei Aufruf des Skripts

¹ »Wir vertrauen darauf, dass Sie die übliche Lektion vom lokalen Systemadministrator erhalten haben. Sie besteht im Wesentlichen aus den folgenden drei Punkten: 1) Respektieren Sie die Privatsphäre anderer. 2) Denken Sie nach, bevor Sie tippen. 3) Große Macht bedingt große Verantwortung.« (Den letzten Satz kennen Sie vielleicht aus »Spider-Man«.)

nacheinander abgearbeitet. Für gewöhnlich erhalten Shell-Skripte die Dateiendung `.sh`. Der Name spielt aber eigentlich keine Rolle; wichtig ist, dass Sie das Skript mittels `chmod` ausführbar machen.

Wie jedes Konsolenprogramm kann ein Shell-Skript Kommandozeilenparameter entgegennehmen. Hinter dem Namen des Skripts kann also beim Aufruf eine durch Leerzeichen getrennte Liste von Zeichenfolgen stehen, die das Skript verarbeiten kann.

Die erste Zeile eines Shell-Skripts enthält die sogenannte *Shebang-Angabe* – ein Kurzwort für die Zeichen `#` (sharp) und `!` (bang), mit denen sie beginnt. An dieser Stelle steht, von welcher konkreten Shell dieses Skript ausgeführt werden soll. Wie Sie bereits erfahren haben, unterscheiden sich die verschiedenen Shells ein wenig in ihrer Syntax; dies gilt für Shell-Skripte erst recht. Hier sehen Sie ein Beispiel für eine Shebang-Zeile; das zugehörige Skript wurde für die `bash` geschrieben:

```
#!/bin/bash
```

Nach dieser Zeile können Sie einen Befehl nach dem anderen untereinander schreiben. Alle bisher besprochenen Systembefehle sind zulässig, darüber hinaus werden einige spezielle programmiertechnische Erweiterungen verwendet.

Die wichtigsten zusätzlichen Befehle und Elemente, die die Shell zur vollwertigen Programmiersprache machen, sind:

- ▶ Variablen
- ▶ Fallentscheidungen
- ▶ Schleifen
- ▶ Funktionen

17.2.2 Das erste Beispiel

Bevor die einzelnen Elemente der Shell-Skripte in den nachfolgenden Unterabschnitten vertieft werden, soll hier ein erstes, typisches Skript-Beispiel vorgestellt werden; die einzelnen Bestandteile werden nur jeweils kurz, aber nicht erschöpfend erläutert. Das Skript gibt einfach untereinander alle Kommandozeilenargumente aus, die Sie beim Aufruf angeben. Sofern diese Parameter Zahlen sind, werden sie zu einer Summe addiert, die am Ende ebenfalls angezeigt wird.

Geben Sie in einem Editor Ihrer Wahl folgenden Code ein, und speichern Sie das Ergebnis unter dem Namen `summe.sh`:

```
#!/bin/bash
let summe=0
for n in $*
do
    case $n in
*[*[!0-9]*)
        echo Keine Zahl: $n
        ;;
*)
        let summe=$summe+$n
        echo Zahl: $n
        ;;
    esac
done
echo Summe: $summe
```

Anschließend müssen Sie das Skript ausführbar machen:

```
$ chmod a+x summe.sh
```

Nun kann es aufgerufen werden; geben Sie als Argumente einfach gemischt Zahlen und Zeichenketten ein. Beispiel:

```
$ ./summe.sh 1 a 2 b 3 c
Zahl: 1
Keine Zahl: a
Zahl: 2
Keine Zahl: b
Zahl: 3
Keine Zahl: c
Summe: 6
```

Im Einzelnen bedeuten die einzelnen Skriptbestandteile Folgendes:

- ▶ `#!/bin/bash`
Die bereits besprochene Shebang gibt die Shell (oder im weiteren Sinne das Programm) an, die das Skript ausführen soll. Im vorliegenden Beispiel wird die Bash angesprochen; die Syntax anderer Shells differiert teilweise.
- ▶ `let summe=0`
Diese Anweisung definiert eine Variable namens `summe` und weist ihr den Anfangswert zu. Es ist wichtig, dass zwischen dem Variablennamen, dem Operator `=` und dem Wert keine Leerzeichen stehen dürfen. Die spezielle Anweisung `let` sorgt dafür, dass die Variable numerisch und nicht als String interpretiert wird.

```
▶ for n in $*
do
...
done
```

Dies ist eine Schleifenkonstruktion. Der Schleifenzähler `n` nimmt nacheinander die Werte des speziellen Arrays `$*` an, das die Kommandozeilenargumente enthält (im Fall von Dateimustern sogar jede einzelne Datei, auf die diese zutreffen). `do` und `done` umschließen einen Block beliebiger Anweisungen.

```
▶ case $n in
...
esac
```

Die `case`-Struktur ist eine Fallentscheidung. Sie vergleicht die angegebene Variable mit einem oder mehreren Mustern, die jeweils in der Form

Muster)

an den Anfang eigener Zeilen geschrieben werden. Die Syntax entspricht den Shell-Dateimustern, gefolgt von einer schließenden Klammer. Der Anweisungsblock für den jeweiligen Fall endet mit einem doppelten Semikolon (`;;`), der gesamte `case`-Block mit der Umkehrung `esac`.

In diesem Beispiel lautet das Muster `*[!0-9]*`, das heißt eine Abfolge aus beliebig vielen beliebigen Zeichen (`*`), gefolgt von etwas, was keine Ziffer ist (alles außer `0-9`), und weiteren beliebigen Zeichen. Mit anderen Worten wird alles, was auch nur eine einzige Nicht-Ziffer enthält, als keine Zahl interpretiert.

Das einzelne `*`), das als zweites Muster dient, gilt für »alle anderen Werte« und wird oft zum Abfangen von Fehlern verwendet. Im vorliegenden Fall gilt es für genau diejenigen Parameter, die keine Fehler enthalten.

```
▶ echo Keine Zahl: $n
echo Zahl: $n
```

Hier wird jeweils der wohlbekannte `echo`-Befehl verwendet, um den Wert von `n` – mit vorangestelltem Einleitungstext – auszugeben.

```
let summe=$summe+$n
```

Diese erneute numerische Variablenzuweisung addiert den Wert von `n` zum bisherigen Wert von `summe`. Da Strings in Shell-Skripten ohne Anführungszeichen geschrieben werden können (solange sie keine Leerzeichen enthalten), müssen Sie Variablen auch außerhalb von Ausgabeanweisungen stets das Dollarzeichen voranstellen, wenn Sie ihren Wert auslesen möchten.

```
echo Summe: $summe
```

Diese Anweisung müssten Sie aus dem bisher Gesagten ohne Weiteres verstehen: Sie gibt den Text »Summe: « aus, gefolgt vom Wert der Variablen `summe`.

17.2.3 Variablen und Ausdrücke

Eines der wichtigsten Elemente jeder Programmiersprache sind *Variablen*, das heißt benannte Speicherplätze. In der Bash wird eine Variable mithilfe einer Anweisung nach folgendem Schema definiert:

```
Variable=Wert
```

Der Wert ist in diesem Fall zunächst immer ein String; setzen Sie ihn in Anführungszeichen, falls er Leerzeichen enthält. Beachten Sie, dass eine auf diese Art definierte Variable nur innerhalb des Shell-Skripts gilt. Angenommen, Sie führen zunächst folgendes Skript aus:

```
#!/bin/bash
text="Hallo Welt"
echo $text
```

Die Ausgabe lautet natürlich »Hallo Welt«. Geben Sie nach Ausführung des Skripts Folgendes ein:

```
$ echo $text
```

Die Ausgabe ist ein leerer String.

Auch umgekehrt werden Variablen, die Sie per Eingabe in der Shell definieren, nicht in Shell-Skripte übernommen. Auch dies können Sie ausprobieren. Geben Sie zunächst Folgendes ein, um eine Variable zu definieren und anschließend ihren Wert auszugeben:

```
$ planet=Erde
$ echo $planet
Erde
```

Nun können Sie folgendes Shell-Skript verfassen:

```
#!/bin/bash
echo $planet
```

Wenn Sie es ausführen, erhalten Sie keine Ausgabe. Geben Sie nun in der Shell diese Anweisung ein:

```
$ export planet
```

Führen Sie das Shell-Skript erneut aus, und Sie erhalten die gewünschte Ausgabe »Erde«.

`export` führt also dazu, dass die Variable in allen Prozessen zur Verfügung steht, die aus der aktuellen Shell heraus gestartet wurden. Die Variable gehört damit zur Umgebung dieser Child-Prozesse – auf dieselbe Weise werden die Umgebungsvariablen für die einzelnen Shells beziehungsweise User durch die Shell-Konfigurationsdateien gesetzt.

Als spezielle Variablen in Shell-Skripten stehen die Kommandozeilenargumente zur Verfügung: einmal durchnummeriert als `$1`, `$2` und so weiter (`$0` ist der Name des Skripts selbst) sowie als Array `$*`, das Sie in einer Schleife durchgehen können (siehe unten). Das folgende Beispiel gibt zunächst seinen eigenen Namen und anschließend das erste Argument aus:

```
#!/bin/bash
echo Ich bin:    $0
echo Sie sagten: $1
```

Speichern Sie dieses Skript beispielsweise unter dem Namen `gruss.sh`, machen Sie es ausführbar, und rufen Sie es wie folgt auf:

```
$ ./gruss.sh hallo
Ich bin:    ./gruss.sh
Sie sagten: hallo
```

Wie oben bereits erwähnt, müssen Sie das spezielle Kommando `let` verwenden, wenn Sie Variablen numerisch meinen. Das folgende Beispiel macht den Unterschied klar – die Variable `a` enthält den String "6+4", während `b` das numerische Ergebnis 10 enthält:

```
$ a=6+4
$ let b=6+4
$ echo $a
6+4
$ echo $b
10
```

Beachten Sie, dass die Variable bei Verwendung von `let` nicht dauerhaft numerisch bleibt. Sobald Sie den normalen Zuweisungsbefehl verwenden, wird sie wieder zum String. Betrachten Sie folgendes Beispiel mit der Variablen `b` aus dem vorigen Beispiel:

```
$ b=$b+10
$ echo $b
10+10
```

Eine Alternative besteht deshalb darin, den Typ einer Variablen mithilfe des Kommandos `typeset` festzulegen. Die allgemeine Syntax lautet:

```
typeset -Typ Variable
```

Das folgende Beispiel legt `c` verbindlich als ganzzahlige Variable (Typ `-i` für Integer) fest, weist ihr das Ergebnis einer Multiplikation zu und gibt dieses aus:

```
$ typeset -i c
$ c=7*6
$ echo $c
42
```

Neben `-i` gibt es einige weitere Variablentypen und `-optionen`, die Sie mithilfe von `typeset` festlegen können; einer der wichtigen ist etwa `-a` für ein Array.

17.2.4 Die Arithmetik-Sprache `bc`

Wenn Sie ansonsten arithmetische Berechnungen in Ihren Shell-Skripten benötigen, können Sie auch `bc` verwenden, den Konsolen-Taschenrechner mit beliebiger Präzision. Außerhalb von Skripten funktioniert `bc` interaktiv; Sie können ihn mithilfe der folgenden Anweisung erst einmal starten:

```
$ bc
bc 1.06
Copyright 1991-1994, 1997, 1998, 2000 Free Software Foundation, Inc.
This is free software with ABSOLUTELY NO WARRANTY.
For details type `warranty'.
```

Nun können Sie beliebige arithmetische Ausdrücke eingeben, die sofort berechnet werden:

```
7*8+6
62
```

Variablen können Sie ganz leicht definieren, indem Sie einfach `Variable=Wert` schreiben. Beispiele:

```
n=5
pi=3.1415926
```

Um den Wert einer Variablen wieder auszulesen, brauchen Sie nur ihren Namen einzugeben:

```
n
5
```

```
pi
3.1415926
```

Darüber hinaus können Sie Variablen auch in beliebigen Ausdrücken verwenden. Das folgende Beispiel berechnet den (ungefähren) Umfang eines Kreises mit dem Radius n :

```
2 * n * pi
31.4159260
```

Wie Sie sehen, ist es in `bc` völlig egal, ob Sie Leerzeichen zwischen den Operanden und Operatoren stehen lassen oder nicht.

Um den Taschenrechner zu verlassen, geben Sie `quit` ein oder betätigen einfach `Strg` + `D`.

`bc` hat eigentlich noch viel mehr Fähigkeiten als die wenigen, die hier dargestellt wurden; es handelt sich im Prinzip um eine ausgewachsene Programmiersprache mit C-ähnlichen Kontrollstrukturen. Außer von der Standardeingabe kann sie auch aus einer Datei lesen, der Aufruf lautet in diesem Fall `bc Dateiname`.

In Shell-Skripten nützt natürlich weder die Standardeingabe noch eine Eingabedatei etwas. Wenn Sie `bc` in einem Skript nutzen möchten, ist es am einfachsten, eine `echo`-Ausgabe über eine Pipe an den Rechner weiterzuleiten. Hier ein interaktives Beispiel:

```
$ echo "3 * 7 + 2" |bc
23
```

Eine einfachere Alternative ist in manchen Fällen `expr`. Dieses Kommando ist weniger leistungsfähig als `bc`, da es beispielsweise nicht mit Fließkommazahlen umgehen kann, aber dafür erlaubt es arithmetische Ausdrücke direkt als Parameter. Beachten Sie, dass Sie den Operanden bei der Multiplikation durch einen Backslash escapen müssen, weil das Sternchen für die Shell eine besondere Bedeutung besitzt. Beispiele:

```
$ expr 17 + 6
23
$ expr 14 \* 3
42
```

17.2.5 Fallentscheidungen

Die Bash unterstützt zwei Arten von Fallentscheidungen: `if` und die weiter oben bereits im Beispiel gezeigte `case`-Struktur. Ein fallentscheidungsähnliches Verhal-

ten zeigen zudem die in Kapitel 4 vorgestellten Operatoren `&&` und `||`, die mehrere Shell-Kommandos per Short-Circuit-Logik verknüpfen und die zweite Anweisung nur bei Erfolg (`&&`) beziehungsweise Misserfolg (`||`) der ersten ausführen.

if – else – fi

Anweisungen, die zwischen `if` Bedingung `then` und `fi` stehen, werden nur ausgeführt, wenn die Bedingung zutrifft – gemäß der Logik der Shell also dann, wenn sie den Rückgabewert 0 (keine Fehler) besitzt.

In der Regel besteht die Bedingung aus dem Vergleich zwischen einer Variablen oder einem Dateinamen und einem bestimmten Wert – ein solcher Ausdruck steht in der Bash in eckigen Klammern, die als Kurzfassung für das Kommando `test` verwendet werden. Wichtig ist, dass Sie nach der öffnenden und vor der schließenden eckigen Klammer jeweils ein Leerzeichen setzen, weil erstere der Kommandoname und letztere ein spezielles Argument ist. Geben Sie beispielsweise folgende Bedingung ein, um zu testen, ob die Variable `zahl` den Wert 10 besitzt:

```
[ $zahl -eq 10 ]
```

Dieses Kommando stellt zahlreiche Testoperationen zur Verfügung, die den Typ und andere Eigenschaften von Dateien sowie die Gleichheit oder Ungleichheit von Strings beziehungsweise Ganzzahlen betreffen. In Tabelle 17.1 sehen Sie die wichtigsten im Überblick.

Testausdruck	Bedeutung	Trifft zu, wenn ...
<code>-e Dateiname</code>	<code>exists</code>	... die angegebene Datei existiert.
<code>-s Dateiname</code>	<code>size</code>	... die Datei existiert und nicht leer ist.
<code>-f Dateiname</code>	<code>file</code>	... die Datei existiert und eine reguläre Datei ist.
<code>-d Dateiname</code>	<code>directory</code>	... die Datei existiert und ein Verzeichnis ist.
<code>int1 -eq int2</code>	<code>equals</code>	... die beiden ganzen Zahlen <code>int1</code> und <code>int2</code> gleich sind.
<code>int1 -ne int2</code>	<code>not equals</code>	... die beiden ganzen Zahlen <code>int1</code> und <code>int2</code> ungleich sind.
<code>int1 -lt int2</code>	<code>less than</code>	... <code>int1</code> kleiner als <code>int2</code> ist.
<code>int1 -le int2</code>	<code>less than or equals</code>	... <code>int1</code> kleiner oder gleich <code>int2</code> ist.

Tabelle 17.1 Die wichtigsten Test-Argumente

Testausdruck	Bedeutung	Trifft zu, wenn ...
<code>int1 -gt int2</code>	greater than	... <code>int1</code> größer als <code>int2</code> ist.
<code>int1 -ge int2</code>	greater than or equals	... <code>int1</code> größer oder gleich <code>int2</code> ist.
<code>-n String</code>	nonzero	... die Länge des Strings größer ist als 0 Zeichen.
<code>-z String</code>	zero	... die Länge des Strings null ist.
<code>str1 = str2</code>	(equals)	... die beiden Strings <code>str1</code> und <code>str2</code> identisch sind.
<code>str1 != str2</code>	(not equals)	... die beiden Strings <code>str1</code> und <code>str2</code> verschieden sind.
<code>(Ausdruck)</code>	–	... der Ausdruck wahr ist.
<code>! Ausdruck</code>	(not)	... der Ausdruck falsch ist.
<code>Ausdr1 -a Ausdr2</code>	and	... beide Ausdrücke wahr sind.
<code>Ausr1 -o Ausdr2</code>	or	... mindestens einer der beiden Ausdrücke wahr ist.

Tabelle 17.1 Die wichtigsten Test-Argumente (Forts.)

Hier ein Beispiel, das eine Meldung ausgibt, wenn die Datei *hallo.txt* existiert:

```
if [ -e hallo.txt ]
then
    echo hallo.txt gefunden
fi
```

Das nächste Beispiel beendet das Skript mit einer Fehlermeldung und einem Exit-Code ungleich 0, wenn kein erstes Kommandozeilenargument übergeben wurde:

```
if [ -z $1 ]
then
    echo Verwendung: $0 \<Argument\>
    exit 1
fi
```

Beachten Sie, dass Sie die Zeichen `<` und `>` jeweils durch den vorangestellten Backslash escapen müssen, weil sie andernfalls dazu dienen, die Ein- beziehungsweise Ausgabe umzuleiten. Dasselbe gilt unter anderem auch für die Pipe (`|`) oder das Dollarzeichen (`$`).

Für numerische Vergleiche können Sie stattdessen auch doppelte runde Klammern verwenden: `((...))`. Darin sind dann die normalen arithmetischen Operatoren erlaubt:

- ▶ `==` für Gleichheit; zum Beispiel `((3 == 3))`
- ▶ `!=` für Ungleichheit, wie etwa `((3 != 4))`
- ▶ `<` für kleiner als, zum Beispiel `((2 < 5))`
- ▶ `<=` für kleiner oder gleich, beispielsweise `((4 <= 4))`
- ▶ `>` für größer als, etwa `((6 > 3))`
- ▶ `>=` für größer oder gleich, zum Beispiel `((5 >= 4))`

Das folgende Beispiel gibt eine Meldung aus, wenn die Variable `punkte` kleiner als 100 ist:

```
if (($punkte < 100))
then
    echo Punktzahl $punkte zu klein.
fi
```

`if` kann auch verwendet werden, um ein beliebiges Kommando zu testen; schreiben Sie den Befehl dazu einfach ohne Klammern oder Ähnliches hinter das Kommando. Wie bereits erwähnt, werden alle Befehle mit einem Exit-Code beendet; 0 bedeutet, dass alles in Ordnung ist, während andere Werte auf Fehler hindeuten. Das folgende Beispiel gibt eine Meldung aus, wenn die Datei `old.txt` erfolgreich gelöscht werden konnte:

```
if rm old.txt
then
    echo old.txt gelöscht
fi
```

Hinter der Anweisung `else` können Sie optional alternative Anweisungen definieren, die ausgeführt werden sollen, wenn die Bedingung nicht zutrifft. Das folgende Beispiel gibt jeweils alternative Meldungen aus, wenn `eintrag` ein Verzeichnis oder eben keines ist (dabei sei dahingestellt, ob es eine normale Datei oder eine andere Art von Verzeichniseintrag ist oder aber gar nicht existiert):

```
if [ -d eintrag ]
then
    echo Verzeichnis
else
    echo Kein Verzeichnis
fi
```

Mithilfe von `elif` (Abkürzung für »else if«) kann sogar eine verschachtelte Bedingung eingesetzt werden. Hier ein Beispiel, das verschiedene Dateitests an eintrag durchführt:

```
if [ -f eintrag ]
then
    echo Normale Datei
elif [ -d eintrag ]
then
    echo Verzeichnis
elif [ -e eintrag ]
then
    echo Existiert, aber ist etwas anderes (Symlink, Pipe o.Ä.)
else
    echo Existiert nicht
fi
```

Hier ein weiteres Beispiel; es führt eine Kette numerischer Vergleiche durch (Punktzahlbereiche in einem Spiel):

```
if (($punkte < 50))
then
    echo Üben, üben, üben!
elif (($punkte < 100))
then
    echo Für den Anfang nicht schlecht!
else
    echo Hervorragend! Hast du geschummelt?
fi
```

case – esac

Die andere wichtige Sorte von Shell-Fallentscheidungen sind die *Einzelfallentscheidungen*: Zwischen `case` und `esac` können Sie verschiedene Muster (kompatibel zu den Dateimustern der Shell) auflisten, denen eine angegebene Variable oder ein String mit Variablen entsprechen könnte. Die vollständige Schreibweise ist `case String in`.

Hinter jedem Muster, das überprüft wird, steht eine schließende Klammer `)`, darauf folgen beliebig viele Anweisungen, die nur ausgeführt werden, wenn das angegebene Muster auf den überprüften String passt. Vor dem nächsten Muster muss eine Befehlssequenz durch `;;` abgeschlossen werden. Praktischerweise können Sie abschließend ein `*` als Universalmuster einsetzen, das auf »alle anderen Fälle« passt – sehr nützlich etwa zum Ausschluss ungültiger Kommandozeilenargumente.

Das folgende Beispiel vergleicht den ersten Kommandozeilenparameter – \$1 – mit diversen festen Werten und führt daraufhin spezielle Operationen mit der als zweiten Parameter (\$2) übergebenen Textdatei durch: Ist der erste Wert -a, so wird ein Kommentar mit Datum und Uhrzeit angehängt, bei -d wird die betreffende Datei gelöscht. -v schließlich gibt nur eine Versionsinformation aus und -h eine Hilfe-Nachricht. Alle anderen Werte führen zu einer Fehlermeldung.

```
#!/bin/bash
# Verwendungsinfo speichern
usage="Verwendung: $0 -a|-d <Datei>"
# Versionsinfo speichern
version="$0 Version 0.01"
# Datum und Uhrzeit speichern
$jetzt=`date "+%d.%m.%Y, %H:%M"`
case $1 in
-a)
  # Testen, ob $2 überhaupt eine reguläre Datei ist
  # (ansonsten Ende mit Fehlermeldung)
  if [ -f $2 ]
  then
    echo $0 war hier am $jetzt >>$2
  else
    echo $0: $2 ist keine Datei
    exit 1
  fi
  ;;
-d)
  # Testen, ob $2 überhaupt eine reguläre Datei ist
  # (ansonsten Ende mit Fehlermeldung)
  if [ -f $2 ]
  then
    rm $2
  else
    echo $0: $2 ist keine Datei
    exit 1
  fi
  ;;
-v)
  echo $version
  ;;
-h)
  echo $usage
  ;;
*)
```

```

    echo Ungültiger Aufruf
    echo $usage
    exit 1
    ;;
esac

```

Angenommen, Sie rufen das Skript – hier *filetest.sh* genannt – völlig ohne Argumente auf:

```
$ ./filetest.sh
```

Da die mit - beginnenden Muster nicht auf den fehlenden Parameter zutreffen, tritt hier der Fall *) in Kraft – »beliebig viele beliebige Zeichen« trifft schließlich auch auf gar keine Zeichen zu. Dasselbe gilt übrigens auch, wenn Sie als ersten Parameter etwas anderes als die vier zulässigen Befehle eingeben.

Wenn Sie das Skript dagegen beispielsweise mit `-a Dateiname` starten, enthält die entsprechende Datei als zusätzliche Zeile den Eintrag, den das Skript hinzugefügt hat:

```

$ ./filetest.sh hallo.txt
$ tail -1 hallo.txt
./filetest.sh war hier am 07.04.2009, 17:34

```

17.2.6 Schleifen

Mitunter müssen bestimmte Anweisungen mehrfach ausgeführt werden. Dafür sind *Schleifen* zuständig. Die *bash* definiert verschiedene Arten von Schleifen; die wichtigsten sind die *for*-Schleife und die *while*-Schleife.

Die for-Schleife

Eine *for*-Schleife geht automatisch eine Liste von Werten durch. Die Werte der Liste werden nacheinander der angegebenen Schleifenvariablen zugewiesen. Die Anweisungen, die in jedem Durchlauf der Schleife ausgeführt werden sollen, stehen zwischen *do* und *done*. Das folgende Beispiel durchwandert eine Reihe von Zahlen, gibt sie einzeln aus und addiert sie:

```

#!/bin/bash
let summe=0
for i in 17 18 19
do
    echo $i
    let summe=$summe+$i
done

```

```
echo ----
echo $summe
```

In der Ausführung sieht es folgendermaßen aus:

```
17
18
19
----
54
```

Alternativ können Sie mittels `for Variable in Muster` ein Dateimuster angeben; die Variable nimmt dann nacheinander den Namen jeder Datei an, auf die dieses Muster passt. Hier ein Beispiel, das die Namen und die Anzahl aller Dateien mit der Endung `.txt` im aktuellen Verzeichnis ausgibt:

```
#!/bin/bash
# Zähler auf 0 setzen
let count=0
# Schleife über alle Textdateien
for txt in *.txt
do
    let count=$count+1
    echo $count. $txt
done
echo -----
echo Gesamt: $count Textdateien.
```

In einem Verzeichnis, in dem einige Dateien dieses Buches stehen, sieht das Ergebnis beispielsweise so aus:

```
$ ./textcount
1. bash.txt
2. chaplist.txt
3. commandref.txt
4. editoren.txt
5. emacs_notes.txt
6. gimpnotes.txt
7. latex.txt
8. mysql.txt
9. prozesse.txt
10. regex.txt
11. sh.txt
12. sudomsg.txt
13. sysfiles.txt
14. test2.txt
```

```

15. testcommands.txt
16. test.txt
17. vi_backup.txt
18. vimport.txt
19. vi.txt
-----

```

Gesamt: 19 Textdateien.

Interessant ist es auch, eine `for`-Schleife über alle Kommandozeilenargumente laufen zu lassen. Die Syntax lautet `for Variable in $*`. Die Schleife geht ein Argument nach dem anderen durch; wenn Dateimuster enthalten sind, werden auch diese aufgelöst. Das folgende Beispiel überprüft für jedes Argument, ob es Vokale enthält, und gibt zum Schluss eine Statistik aus:

```

#!/bin/bash
# Zählervariablen definieren
let mitvok=0
let ohnevok=0
# Schleife über alle Kommandozeilenargumente
for arg in $*
do
    case $arg in
    *[aeiou]*)
        # Mit Vokalen
        echo $arg: Mit Vokalen
        let mitvok=$mitvok+1
        ;;
    *)
        # Ohne Vokale
        echo $arg: Ohne Vokale
        let ohnevok=$ohnevok+1
        ;;
    esac
done
echo $mitvok Wörter mit Vokalen, $ohnevok ohne.
let summe=$mitvok+$ohnevok
echo $summe Wörter insgesamt.

```

Hier ein Beispiel für die Ausführung dieses Skripts:

```

$ ./vokzaehler PHP Perl C Java XML Python C++
3 Wörter mit Vokalen, 4 ohne.
7 Wörter insgesamt.

```

Zu guter Letzt können Sie eine `for`-Schleife auch über die Ausgabe eines Befehls iterieren lassen, indem Sie beispielsweise Backticks verwenden.

Ein Problem besteht nur darin, dass solche Ausgaben standardmäßig an beliebigem Whitespace (Leerzeichen, Tabs und Zeilenumbrüche) zerteilt werden. Zuständig ist die Umgebungsvariable `IFS` (Abkürzung für »input field separator«), deren Standardwert `' \t\n'` lautet. Wenn Sie ganze Zeilen verarbeiten oder Felder bei anderen Trennzeichen teilen möchten, können Sie sie vorübergehend auf einen anderen Wert setzen. Das folgende Beispiel speichert den bisherigen Wert in einer selbst definierten Variablen namens `ifscopy`, setzt `IFS` auf den Zeilenumbruch und stellt den ursprünglichen Wert nach getaner Arbeit wieder her:

```
ifscopy=$IFS
IFS=$'\n'
# Zeilen verarbeiten
# ...
# IFS zurücksetzen
IFS=$ifscopy
```

Andere beliebte Ersatzwerte sind etwa `':'`, um die Bestandteile von `/etc/passwd`- oder `PATH`-Einträgen zu verarbeiten, oder `'/'`, um Pfade zu zerlegen.

Hier nun ein ausführliches Beispiel dazu: Es zählt anhand des Beginns von Einträgen in der Ausgabe von `ls -l` die Verzeichnisse (die Einträge beginnen mit `d`), Symlinks (beginnen mit `l`) und gewöhnlichen Dateien (beginnen mit `-`):

```
#!/bin/bash
# Verzeichnis aus Kommandozeilenparameter lesen
# oder auf '.' (akt. Verzeichnis) setzen
testdir=$1 || testdir=.
# Original-Eingabeseparator speichern
ifscopy=$IFS
# Eingabeseparator auf Zeilenumbrüche beschränken
IFS=$'\n'
# Zählervariablen initialisieren
let direcs=0
let links=0
let files=0
for entry in `ls -l $testdir |sort`
do
  case $entry in
    d*)
      # Verzeichnis
      let direcs=$direcs+1
      ;;
    l*)
      # Symlink
      let links=$links+1
```

```

    ;;
-*)
    # Normale Datei
    let files=$files+1
    ;;
esac
done
# Eingabeseparator zurücksetzen
IFS=$ifscopy
echo $direcs Verzeichnisse.
echo $files normale Dateien.
echo $links Symlinks.

```

Interessant ist in diesem Beispiel zusätzlich die folgende Formulierung:

```
testdir=$1 || testdir=.
```

Hier wird versucht, `testdir` durch Auslesen des ersten Kommandozeilenparameters zu setzen. Falls dieser nicht verfügbar ist, schlägt diese Aktion fehl, und der zweite Teil der Oder-Verknüpfung setzt als Fallback-Wert das aktuelle Verzeichnis (`.`).

Hier ein Beispielaufruf für das Systemverzeichnis `/etc`:

```

$ ./typ.sh /etc
108 Verzeichnisse.
206 normale Dateien.
7 Symlinks.

```

Die while-Schleife

`while`-Schleifen verwenden eine Bedingung im gleichen Format wie `if`; der einzige Unterschied besteht darin, dass die Anweisungen mehrmals ausgeführt werden, solange die Bedingung noch zutrifft. Auch bei `while`-Schleifen werden `do` und `done` verwendet.

Hier ein einfaches Beispiel, das Interaktivität einsetzt – `read Variable` liest die angegebene Variable von der Standardeingabe. Nach Anzeigen eines Menüs wird eine Benutzereingabe verlangt; je nach Art der Eingabe erfolgt die Ausgabe von Datum oder Uhrzeit:

```

#!/bin/bash
echo "(D)atum, (U)hrzeit, oder (E)nde?"
while read input
do
    case $input in
    [dD])

```

```

    heute=`date "+%d.%m.%Y"`
    echo $heute
    ;;
[uU])
    jetzt=`date "+%H:%M"`
    echo $jetzt
    ;;
[eE])
    break
    ;;
*)
    echo Ungültige Auswahl
    ;;
esac
done

```

Die Schleifenbedingung

```
while read input
```

liest den Inhalt der Variablen `input` so lange aus der Standardeingabe, bis keine weitere Eingabe mehr erfolgt – mit anderen Worten bis EOF. Daher können Sie das Skript auf Wunsch auch mit einer Eingabedatei füttern.

Die spezielle Anweisung `break` wird verwendet, um Schleifen – in diesem Fall die `while`-Schleife – zu verlassen. Anders als in C und vielen anderen Programmiersprachen hat `break` in Shell-Skripten nichts mit `case`-Strukturen zu tun.

Die `case`-Muster `[dD]`, `[uU]` und `[eE]` sorgen dafür, dass der Benutzer sowohl Groß- als auch Kleinbuchstaben eingeben kann. Hier eine komplette »Beispielsitzung«:

```

$ ./uhr.sh
(D)atum, (U)hrzeit oder (E)nde?
d
09.09.2009
u
10:52
x
Ungültige Auswahl
e

```

17.2.7 Funktionen

Wie in vielen anderen Programmiersprachen können Sie auch in der Shell Funktionen definieren. Diese ermöglichen es Ihnen, wiederkehrende Aufgaben nur

einmal hinschreiben zu müssen. Funktionen werden durch Klammern hinter dem Bezeichner deklariert; der Funktionsrumpf (die auszuführenden Anweisungen) steht in geschweiften Klammern. Schematisch sieht dies folgendermaßen aus:

```
Funktionsname()
{
    Anweisung(en)
}
```

Statt des Namens mit den Klammern können Sie alternativ auch das Schlüsselwort `function` verwenden. Das entsprechende Syntaxschema lautet wie folgt:

```
function Funktionsname
{
    Anweisung(en)
}
```

Hier zunächst ein sehr einfaches Beispiel mit einer Funktion, die Datum und Uhrzeit ausgibt:

```
#!/bin/bash
jetzt()
{
    date "+%d.%m.%Y, %H:%M"
}
Echo Heute ist:
jetzt
```

Das einfache `jetzt` ist der Funktionsaufruf. Die Ausgabe lautet etwa:

```
Heute ist:
10.09.2009, 13:52
```

Besonders nützlich werden Funktionen, wenn sie Parameter entgegennehmen können. Shell-Funktionen können dies automatisch, ohne dass Sie Parametervariablen zu deklarieren brauchen. Die Namen der automatischen Parametervariablen kennen Sie auch schon: Es handelt sich um `$1`, `$2` und so weiter, das heißt um dieselben Bezeichner, die im »Hauptskript« für die Kommandozeilenparameter verwendet werden.

Das folgende Beispiel versucht, die Benutzereingabe als `grep`-Muster für den Inhalt des aktuellen Verzeichnisses zu verwenden:

```
#!/bin/bash
dirgrep()
{
```

```

    ls |grep "$1"
}
echo "Bitte ein Muster eingeben, nach dem gesucht werden soll:"
read muster
dirgrep $muster

```

Speichern Sie dieses Skript beispielsweise als *dirgrep.sh*, machen Sie es ausführbar und starten Sie es. Sofort werden Sie nach einem regulären Ausdruck gefragt, nach dem die Dateiliste des aktuellen Verzeichnisses durchsucht werden soll. Hier ein Ausführungsbeispiel:

```

$ ./dirgrep.sh
Bitte ein Muster eingeben, nach dem gesucht werden soll:
.pdf$
latexinfo.pdf
latex_sample.pdf
testdokument.pdf

```

Das eingegebene Muster *.pdf\$* steht für alle Dateien, die mit *.pdf* enden (\$ bezeichnet das Ende eines Ausdrucks).

17.2.8 sed und awk

Shell-Skripte sind für sich allein sehr schlecht zur Verarbeitung von Textdaten geeignet. In der Regel ist es am praktischsten, dafür auf externe Hilfsprogramme zuzugreifen. *grep* für die Suche nach regulären Ausdrücken haben Sie bereits in Kapitel 4 kennengelernt. Hier werden anhand kurzer Beispiele zwei weitere Werkzeuge vorgestellt: *sed* und *awk*.

Der *Stream-Editor* *sed* ist ein befehlsorientierter, nichtinteraktiver »Editor« zur beliebigen Umwandlung von Texten. Er basiert auf einem erweiterten Satz regulärer Ausdrücke. Die meisten von ihnen werden auch von Perl unterstützt und daher weiter unten vorgestellt.

Die allgemeine Kommandozeilen-Syntax von *sed* lautet:

```
sed 'Kommando' [Eingabedatei]
```

Wenn Sie keine Eingabedatei angeben, liest *sed* von der Standardeingabe; die Ausgabe erfolgt auf jeden Fall auf die Standardausgabe. In vielen Fällen sind Sie daher auf Ausgabeumleitung angewiesen, um etwas Sinnvolles damit zu erledigen.

Das folgende interaktive Beispiel ersetzt alle Vokale im Eingabetext mithilfe der Substitutionsanweisung *s///* durch ***; die Optionen *gi* bedeuten »global« (alle

ersetzen) und »(case) insensitive«, das heißt ohne Beachtung von Groß- und Kleinschreibung:

```
$ sed 's/[aeiou]*/gi'
Hallo Welt hier
H*ll* W*lt h**r
```

+

Der Operator `s///` wird weiter unten für Perl ausführlicher besprochen.

Wenn Sie umfangreichere Abfolgen von `sed`-Anweisungen verwenden möchten, können Sie diese auch in eine Skriptdatei schreiben und mithilfe der Option `-f` ausführen. Das folgende Beispiel bearbeitet die Textdatei `test.txt` mit den Anweisungen des `sed`-Skripts `work.sed` und gibt das Ergebnis in `test2.txt` aus:

```
$ sed -f work.sed test.txt >test2.txt
```

`awk` (benannt nach seinen Entwicklern *Alfred Aho*, *Brian Kernighan* und *Peter Weinberger*) ist eine etwas umfangreichere Sprache, die auch Variablen und Kontrollstrukturen unterstützt, aber eine ähnliche Zielsetzung verfolgt wie `sed`.

Ein wichtiger Unterschied zu `sed` besteht darin, dass die einzelnen Zeilen der Standardeingabe oder einer Eingabedatei zwar ebenfalls automatisch eingelesen, aber nicht unbedingt wieder ausgegeben werden. Die Eingabe steht außerdem nicht nur durch die `sed`-ähnlichen Muster, sondern auch in Form der Variablen `$0` (die ganze Zeile) sowie `$1` bis `$n` (einzelne Tokens) zur Verfügung; Sie können Teile davon, bei Bedarf per RegExp-Kommandos modifiziert, mit `print` wieder ausgeben.

Das folgende Beispiel müssen Sie als Skript speichern. Es gibt aufgeschlüsselte Typinformationen und Dateinamen aus einem Standardlisting von `ls -l` aus:

```
{
  if (/^d/) {
    print "Verzeichnis:";
  } else if (/^l/) {
    print "Symlink:";
  } else if (/^\-/ ) {
    print "Datei: ";
  } else {
    print "Sonstiges: ";
  }
  print $8;
}
```

Wenn Sie dieses Skript für den Inhalt des aktuellen Verzeichnisses aufrufen möchten, können Sie Folgendes eingeben:

```
$ ls -l |awk -f dirtest.awk
```

Die automatisch generierte Variable `$8` steht in diesem Fall für das achte (letzte) Token des Verzeichniseintrags, nämlich den Dateinamen. Genau wie in der Shell gilt auch in `awk` standardmäßig jeder beliebige Whitespace als Feldseparator.

Die Syntax von Kontrollstrukturen wie `if()` entspricht in `awk` übrigens eins zu eins der Programmiersprache C; das ist nicht weiter verwunderlich, weil Brian Kernighan an der Entwicklung beider Sprachen mitgewirkt hat.

Da Sie den jeweiligen Interpreter explizit als Kommando aufrufen, brauchen Sie `sed`- und `awk`-Skripte übrigens nicht ausführbar zu setzen.

17.3 Die Alternative Perl

Nachdem Sie in diesem Kapitel manches über das Schreiben von Shell-Skripten erfahren haben, muss ich Ihnen doch ein Geständnis machen: Für meine tägliche Arbeit schreibe ich so gut wie nie Shell-Skripte. Das liegt nicht etwas daran, dass ich die entsprechende Funktionalität nicht benötigen würde, sondern hat damit zu tun, dass ich vor einigen Jahren beschlossen habe, *Perl* zu lernen. Diese mächtige Sprache kombiniert die Flexibilität von Shell-Skripten mit den Funktionen von `sed` und `awk` und fügt sogar mancherlei Funktionen hinzu, die sonst nur in C verfügbar sind.²

Perl wurde 1987 von *Larry Wall* entwickelt, als er im Rahmen eines Programmierauftrags umfangreiche Datenbestände koordinieren musste, die auf Rechnern an beiden Küsten der USA verteilt waren. Heute wird Perl vorwiegend von UNIX-Systemadministratoren verwendet, um ihre tägliche Arbeit zu erleichtern. Außerdem ist Perl die bevorzugte Sprache für CGI-Skripte, die klassische Form der Web-Anwendung. Die Perl-Mottos »There's more than one way to do it« und »Make easy things easy and hard things possible« geben einen Eindruck von der Vielseitigkeit, Praxisnähe und Leistungsfähigkeit dieses Werkzeugs.

² Inzwischen erledige ich die meisten Automatisierungsaufgaben auch nicht mehr mit Perl, sondern mit Ruby. Das hat mit dessen eleganterer Syntax und der konsequenten Objektorientierung zu tun – vernünftige Objektorientierung ist in Perl erst für Version 6 geplant. Ruby übernimmt aber gerade im Bereich der Systemverwaltung zahlreiche Perl-Konzepte. Wenn Sie sich für Ruby interessieren, können Sie die Einführung in der Online-Ausgabe meines Buches »IT-Handbuch für Fachinformatiker« (http://www.galileocomputing.de/openbook/it_handbuch) lesen.

Die Abkürzung Perl steht eigentlich für *Practical Extraction and Report Language*, wird aber von Larry Wall und den anderen Perl-Hackern manchmal auch als *Pathologically Eclectic Rubbish Lister* (etwa: krankhaft sammelwütiger Auflister für jeden Mist) bezeichnet. Solcher Humor ist ein typisches Merkmal der Perl-Gemeinde und hat auch die Sprache selbst beeinflusst (und umgekehrt!).

Die derzeitige Version von Perl ist 5.10 (aktuelle Release im Juli 2008: 5.10.0). Die kommende Version 6 wird viele Neuerungen bringen, insbesondere eine virtuelle Maschine namens *Parrot*, auf die neben Perl auch andere Sprachen aufsetzen können.

Um ein Perl-Programm auf Ihrem Rechner auszuführen, müssen Sie es dem Perl-Interpreter übergeben. Dies geschieht wie bei Shell-Skripten durch die Shebang-Zeile zu Beginn des Skripts:

```
#!/usr/bin/perl
```

Eine auf diese Weise präparierte Datei müssen Sie nur noch mittels `chmod +x` ausführbar setzen, um sie zu starten. Empfehlenswert ist übrigens die folgende Variante der Shebang:

```
#!/usr/bin/perl -w
```

Die Option `-w` schaltet ausführliche Warnmeldungen ein, die Sie auf mögliche Ungenauigkeiten oder Zweideutigkeiten hinweisen können. Dies kann auf Dauer Ihren Programmierstil erheblich verbessern.

17.3.1 Das erste Beispiel

Damit Sie eine Vorstellung von der Funktionsweise der Programmiersprache erhalten, sehen Sie hier zunächst ein vollständiges »Hallo Welt«-Programm mit Benutzereingabe, Zeilenausgabe sowie Datum und Uhrzeit:

```
#!/usr/bin/perl -w
print "Hallo Welt!\n";
$jetzt = scalar localtime;
print "Es ist $jetzt.\n";
print "Ihr Name, bitte: ";
$name = <>;
chomp $name;
print "Hallo $name!\n";
```

Speichern Sie das Skript unter *hallo.pl*, machen Sie es ausführbar, und starten Sie es. Hier alle entsprechenden Schritte im Überblick:

```
$ chmod a+x hallo.pl
$ ./hallo.pl
```

```
Hallo Welt!
Es ist Thu Sep 10 12:24:05 2009.
Ihr Name, bitte: Sascha
Hallo Sascha!
```

Ähnlich wie Shell-Skripte müssen auch Perl-Skripte keine äußeren Formalitäten einhalten: Es gibt kein explizites Hauptprogramm; ein Skript ist im einfachsten Fall lediglich eine Folge normaler Anweisungen. Im Folgenden werden die Zeilen des »Hallo Welt«-Skripts nacheinander beschrieben:

- ▶ `print "Hallo Welt!\n";`
`print "Ihr Name, bitte: ";`
 Mithilfe von `print` werden Inhalte auf die Standardausgabe geschrieben. In Perl können Sie die Klammern um Funktionsargumente weglassen; dies ist nicht immer übersichtlich, aber speziell bei `print` allgemein üblich.
- ▶ `$jetzt = scalar localtime;`
 In Perl beginnen die Namen der verschiedenen Arten von Variablen mit speziellen Zeichen. Eine Variable, die einen einzelnen Wert speichern kann (Skalar genannt), fängt mit einem Dollarzeichen an; die beiden anderen wichtigen Typen werden weiter unten behandelt. In der Variablen `$jetzt` werden an dieser Stelle Datum und Uhrzeit gespeichert: Die Funktion `localtime` ermittelt aus einer ganzen Zahl, die als Sekunden seit `EPOCH` interpretiert wird, den entsprechenden Zeitpunkt unter Berücksichtigung von Zeitzone und etwaiger Sommerzeit; wenn kein Argument übergeben wird, verwendet sie die aktuelle Systemzeit. Die Helferfunktion `scalar` macht daraus einen einzelnen, für Menschen lesbaren String.
- ▶ `print "Es ist $jetzt.\n";`
 Genau wie die Shell führt auch Perl Variablensubstitution durch. In diesem String wird `$jetzt` also vor der Ausgabe durch seinen Wert ersetzt.
- ▶ `$name = <>;`
 Mithilfe des Operators `<>` wird eine Zeile Text von der Standardeingabe gelesen. Wie weiter unten besprochen wird, kann zwischen den Klammern auch eine Datei angegeben werden, aus der gelesen werden soll. Im vorliegenden Fall wird die gelesene Zeile in der Variablen `$name` gespeichert.
- ▶ `chomp $name;`
 Die Funktion `chomp` entfernt den Zeilenumbruch vom Ende eines Strings. Der Zeilenumbruch, mit dem die eingegebene Zeile endet, wird nämlich von `<>` mit eingelesen und stört in vielen Fällen.
- ▶ `print "Hallo $name!\n";`
 Noch eine Ausgabe mit Variablensubstitution.

17.3.2 Elemente der Sprache Perl

Im Folgenden werden verschiedene wichtige Bestandteile von Perl systematisch vorgestellt.

Variablen

Wie die meisten interpretierten Skriptsprachen ist auch Perl eine untypisierte Sprache. Variablen besitzen keinen festen Datentyp, sondern ihr gespeicherter Inhalt wird je nach Kontext interpretiert. Außerdem können sie nacheinander verschiedene Werte annehmen, sodass Zuweisungen wie die folgenden erlaubt sind:

```
$a = "hallo";    # String
$a = 3;         # ganze Zahl
$a = 4.678;    # Fließkommazahl
```

Eine Variable existiert automatisch durch die erste Wertzuweisung und muss nicht deklariert werden. Dadurch werden Programme einerseits kürzer und kompakter, aber andererseits werden Fehler nicht so leicht bemerkt, weil Perl sich nicht beschwert, wenn Sie sich bei einem Variablennamen verschreiben. Eine Variable, die in einem Ausdruck verwendet wird, ohne dass ihr ein Wert zugewiesen wurde, hat den speziellen Wert `undefined`. Dies führt möglicherweise zu unerwarteten Ergebnissen, nicht aber zu einer Fehlermeldung.

Für längere Programme bietet es sich deshalb an, die *strenge Typüberprüfung* zu aktivieren. Diese Option verlangt, dass jede Variable bei ihrer ersten Verwendung explizit deklariert wird. Die Typüberprüfung wird zu Beginn eines Programms (meist unmittelbar unter der Shebang) folgendermaßen eingeschaltet:

```
use strict;
```

Innerhalb eines normalen Perl-Skripts werden Variablen bei strenger Typüberprüfung durch ein vorangestelltes `my` deklariert; in den weiter unten vorgestellten Subroutinen mit `local`:

```
#!/usr/bin/perl
use strict;
my $zahl1;    # explizite Deklaration
my $zahl2 = 3; # implizite Deklaration bei Wertzuweisung
$zahl3 = 7;   # Fehlermeldung: Nicht deklariert!
```

Perl kennt drei verschiedene Arten von Variablen: Skalare, Arrays und Hashes.

- ▶ Ein *Skalar* ist eine normale »Einzelvariable«. Ihr Wert ist ein einzelner String, eine Zahl oder eine Referenz auf ein beliebiges Element. Die Namen von Skalaren beginnen mit einem Dollarzeichen, zum Beispiel \$wert.
- ▶ Ein *Array* ähnelt der gleichnamigen Datenstruktur in C und Java, ist aber erheblich flexibler. Die Anzahl der Elemente im Array kann dynamisch erhöht oder vermindert werden. Der Name der gesamten Array-Variablen beginnt mit dem Zeichen @, beispielsweise @werte.
- ▶ Ein *Hash*, auch assoziatives Array genannt, ähnelt einem gewöhnlichen Array. Das Besondere daran ist allerdings, dass die Indizes für die einzelnen Felder keine aufeinanderfolgenden Integer-Werte sind, sondern beliebige Strings sein können. Der Name eines Hashs beginnt mit % wie etwa %wochentage.

Ein Array kann auf zwei verschiedene Arten erzeugt werden: Entweder weisen Sie dem gesamten Array eine *Liste* als Wert zu, oder Sie erstellen sofort ein einzelnes Element des Arrays:

```
@tiere = ("Tux", "Geeko", "Sakila"); # Liste zuweisen
$distro[0] = "openSUSE";           # Einzelnes Element
$distro[1] = "Ubuntu";
$distro[2] = "Fedora Core";
$distro[3] = "Debian";
```

Da die Elemente von Arrays Skalare sind, ist das Vorzeichen für ein einzelnes Element kein @, sondern ein \$.

Bei der Formulierung ("Tux", "Geeko", "Sakila") handelt es sich um eine *Liste*. Listen können nicht nur in Arrays gespeichert werden, sondern auch ein interessantes Eigenleben führen. Sie können sogar einer Liste skalarer Variablen einen Wert zuweisen. Dies ermöglicht beispielsweise die folgende praktische Schreibweise für den Tausch der Werte zweier Variablen:

```
($a, $b) = ($b, $a);
```

In den meisten anderen Sprachen benötigen Sie für einen solchen Tausch eine dritte Variable als Zwischenspeicher und müssen drei Anweisungen schreiben.

Perl definiert darüber hinaus eine Reihe nützlicher Funktionen zur Behandlung von Arrays und Listen. Einige wichtige Beispiele sind folgende:

```
$w = pop(@array); # Den letzten Wert des Arrays ab-
                  # trennen und zurückgeben
push (@array, $w); # Skalar an ein Array anhängen
$w = shift (@array); # Den ersten Wert des Arrays ab-
                    # trennen und zurückgeben
```

```
unshift (@array, $w); # Skalar vor das erste Element
                    # eines Arrays setzen
```

Mithilfe der Funktionen `pop`, `push`, `shift` und `unshift` können Sie allerdings nicht nur einzelne Elemente von Arrays abtrennen oder ihnen hinzufügen, sondern auch Listen. Beispielsweise hängt die folgende Anweisung eine Liste mit zwei weiteren Programmiersprachen an das weiter oben definierte Array `@tiere` an:

```
push (@tiere, ("Konqui", "Kamel"));
```

Für eine automatische Schleife über alle Elemente eines Arrays wird die spezielle Kontrollstruktur `foreach` verwendet. Das folgende Beispiel gibt die Namen dreier in einem Array gespeicherter Prozessoren untereinander aus:

```
@prozessoren = ("Intel i7", "Athlon II", "PowerPC");
foreach $prozessor(@prozessoren) {
    print "$prozessor\n";
}
```

Wenn Sie die Angabe der Schleifenzählervariablen (hier `$prozessor`) weglassen, stellt Perl übrigens automatisch die spezielle Variable `$_` dafür zur Verfügung. Diese wird von vielen Funktionen automatisch verwendet, wenn kein anderes Argument angegeben wird. Die obige `foreach`-Schleife ließe sich also auch so schreiben:

```
foreach (@prozessoren) {
    print;          # $_ ausgeben
    print "\n";    # der fehlende Zeilenumbruch
}
```

Sie können ein Array auch zu einem Skalar zusammenfassen und umgekehrt. Die Anweisung `split (/Muster/, String)` zerlegt den angegebenen String an den Stellen, an denen das Muster (ein regulärer Ausdruck) vorkommt, und konvertiert ihn in ein Array. `join (Ausdr, Array)` fasst die Elemente des angegebenen Arrays zu einem String zusammen und setzt jeweils den angegebenen Ausdruck dazwischen:

```
$distlist = "openSUSE,Ubuntu,Debian";
@systeme = split (/,/, $distlist);
                    # ergibt Array mit Distributionen
@sprachen = qw(C Java Perl);
$sprachlist = join (" ", @sprachen);
                    # ergibt "C, Java, Perl"
```

Ein *Hash* kann ebenfalls durch die Definition eines einzelnen Elements eingerichtet werden:

```
$wtage{'Mo'} = 'Montag';
```

Auch einzelne Elemente von Hashes sind skalar und beginnen deshalb mit einem `$`. Der Index eines Hash-Elements wird als *Schlüssel* (key) bezeichnet. Alternativ können Sie das ganze Hash auf einmal definieren:

```
%wtage = (Mo => 'Montag',   Di => 'Dienstag',
          Mi => 'Mittwoch', Do => 'Donnerstag',
          Fr => 'Freitag',  Sa => 'Samstag',
          So => 'Sonntag');
```

Die spezielle Schreibweise `Schlüssel => Wert` erlaubt es, die Anführungszeichen um die Schlüsselnamen wegzulassen, solange diese keine problematischen Sonderzeichen enthalten.

Auch über die Elemente eines Hashs lässt sich eine `foreach`-Schleife bilden. Dazu wird dem Namen des Hashs das spezielle Schlüsselwort `keys` vorangestellt. Die oben definierten Wochentage können Sie zum Beispiel so ausgeben:

```
foreach $key (keys %wtage) {
    print "$key ist die Abkürzung für ${wtage{$key}}\n";
}
```

Wie Sie sehen, ermöglichen geschweifte Klammern hinter dem `$`-Zeichen die Auflösung komplexer Variablenkonstrukte innerhalb von Strings.

Ein weiteres interessantes Konstrukt im Zusammenhang mit Variablen ist die *Referenz*. Sie ist im Wesentlichen mit einem Zeiger in C oder einer Objektreferenz in Java vergleichbar: Referenzen verweisen auf beliebige Variablen, Subroutinen oder sonstige Elemente; eine Referenz wird durch einen `\` dargestellt. Beispiele:

```
$v_ref = \ $var;      # Referenz auf ein Skalar
$a_ref = \@array;    # Referenz auf ein Array
$h_ref = \%hash;     # Referenz auf ein Hash
```

Die umgekehrte Operation heißt *Dereferenzierung*. Sie wird in Perl durchgeführt, indem Sie dem Namen der Referenz das Zeichen für den Variablentyp voranstellen, auf den die Referenz verweist:

```
$$v_ref = 3;         # indirekt auf $var zugreifen
@$a_ref.push("hi"); # indirekt auf @array zugreifen
%$h_ref = (name => 'Klaus', alter => 42);
                  # indirekt auf %hash zugreifen
```

```

$a_ref[0] = "hallo"; # indirekt auf ein Element
                  # von @array zugreifen

```

Referenzen sind unter anderem als Übergabewerte für Subroutinen nützlich (siehe unten). Eine weitere Möglichkeit, die sie bieten, ist die Konstruktion verschachtelter Arrays und Hashes, für die es keine andere Lösung gibt:

```

@schmitz = ("Klaus", "Erika", "Michael");
@mueller = ("Heinz", "Gisela", "Claudia");
@familien = (\@schmitz, \@mueller);

```

Um nun beispielsweise auf das erste Mitglied der zweiten Familie zuzugreifen, können Sie `$familien[1][0]` schreiben und erhalten "Heinz".

Ausdrücke und Operatoren

Die einfachsten Bestandteile von Ausdrücken sind die *Literale*. Es handelt sich dabei um Werte, die nicht weiter berechnet oder ersetzt werden müssen. Perl unterscheidet die folgenden drei Arten von Literalen:

► *Integer-Literale*

Sie dienen der Darstellung ganzzahliger Werte. Normalerweise werden sie dezimal notiert; dazu wird einfach die entsprechende Zahl mit optionalem Vorzeichen geschrieben. Wenn Sie einem Integer-Wert eine 0 voranstellen, wird er als Oktalzahl interpretiert: 033 bedeutet demzufolge nicht 33, sondern 27. Ein vorangestelltes 0x kennzeichnet dagegen eine Hexadezimalzahl: 0x33 steht also für den Wert 51.

In Perl können Sie zusätzlich einen Unterstrich als Tausendertrennzeichen einsetzen, um große Zahlen übersichtlicher zu machen:

```

$milliarde = 1_000_000_000;          # Wert: 1000000000

```

► *Fließkomma-Literale*

Diese Literale repräsentieren Fließkommawerte. Beachten Sie, dass in Perl – wie in den meisten anderen Programmiersprachen – kein Komma als Dezimaltrennzeichen verwendet wird, sondern ein Punkt. Alternativ können Sie Fließkomma-Literale in wissenschaftlicher Schreibweise (Exponentialschreibweise) angeben: 3.5E+5 steht beispielsweise für $3,5 \times 10^5$ (350.000); 4.78E-4 hat dagegen den Wert $4,78 \times 10^{-4}$ (0,000478).

► *String-Literale*

String-Literale enthalten Zeichenketten, das heißt beliebig lange Textblöcke. Sie müssen in einfachen oder doppelten Anführungszeichen stehen, etwa 'hallo' oder "hallo".

In doppelten Anführungszeichen wird die bereits angesprochene Variablensubstitution durchgeführt, während Strings in einfachen Anführungszeichen stets wörtlich erhalten bleiben:

```
$geld = 100;
print "Ich habe $geld \$.\\n"; # Wert: Ich habe 100 $.
print 'Ich habe $geld $\\n'; # Wert: Ich habe $geld $\\n
```

In einfachen Ausführungszeichen wird also nicht nur die Variablensubstitution, sondern auch die Ersetzung von Escape-Sequenzen wie `\\n` unterdrückt. Lediglich `\\'` für das einfache Anführungszeichen selbst und `\\\\` für den Backslash werden aufgelöst.

In doppelten Anführungszeichen müssen Sie dagegen eine Reihe spezieller Zeichen als Escape-Sequenz schreiben, wenn Sie diese Zeichen als solche verwenden möchten. Neben den zu erwartenden Fällen `\\` und `\\\\` betrifft dies auch `\\$` (das Dollarzeichen), `\\@` (für ein `@`) und `\\%`.

Neben den beiden Arten von Anführungszeichen definiert Perl noch weitere Schreibweisen für String-Literale. Diese haben den Vorteil, dass Sie die Zeichen `"` oder `'` – je nachdem – einfach als solche verwenden können:

- ▶ `q/String/` ersetzt die einfachen Anführungszeichen, steht also für `'String'`.
- ▶ `qq/String/` wird statt doppelter Anführungszeichen gesetzt, ist also ein Ersatz für `"String"`.
- ▶ `qw/Ausdr1 Ausdr2 Ausdr3/` stellt keinen String dar, sondern ist eine kompakte Schreibweise für eine Liste einzelner Wörter. Die Langschreibweise dieser Liste wäre `('Ausdr1', 'Ausdr2', 'Ausdr3')`.

Das besonders Interessante an diesen speziellen Quoting-Verfahren ist, dass Sie statt des Slashes (`/`) auch verschiedene andere Zeichen verwenden können, um den Inhalt abzugrenzen. Sie können sich individuell für Zeichen entscheiden, die im jeweiligen String selbst nicht vorkommen. Gängige Beispiele sind `q!String!`, `q(String)` oder `q{String}`. Entweder werden also zwei gleiche Zeichen wie `/.../`, `!...!` oder `|...|` verwendet oder aber zwei entgegengesetzte Klammern einer beliebigen Sorte.

Bei Operationen berücksichtigt Perl das Problem, dass die Inhalte von Variablen als unterschiedliche Datentypen interpretiert werden können, und definiert deshalb zwei Sätze von Vergleichsoperatoren. Die »normalen« Vergleichsoperatoren sind `==`, `!=`, `<`, `<=`, `>` und `>=`; sie sind ausschließlich für numerische Vergleiche vorgesehen. Für String-Vergleiche werden die folgenden separaten Operatoren definiert:

In doppelten Anführungszeichen wird die bereits angesprochene Variablensubstitution durchgeführt, während Strings in einfachen Anführungszeichen stets wörtlich erhalten bleiben:

```
$geld = 100;
print "Ich habe $geld \$.\\n"; # Wert: Ich habe 100 $.
print 'Ich habe $geld $\\n'; # Wert: Ich habe $geld $\\n
```

In einfachen Ausführungszeichen wird also nicht nur die Variablensubstitution, sondern auch die Ersetzung von Escape-Sequenzen wie `\\n` unterdrückt. Lediglich `\\'` für das einfache Anführungszeichen selbst und `\\\\` für den Backslash werden aufgelöst.

In doppelten Anführungszeichen müssen Sie dagegen eine Reihe spezieller Zeichen als Escape-Sequenz schreiben, wenn Sie diese Zeichen als solche verwenden möchten. Neben den zu erwartenden Fällen `\\` und `\\\\` betrifft dies auch `\\$` (das Dollarzeichen), `\\@` (für ein `@`) und `\\%`.

Neben den beiden Arten von Anführungszeichen definiert Perl noch weitere Schreibweisen für String-Literale. Diese haben den Vorteil, dass Sie die Zeichen `"` oder `'` – je nachdem – einfach als solche verwenden können:

- ▶ `q/String/` ersetzt die einfachen Anführungszeichen, steht also für `'String'`.
- ▶ `qq/String/` wird statt doppelter Anführungszeichen gesetzt, ist also ein Ersatz für `"String"`.
- ▶ `qw/Ausdr1 Ausdr2 Ausdr3/` stellt keinen String dar, sondern ist eine kompakte Schreibweise für eine Liste einzelner Wörter. Die Langschreibweise dieser Liste wäre `('Ausdr1', 'Ausdr2', 'Ausdr3')`.

Das besonders Interessante an diesen speziellen Quoting-Verfahren ist, dass Sie statt des Slashes (`/`) auch verschiedene andere Zeichen verwenden können, um den Inhalt abzugrenzen. Sie können sich individuell für Zeichen entscheiden, die im jeweiligen String selbst nicht vorkommen. Gängige Beispiele sind `q!String!`, `q(String)` oder `q{String}`. Entweder werden also zwei gleiche Zeichen wie `/.../`, `!...!` oder `|...|` verwendet oder aber zwei entgegengesetzte Klammern einer beliebigen Sorte.

Bei Operationen berücksichtigt Perl das Problem, dass die Inhalte von Variablen als unterschiedliche Datentypen interpretiert werden können, und definiert deshalb zwei Sätze von Vergleichsoperatoren. Die »normalen« Vergleichsoperatoren sind `==`, `!=`, `<`, `<=`, `>` und `>=`; sie sind ausschließlich für numerische Vergleiche vorgesehen. Für String-Vergleiche werden die folgenden separaten Operatoren definiert:

Wenn die Schluss-Markierung in doppelten Anführungszeichen steht, wie in den beiden Beispielen, wird innerhalb des Textes übriges Variablensubstitution durchgeführt.

Eine weitere interessante Besonderheit von Perl besteht darin, dass Sie logische Operatoren nicht nur für Ausdrücke verwenden können, sondern wie in der Shell auch zur logischen Verknüpfung von Anweisungen. Bei der Verknüpfung zweier Anweisungen mittels `||` (oder dem Synonym `or`) wird die zweite Anweisung nur dann ausgeführt, wenn die erste fehlschlägt. Dies liegt an der Short-Circuit-Logik der logischen Operatoren: Wie die meisten Programmiersprachen hört Perl mit der Auswertung eines logischen Operators auf, sobald das Ergebnis feststeht. Bei einer Oder-Verknüpfung steht das Ergebnis »wahr« fest, wenn der erste Ausdruck wahr ist. Dies ist bei einer erfolgreich ausgeführten Anweisung der Fall. Bekannt ist beispielsweise die folgende Formulierung zum Öffnen einer Datei:

```
open (FILE, "<datei.txt") or die "Kann nicht öffnen!\n";
```

Dies lässt sich in der Tat mit »Öffne die Datei oder stirb!« beschreiben – die Anweisung die bricht die Ausführung eines Skripts mit einer Fehlermeldung ab.

Umgekehrt wird die zweite Anweisung bei einer `&&`- beziehungsweise `and`-Verknüpfung nur dann ausgeführt, wenn die erste Erfolg hatte – andernfalls steht das Gesamtergebnis »falsch« bereits fest.

Kontrollstrukturen

Zur Flusskontrolle bietet Perl die übliche Kontrollstrukturen `if()`, `for()` und `while()`; nur `case` existiert nicht. Die zusätzliche Kontrollstruktur `foreach()` wurde bereits weiter oben vorgestellt. Die Syntax der Standard-Kontrollstrukturen entspricht der Programmiersprache C – mit einer Ausnahme: Sie müssen immer geschweifte Klammern setzen, auch wenn nur eine Anweisung von der Kontrollstruktur abhängt.³

Der grundsätzliche Unterschied zur Shell besteht bei Bedingungsprüfungen darin, dass in Perl (und den meisten anderen Sprachen) der Wert 0 als falsch gilt, alle anderen Werte dagegen als wahr.

Ein einfaches `if()` sieht beispielsweise so aus:

```
if ($a == 1) {
    print "Ja! Es ist 1!";
}
```

³ Das sollten gute Programmierer aber sowieso tun ...

Wenn Sie möchten, dass bei Nichtzutreffen der Bedingung etwas anderes geschieht, können Sie einen `else`-Block verwenden:

```
if ($a == 1) {
    print "Ja! Es ist 1!";
} else {
    print "Nein, dies ist keine 1.";
}
```

Für verschachtelte Fallentscheidungen steht zudem die Anweisung `elsif` zur Verfügung. Sie tritt in Kraft, wenn die Bedingung des vorherigen `if` oder `elsif` falsch ist, und überprüft ihrerseits eine weitere Bedingung. Dieses Beispiel zeigt nebenbei, dass einige der für Shell-Skripte gezeigten Dateitest-Operationen auch in Perl existieren:

```
# $entry sei ein String, der einen beliebigen
# Verzeichniseintrag (oder auch nicht) darstellt
if (-f $entry) {
    print "$entry ist eine normale Datei\n";
} elsif (-d $entry) {
    print "$entry ist ein Verzeichnis\n";
} elsif (-e $entry) {
    print "$entry ist ein sonstiger Verzeichniseintrag\n";
} else {
    print "$entry existiert nicht\n";
}
```

Eine Alternative für ein `if`, von dem nur eine Anweisung abhängt und das kein `else` besitzt, ist das nachgestellte `if`:

```
print "\$a ist 0!\n" if !$a;
```

ist ein Synonym für folgende Fallentscheidung:

```
if (!$a) {
    print "\$a ist 0!\n";
}
```

Eine weitere Perl-Spezialität ist die zusätzliche Kontrollstruktur `unless`, die das Gegenteil von `if` darstellt und ebenfalls vor- oder nachgestellt sein kann. Die abhängigen Anweisungen werden genau dann ausgeführt, wenn die Bedingung *nicht* zutrifft. Genau wie `if` kann auch `unless` sowohl blockbildend als auch – wie im nachfolgenden Beispiel – nachgestellt sein:

```
die "\$b existiert nicht!\n" unless defined $b;
```

Die Funktion `defined` prüft übrigens, ob die angegebene Variable existiert.

Subroutinen

Anstelle von Funktionen können Sie in Perl sogenannte *Subroutinen* definieren. Sie werden durch das Schlüsselwort `sub` eingeleitet und durch ihren Namen aufgerufen. Das folgende einfache Beispiel zeigt, wie eine Subroutine definiert und aufgerufen wird:

```
tuwas;                # Subroutine tuwas aufrufen
...

sub tuwas {
    print "Hi! Ich tue etwas: Text ausgeben.\n";
}
```

Es ist vollkommen egal, an welcher Stelle Sie Subroutinen in einem Perl-Skript definieren. Allerdings ist es üblich, sie am Ende des Skripts unter dem globalen Code unterzubringen.

Subroutinen können Argumente entgegennehmen. Sämtliche übergebenen Werte befinden sich in dem speziellen Array `@_`, dessen einzelne Elemente Sie sinnvollerweise mittels `shift` abtrennen sollten, weil Sie sie dadurch in der richtigen Reihenfolge einlesen können. Dazu genügt sogar die Angabe von `shift` ohne Argument, weil diese Funktion in diesem Fall automatisch auf `@_` zugreift. Das folgende Beispiel definiert eine Subroutine namens `print_summe`, die zwei übergebene Werte addiert und ausgibt:

```
sub print_summe {
    $w1 = shift;
    $w2 = shift;
    $summe = $w1 + $w2;
    print "Summe: $summe\n";
}
```

Diese Subroutine können Sie beispielsweise folgendermaßen aufrufen:

```
print_summe (3, 7);
```

Die folgende Subroutine erhöht jedes Element eines als Referenz übergebenen Arrays um 1:

```
sub list_inc {
    $listref = shift;
    foreach (@$listref) {
        $_++;
    }
}
```

Ein Aufruf dieser Subroutine sieht etwa so aus:

```
@list = qw(2 4 6 8 10);
list_inc (\@list);    # @list ist nun (3, 5, 7, 9, 11)
```

Eine Subroutine kann übrigens auch Werte zurückgeben. Dazu wird die Anweisung `return` verwendet. Die folgende Subroutine gibt beispielsweise das Quadrat des ersten Arguments zurück:

```
sub quadrat {
    $val = shift;
    return $val * $val;
}
```

Aufrufen könnten Sie diese Subroutine zum Beispiel in folgendem Kontext:

```
print "Bitte eine Zahl eingeben: ";
$zahl = <>;
chomp $zahl;
$quad = quadrat ($zahl);
print "Ergebnis: $zahl^2 = $quad";
```

Wenn Sie dieses Skript ausführen, könnte sich etwa Folgendes ergeben:

```
Bitte eine Zahl eingeben: 256
Ergebnis: 256^2 = 65536
```

Mit Dateien arbeiten

Perl besitzt eine besonders einfache Syntax für das Lesen und Schreiben von Dateien. Dies ist eine der wichtigsten Eigenschaften der Sprache für Systemadministratoren.

Um eine Datei zu öffnen, wird die folgende Syntax verwendet:

```
open (DATEIHANDLE, "...Pfad");
```

Statt `...` steht vor dem Pfad die Angabe des Zugriffsmodus: `<` öffnet die Datei zum Lesen, `>` zum Schreiben (wobei eine existierende Datei dieses Namens geleert würde) oder `>>` zum Anhängen.

Das `DATEIHANDLE` ist ein Bezeichner, über den Sie auf die Datei zugreifen können. Im Gegensatz zu Variablen besitzt es kein einleitendes Zeichen und wird üblicherweise vollständig in Großbuchstaben geschrieben.

Das folgende Beispiel versucht, die Datei `test.txt` im aktuellen Verzeichnis zum Lesen zu öffnen, liest eine Zeile daraus und gibt diese aus. Anschließend wird die Datei mittels `close` geschlossen:

```

open (FILE, "<test.txt")
    or die "Kann test.txt nicht öffnen!";
$line = <FILE>;
chomp $line;
print "Gelesen: $line\n";
close FILE;

```

Wenn Sie alle Zeilen aus der Datei lesen möchten, bietet sich die folgende Kurzfassung an:

```

while ($line = <FILE>) {
    chomp $line;
    print "Gelesen: $line\n";
}

```

Die Bedingung bleibt dabei automatisch wahr, solange das Dateiende noch nicht erreicht ist.

Wenn Sie eine Datei zum Schreiben (oder Anhängen) öffnen, können Sie folgendermaßen eine Zeile Text hineinschreiben:

```

open (FILE, ">test.txt")
    or die "Kann test.txt nicht öffnen!";
print FILE "Eine Zeile Text.\n";
close FILE;

```

Es ist übrigens *sehr wichtig*, dass zwischen dem Namen des Dateihandles und dem Text, den `print` ausgeben soll, kein Komma, sondern nur ein Leerzeichen stehen darf.

Auf ähnlich einfache Weise können Sie übrigens Verzeichnisinhalte lesen. Das folgende Beispiel gibt die Namen sämtlicher Dateien des aktuellen Verzeichnisses (repräsentiert durch ".") aus:

```

opendir (DIR, ".");
while ($filename = readdir DIR) {
    print "$filename\n";
}
closedir DIR;

```

17.3.3 In Perl mit regulären Ausdrücken arbeiten

Perl bietet traditionellerweise die beste Unterstützung für reguläre Ausdrücke, denn die Sprache wurde vor allem zu diesem Zweck entworfen. Die Syntax für reguläre Ausdrücke, die in Perl eingesetzt wird, geht weit über die Möglichkeiten

von `grep` hinaus. Sie stammen zum großen Teil aus den oben angesprochenen Textverarbeitungssprachen `sed` und `awk`.

In Perl werden vor allem zwei wichtige Operatoren zusammen mit regulären Ausdrücken verwendet:

- ▶ `m/RegExp/` sucht in einem String nach dem Vorkommen von `RegExp`. In vielen Fällen kann das `m` vor dem `/` einfach weggelassen werden.
- ▶ `s/RegExp/String/` ersetzt ein gefundenes Vorkommen von `RegExp` durch den angegebenen String.

Es ist nicht so schwierig, das Funktionieren dieser Operatoren als solches zu verstehen; wichtiger ist es, zu lernen und sich zu merken, welche Formulierungen in regulären Ausdrücken zulässig sind.

Muster für reguläre Ausdrücke

Grundsätzlich existieren die folgenden Komponenten für einzelne Zeichen, aus denen Sie in Perl reguläre Ausdrücke zusammensetzen können:

- ▶ `abc` steht einfach für die genannten Zeichen selbst. `ee` passt auf »Tee« und »See«, aber nicht auf »Tea«.
- ▶ `[abc]` steht für ein beliebiges Zeichen aus der Liste. `[MH]aus` passt also auf »Haus« und »Maus«, aber nicht auf »raus«.
- ▶ `[a-z]` steht für die gesamte angegebene Zeichenfolge im Zeichensatz. Beispielsweise bedeutet `[a-z]`, dass alle kleinen Buchstaben gemeint sind, `[0-9]` steht dagegen für alle Ziffern und so weiter. Natürlich müssen Sie nicht immer eine ganze Zeichenkategorie angeben: `[A-K]` repräsentiert zum Beispiel alle großen Buchstaben von A bis K. Außerdem können Sie innerhalb desselben Paares eckiger Klammern sowohl einen Bereich als auch einzelne Zeichen eingeben. `[a-dp]` trifft etwa auf alle Zeichen von a bis d sowie auf das p zu. Auch mehrere Bereiche sind möglich: Beispielsweise stellt `[0-9a-fA-F]` sämtliche Hexadezimalziffern dar.
- ▶ `[^abc]` steht für jedes beliebige Zeichen außer denjenigen in der Liste. So trifft `[^M]aus` auf »Haus« und »raus«, aber nicht auf »Maus« zu. Übrigens passt dieser Ausdruck auch nicht auf »aus«, weil auf jeden Fall ein Zeichen davor erwartet wird – es darf nur kein M sein.
- ▶ `.` (ein Punkt) steht für genau ein beliebiges Zeichen.
- ▶ `?` bedeutet, dass das links davon stehende Zeichen beziehungsweise die Liste optional ist; sie darf also vorkommen oder auch nicht. `Hau?se` kann also für »Hase« oder für »Hause« stehen, aber nicht für »Hanse«.

- ▶ * bedeutet, dass das links davon stehende Zeichen beziehungsweise die Liste beliebig oft vorkommen darf, also gar nicht, einmal oder öfter. 12^*3 könnte also etwa »13«, »123« oder »12223« heißen. $.^*$ steht übrigens für beliebig viele beliebige Zeichen.
- ▶ + bedeutet, dass das links davon stehende Zeichen beziehungsweise die Liste mindestens einmal vorkommen muss, aber auch öfter vorkommen darf. $0,3^+$ kann also etwa für »0,3«, »0,333« oder »0,333333333« stehen.

Achtung! Die Modifikatoren * und + sind »gierig« (greedy): Wenn sie die Wahl haben, passen sie auf die maximal mögliche Anzahl von Zeichen. Falls Sie etwa mittels $.^*$ nach Text in Anführungszeichen suchen, passt dieser Ausdruck auf sämtlichen Text zwischen dem ersten und dem allerletzten Anführungszeichen im aktuellen Bereich! Deshalb existiert alternativ eine nicht-gierige (non-greedy) Variante: Schreiben Sie $^+?$ beziehungsweise $^*?$, wenn Sie nur jeweils den kleinstmöglichen Treffer erhalten möchten. Für einen einzelnen Textblock in Anführungszeichen müsste der Ausdruck also $\". *?\"$ lauten.

- ▶ $\{n\}$ bedeutet, dass das links davon stehende Zeichen beziehungsweise die Liste genau n -mal vorkommen muss. Beispielsweise ist $[0-9]\{5\}$ das Muster für eine deutsche Postleitzahl: fünf Ziffern zwischen 0 und 9.
- ▶ $\{m,n\}$ bedeutet, dass das links davon stehende Zeichen beziehungsweise die Liste mindestens m -mal und höchstens n -mal vorkommen darf. Ein sinnvolles Muster für Telefonnummern ohne Vorwahl ist zum Beispiel $[0-9]\{4,8\}$.

Damit Sie die bisherigen Informationen leichter verarbeiten können, sehen Sie hier zunächst ein kurzes Beispiel. Das folgende Programm überprüft bei einer eingegebenen Zeile, ob sie das Wort »Perl« enthält:

```
#!/usr/bin/perl -w
print "Geben Sie etwas ein: ";
$zahl = <>;
if ($zahl =~ /Perl/) {
    print "Danke für Perl!\n";
} else {
    print "Da war kein Perl!\n";
}
```

Wenn Sie in regulären Ausdrücken bestimmte Zeichen als solche verwenden möchten, müssen Sie ihnen einen Backslash (\) voranstellen, weil sie eine spezielle Bedeutung haben. Es handelt sich um folgende Zeichen: $- + ? * () [] \{ \} / \ | \ . \ ^ \ \$$. Darüber hinaus müssen Sie in Perl noch % und @ schützen, weil sie ein Hash beziehungsweise Array einleiten.

Neben den bisher besprochenen Formen für einzelne Zeichen existieren etliche interessante Kurzfassungen und spezielle Formulierungen, von denen hier nur die wichtigsten wiedergegeben werden:

- ▶ `\s` steht für beliebigen Whitespace, das heißt für Leerzeichen, Tabulatoren oder Zeilenumbrüche (falls überhaupt mehr als eine Zeile auf einmal verarbeitet wird; siehe unten).
- ▶ `\w` repräsentiert sämtliche Arten von Zeichen, die in Wörtern vorkommen können. Genauer gesagt sind alle Zeichen zulässig, die in Perl-Bezeichnern vorkommen können: Buchstaben, Ziffern und `_` (Unterstrich).
- ▶ `\W` ist das Gegenteil von `\w`: Es trifft auf sämtliche anderen Zeichen zu.
- ▶ `\d` steht für sämtliche Ziffern, ist also eine Abkürzung für `[0-9]`.
- ▶ `\D` stellt dagegen sämtliche Zeichen dar, die keine Ziffern sind; dies ist eine Abkürzung für `[^0-9]`.
- ▶ `\b` trifft auf Wortgrenzen zu. Diese Komponente ist also überall dort zu finden, wo ein Wort beginnt oder aufhört. Zum Beispiel findet `\ber` die Wörter »er« oder »erklären«, aber nicht »lernen«.
- ▶ `\B` betrifft dagegen sämtliche Stellen innerhalb von Wörtern; ein Muster, das Sie mit `\B` kombinieren, wird also nur dann gefunden, wenn es nicht am Anfang beziehungsweise am Ende eines Wortes steht. `\Ber` findet also etwa »lernen«, aber nicht »erklären«.
- ▶ `()` dient dazu, einen Teilausdruck zu einer Gruppe zusammenzufassen. Dies hat zum einen den Zweck, ganze Gruppen mit Zählern wie `+`, `?` und `*` zu versehen oder mittels `|` Optionen anzugeben; zum anderen merkt sich die Perl-RegExp-Engine geklammerte Ausdrücke in den speziellen Variablen `$1` bis `$9`, um sie beim Ersetzen in den Ersatztext mit aufzunehmen.
- ▶ `^` außerhalb eckiger Klammern steht für den Zeilenanfang. `^H` trifft also beispielsweise auf den Text »Hallo Welt!« zu, aber nicht auf den Satz »Sag Hallo zur Welt!«.
- ▶ `$` steht entsprechend für das Zeilenende. Beispielsweise trifft `\d+0$` nur auf Zahlen zu, die mit einer 0 enden.

Der folgende Ausdruck trifft nur zu, wenn die gesamte Zeile aus einer Ganzzahl besteht:

```
/^ \-? \d+ $/
```

`^` markiert den Zeilenanfang, `\-?` steht für ein Minuszeichen oder auch keines, dahinter bezeichnet `\d+` eine oder mehrere Ziffern, und das Zeilenende wird durch `$` gekennzeichnet.

Mithilfe geklammerter Ausdrücke können Sie eine noch größere Flexibilität erreichen. Beispielsweise passt der nächste Ausdruck auf »Hund«, »Katze« oder »Maus«:

```
/(Hund|Katze|Maus)/
```

Ein weiteres interessantes Hilfsmittel bei der Arbeit mit regulären Ausdrücken sind diverse Modifikatoren, die hinter den schließenden Slash geschrieben werden:

- ▶ `/i` verzichtet auf die Unterscheidung zwischen Groß- und Kleinschreibung. `[a-z]+/i` findet also beispielsweise beliebige Kombinationen von Buchstaben in beiden Schreibweisen.
- ▶ `/g` findet mehr als einen Treffer, kann also etwa in einer Schleife abgearbeitet werden oder bei Ersetzungen dafür sorgen, dass nicht nur das erste Vorkommen ersetzt wird, sondern alle.
- ▶ `/e` erlaubt die Verwendung eines auszuwertenden Ausdrucks als Ersatztext. Normalerweise ist nur ein einfacher String gestattet. Bei aller Perl-Kompatibilität der RegExp-Implementierungen anderer Sprachen und Programme funktioniert diese eine Option ausschließlich in Perl.

Programmierung mit regulären Ausdrücken

Nachdem Sie nun die einzelnen Bestandteile regulärer Ausdrücke kennengelernt haben, müssen Sie als Nächstes die verschiedenen Perl-Funktionen erlernen, in denen reguläre Ausdrücke verwendet werden können.

Die einfachste dieser Funktionen ist der Matching-Operator `/`, der überprüft, ob ein angegebener regulärer Ausdruck in einem String vorkommt oder nicht. Er wird in der Regel zusammen mit einem der beiden Operatoren `==` oder `!~` verwendet. Der Wert des Operators `==` ist wahr, wenn der reguläre Ausdruck auf der rechten Seite zu dem Ausdruck auf der linken Seite passt. Das Gegenteil ist bei `!~` der Fall: Dieser Operator liefert den Wert wahr, wenn der reguläre Ausdruck auf der rechten Seite nicht zum untersuchten Ausdruck links passt.

Beispielsweise können Sie folgendermaßen überprüfen, ob in der Variablen `$test` Ziffern vorkommen:

```
print "Ziffern gefunden!" if $test == /\d/;
```

Mithilfe der folgenden Anweisung lässt sich dagegen testen, ob `$test` keine Buchstaben enthält:

```
print "Buchstabenfrei!" if $test !~ /[a-z]/i;
```

Wie oben erwähnt, wird das `m` vor dem Ausdruck in der Regel weggelassen. Es ist nur dann wichtig, wenn Sie statt des `/` ein anderes Zeichen verwenden, was besonders nützlich ist, wenn der `/` selbst im regulären Ausdruck vorkommt. Sie können sämtliche Zeichen benutzen, die auch für Quoting-Operatoren wie `q//` oder `qw//` zulässig sind, in einem solchen Fall ist das vorangestellte `m` aber zwingend erforderlich. Beispielsweise können Sie folgendermaßen überprüfen, ob es sich beim Inhalt der Variablen `$pfad` um einen absoluten UNIX-Pfad handelt:

```
print "UNIX-Pfad!" if $pfad =~ m|^(/[^\s/]+)+|;
```

In der normalen Schreibweise wäre es erheblich komplizierter, diese Anweisung zu schreiben, und obendrein wäre sie viel schwerer lesbar:

```
print "UNIX-Pfad!" if $pfad =~ /^[^\s/]+|/;
```

Der Ausdruck bedeutet in beiden Fällen: Zu Beginn des Ausdrucks (^) steht ein Slash (/), gefolgt von einem oder mehreren Zeichen außer dem Slash ([^\s/]+). Diese Abfolge wiederholt sich bei Bedarf mehrere Male (+ außerhalb der Klammern). Im zweiten Fall wirkt es sehr störend, dass der Slash selbst nicht einfach verwendet werden kann, sondern durch einen vorangestellten Backslash zum normalen Zeichen erklärt werden muss.

Eine weitere Anwendungsmöglichkeit regulärer Ausdrücke ist das Ersetzen der jeweiligen Fundstellen durch einen Ersatztext. Dies geschieht in Perl mithilfe des Operators `s///`. Beispielsweise ersetzt die folgende Anweisung jedes Vorkommen von »Düsseldorf« oder »Leverkusen« in der Variablen `$text` durch »Köln«:

```
$text =~ s/(Düsseldorf|Leverkusen)/Köln/g;
```

Der bereits erwähnte Modifikator `g` sorgt dafür, dass jedes Vorkommen des gesuchten regulären Ausdrucks ersetzt wird.

Durch die Verwendung von Klammern im regulären Ausdruck können Sie außerdem die Speicherung einzelner Teile des gefundenen Textes veranlassen, diese Teile können Sie innerhalb des Ersatztextes verwenden. Beispielsweise könnten Sie eine Angabe von Postleitzahl und Ort nach dem Schema »53227 Bonn« in die Schreibweise »Postleitzahl: 53227, Ort: Bonn« umwandeln:

```
$ortsangabe =
    s/((\d{5})\s+([a-z]+)/Postleitzahl: $1, Ort: $2/i;
```

Auch zum Extrahieren bestimmter Informationen aus einem Gesamttext sind die Platzhalter `$1` bis `$9` hervorragend geeignet. Zum Beispiel können Sie folgendermaßen den Inhalt der Textdatei `info.txt` nach Angaben durchforsten, bei denen es sich sehr wahrscheinlich um E-Mail-Adressen handelt:

```

open (FH, "<info.txt") || die "Kann nicht öffnen!\n";
while ($line = <FH>) {
    chomp $line;
    while ($line =~ /([a-z0-9\.\-_]+\@[a-z0-9\.\-_]+)/) {
        push (@mails, $1);
    }
}
close FH;
print "Gefundene Mail-Adressen:\n";
foreach $addr(@mails) {
    print "$addr\n";
}

```

Der reguläre Ausdruck, der hier für E-Mail-Adressen verwendet wird, basiert auf einer Wahrscheinlichkeitsannahme und ist nicht absolut genau. Gefunden wird eine Sequenz aus Buchstaben, Ziffern, Bindestrichen und Unterstrichen, ein @-Zeichen und eine weitere Sequenz aus diesen Zeichen. Einige Administratoren erlauben in E-Mail-Benutzernamen noch andere Zeichen, solche nicht standardkonformen E-Mail-Adressen bleiben unentdeckt. Außerdem würde beispielsweise ein Punkt hinter einer E-Mail-Adresse, die am Ende eines Satzes steht, mit aufgenommen.

Eine ähnliche Funktion wie `s///` erledigt übrigens der Operator `tr///`, der Translations-Operator: Er ersetzt jedes Vorkommen des Strings im ersten Teil durch den String im zweiten Teil. Insofern hat er allerdings nichts mit regulären Ausdrücken zu tun.

17.3.4 Administrationsbeispiel: Webserver-Logdateien auswerten

Hier wird ein sehr nützliches Perl-Administrations-Skript vorgestellt; es stammt aus meinem Buch »Apache 2«. Nebenbei lernen Sie ein weiteres Beispiel für die oben diskutierten Logdateien kennen. Es handelt sich um ein Skript, das eine einfache Statistik aus einer Webserver-Logdatei generiert. Wenn Sie einen Account bei einem Hoster oder gar einen eigenen Webserver betreiben, haben Sie solche Access-Logdateien wahrscheinlich schon einmal gesehen. Möglicherweise stellt Ihr Hoster auch eine grafisch aufbereitete Statistik zur Verfügung.

Traditionellerweise führt ein Webserver wie Apache (siehe Kapitel 15, »LAMP«) mindestens zwei Logdateien: die Fehler-Logdatei (`ErrorLog`), in der sämtliche unerwünschten Vorkommnisse dokumentiert werden, und die Zugriffs-Logdatei (`TransferLog` oder auch `AccessLog`), die alle Client-Zugriffe protokolliert. An dieser Stelle soll Letztere untersucht werden.

Die TransferLog-Datei besitzt üblicherweise entweder das *Common Log Format* (CLF) oder das erweiterte Log-Format des NCSA HTTPd (*Combined Log Format*). Ein Eintrag im Common Log Format sieht beispielsweise so aus:

```
163.15.41.28 - - [29/Sep/2009:15:35:07 +0200] "GET / HTTP/1.1" 200
2263
```

Die einzelnen Werte in einer Zeile haben (getrennt durch Leerzeichen) die folgenden Bedeutungen:

- ▶ IP-Adresse des entfernten Hosts
- ▶ RFC 1413-Identität (hier ein -, da keine Identitätsprüfung stattfand)
- ▶ Benutzername (ebenfalls nur ein -, weil auch keine Authentifizierung durchgeführt wurde)
- ▶ Datum und Uhrzeit im weiter oben hergestellten komplexen Format; komplett in eckigen Klammern
- ▶ die komplette Client-Anfrage in Anführungszeichen
- ▶ der Statuscode der Server-Antwort
- ▶ die Größe des Antwort-Body

Das Combined Log Format enthält noch zwei zusätzliche Felder: den Referer (die URL der zuvor besuchten Seite) sowie den User-Agent (den Namen des Client-Programms). Ein typischer Eintrag hat diese Form:

```
163.15.41.28 - - [29/Sep/2009:15:36:24 +0200] "GET /seite2.html
HTTP/1.1" 200
2326 "http://www.test.local/" "Mozilla/4.0 (compatible; MSIE 6.0;
Windows NT 4.0; .NET CLR 1.0.3705)"
```

Vor dem eigentlichen Skript werden hier noch die beiden regulären Ausdrücke vorgestellt, die das Common beziehungsweise das Combined Log Format erkennen. Für das CLF gilt das folgende Ungetüm:

```
^([\s]+)\s+([\s]+)\s+([\s]+)\s+\[[^\]]+\]\s+"([\s"]+)"\s+
(\d+)\s+(\[\d\-\]+\)$
```

Die Teilausdrücke in den runden Klammern bilden die einzelnen Komponenten, die tatsächlich gefunden werden sollen. Viele von ihnen werden einfach durch `[\s]+` – ein oder mehrere Nicht-Leerzeichen – beschrieben, da sie jeweils durch Leerzeichen voneinander getrennt werden. Hinzu kommen die Sonderzeichen, die manche Bestandteile umschließen: Die eckigen Klammern um das Datum sowie die Anführungszeichen um die Anfrage (genau wie Referer und User-Agent beim Combined Log Format) stehen außerhalb der runden Klammern, werden

also nicht mitgespeichert. Vorsichtshalber werden die Felder jeweils durch »ein oder mehrere Leerzeichen« (\s+) getrennt. Wichtig ist das Dollarzeichen als Abschlussmarkierung, da das RegExp ansonsten auch auf das Combined Log Format passen würde. Dessen regulärer Ausdruck sieht entsprechend so aus:

```
^([\s]+)\s+([\s]+)\s+([\s]+)\s+\[([^\]]+)\]\s+"([\s]+)"\s+
([\d+)]\s+([\d\-\+)]\s+"([\s]+)"\s+"([\s]+)"
```

Das hier vorgestellte Perl-Skript beschränkt sich beispielhaft auf zwei typische Werte von Web-Statistiken: die treuesten Besucher(-Adressen) und die am häufigsten besuchten URLs, jeweils mit ihrer Anzahl. Als Pflichtparameter müssen Sie wieder den Namen einer Logdatei angeben; optional können Sie zusätzlich die gewünschte Höchstzahl der Ergebnisse setzen (ansonsten wird 20 verwendet). Ein typischer Aufruf dieses Skripts mit seinem Ergebnis sieht so aus:

```
$ ./log_stat.pl access_log 10
```

Die (bis zu) 10 haeufigsten Besucher:

```
1. 66.246.218.107: 696
2. 213.160.11.114: 683
3. 66.249.65.67: 386
4. 65.54.188.71: 379
5. 81.173.174.49: 306
6. 213.196.248.98: 271
7. 66.249.65.133: 270
8. 213.196.241.196: 237
9. 66.249.65.79: 157
10. 87.78.46.0: 151
```

Die (bis zu) 10 haeufigsten erfolgreichen Anfragen:

```
1. /main.css: 1791
2. /pixel.gif: 1771
3. /galileo.gif: 1593
4. /apache2/c_168_240.jpg: 1158
5. /apache2/noepatents.jpg: 1090
6. /it_komp/c_180_240.jpg: 409
7. /it_komp/noepatents.jpg: 345
8. /: 261
9. /fachinfo/c_190_240.jpg: 184
10. /apache2/mod_ssl.html: 170
```

Im Folgenden sehen Sie den Quellcode des Beispiels. Die Funktionsweise dürfte durch die relativ ausführlichen Kommentare einigermaßen klar werden.

```

#!/usr/bin/perl -w
use strict;

# *****
# * Skript: log_stat.pl *
# * Einfache Statistik für Web-Logdateien *
# *****

# Namen der Logdatei von der Kommandozeile lesen
my $logfile = $ARGV[0] || die "Verwendung: $0 Logdatei [Max]\n";
# Gewünschte Höchst-Zeilenzahl lesen oder auf 20 setzen
my $maxlines = $ARGV[1] || 20;

# Versuchen, die Logdatei zu öffnen
open (LOG, "<$logfile") || die "Kann $logfile nicht öffnen\n";

# Arrays für die Logdatei-Bestandteile
my (@hosts, @ids, @uids, @dates, @requests, @status, @sizes,
    @referers, @useragents);

# Zeilenweise einlesen und auswerten
while (my $line = <LOG>) {
    chomp $line;
    if ($line =~ /^(([\s]+\s+([\s]+\s+([\s]+\s+
        \[[^\]]+\])\s+"([^\"]+)"\s+(\d+)\s+(\[\d\-\]+\s)/) {
        push @hosts, $1;
        push @ids, $2;
        push @uids, $3;
        push @dates, $4;
        push @requests, $5;
        push @status, $6;
        push @sizes, $7;
        push @referers, "-";
        push @useragents, "-";
    } elsif ($line =~ /^(([\s]+\s+([\s]+\s+([\s]+\s+
        \[[^\]]+\])\s+"([^\"]+)"\s+(\d+)\s+(\[\d\-\]+\s+
        "([^\"]+)"\s+"([^\"]+)"/) {
        push @hosts, $1;
        push @ids, $2;
        push @uids, $3;
        push @dates, $4;
        push @requests, $5;
        push @status, $6;
        push @sizes, $7;
    }
}

```

```

        push @referers, $8;
        push @useragents, $9;
    }
}

# Anzahlen unterschiedlicher Hosts ermitteln
my %sorthosts;
foreach my $host(@hosts) {
    # Im Hash %sorthosts ist der Host der Schlüssel;
    # der Wert dient als Zähler und wird um 1 erhöht
    $sorthosts{$host}++;
}
print "Die (bis zu) $maxlines haeufigsten Besucher:\n\n";
my $c = 1;
foreach my $host
    (sort {$sorthosts{$b} <=> $sorthosts{$a}} (keys %sorthosts)) {
    print "$c. $host: ${sorthosts{$host}}\n";
    $c++;
    last if $c > $maxlines;
}
print "\n";

# Anzahlen unterschiedlicher erfolgreich angeforderter URLs
my %sorturls;
my $i = 0;
foreach my $request(@requests) {
    # URL aus dem Request ermitteln
    my $url;
    if ($request =~ /[A-Z]+\s+([\^\s]+)/) {
        $url = $1;
    }
    # War diese Anfrage erfolgreich?
    if ($status[$i] == 200) {
        # URL zählen
        $sorturls{$url}++;
    }
    $i++;
}
print "Die (bis zu) $maxlines haeufigsten erfolgreichen Anfragen:\n\n";
$c = 1;
foreach my $url
    (sort {$sorturls{$b} <=> $sorturls{$a}} (keys %sorturls)) {
    print "$c. $url: ${sorturls{$url}}\n";
}

```

```

$c++;
last if $c > $maxlines;
}

```

Ein wenig kompliziert ist in diesem Beispiel das Verfahren zur sortierten Ausgabe der Hashes – diese sollen nämlich nicht nach ihren Schlüsseln (den IP-Adressen beziehungsweise URL-Pfaden), sondern nach den Werten sortiert werden. Die Perl-Funktion `sort()` erlaubt für solche Zwecke die Angabe einer Sortiervorschrift in Form einer eingebetteten Funktion. Die stilisierten Variablen `$a` und `$b` geben die Reihenfolge an. Für numerische Vergleiche wird der spezielle Operator `<=>` verwendet, während die String-Sortierung durch `cmp` geregelt wird. Beispielsweise werden die URL-Pfade wie folgt absteigend nach ihrer Anzahl sortiert:

```
sort {$sorturls{$b} <=> $sorturls{$a}} (keys %sorturls)
```

17.4 Zusammenfassung

Dieses Kapitel ist gewissermaßen die Quintessenz aus der UNIX-Philosophie: Aus den zahllosen kleinen, spezialisierten und dennoch gut zusammenpassenden Bausteinen lassen sich die imposantesten Türme errichten.

Im ersten Abschnitt wurden zunächst einige neue Bausteine vorgestellt: Aliasse als »kleine Geschwister« der Shell-Skripte, mit denen Sie manche einfachen Aufgaben effizient lösen können; Cronjobs für wiederkehrende, automatisierte Aufgaben; Logdateien zur Protokollierung sowie umgekehrt zur Fehlersuche; `sudo`, um einzelne Kommandos als `root` auszuführen.

Dass die Shell mehr ist als ein linearer Befehlsempfänger, haben Sie wahrscheinlich schon in Kapitel 4 bemerkt. Gerade die `bash` ist zusätzlich eine voll ausgestattete Programmiersprache, was im zweiten Abschnitt dieses Kapitels genauer untersucht wurde. Sobald Sie die kleinen Eigenwilligkeiten der Syntax verstanden haben, werden Sie mit Shell-Skripten hervorragende Ergebnisse erzielen. Im Einzelnen haben Sie hier vor allem Variablen, Kontrollstrukturen und Funktionen kennengelernt.

Oder lesen Sie den letzten Abschnitt dieses Kapitels und lernen Sie Perl. Die Chance ist groß, dass Sie Shell-Skripte, `sed` und `awk` dann nur noch in Ausnahmefällen nutzen werden – diese Sprache ist einfach noch erheblich leistungsfähiger als die drei anderen zusammen. Zudem kommt die Syntax den Gewohnheiten von Programmierern etwas näher als diejenige der Shell. Als praktisches Beispiel wurde zum Schluss ein längeres Skript präsentiert, mit dem Sie Webserver-Logdateien auswerten können – eine Alltagsaufgabe für jeden Administrator.

»Schluckst du die blaue Kapsel, ist alles aus. Du wachst in deinem Bett auf und glaubst an das, was du glauben willst. Schluckst du die rote Kapsel, bleibst du im Wunderland, und ich führe dich in die tiefsten Tiefen des Kaninchenbaus!«
– Morpheus zu Neo in »The Matrix«

18 Hinter den Kulissen

In diesem Kapitel werden einige Aspekte der Funktionsweise des Linux-Kernels vorgestellt, was Ihnen das Verständnis verschiedener in diesem Buch besprochener Aspekte erleichtern wird. Das Thema ist allerdings zu umfangreich, um es innerhalb eines einzelnen Kapitels umfassend darzustellen; falls Sie also wirklich in die »tiefsten Tiefen des Kaninchenbaus« vordringen möchten, brauchen Sie ein spezielles Buch über dieses Thema. Empfehlenswert sind unter anderem die beiden folgenden Werke, die ich auch zur Recherche für dieses Buch verwendet habe:

- ▶ »Understanding the Linux Kernel«, 3. Auflage, von Daniel P. Bovet und Marco Cesati, Sebastopol 2006, O'Reilly Media (Englisch)
- ▶ »Linux Kernelarchitektur« von Wolfgang Mauerer, München 2004, Carl Hanser Verlag (Deutsch)

Wie fast alle Betriebssystemkernel wurde auch der Linux-Kernel zum größten Teil in der Programmiersprache C geschrieben, bis auf einige besonders hardwarenahe Routinen, die im Assembler der jeweiligen Prozessorarchitektur verfasst wurden. Wenngleich fortgeschrittene C-Kenntnisse Ihnen helfen, dieses Kapitel zu verstehen, werden hier eher allgemeine Konzepte erläutert.

18.1 Die zugrundeliegende Hardware

Bevor Sie sich mit der technischen Funktionsweise eines Betriebssystems auseinandersetzen können, ist es wichtig, die grundlegenden Arbeitsvorgänge der Hardware zu verstehen. Insbesondere die Komponenten der Zentraleinheit (als Bauteil: das Mainboard) des Rechners spielen für die Funktion des Betriebssys-

tems eine wichtige Rolle. Im Einzelnen sind vor allem folgende Komponenten zu unterscheiden:

- ▶ Der *Mikroprozessor* (Central Processing Unit oder CPU) ist das eigentliche Herzstück des Computers, das für die Ausführung der Programme sowie für die zentrale Steuerung und Verwaltung der Hardware zuständig ist. Die meisten Desktop-PCs besitzen nur einen Mikroprozessor, maximal sind es in diesem Bereich zwei. Highend-Workstations, beispielsweise für professionelle 3-D-Grafik, sowie leistungsfähige Serverrechner haben dagegen oft zwei, vier, acht oder noch mehr parallel arbeitende Prozessoren. Allerdings besitzen moderne Desktop-Prozessoren mehrere Kerne (meist zwei, manchmal auch vier), sodass sie sich wie mehrere Prozessoren verhalten. Wichtig ist in allen diesen Fällen, dass das verwendete Betriebssystem und die Anwendungsprogramme die Verteilung der Arbeit auf mehrere Prozessoren (Symmetric Multi Processing oder kurz SMP) überhaupt unterstützen – Linux tut dies in besonderem Maße seit der Kernel-Version 2.4.
- ▶ Der *Arbeitsspeicher* (das Random Access Memory, kurz RAM) enthält während der Laufzeit die Programme, die gerade ausgeführt werden, sowie die von ihnen verwendeten Daten. Linux unterstützt wie die meisten modernen Betriebssysteme die virtuelle Speicheradressierung, die die von Programmen verwendeten Speicheradressen von den physikalischen Adressen abstrahiert und auf diese Weise jedem Prozess vorgaukeln kann, er besäße das gesamte RAM für sich allein; außerdem wird das Auslagern nicht benötigter Inhalte auf die Festplatte ermöglicht.
- ▶ Der *ROM-Speicher* (für Read-Only Memory, also Nur-Lese-Speicher) ist bei modernen PCs weniger wichtig als früher. Statt des gesamten Betriebssystems enthält es heutzutage fast nur noch ein Programm, das beim Einschalten die wichtigsten Hardwarekomponenten überprüft und dann das Booten des Betriebssystems von einem Datenträger in Gang setzt. Dieses Programm wird bei Intel-PCs *BIOS* genannt. Neben der genannten Selbsttest- und Start-Funktionalität enthält es auch Routinen für den Hardwarezugriff. Allerdings greifen moderne Betriebssysteme nicht mehr über das BIOS auf die Hardware zu; alle verwenden ihre eigenen, effizienteren Routinen dafür.

Fest ins ROM eingebaute Betriebssysteme oder Anwendungsprogramme besitzen heute nur noch Spezialcomputer: kleine, leicht konfigurierbare Router-Boxen, Industrie-PCs, die aufgrund der unwirtlichen Umgebung ohne mechanische Teile auskommen müssen, oder auch die weitverbreiteten Embedded Systems, also die eingebauten Computer in Automotoren, Spülmaschinen oder Produktionsanlagen. Für solche Einsatzgebiete gibt es die spezielle Linux-Variante Embedded Linux.

- ▶ Der *Chipsatz* (englisch »chipset«) ist in der Regel fest auf dem Mainboard verlötet. Es handelt sich um eine Gruppe von Schaltkreisen, die spezielle Steuerungsaufgaben übernehmen. Sie enthalten vor allem die Steuerfunktionen für sämtliche Anschlüsse, die das Mainboard zu bieten hat. Die Qualität der unterschiedlichen Chipsätze hat einen erheblichen Einfluss auf die Performance eines Rechners.
- ▶ Die verschiedenen *Busse* und *Schnittstellen* dienen zum einen der Kommunikation zwischen den Bestandteilen des Mainboards, zum anderen dem Anschluss aller Arten von Peripheriekomponenten, angefangen bei den diversen Steckkarten wie Sound-, Grafik- oder Netzwerkkarten über verschiedene Arten von Laufwerken bis hin zu Druckern, Scannern oder Digitalkameras.

18.1.1 Aufbau und Aufgaben des Prozessors

Mikroprozessoren sind komplexe integrierte Schaltkreise. Die ersten bestanden aus einigen Tausend Transistoren; heutige Modelle setzen sich sogar aus mehreren Millionen Transistoren zusammen, aber ihre Größe hat sich kaum verändert: Wegen des Fortschritts der fotolithografischen Verfahren, mit denen die Schaltungen auf die Siliziumscheiben aufgebracht werden, wird die Integrationsdichte immer höher.

Der Mikroprozessor wird auf die Hauptplatine aufgesteckt. Die verschiedenen Bauformen und Prozessorsockel haben sich im Lauf der Jahre stark verändert. Selbst innerhalb der Welt der Intel-kompatiblen PCs passt bei Weitem nicht jeder Prozessor auf jedes Mainboard – seit der zunehmenden Aufteilung des Marktes in 32- und 64-Bit-Varianten erst recht nicht mehr.

Technischer Überblick

Schematisch gesehen besitzt ein Mikroprozessor die folgenden Bestandteile:

- ▶ Die *ALU* (Arithmetic-Logical Unit oder zu Deutsch Arithmetisch-logische Einheit) ist das Rechenwerk des Prozessors (der Begriff »Rechenwerk« stammt aus der schematischen Computerarchitektur, die *John von Neumann* in den 1940er-Jahren erdachte). Die ALU ist für mathematische Operationen und logische Verknüpfungen zuständig. Moderne Prozessoren besitzen in der Regel getrennte ALUs oder ALU-Teile für ganzzahlige Operationen und für Fließkommaoperationen. Bei früheren Prozessoren mussten die Fließkommaoperationen durch komplexe Ganzzahlberechnungen simuliert werden; später verwendete man externe Fließkommaeinheiten, die als arithmetische Koprozessoren bezeichnet wurden.

- ▶ Die *Register* sind einzelne Speicherstellen innerhalb des Prozessorkerns; einige von ihnen haben spezifische Aufgaben. Die ALU kann am schnellsten mit Werten rechnen, die sich in den Registern befinden. Register sind definitiv *kein* Arbeitsspeicher; ein typischer Prozessor besitzt nur relativ wenige von ihnen (zum Beispiel 32 Stück) und verwendet sie nicht zur längerfristigen Ablage von Informationen.
- ▶ Das *Steuerwerk* übernimmt die Kontrolle über die Ausführung des Programmcodes und initiiert andere Steuerungsfunktionen. Der Befehlszeiger, ein spezielles Register, verweist auf die Speicheradresse, aus der der nächste Programmbefehl gelesen wird. Bei einem Sprung im Programm muss der Befehlszeiger auf die richtige neue Adresse gesetzt werden. Ein weiteres Register, das vom Steuerwerk verwaltet wird, ist der weiter unten erläuterte Stack-Zeiger.

Die heutigen Steuerwerke sind recht komplizierte Bauteile, weil bereits auf Prozessorebene Unterstützung für die abwechselnde, gleichzeitige Ausführung mehrerer Prozesse (Multitasking) eingebaut ist. Aus diesem Grund genügt es nicht, dass das Steuerwerk sich um den Gang eines linear ablaufenden, höchstens durch Sprünge verzweigenden Programms kümmert, sondern es muss in Zusammenarbeit mit Mechanismen des Betriebssystems dafür sorgen, dass ein sauberer Wechsel zwischen den verschiedenen Prozessen stattfindet. Das Konzept wird weiter unten genauer erläutert.

- ▶ Die *Befehlstabelle* (Instruction Table) ermöglicht die Decodierung der verschiedenen Maschinenbefehle in einem Computerprogramm: Jeder Befehl, der aus dem laufenden Programm gelesen wird, besitzt einen bestimmten numerischen Wert. Je nach Befehlsnummer werden unterschiedliche Schaltungen aktiviert, die für ein bestimmtes Verhalten des Prozessors sorgen.
- ▶ Über verschiedene *Busse* (Datenleitungen) ist der Prozessor mit der Außenwelt, also mit den anderen Komponenten des Mainboards, verbunden: Der *Datenbus* dient dem Austausch von Dateninhalten mit dem Arbeitsspeicher, der *Adressbus* überträgt die zugehörigen Speicheradressen, und der *Steuerbus* kümmert sich um die Ansteuerung der Peripherie-Anschlüsse.

In heutigen Prozessoren sind darüber hinaus die sogenannten *Cache*-Speicher untergebracht. Es handelt sich um kleine, aber sehr schnelle Zwischenspeicher, in denen Befehle oder Daten abgelegt werden können, die bald wieder benötigt werden. Die Verwendung von Cache-Speichern macht den Bau von PCs erheblich wirtschaftlicher: Es wäre schlicht zu teuer, den gesamten Hauptspeicher aus den schnellen Bausteinen aufzubauen, aus denen der Cache besteht, und würde darüber hinaus zu viel Strom verbrauchen. Deshalb wird eine mehrstufige Speicherarchitektur verwendet:

- ▶ Der Level-1-Cache ist unmittelbar im Prozessorkern selbst untergebracht und wird mit derselben Taktrate (siehe unten) betrieben wie der Prozessor selbst. Er ist sehr klein, beim Intel Core 2 ist er beispielsweise nur 64 KByte (32 KByte für Befehle, 32 KByte für Daten) groß, beim AMD Athlon dagegen 128 KByte (je 64 KByte für Befehle und Daten). Den L1-Cache verwendet der Prozessor vornehmlich bei der Ausführung sehr kurzer Schleifen aus wenigen Befehlen.
- ▶ Der Level-2-Cache war bei älteren Prozessoren außerhalb des eigentlichen Prozessorgehäuses auf dem Mainboard untergebracht. Inzwischen befindet er sich in der Regel im Prozessorgehäuse, gehört aber formal nicht zum Prozessorkern. Er wird mit einem Vielfachen der Geschwindigkeit des normalen Arbeitsspeichers betrieben, aber mit einem Bruchteil der Prozessorgeschwindigkeit. Dafür ist er erheblich größer als der L1-Cache; bei aktuellen Intel- und AMD-Produktlinien beispielsweise 2 bis 12 MByte.
- ▶ Der eigentliche Arbeitsspeicher (RAM) dient der normalen Speicherung der Programme, die zurzeit ausgeführt werden, und der von ihnen verarbeiteten Daten. Die RAM-Bausteine, aus denen der Arbeitsspeicher zusammengesetzt ist, arbeiten bei älteren PCs mit derselben Taktrate wie das Mainboard selbst, bei sehr modernen Computern dagegen mit dem 2-, 4- oder sogar 8-fachen davon.
- ▶ Falls auch der normale Arbeitsspeicher nicht mehr ausreicht, um alle Daten der aktuell geladenen Programme aufzunehmen, werden Inhalte aus dem RAM, die nicht besonders dringend benötigt werden, auf die Festplatte ausgelagert. Dieser Vorgang wird je nach Prozessorarchitektur und Betriebssystem als Swapping oder Paging bezeichnet; Linux verwendet die in Kapitel 2 und 9 beschriebenen Swap-Partitionen dafür. Moderne Prozessoren unterstützen eine solche virtuelle Speicherverwaltung durch eine eingebaute Komponente namens Memory Management Unit (MMU).

Programmierer haben übrigens so gut wie keinen Einfluss darauf, welche Daten der Prozessor in einem der Cache-Speicher ablegt. Eine wichtige Entscheidungsgrundlage ist dagegen eine Einheit in der CPU, die als Sprungvorhersage (Branch Prediction) bezeichnet wird: Während der Ausführung von Programmen berechnet der Prozessor, wohin der nächste Sprung im Programm am wahrscheinlichsten führen wird. Aufgrund dieser Daten kann der Prozessor jeweils entscheiden, ob es sich lohnt, bestimmte Programmteile oder Daten im Cache abzulegen oder nicht.

Leistungsmerkmale von Prozessoren

Dass Prozessoren im Lauf der Jahre immer leistungsfähiger wurden, dürfte allgemein bekannt sein und ist für technische Geräte fast selbstverständlich: Moores

Gesetz von 1965 besagt, dass sich die Integrationsdichte und damit die Leistungsfähigkeit integrierter Schaltkreise etwa alle zwei Jahre verdoppelt; die reale Entwicklung hat dies bisher sogar leicht übertroffen. Es ist allerdings wichtig, die verschiedenen Leistungsmerkmale zu kennen und zuordnen zu können.

Die wichtigste Information über die Leistungsfähigkeit eines Mikroprozessors ist seine *Wortbreite*. Dieser Wert gibt an, aus wie vielen Bits ein Maschinenwort dieses Prozessors besteht. Je breiter ein solches Maschinenwort ist, desto mehr unterschiedliche Zustände oder Werte kann der Prozessor im gleichen Durchgang bearbeiten. Verschiedene Komponenten eines Prozessors können unterschiedliche Wortbreiten aufweisen (auch wenn dies seit Langem unüblich geworden ist):

- ▶ Die Wortbreite der *Register* betrifft die Rechenfähigkeiten der ALU, nämlich die mögliche Größe von Ganzzahlen und die Genauigkeit von Fließkommawerten.
- ▶ Die Breite des *Datenbusses* bestimmt, wie viele Bits gleichzeitig aus dem Arbeitsspeicher gelesen oder in ihn geschrieben werden können. Da dieser Wert also den Datenaustausch mit Programmen betrifft, ist er für Programmierer relevant und wird deshalb als Wertangabe für die Datenbreite des Prozessors selbst verwendet.
- ▶ Die Breite des *Adressbusses* regelt die maximale Größe von Speicheradressen und bestimmt deshalb, wie viel Arbeitsspeicher ein Prozessor überhaupt adressieren kann.
- ▶ Die Breite des *Steuerbusses* ist schließlich relevant dafür, mit welchen Arten von Peripherieanschlüssen ein Prozessor umgehen kann. Erst die Einführung der 32-Bit-Prozessoren ermöglichte deshalb die Entwicklung leistungsfähiger Peripherieschnittstellen wie PCI und AGP.

In Tabelle 18.1 finden Sie eine Übersicht über die verschiedenen Prozessorgenerationen. Das angegebene Entwicklungsjahr bezeichnet jeweils die Entstehung des ersten verfügbaren Prozessors mit der entsprechenden Wortbreite.¹

Neben der Wortbreite gibt es noch verschiedene andere Kriterien, die die Geschwindigkeit von Prozessoren bestimmen. Die bekannteste Angabe ist die vom Intel- und AMD-Marketing gern zum Hauptgesichtspunkt erklärte *Taktfrequenz* (Clock Rate). Die Taktfrequenz wird nicht vom Prozessor selbst bestimmt, sondern ist ein Vielfaches des Mainboard-Taktes (*Front Side Bus Clock Rate*).

¹ Der Intel 80386 SX besaß zwar 32 Bit breite Register, aber nur einen 16-Bit-Datenbus. Er konnte also schneller rechnen, aber nach außen nicht anders programmiert werden als ein 16-Bit-Prozessor. Aufgehoben wurde diese erst beim 386 DX.

Jahr	Wortbreite	Anzahl Zustände	Beispiele
1971	4 Bit	16	Intel 4004
1974	8 Bit	256	Intel 8080 Zilog Z80 MosTek 6502
1978	16 Bit	65.536	Intel 8086 Motorola 68000
1985	32 Bit	> 2 Milliarden	Intel 80386 PowerPC
1992	64 Bit	> 18 Trillionen	DEC Alpha AMD64 Intel64 PowerPC G5

Tabelle 18.1 Übersicht über die Entwicklung der verschiedenen Prozessorgenerationen

Die Taktrate sagt viel weniger über die tatsächliche Arbeitsgeschwindigkeit einer CPU aus, als die Intel-Werbung glauben machen möchte. Je nach Komplexität bestimmter Befehle dauert ihre Ausführung mehrere Taktzyklen; unterschiedliche Prozessorfamilien lösen verschiedene Aufgaben nicht immer mit derselben Effizienz. Außerdem verbringen Prozessoren einen Großteil ihrer Zeit mit Warten: Da Ein- und Ausgabe verhältnismäßig langsam stattfinden, hat ein Prozessor häufig nichts zu tun, weil die erforderlichen Daten nicht schnell genug nachkommen.

Zu wenig Arbeitsspeicher führt beispielsweise zu erheblich größeren Geschwindigkeitseinbußen als ein etwas langsamerer Prozessor: Das Auslagern nicht benötigter Speicherinhalte auf die Festplatte und das Laden der als Nächstes erforderlichen verbraucht relativ viel Zeit. Ein speicherhungriges Programm wie die in Kapitel 8, »Desktop-Software«, vorgestellte Bildbearbeitung GIMP läuft auf einem älteren Rechner mit nur 256 oder 512 MByte RAM zum Beispiel nicht sonderlich schnell.

Im Zusammenhang mit der Taktfrequenz ist die oben angesprochene Tatsache von Bedeutung, dass die meisten CPUs heute Mehrkern-Prozessoren (Multicore) sind. Deshalb wird die Taktfrequenz beispielsweise als 2×2.600 MHz angegeben, weil jeder der beiden Kerne mit 2.600 MHz arbeitet. Kurz vor der Einführung der Multicore-CPU's lagen die Taktraten etwas höher, was sich durch einen höheren Stromverbrauch auswirkte.

Die tatsächliche Effizienz von Prozessoren lässt sich übrigens durch andere Werte besser angeben als durch die Taktrate:

- ▶ Die Anzahl der Befehle, die in einer Sekunde ausgeführt werden können, genannt *MIPS* – Million Instructions per Second. Dieser Wert wird durch Benchmark-Tests (möglichst an realistische Anwendungen angelehnte Abfolgen von Befehlen) gemessen. Wichtig ist, dass für einen realistischen Vergleich unabhängige Benchmark-Programme verwendet werden und nicht diejenigen der Prozessorhersteller.
- ▶ Insbesondere für die Multimedia-Fähigkeiten eines Prozessors liefert die Anzahl der pro Sekunde durchführbaren Fließkommaoperationen, *FLOPS* (Floating Point Operations Per Second), einen guten Anhaltspunkt: 3-D-Grafik, Audio- und Videoperformance sind auf die Fähigkeit zu möglichst schnellen Fließkommaberechnungen angewiesen.

Prozessorarchitekturen

Es lassen sich zwei grundlegende Prozessorarchitekturen voneinander unterscheiden. Der Unterschied besteht in der Ausstattung der Befehlstabelle:

- ▶ Das klassische Modell, das auf Entwicklungen aus den 70er-Jahren basiert, versucht tendenziell, immer mehr und immer komplexere Anweisungen unmittelbar durch einzelne Prozessorinstruktionen zu verwirklichen. Aus diesem Grund wurde es – zur Unterscheidung vom später entwickelten zweiten Modell – nachträglich als *CISC* (Complex Instruction Set Computer), also als Rechner mit komplexem Befehlssatz, bezeichnet.
- ▶ Ein modernerer, in den 80er-Jahren entwickelter Ansatz versucht im Gegenteil, die Struktur des Prozessors zu vereinfachen. Dazu wird der Befehlssatz auf wenige, besonders schnell und einfach auszuführende Befehle vermindert. Komplexere Funktionen lassen sich durch mehrere solcher einfachen Instruktionen verwirklichen. Dieser Konstruktionsansatz beschleunigt die Ausführung der einfachen Befehle erheblich: Sie lassen sich durch ihre simple Struktur in Pipelines (Warteschlangen) anordnen. Dadurch kann auf effiziente Weise ein Befehl nach dem anderen ausgeführt werden. Beim klassischen Prozessordesign konnte dagegen selbst das Lesen des folgenden Befehls erst beginnen, wenn der vorherige vollständig abgeschlossen war. Diese Architektur wird als *RISC* (Reduced Instruction Set Computer) bezeichnet.

Die Hauptvertreter der CISC-Architektur sind die Prozessoren von Intel und Kompatible wie die diversen AMD-CPU's. Allerdings wurden bereits in den 1993 vorgestellten Pentium-Prozessor einige Funktionen eingebaut, die bis dahin nur in RISC-Prozessoren verwirklicht worden waren, beispielsweise die oben

genannten Pipelines. Nach und nach gelang es Intel und AMD, immer mehr RISC-Vorzüge in ihre Prozessoren einzubauen. Vor allem werden die komplexen CISC-Befehle in einzelne, RISC-artige Mikroinstruktionen zerlegt, die anschließend vom eigentlichen Prozessorkern mit erheblich höherer Effizienz ausgeführt werden. Deshalb dominieren heute, nach einer vorübergehenden RISC-Hochphase, die Weiterentwicklungen der klassischen CISC-Prozessoren – selbst Apple ist vor einigen Jahren vom PowerPC auf Intel-CPU's umgestiegen.

Wie Prozessoren arbeiten

Nach dieser theoretischen Aufbaubeschreibung von Mikroprozessoren ist es natürlich auch interessant zu erfahren, wie ein Prozessor eigentlich arbeitet. Schematisch betrachtet geschieht bei der Ausführung eines Programms durch den Prozessor Folgendes:

1. Der aktuelle Befehl wird aus dem Programm gelesen; die Stelle wird durch den Befehlszeiger des Prozessors angezeigt.
2. Der Prozessor schlägt die Nummer des erhaltenen Befehls in der Befehlstabelle nach und liest je nach Befehl die passende Anzahl darauffolgender Bytes als Parameter dieses Befehls. Dabei rückt der Befehlszeiger hinter das letzte Parameter-Byte, um für das Lesen des nächsten Befehls bereit zu sein.
3. Der Befehl wird ausgeführt. Dies ist der komplexeste Teil der Prozessortätigkeit, denn je nach konkretem Befehl kann dazu das Lesen von Daten aus dem Arbeitsspeicher, die Ansteuerung von Peripherieschnittstellen, das Rechnen in der ALU oder die Durchführung eines Sprungs im Programm gehören.
4. Falls ein Sprung stattfindet, wird der Befehlszeiger an die entsprechende neue Position gesetzt. Andernfalls geht es an der nach dem Lesen der Parameter ermittelten Stelle weiter.

Es lohnt sich übrigens, die Funktionsweise von Sprüngen etwas genauer zu betrachten. Prozessoren beherrschen nämlich grundsätzlich zwei verschiedene Arten von Sprüngen: Ein unbedingter Sprung wird immer ausgeführt, sobald der entsprechende Befehl gelesen wird. Bedingte Sprünge werden dagegen nur dann ausgeführt, wenn bestimmte Bedingungen zutreffen. Diese Bedingungen betreffen meist die Zustände eines Flag-Registers. Flags sind Statusbits, deren Werte durch Vergleichsoperationen, Fehler oder direkte Manipulation durch ein Programm gesetzt werden.

Eine noch etwas komplexere Variante betrifft den Aufruf von Unterprogrammen: Anstatt ohne Wiederkehr zu einer bestimmten Programmstelle zu springen, wird hier die auf den Sprungbefehl folgende Adresse gespeichert. Dazu dient ein Last-in-First-out-Speicher, der als Stack (Stapel) bezeichnet wird. Wenn am Ende des

Unterprogramm der Rücksprungbefehl erfolgt, holt sich das Programm den obersten Wert vom Stack und springt zur angegebenen Programmadresse. Die Adresse, die zurzeit den Abschluss des Stacks bildet, wird von einem speziellen Steuerregister angezeigt, dem Stack Pointer.

Ein weiterer Grund, warum Programme nicht immer linear nacheinander ausgeführt werden, sind die weiter oben bereits erwähnten Prozesse: Eine CPU führt üblicherweise abwechselnd Befehle mehrerer Programme aus. Vor dem Wechsel zu einem anderen Prozess wird dessen Zustand gespeichert, das heißt, die Inhalte der Prozessorregister werden gesichert, um sie bei der späteren Wiederaufnahme dieses Prozesses zurückzusetzen – aus der Sicht des Prozesses findet seine Ausführung also ohne Unterbrechung statt.

Im Zusammenhang mit Ein- und Ausgabevorgängen werden darüber hinaus sogenannte Interrupts verwendet: Da die Anfragen, die die Hardware an den Prozessor stellt, asynchron auftreten, muss der Prozessor sich immer wieder darüber informieren, ob ein Gerät auf Antwort wartet. Deshalb fragt er die verschiedenen Geräte in regelmäßigen Abständen ab und unterbricht unter Umständen den laufenden Prozess zugunsten der Hardwarekommunikation.

Maschinenbefehle

Die Maschineninstruktionen, von denen hier die Rede ist, lassen sich nicht mit den mächtigen Befehlen höherer Programmiersprachen vergleichen. Sie bewegen sich auf einem viel niedrigeren Niveau. Der Prozessor »weiß« nicht, dass er den Buchstaben A in 16 Punkt Garamond auf den Bildschirm zeichnet, einen Videoclip abspielt oder das Spielprogramm »Tomb Raider« ausführt. Aus der Sicht der CPU geht es immer nur um Rechenschritte, also um die Manipulation irgendwelcher numerischen Werte.

Typische Maschinenbefehle lauten beispielsweise folgendermaßen:

- ▶ Hole den Wert aus der Speicherstelle mit der Nummer 9A33 und lege ihn im Rechenregister BX ab. In Intel-Assembler lautet diese Anweisung übrigens folgendermaßen: `MOV BX, $9A33`.
- ▶ Addiere den Wert 10 zum Inhalt des Rechenregisters BX: `ADD BX, 10`.
- ▶ Vergleiche das Register BX mit dem Wert 20: `CMP BX, 20`.
- ▶ Falls der Vergleichsbefehl »gleich« ergeben hat (ein bestimmtes Flag enthält den Wert 0), springe zur Programmadresse C9A4: `JE $C9A4` (je steht für »jump if equals«, also »Sprung, falls gleich«).

Beachten Sie, dass die Assembler-Sprache bereits eine vereinfachende Abstraktion der Maschinensprache ist; in Wirklichkeit bestehen alle Befehle aus Zahlen.

Es gibt zum Beispiel nicht »den« MOV-Befehl zum Verschieben von Speicherinhalten. Es handelt sich um eine ganze Sammlung von Befehlen, die lediglich gemeinsam haben, dass sie einen bestimmten Wert an einer bestimmten Stelle ablegen sollen: Das an erster Stelle angegebene Ziel kann ein Prozessorregister, eine Speicheradresse oder eine indirekte Adresse (eine Speicheradresse, deren Inhalt wiederum als Speicheradresse interpretiert werden soll) sein. Ebenso kann der Wert, der an der angegebenen Stelle gespeichert werden soll, ein konkreter Wert, ein Register oder eine Speicheradresse sein.

Andere Prozessorhersteller haben ihre Assembler-Sprachen nicht notwendigerweise genauso definiert wie Intel, die Aufteilung der Befehle in Gruppen mit demselben Namen kann vollkommen anders geregelt sein. Letzten Endes interessiert das alles den Prozessor ohnehin nicht, weil er gar kein Assembler versteht. Es handelt sich lediglich um eine bequemere Schreibweise der Maschinensprache für Menschen. Programme, die in Assembler geschrieben werden, müssen von einem geeigneten Programm in die eigentliche Maschinensprache übertragen (assembliert) werden. Natürlich ist dieser Vorgang einfacher als die Übersetzung einer höheren Programmiersprache, weil in der Regel jeder Assembler-Befehl für genau eine wohldefinierte Maschineninstruktion steht.

Was die Übersetzung von Hochsprachen angeht, ist zudem noch folgender Umstand interessant: Wenn Sie ein Programm schreiben und mit einem Compiler übersetzen, entsteht ein Programm, das nur auf einem bestimmten Prozessor läuft, weil es Maschinenbefehle für diesen Prozessor enthält. Interessanterweise funktioniert dies aber nicht über Betriebssystemgrenzen hinweg: Wenn Sie auf Ihrem Intel-PC beispielsweise Linux und Windows installieren, unter Linux ein C-Programm schreiben und kompilieren und anschließend Windows booten, kann das Programm dort nicht funktionieren. Programme, die in Hochsprachen geschrieben werden, bestehen nämlich nicht nur aus neutralen Maschinensprachanweisungen. Zahlreiche Funktionen, etwa zur Ein- und Ausgabe, werden durch Routinen des Betriebssystems bereitgestellt, die das Programm durch sogenannte Systemaufrufe anspricht. Dieser Mechanismus wird im nächsten Abschnitt genauer erläutert.

18.1.2 Der Arbeitsspeicher

Der Arbeitsspeicher eines Computers enthält die Programme, die zurzeit ausgeführt werden, und die Daten, die von ihnen aktuell verarbeitet werden. Weiter oben wurde bereits die mehrstufige Speicherarchitektur angesprochen, die vom Prozessor in Zusammenarbeit mit dem Betriebssystem verwaltet wird. In diesem

Unterabschnitt geht es dagegen konkret um den Arbeitsspeicher als Hardwarekomponente.

Der Arbeitsspeicher besteht aus Speicherbausteinen, die als *RAM* bezeichnet werden; die Abkürzung steht für Random Access Memory (Speicher mit wahlfreiem Zugriff). Der Begriff »Random Access« bedeutet in diesem Zusammenhang zweierlei:

- ▶ Die Inhalte dieses Speichers können sowohl gelesen als auch verändert werden. Den Gegenbegriff bildet das ROM (Read Only Memory), das weiter unten im Zusammenhang mit dem BIOS behandelt wird.
- ▶ Auf jedes Byte des Speichers kann einzeln in beliebiger Reihenfolge zugegriffen werden; in diesem Zusammenhang ist Random Access der Gegenbegriff zum sequenziellen Zugriff, der beispielsweise bei Magnetbandspeichern eingesetzt wird.

Alle RAM-Bausteine haben gemeinsam, dass ihr Inhalt flüchtig ist. Sie müssen ständig mit Strom versorgt werden, weil es ansonsten zum vollständigen Datenverlust kommt. Es gibt allerdings zwei verschiedene konkrete Bauformen:

- ▶ Das Dynamic RAM (*DRAM*) benötigt nicht nur das Anliegen einer Spannung, sondern der Inhalt jeder einzelnen Speicherstelle muss mit jedem Taktzyklus aufgefrischt werden (Refresh). Es ist vergleichsweise günstig herzustellen und hat einen niedrigeren Stromverbrauch.
- ▶ Das Static RAM (*SRAM*) bedarf lediglich einer Spannung. Es arbeitet erheblich schneller als DRAM, verbraucht aber mehr Strom und ist viel teurer. SRAM-Bausteine werden deshalb niemals zur Realisation des ganzen Arbeitsspeichers eingesetzt, sondern nur für die Cache-Speicher.

RAM ist stets in einzelnen Speicherzellen organisiert, die 1 Byte groß sind und jeweils eine eigene Adresse haben. Die Art und Weise, wie Speicher konkret adressiert wird, hängt vom Prozessor und indirekt vom Betriebssystem ab. Der Mechanismus der Speicheradressierung bei Intel-CPUs wird weiter unten – im Zusammenhang mit der Linux-Speicherverwaltung – kurz erläutert.

Der eigentliche Arbeitsspeicher wird in Slots senkrecht auf das Mainboard aufgesteckt. Es handelt sich um kleine, rechteckige Platinen, die mit mehreren konkreten Chips versehen sind. Auf einer Seite befindet sich eine Reihe von Kontakten, die in den entsprechenden Slot gehören. Bei aktuellen *DIMM*-Modulen (Double Inline Memory Modules) für DDR-RAM müssen Sie dazu einen Hebel zur Seite ziehen und das Speichermodul fest (aber vorsichtig!) in den Slot drücken, bis es merklich einrastet und der Hebel sich automatisch in die aufrechte Position begibt.

Es existieren verschiedene Bauformen von RAM-Bausteinen. Gängig ist zurzeit vor allem das etwas unglücklich bezeichnete *DDR-RAM*² (die Abkürzung steht für »Double Data Rate«). Die Besonderheit besteht – wie der Name schon sagt – darin, dass diese Speicherbausteine mit der doppelten Datenrate von SD-RAMs arbeiten. Pro Taktzyklus können sie mit anderen Worten doppelt so viel Inhalt aufnehmen oder abgeben und beschleunigen so einen der schlimmsten Engpässe des PC-Designs. Inzwischen sind vor allem die nochmals verbesserten Varianten DDR2 und DDR3 im Einsatz.

Für die Performance eines Rechners ist die Menge des Arbeitsspeichers erheblich wichtiger als die Speicherbausteintechnologie. Für einen gewöhnlichen Büro-PC ist 1 GByte RAM ein vernünftiger Wert. Für Server, Workstations oder Multimedia-Anwendungen sollten es dagegen mindestens 2 bis 4 GByte sein. Denken Sie daran, dass all diese Werte einem sehr starken Wandel unterworfen sind – die vorliegenden Angaben gelten für Mitte 2008 und werden bald überholt sein.

18.1.3 Bus- und Anschlusssysteme

Ein weiterer wichtiger Aspekt der Hardware – die Schnittstelle zwischen Zentraleinheit und Peripherie – sind die diversen Bus- und Anschlusssysteme. Dazu gehören die Slots für Erweiterungskarten, die internen Anschlüsse für Laufwerke sowie die zahlreichen externen Anschlüsse für Tastatur, Maus, Monitor, Modem und viele andere Geräte.

Serielle und parallele Datenübertragung

Das relevanteste Unterscheidungsmerkmal zwischen verschiedenen Arten von Anschlüssen ist die Frage, ob sie Daten seriell oder parallel übertragen.

Bei der *seriellen Datenübertragung* werden die einzelnen Bits nacheinander, bei der *parallelen Übertragung* werden die Bits dagegen auf mehreren nebeneinanderliegenden Leitungen gleichzeitig übertragen (je nach Anschlussart sind es 8, 16, 32 oder 64 Bit).

Auf der untersten Ebene muss zwischen »Sender« und »Empfänger« zunächst einmal Einigkeit darüber herrschen, wie die transportierten Ströme überhaupt als Daten interpretiert werden sollen. Es geht um die Frage, welche elektrischen Ereignisse (zum Beispiel Spannungswechsel oder Spannungszustände) überhaupt als 1 oder als 0 interpretiert werden sollen. Statt »Strom« müsste man allgemei-

2 Auch wenn die Bezeichnung DDR-RAM nichts damit zu tun hat: In der damaligen DDR wurde der allererste Speicherchip entwickelt, der eine Kapazität von einem Megabit besaß.

ner eigentlich von Energie sprechen; beispielsweise übertragen Lichtwellenleiter keinen Strom, sondern Licht.

Bei seriellen Leitungen muss als Nächstes die Frage geklärt werden, in welcher Reihenfolge die aufeinanderfolgenden Einsen und Nullen überhaupt zu ganzen Bytes zusammengesetzt werden sollen (bei parallelen Leitungen ist dies natürlich irrelevant).

Angenommen, der Wert 77 soll als 8-Bit-Sequenz (binär 01001101) übertragen werden. Falls der Sender die Sequenz in dieser »natürlichen« Reihenfolge (die üblicher ist) überträgt, der Empfänger sie aber falsch herum interpretiert, erhält Letzterer statt 77 den Wert 178 (10110010)!

Abgesehen davon werden in der Regel nicht nur aufeinanderfolgende Datenbits übertragen, sondern dazwischen noch zusätzliche Kontrollbits. Es gibt drei grundsätzliche Arten solcher Zusatzbits:

- ▶ Ein *Startbit* zeigt den Beginn einer neuen Übertragungssequenz an, wenn es den Wert 1 hat. Startbits werden nur noch sehr selten verwendet.
- ▶ Ein *Stoppbit* kennzeichnet entsprechend das Ende einer Sequenz, wenn es den Wert 1 besitzt.
- ▶ Ein *Prüfbit* oder *Parity-Bit* sorgt für eine Plausibilitätskontrolle auf der untersten Ebene: Die Anzahl der Einsen in den Datenbits wird gezählt. Das Parity-Bit (wörtlich: Geradheits-Bit) wird derart auf 1 oder 0 gesetzt, dass sich insgesamt immer eine gerade Anzahl von Einsen ergibt. Wenn die Paritätsprüfung beim Empfänger einen Fehler ergibt (beispielsweise drei empfangene Einsen, Parity 0), liegt auf jeden Fall ein Übertragungsfehler vor. Andererseits bedeutet ein korrektes Parity nicht unbedingt, dass die Übertragung fehlerfrei funktioniert hat. Aus diesem Grund verwenden die meisten Übertragungsverfahren auf einer höheren Ebene Prüfsummen.

Die Verwendung von Parity-Bits ist mittlerweile eher unüblich geworden. Sie war vor allen Dingen für die Datenübertragung über analoge Telefonleitungen mit geringer Qualität (starken Nebengeräuschen) wichtig, weil Übertragungsfehler bei ihnen recht häufig vorkamen und auf diese Weise verringert werden konnten.

Die Leitungskonventionen werden üblicherweise durch drei aufeinanderfolgende Werte ausgedrückt, und zwar folgendermaßen:

- ▶ Wie viele Datenbits werden pro Sequenz verwendet? In der Regel sind es 8. Das früher übliche Telex-Verfahren verwendete dagegen nur 5 Bits; manchmal gibt es auch Übertragungsverfahren mit 7 oder 9 Datenbits.

- ▶ Ein N (No Parity) zeigt an, dass keine Paritätsprüfung verwendet wird, während ein P für Parity steht.
- ▶ Der dritte Wert gibt an, wie viele Stoppbits verwendet werden (0, 1 oder sogar 2).

Ein übliches Verfahren ist beispielsweise das für Modemverbindungen zum Internetprovider verwendete 8N1 (8 Datenbits, kein Parity, 1 Stoppbit). Eine Sequenz, die mithilfe dieser Methode übertragen wird, könnte zum Beispiel so aussehen:

```
01001000 0 01000001 0 01001100 0 01001100 0 01001111 1
```

Es handelt sich um die Werte 72, 65, zweimal 76 und 79 – als ASCII-Zeichen interpretiert ist es der Text »HALLO«. Die 1 hinter dem letzten Datenbit ist das gesetzte Stoppbit und besagt, dass die Übertragung nun beendet ist.

In den letzten Jahren ist zu beobachten, dass die Hersteller immer stärker auf serielle Datenleitungen setzen. So werden Drucker nur noch an die serielle USB-Schnittstelle statt an den veralteten Parallelport angeschlossen; der SCSI-Anschluss, der früher häufig für externe Laufwerke verwendet wurde, wird fast immer durch FireWire oder USB 2.0 ersetzt. Selbst der interne Festplattenanschluss EIDE ist im Aussterben begriffen; seine serielle Variante Serial ATA hat sich durchgesetzt.

Für diesen Wechsel zu seriellen Leitungen gibt es gute Gründe: Sie benötigen vor allem weniger Strom und können längere Entfernungen überwinden als parallele Leitungen. Das Problem der geringeren Datenübertragungsleistung, das die Entwickler früher in manchen Bereichen zur Verwendung der parallelen Datenübertragung zwang, konnte dadurch gelöst werden, dass die hervorragende Verarbeitungsqualität heutiger Leitungen den Transport von Daten mit hoher Frequenz ermöglicht.

Noch ein Wort zur Angabe der Übertragungsgeschwindigkeiten verschiedener Leitungen: Bei parallelen Leitungen lässt sie sich einfach in Byte pro Sekunde beziehungsweise in den entsprechenden höheren Einheiten (Kilobyte/s, Megabyte/s und so weiter) angeben. Diese Einheiten werden jeweils mit 2¹⁰ (1.024) multipliziert, um zur nächsthöheren zu gelangen – 1 Kilobyte ist also 1.024 Byte, 1 Megabyte 1.024 Kilobyte und so weiter.

Serielle Leitungen verwenden, wie oben erläutert, unterschiedliche Bitfolgen zur Darstellung der Datenbytes. Ein tatsächliches Datenbyte kann je nach Übertragungsstandard durch eine Bitfolge von 8, 9 oder sogar 10 Bits dargestellt werden. Deshalb lässt sich die Geschwindigkeit einer seriellen Leitung nicht in Byte/s oder Vielfachen davon angeben. Sie wird stattdessen in Bit pro Sekunde (bps)

gemessen. Beachten Sie, dass die Vervielfältigungseinheiten von Bit (Kilobit, Megabit, Gigabit) nicht mit dem Faktor 1.024, sondern mit der dezimalen 1.000 gebildet werden.

Hat zum Beispiel ein Modem eine Übertragungsrate von 56,6 kbps (Kilobit pro Sekunde), dann bedeutet dieser Wert, dass in einer Sekunde 56.600 Bit übertragen werden. Wird der Übertragungsstandard 8N1 verwendet, dann muss dieser Wert durch 9 geteilt werden, um auf die maximal transportierbaren Bytes zu kommen: Es sind etwa 6.289 Byte oder 6,14 Kilobyte. In der Praxis vereinfacht man solche Rechnungen allerdings häufig und legt 8 Datenbits zugrunde. 56.600 Bit wären demnach 7.075 Byte oder 6,91 Kilobyte.

Hardware-Ressourcen

Zur Regelung der geordneten Kommunikation zwischen Prozessor, Arbeitsspeicher und Peripheriegeräten werden verschiedene Kommunikationsressourcen verwendet. Es handelt sich um Kanäle für Hardware-Interrupts, um Speicheradressen, über die der Datenaustausch geregelt wird, und um sogenannte DMA-Kanäle für die direkte Übertragung von Daten in den Arbeitsspeicher und zurück.

Damit sich die Anfragen der verschiedenen Geräte nicht in die Quere kommen und der Prozessor sie unterscheiden kann, werden verschiedene IRQs (Interrupt Requests) verwendet. Dafür verwendet jedes Gerät, das kommunizieren möchte, ein einmaliges und eindeutiges Signal. Von diesen Signalen gibt es 16 verschiedene, die von 0 bis 15 durchnummeriert werden; der Prozessor untersucht sie jeweils der Reihe nach, um festzustellen, ob über den jeweiligen IRQ Kommunikation gewünscht wird.

Bei modernen PCI-Mainboards ist IRQ-Sharing möglich. Zwei oder mehr moderne Geräte können sich denselben IRQ teilen und mithilfe anderer Techniken erkennbar machen, um welches dieser Geräte es sich tatsächlich handelt. Zu diesem Zweck enthält der Chipsatz eine Komponente, die als programmierbarer Interrupt-Controller (PIC) bezeichnet wird. Bei Motherboards mit der veralteten ISA-Architektur ist IRQ-Sharing dagegen vollkommen ausgeschlossen.

Einige IRQs sind standardmäßig reserviert. Bei manchen von ihnen kann die voreingestellte Belegung verändert werden, andere sind dagegen fest zugewiesen. Tabelle 18.2 zeigt eine entsprechende Übersicht.

Eine weitere wichtige Kommunikationsressource ist die *I/O-Basisadresse*: Diese Speicheradresse markiert den Beginn eines Adressblocks, der für den Austausch von Konfigurations- und Steuerungsinformationen zwischen dem Prozessor und dem jeweiligen Gerät verwendet wird. Die jeweiligen Adressen werden hexadezimal angegeben und liegen ganz unten im adressierbaren Bereich. Die Adress-

bereiche werden in der Regel im Abstand von 0x20 (dezimal 32 Byte) belegt, also zum Beispiel 0x0200, 0x0220, 0x0240 und so weiter. Beachten Sie, dass es sich nicht wirklich um Adressen im Arbeitsspeicher handelt, sondern um spezielle Geräteschnittstellen, die vom Prozessor nach derselben Logik angesprochen werden wie Speicheradressen.

IRQ	Gerät	Änderung möglich
0	Systemtaktgeber	nein
1	Tastatur	nein
2	Programmierbarer Interrupt-Controller	nein
3	Serieller Port COM 2	ja
4	Serieller Port COM 1	ja
6	Diskettenlaufwerk	ja
7	Parallelport	ja
8	Echtzeituhr	nein
12	PS/2-Myyyaus	ja
13	Koprozessor	nein
14	Erster IDE-Controller	ja
15	Zweiter IDE-Controller	ja

Tabelle 18.2 Fest belegte IRQs

Eine nicht mehr besonders wichtige Ressource sind die *DMA*-Kanäle (Direct Memory Access). Bei DMA handelt es sich um ein Verfahren zur direkten Übertragung von Gerätedaten in den Arbeitsspeicher und umgekehrt, ohne dass jedes einzelne Datenbyte den Prozessor passieren muss. Diese Technik beseitigt einen der bedeutendsten Engpässe der klassischen Ein- und Ausgabesteuerung, ist also alles andere als unwichtig. Allerdings wird die DMA-Steuerung der meisten Geräte heute nicht mehr über die klassischen DMA-Kanäle durchgeführt, sondern über ein verbessertes Verfahren, das als Bus Mastering bezeichnet wird. DMA-Kanäle werden eigentlich nur noch für Soundkarten und Diskettenlaufwerke verwendet, vor allem, weil es sich nur um 8- und 16-Bit-Kanäle handelt.

Die Zuweisung der Ressourcen musste früher umständlich von Hand vorgenommen werden. Zunächst mussten sie am Gerät selbst eingestellt werden – anfangs durch Jumper, später per Software. Anschließend wurde dem Betriebssystem mitgeteilt, welche Ressourcen das jeweilige Gerät verwendete. Allzu oft kam es

zu Ressourcenkonflikten, vor allem waren chronisch zu wenige IRQs vorhanden. Heutzutage erledigt das Betriebssystem diese Aufgaben in aller Regel ohne großartige Eingriffe (Plug-and-Play-Verfahren) – nur bei sehr alter Hardware (vor allem ISA) müssen Sie manchmal noch Ressourcen von Hand eingeben.

Steckplätze für Erweiterungskarten

Viele wichtige Peripheriegeräte werden schon seit den Anfangstagen der PCs als Einsteckkarten realisiert. Das hat vor allen Dingen den Vorteil, dass diese Karten Anschlüsse nach außen führen können: Grafikkarten, Netzwerkkarten oder Soundkarten sind jeweils mit spezialisierten Schnittstellen ausgestattet, an die ein Monitor, ein Netzwerkkabel beziehungsweise Ton-Ein- und -Ausgabegeräte angeschlossen werden.

Eine Karte verfügt am unteren Rand über eine lange Reihe von Anschlüssen, mit denen sie in den jeweils passenden Slot auf dem Mainboard gesteckt wird. Den Abschluss nach außen, zur Geräterückseite, bildet ein Slotblech, auf dem sich die genannten externen Anschlüsse befinden. Der obere Rand des Slotblechs ist nach außen gebogen und wird mit einer Schraube fixiert.

Es gibt folgende verschiedene Arten von Kartenschnittstellen:

- ▶ Der *PCI-Anschluss* (Peripheral Component Interface) ist der Standard-Kartenanschluss für PC und Mac. Der PCI-Bus wird mit einer Taktfrequenz von 33 MHz betrieben und besitzt eine Datenbreite von 32 Bit; neuere PCI-Varianten bieten 64 Bit. Die Konfiguration erfolgt bei entsprechend ausgestatteten Betriebssystemen per Plug & Play.
- ▶ Der *AGP-Anschluss* (Accelerated Graphics Port) ist ein spezieller Anschluss für Grafikkarten. Die Taktfrequenz ist höher als bei PCI, sie beträgt mindestens 66 MHz. Es gibt inzwischen auch Varianten mit 100 und 133 MHz. Die Datenbreite wurde auf 64 Bit verdoppelt.
- ▶ *PCI-Express* (PCIe) ist die neueste Anschlussart, die auf modernen Mainboards sowohl PCI als auch AGP ersetzt. Ihre Taktrate beträgt 1,25 GHz, aber der Hauptvorteil besteht darin, dass jeder einzelne PCIe-Anschluss über eine eigene Punkt-zu-Punkt-Verbindung verfügt, während sich alle PCI-Anschlüsse einen gemeinsamen Bus teilen.

Laufwerksanschlüsse

Für Festplatten, CD-ROM-Laufwerke und andere Massenspeicher gibt es zwei verschiedene wichtige Hauptarten von Anschlüssen: EIDE/SATA und SCSI.

EIDE (Enhanced Integrated Device Electronics) wird auch *ATA* (Advanced Technology Attachments) genannt und war bis vor Kurzem auf den meisten Mainboards integriert. In der Regel sind auf solchen Boards zwei Anschlüsse für 40-polige Flachbandkabel vorhanden; ein neuerer Standard namens Ultra DMA ermöglicht höhere Datenraten, benötigt aber spezielle 80-polige Kabel. Mit jedem der beiden Anschlüsse können je zwei Geräte verbunden werden. Eines der beiden Geräte wird als Master bezeichnet, das andere als Slave. Der Master hat beim Datenaustausch Priorität, im Zweifelsfall muss der Slave auf Daten warten. Deshalb sollten CD- und DVD-Brenner niemals als Slave betrieben werden, weil ein Abreißen des Schreibdatenstroms den beschreibbaren Datenträger zerstören kann.

Die Unterscheidung zwischen Master und Slave hat übrigens nichts mit der Reihenfolge am Kabel zu tun, auch wenn es bei manchen Geräten noch die historische Einstellmöglichkeit »Cable Select« gibt. Die Einstellung, ob ein Gerät als Master oder als Slave betrieben wird, erfolgt durch einen Jumper oder DIP-Schalter am Gerät selbst.

Damit die Stecker der Flachbandkabel korrekt eingesteckt werden, besitzt das Kabel eine rote Ader; diese Seite gehört an den Pin mit der Nummer 1. Sowohl das Mainboard als auch die Geräte sind meist entsprechend beschriftet. Bei den Laufwerken befindet sich die 1 meistens neben dem Stromanschluss.

Der wichtigste Vorteil gegenüber SCSI besteht darin, dass *EIDE*-Geräte erheblich günstiger sind. Im normalen Alltagsbetrieb an PCs ist die Leistungsfähigkeit darüber hinaus vergleichbar; SCSI ist nicht schneller, sondern nur belastbarer und daher für Server und andere Hochleistungsmaschinen besser geeignet. Der größte Nachteil von *EIDE* besteht dagegen darin, dass der Anschluss elektrisch instabil ist und die maximale Kabellänge somit nur 60 cm beträgt, weshalb es keine externen *EIDE*-Geräte gibt.

Vor einigen Jahren begann man mit der Entwicklung einer seriellen Variante des *EIDE*-Anschlusses, genannt *Serial ATA* (*S-ATA*). Um Verwechslungen vorzubeugen, wurde das herkömmliche *ATA* inzwischen in *Parallel ATA* (*P-ATA*) umbenannt.

Hauptvorteile der neuen Version sind die Möglichkeit längerer Kabel (bis zu einem Meter), eine vereinfachte Anschluss technik mit dünneren, flexibleren Kabeln und kleineren Steckern sowie die von USB oder FireWire bekannten Hot-Plugging-Fähigkeiten (Gerätewechsel im laufenden Betrieb, siehe unten). Inzwischen hat *S-ATA* sich weitestgehend durchgesetzt.

SCSI ist die Abkürzung für Small Computer System Interface. Wie der Name vermuten lässt, wurde diese Schnittstelle ursprünglich nicht für PCs konzipiert, sondern eben für Kleincomputer (die etwa so groß wie Kleiderschränke waren. Durch S-ATA für interne und USB oder FireWire für externe Geräte verliert SCSI vor allem im Desktop-Bereich zunehmend an Bedeutung, im Server- und Workstation-Bereich ist es dagegen noch verbreitet.

Eine SCSI-Schnittstelle bietet die Möglichkeit, 7 Geräte anzuschließen; der neuere Wide-SCSI-Standard erlaubt sogar 15. Bei allen älteren Macs bis zum ersten G3-PowerMac von Mitte 1998 war SCSI integriert und wurde auch für die bereits eingebauten Festplatten und CD-ROM-Laufwerke verwendet. Bei PCs war SCSI nur sehr selten onboard und wurde meist als PCI-Einsteckkarte nachgerüstet; die meisten dieser SCSI-Karten stammten von der Firma Adaptec.

Ein SCSI-Controller verfügt in der Regel über drei Anschlüsse: zwei interne für 50-polige Flachbandkabel sowie einen externen, an den entweder ein 25-poliges Centronics-Kabel (optisch identisch mit dem klassischen Parallelport eines PCs, aber technisch abweichend) oder ein modernes SUB-D-Kabel angeschlossen werden kann.

Beachten Sie, dass Sie von diesen drei Anschlüssen nur zwei verwenden dürfen! Am SCSI-Controller dürfen Geräte nur busförmig angeschlossen werden, das heißt hintereinander in einer Kette, und nicht etwa sternförmig (mit drei Abzweigungen).

Das erste und das letzte Gerät in der Kette benötigen jeweils einen Abschlusswiderstand, auch Terminator genannt. Dieser sieht je nach Geräteart unterschiedlich aus: Die meisten internen Geräte wie Festplatten oder CD-ROM-Laufwerke verwenden hierfür einen Jumper. Bei externen Geräten (Scanner, externe Festplatten, CD-Brenner und so weiter) gibt es hingegen oft die Möglichkeit, über einen durchgeschleiften Anschluss ein weiteres Gerät anzuschließen. In diesem Fall ist der Terminator oftmals nur ein großer Stecker, der auf diesen Anschluss gesetzt wird, oder manchmal auch ein DIP-Schalter.

Wenn der SCSI-Controller selbst ein Ende der Kette bildet, weil nur einer der drei Anschlüsse genutzt wird, dann muss er ebenfalls terminiert werden: Dies geschieht oft mittels eines Jumpers oder DIP-Schalters, bessere Controller terminieren sich in diesem Fall automatisch selbst.

Damit SCSI-Geräte angesteuert werden können, benötigen sie eine eindeutige Nummer, die als SCSI-ID bezeichnet wird. Diese hat nichts mit der Reihenfolge in der Kette zu tun; wichtig ist nur, dass jede verwendete Nummer einmalig ist. Beim klassischen SCSI existieren die IDs 0 bis 7, bei Wide SCSI dagegen 0 bis 15.

Je höher die Nummer eines Gerätes, desto höher ist seine Priorität. Der Controller selbst hat daher meist die ID 7 beziehungsweise 15.

Je nach Gerätesorte wird die ID auf unterschiedliche Weise eingestellt: Bei den meisten externen Geräten gibt es einen Drehschalter mit vielen Auswahlmöglichkeiten oder einen DIP-Schalter, der nur zwei Optionen bietet. Bei internen SCSI-Geräten wird die ID oft durch einen oder zwei Jumper eingestellt. Manchmal besitzen sie auch einfach eine festgelegte ID. Vor dem Kauf eines weiteren Gerätes müssen Sie deshalb darauf achten, dass Sie keines mit derselben festen SCSI-ID kaufen. Bei neueren Geräten erfolgt die Konfiguration meist bequem per Software.

Auch SCSI ist inzwischen in einer seriellen Variante namens SAS (Serial Attached SCSI) verfügbar. Äußerlich ist der Anschluss baugleich mit S-ATA, und anders als beim busförmigen klassischen SCSI handelt sich um ein Punkt-zu-Punkt-Anschlusssystem.

USB und FireWire

Alle modernen Rechner sind mit neuartigen seriellen Schnittstellen für externe Geräte ausgestattet. Gegenüber dem schwierig einzurichtenden, fehleranfälligen SCSI-Anschluss besitzen sie den Vorteil, dass sie völlig ohne Konfigurationsaufwand benutzt werden können.

Der *USB-Anschluss* (Universal Serial Bus) existiert in zwei verschiedenen Versionen: USB 1.0 und 1.1 arbeiten mit einer Datenübertragungsrate von 12 MBit/s, USB 2.0 vervielfacht diesen Wert auf 480 MBit/s. An die USB-Anschlüsse eines Rechners können insgesamt 127 Geräte angeschlossen werden, wozu allerdings sogenannte USB-Hubs als Verteilungsgeräte notwendig sind.

Ende 2008 wurde der neue USB-3.0-Standard spezifiziert. Er soll Datenraten von bis zu 5 GBit/s ermöglichen. Mit dem aktuellen Kernel 2.6.31 ist Linux das erste Betriebssystem, das den neuen Anschluss unterstützt, wenngleich praktisch noch keine entsprechende Hardware verfügbar ist.

Die *FireWire*-Schnittstelle (früher wegen der Markenrechte von Apple meist *IEEE-1394* genannt) arbeitet mit einer Übertragungsrate von 400 MBit/s. Die neuere Variante FireWire 800 unterstützt die doppelte Geschwindigkeit. Der Hauptverwendungszweck dieses Anschlusses ist der digitale Videoschnitt, deshalb wird er auch als DV-Schnittstelle bezeichnet. An den FireWire-Port lassen sich bei entsprechender Vielfältigkeit bis zu 63 Geräte anschließen.

Beide Schnittstellen unterstützen das sogenannte *Hot-Plugging-Verfahren*: Es ist vorgesehen, dass Geräte im laufenden Betrieb ein- und ausgesteckt werden kön-

nen. Das Betriebssystem lädt dann automatisch die passenden Treiber und sollte sie nach Gebrauch auch wieder entfernen.

USB wird nicht nur für Drucker, Modems, Scanner, Digitalkameras, externe Festplatten, CD-Brenner oder die allgegenwärtigen USB-Sticks eingesetzt, sondern zunehmend auch für Tastatur und Maus.

Einige ältere PCs sind neben USB und FireWire noch mit den klassischen seriellen und parallelen Schnittstellen ausgestattet, die allerdings kaum noch eine Bedeutung haben. Noch etwas verbreiteter, aber ebenfalls im Aussterben begriffen sind die PS/2-Anschlüsse. PCs, die diese Anschlüsse nicht aufweisen, werden als »legacy-free« bezeichnet.

PS/2-Anschlüsse werden, falls vorhanden, ausschließlich für Tastatur und Maus verwendet. Zu beachten ist, dass PS/2-Kabel *niemals* im laufenden Betrieb ein- oder ausgestöpselt werden dürfen, dies kann das Mainboard ernsthaft beschädigen!

Die alte serielle Schnittstelle (RS-232 oder V.24) wurde ursprünglich für den Anschluss von Terminals an Kleincomputer konzipiert. Beim PC diente sie lange Jahre als Maus- und Modemanschluss. Heute hat ihre praktische Bedeutung dagegen stark nachgelassen. Am ehesten wird sie vielleicht noch für Messgeräte verwendet, die ihre Daten zur Auswertung an spezielle Software übermitteln. Alte Macs verfügten über eine technisch sehr ähnliche Schnittstelle, die als RS-422 bezeichnet wurde, die allerdings rund war wie eine PS/2-Schnittstelle.

Ein weiterer klassischer Anschluss ist der Centronics-Parallelport (benannt nach einem längst vergessenen Druckerhersteller). Er wurde bis vor einigen Jahren vor allem für Drucker eingesetzt, inzwischen aber fast vollständig durch USB verdrängt.

Drahtlose Schnittstellen

Ein moderner Rechner mit seinen zahlreichen Peripheriegeräten verursacht ein kaum zu bändigendes Durcheinander von Kabeln. Aus diesem Grund wächst die Beliebtheit von Lösungen, die den Anschluss von Geräten ohne Kabel ermöglichen. Grundsätzlich gibt es zwei verschiedene Ansätze: Infrarot- und Funkanschlüsse.

Die Infrarot-Technologie ist von der Fernseh-Fernbedienung bekannt. Sie benötigt Sichtkontakt zwischen Gerät und Empfangsstation und arbeitet verhältnismäßig langsam. Andererseits kann es bei Infrarot-Schnittstellen nicht zu den Störungen kommen, denen Funkverbindungen ausgesetzt sein können. Infrarot-Anschlüsse sind vor allem für Tastaturen und Mäuse geeignet. Der verbreitetste

Standard für Infrarot-Anschlüsse, der auch bei Handys und PDAs gängig ist, wird IrDA genannt.

Funkanschlüsse arbeiten fast immer mit Mikrowellen im Frequenzbereich von 2,4 GHz – dieses Frequenzband hat den Vorteil, dass es lizenzfrei ist, weil es auch von Mikrowellenöfen genutzt wird. Allmählich kristallisiert sich ein einheitlicher Datenfunk-Standard für den Anschluss von Peripheriegeräten heraus: die Bluetooth-Technologie. Es handelt sich um Funkverbindungen mit einer maximalen Reichweite von etwa 10 Metern und einer Übertragungsrate von 1 MBit/s. Es gibt inzwischen Unmengen von Geräten, die über Bluetooth angeschlossen werden können, beispielsweise Tastaturen, Mäuse, Modems oder Mobiltelefone.

Sowohl Infrarot- als auch Bluetooth-Anschlüsse sind nicht auf allen Mainboards eingebaut, aber inzwischen recht günstig als PCI-Erweiterungskarten oder USB-Stecker erhältlich.

Erheblich höhere Datenübertragungsraten bieten übrigens die drahtlosen Netzwerke, die in Kapitel 13, »Netzwerkgrundlagen«, ausführlich behandelt werden.

18.2 Die Linux-Systemarchitektur

Dieser Abschnitt beschreibt die Grundfunktionalität des Betriebssystems Linux. Jedes Thema wird zunächst aus allgemein-funktionaler Sicht beleuchtet, das heißt, Sie erfahren, warum ein Betriebssystem die jeweilige Funktionalität beherrschen muss. Anschließend wird dann auf die Linux-spezifische Implementierung des jeweiligen Themas eingegangen. Für manche Konzepte werden sogar einfache Programmierbeispiele geliefert.

18.2.1 Allgemeiner Aufbau des Systems

Linux besteht wie beinahe jedes neuere Betriebssystem aus dem Kernel, den mehr oder weniger fest zu diesem gehörenden Gerätetreibern, den Systemprogrammen, einer Schnittstelle für Anwendungsprogramme und den diversen Benutzeroberflächen.

Der Kernel

Der *Kernel* (englisches Wort für einen Obstkern) ist das grundlegende Computerprogramm, das unmittelbar auf dem Prozessor des Rechners ausgeführt wird. Er läuft bis zum Herunterfahren des Systems permanent im Hintergrund und steuert alle anderen Betriebssystemkomponenten sowie den Start und den Ablauf der Anwendungsprogramme. Der Kernel initialisiert die Zusammenarbeit mit der

Hardware, indem er die Treiber lädt und koordiniert. Aus einer technischen Perspektive kann man sich vorstellen, dass der Kernel das einzige echte Programm ist, das permanent ausgeführt wird, während alle anderen Programme, die später geladen werden, nur Unterprogramme sind, die vom Kernel geladen und aufgerufen werden und die Kontrolle durch einen Rücksprung wieder abgeben.

Es gibt verschieden konzipierte Kernels. Das ältere Kernel-Konzept ist der sogenannte *monolithische Kernel*, der so viel Funktionalität wie möglich selbst erledigt. Moderner ist das Konzept des *Mikrokernels*, der so wenig wie möglich selbst tut und die meisten Aufgaben an Prozesse delegiert, die im Benutzermodus laufen wie gewöhnliche Anwendungsprogramme.

Mikrokernel lassen sich insgesamt »sauberer« und eleganter implementieren als monolithische Kernel und waren daher lange das Lieblingskind der akademischen Betriebssystemforschung (genau dies war auch der Gegenstand des in Kapitel 1 beschriebenen Streits zwischen Andrew Tanenbaum und Linus Torvalds). Da sie die einzelnen Teile des Betriebssystems nur bei Bedarf im Speicher halten, müssten Betriebssysteme auf Mikrokern-Basis theoretisch schneller und effizienter laufen als Systeme mit monolithischen Kernen. Allerdings wird dabei oft vergessen, dass der ständige Wechsel zwischen Benutzer- und Kernelmodus Zeit und Ressourcen verbraucht.

Linux kann übrigens trotz seines monolithischen Kernels von einem der entscheidenden Vorteile des Mikrokernels profitieren: Die meisten Gerätetreiber sind modular, können also je nach Bedarf geladen und wieder aus dem Speicher entfernt werden. Dies ist besonders wichtig für hot-plugging-fähige Schnittstellen wie USB, FireWire oder Bluetooth.

Ein weiterer Fortschritt ist das *Threading*-Modell, das in immer mehr Betriebssystemen zum Einsatz kommt. Die schwerfälligen Prozesse werden durch eine leichtgewichtige Alternative namens Threads ergänzt, was die Arbeit des Kernels weiter beschleunigt.

Wenn ein Computer eingeschaltet wird, führt das BIOS des Rechners zunächst einige Überprüfungen durch und übergibt die Kontrolle anschließend dem Bootloader eines Betriebssystems; bei Linux meist Grub oder die ältere Variante LILO (siehe Kapitel 9, »Datenträger und Dateisysteme«). Der Bootloader ermöglicht entweder die Auswahl mehrerer Betriebssysteme, die auf den Datenträgern des Rechners installiert sind, oder startet unmittelbar ein bestimmtes System. Das Booten (kurz für *Bootstrapping* – die »Stiefel schnüren«) eines Betriebssystems bedeutet zunächst, dass der Kernel geladen und ausgeführt wird. Dieser erledigt alle weiteren erforderlichen Aufgaben.

Wichtig ist es, die Trennung zwischen Kernelmodus und Benutzermodus zu verstehen. Ein Prozess, der im Kernelmodus läuft, besitzt gewisse Privilegien, die im Benutzermodus nicht gegeben sind. Bei den meisten Computern werden für die beiden Modi unterschiedliche Betriebsmodi des Prozessors selbst verwendet. Beispielsweise besitzen Intel-Prozessoren und Kompatible seit dem 386er vier verschiedene Modi, die sich insbesondere durch einen unterschiedlich starken Schutz vor Interrupts, das heißt Unterbrechungsanforderungen durch Hardware oder bestimmte Programmschritte, voneinander unterscheiden. Der Modus mit dem stärksten Schutz (bei Intel-CPU's 0) wird als Kernelmodus und der mit dem geringsten (Intel: Modus 3) als Benutzermodus verwendet.

Prozesse im Kernelmodus führen wichtige Betriebssystemaufgaben durch, die nicht durch Prozesse im Benutzermodus unterbrochen werden dürfen. Beispielsweise sorgen sie für die eigentliche Verarbeitung von Hardware-Interrupts, das Öffnen und Schließen von Dateien oder die Speicherverwaltung. Auch wenn ein Prozess im Kernelmodus nicht von außen unterbrochen werden kann, kann er freiwillig die Kontrolle an einen anderen Prozess abgeben. In der Regel ruft er den *Task Scheduler* auf, der ebenfalls im Kernelmodus läuft und für die Verteilung der Rechenzeit an die verschiedenen Prozesse zuständig ist.

Eine der hervorstechendsten Neuerungen der Linux-Kernel-Version 2.6 ist übrigens der sogenannte *O(1)-Scheduler*. Die Komplexitätsklasse $O(1)$ besagt, dass die Rechenzeit nicht von der Anzahl der zu bearbeitenden Elemente abhängt – es ist mit anderen Worten egal, wie viele Prozesse zurzeit laufen; der Scheduler benötigt stets in etwa gleich lange (oder besser gesagt kurz), um auszuwählen, welcher als nächster an die Reihe kommt.

Ein Prozess im Benutzermodus kann jederzeit unterbrochen werden, etwa durch einen Hardware-Interrupt, durch einen aufwachenden Kernelprozess oder dadurch, dass er selbst einen Befehl aufruft, der nur im Kernelmodus ausgeführt werden kann. Letzteres sind die sogenannten *Systemaufrufe* (System Calls), die es Programmierern ermöglichen, die eingebauten Funktionen des Betriebssystems zu nutzen. Auf allen UNIX-Systemen werden diese Systemaufrufe vor allem durch große Teile der C-Standardbibliothek bereitgestellt.

Der auffälligste Unterschied zwischen einem modernen System mit ordentlicher Trennung zwischen Benutzer- und Kernelmodus auf der einen Seite und altmodischen Betriebssystemen wie MS-DOS oder dem klassischen Mac OS andererseits besteht darin, dass Letztere nur das veraltete *kooperative Multitasking* verwenden. Ein Prozess entscheidet selbst, wann er die Kontrolle an das Betriebssystem zurückgeben möchte. Stürzt ein Programm ab, das in einem solchen Prozess läuft, dann ist sehr wahrscheinlich das gesamte Betriebssystem instabil gewor-

den. Das in modernen Systemen eingesetzte *präemptive Multitasking* entscheidet dagegen selbst, wie lange Prozesse im Benutzermodus die Rechenzeit behalten dürfen, und entzieht sie ihnen bei Bedarf wieder.

Außerdem besitzen veraltete Betriebssysteme kein richtiges Speichermanagement; Prozesse können gegenseitig auf ihre Speicherbereiche zugreifen und sie versehentlich überschreiben.

Gerätetreiber

Die *Gerätetreiber* (Device Drivers) sind spezielle kleine Programme, die sich um die Steuerung einzelner Hardwarekomponenten kümmern. Unter Linux können weitaus die meisten Treiber (bis auf einige grundlegende, ohne die das System nicht starten kann) wahlweise als fester Bestandteil des Kernels oder aber als Module vorliegen, die sich bei Bedarf laden und wieder aus dem Speicher entfernen lassen.

Es ist eine der schwierigsten Aufgaben für Programmierer, Gerätetreiber zu schreiben. Der Treiber bildet die Schnittstelle zwischen Betriebssystem und Hardware. Er muss die allgemeinen Anforderungen des Betriebssystems an eine bestimmte Geräteklasse in die spezifische Sequenz von Steuerbefehlen umsetzen, die das Gerät eines bestimmten Herstellers versteht, und umgekehrt die Antworten des Geräts wieder in eine allgemein verständliche Form bringen.

Aus der Sicht von Treibern lassen sich zwei grundsätzliche Arten von Geräten unterscheiden. *Zeichengeräte* (Character Devices) tauschen Daten mit ihrer Umgebung als sequenzielle Datenströme aus. Die Daten werden also Zeichen für Zeichen nacheinander ausgelesen. Typische Beispiele sind die Tastatur, ein Drucker oder ein Bandlaufwerk. *Blockgeräte* (Block Devices) stellen sich dagegen ähnlich dar wie der Arbeitsspeicher: Der Zugriff auf den Inhalt des Geräts kann in beliebiger Reihenfolge blockweise erfolgen (Random Access). Zu den Blockgeräten zählen vornehmlich die meisten Laufwerke wie Festplatte oder CD-ROM-Laufwerk, aber auch zum Beispiel Grafikkarten.

Über die Verfügbarkeit – oder auch die gelegentliche Nichtverfügbarkeit – von Gerätetreibern für Linux war bereits in Kapitel 1, »Einführung«, die Rede.

Systemprogramme

Die Bestandteile des Betriebssystems, die nicht zum Kernel gehören, liegen – besonders unter Linux und anderen UNIX-Varianten – als unabhängige Programme vor, die willkürlich geladen, ausgeführt und wieder beendet werden können. Bei einem Betriebssystem mit Konsolenoberfläche müssen Sie die Namen dieser Programme kennen, weil sie durch Eingabe ihres Namens aufgeru-

fen werden. In einer grafischen Benutzeroberfläche werden sie dagegen hinter den Kulissen automatisch aufgerufen, wenn Sie die entsprechenden Menübefehle aufrufen oder Aufgaben per Drag and Drop erledigen, also durch das Ziehen von Symbolen und ihre Ablage an einer bestimmten Stelle.

Typische Systemprogramme wurden in diesem Buch zu Dutzenden vorgestellt – insbesondere in Kapitel 4, »Mit der Shell arbeiten«. Es sind beispielsweise Befehle zur Manipulation von Dateien und Verzeichnissen, etwa für das Umbenennen, Löschen oder Kopieren. Außerdem gehören allerlei Steuerungs- und Analysewerkzeuge dazu. Im nächsten Kapitel finden Sie noch einmal eine systematische Referenz.

Verwechseln Sie Systemprogramme übrigens nicht mit den weiter unten besprochenen Systemaufrufen. Letztere werden von Prozessen in Gang gesetzt, die auf Dienstleistungen des Kernels zugreifen müssen. Viele Systemprogramme verwenden letztendlich Systemaufrufe, um ihre Aufgabe zu erfüllen, aber nicht alle. Ebenso wenig sollten Sie Systemprogramme mit den einfachen Anwendungsprogrammen durcheinanderbringen, die mit Ihrer Distribution geliefert werden. Taschenrechner, Texteditor oder MP3-Player sind keine Systemprogramme, sondern Anwendungen.

Die Schnittstelle für Anwendungsprogramme

Jedes Betriebssystem bietet Anwendungsprogrammen die Möglichkeit, seine Dienstleistungen in Anspruch zu nehmen. Dies ermöglicht es Programmierern, bestimmte aufwendige und hardwareabhängige Aufgaben an das Betriebssystem zu delegieren. Bei den meisten aktuellen Systemen bleibt ihnen auch gar nichts anderes übrig, weil der direkte Zugriff auf die Hardware durch Anwendungsprogramme verhindert wird.

Um eine Funktion des Betriebssystems zu verwenden, muss ein Programm einen *Systemaufruf* (System Call) durchführen. Das Betriebssystem reagiert darauf, indem es den aktuellen Prozess unterbricht, den geforderten Systembefehl im Kernelmodus ausführt und dessen Ergebnis an den aufrufenden Prozess zurückliefert.

UNIX-Systeme wie Linux bieten nur verhältnismäßig wenige, dafür aber sehr mächtige Systemaufrufe an. Einige von ihnen sind auf Dateien und andere Ein- und Ausgabekanäle bezogen, beispielsweise `create()` zum Erzeugen einer neuen Datei, `open()` zum Öffnen, `read()` zum Lesen, `write()` zum Schreiben oder `close()` zum Schließen. Andere Systemaufrufe beschäftigen sich mit der Prozessverwaltung, zum Beispiel erzeugt `fork()` einen neuen Prozess als Kopie des bestehenden, und `kill()` sendet Signale an Prozesse.

Neben den eigentlichen Systemaufrufen basiert jedes Betriebssystem auf der *Bibliothek* der Programmiersprache, in der es geschrieben wurde. Nach wie vor werden die meisten Betriebssysteme zu großen Teilen in der Programmiersprache C geschrieben (nur einige sehr hardwarenahe Teile des Kernels werden in Assembler verfasst). Aus diesem Grund basiert die Arbeitsweise vieler Systembereiche auf Funktionen der *C-Standardbibliothek* – die, wie bereits erwähnt, auch die Systemaufrufe bereitstellt. UNIX, Windows und viele andere Systeme behandeln vieles aus der Sicht von Anwendungsprogrammierern recht ähnlich, weil es mithilfe der entsprechenden Bibliotheksroutinen realisiert wurde.

Ein Beispiel soll an dieser Stelle genügen, um einen Eindruck vom Einfluss der C-Standardbibliothek zu geben: Fast alle Betriebssysteme speichern Datum und Uhrzeit als die Anzahl der Sekunden seit dem 01. Januar 1970, 00:00 Uhr UTC. Dieses Datum wird als *EPOCH* bezeichnet, weil es als der »Epoche machende« Erfindungszeitpunkt von UNIX gilt. Diese Art der Speicherung von Datum und Uhrzeit ist in der C-Bibliotheksdatei *time.h* definiert.

18.2.2 Prozessverwaltung

Jedes moderne Betriebssystem ist in der Lage, scheinbar mehrere Aufgaben gleichzeitig auszuführen. Diese Fähigkeit wird allgemein als *Multitasking* bezeichnet. Es geht dabei nicht nur um den bequemen Nebeneffekt, dass Sie mehrere Anwendungsprogramme geöffnet halten und zwischen ihnen hin- und herschalten können, sondern vor allem um Aufgaben, die das Betriebssystem im Hintergrund erledigen muss, während Sie nur eines der Programme verwenden.

Jede der einzelnen gleichzeitig stattfindenden Aufgaben wird unter Linux und den meisten anderen Betriebssystemen durch einen *Prozess* realisiert. Einem Prozess stehen aus seiner eigenen Sicht alle Ressourcen des Rechners exklusiv zur Verfügung: die gesamte, ungeteilte Rechenzeit des Prozessors, der vollständige Arbeitsspeicher (bis auf den Kernel-Speicher am oberen Ende des virtuellen Adressraums) und der alleinige Zugriff auf sämtliche Ein- und Ausgabekanäle. Es ist Sache des Betriebssystems, die Ressourcen hinter den Kulissen zu verteilen. Ein Prozess, der auf eine Ressource wartet, muss in einen Wartezustand versetzt und später wieder aufgerufen werden.

Dieser Service eines Betriebssystems erleichtert es Anwendungsprogrammierern, sich auf ihre eigentlichen Aufgaben zu konzentrieren. Wenn eine Bedingung eintritt, für die ein bestimmter Prozess nicht zuständig ist, übernimmt das System automatisch die Kontrolle, legt den Prozess »schlafen«, löst das anstehende Problem und ruft den Prozess anschließend wieder auf.

Stellen Sie sich zur Verdeutlichung dieses Sachverhalts vor, Sie wohnen in einem Haus, in dem es für alle Wohnungen nur einen einzigen Klingelknopf an der Haustür gäbe. Ein Druck auf diesen Knopf würde dafür sorgen, dass es in allen Wohnungen klingelt. In diesem Haus müssten alle Mieter auf das Klingeln reagieren und überprüfen, ob es für sie bestimmt ist.

Ähnlich sähe es auf einem Computer aus, wenn es kein Prozessmanagement gäbe: Jedes einzelne Programm müsste sämtliche Bedingungen überprüfen, die auf dem Rechner eintreten können, und keines könnte sich mehr auf seine Tätigkeit konzentrieren.

Das Prozessverwaltungssystem von UNIX und damit auch Linux ist besonders gut verständlich, weil das Konzept recht einfach ist. Für UNIX-Prozesse gelten die folgenden Aussagen:

- ▶ Jeder Prozess ist durch eine eindeutige, ganzzahlige Nummer gekennzeichnet, seine Prozess-ID (PID).
- ▶ Der erste Prozess, der auf dem Rechner gestartet wird, heißt `init`, hat die PID 1 und erzeugt alle anderen Prozesse direkt oder indirekt.
- ▶ Jeder Prozess läuft entweder im Kernelmodus oder im Benutzermodus, und zwar ein für alle Mal. Keiner kann den Modus nachträglich wechseln. Ein Anwendungsprogramm kann niemals selbst einen Prozess starten, der im Kernelmodus läuft – dafür gibt es Systemaufrufe.
- ▶ Ein neuer Prozess wird durch einen speziellen Systemaufruf namens `fork()` erzeugt. Dieser Systemaufruf erzeugt eine identische Kopie des Prozesses, der ihn gestartet hat – der neue Prozess kann sich sogar daran »erinnern«, `fork()` aufgerufen zu haben. Lediglich die PID ist eine andere. In der Regel wird der neue Prozess anschließend für eine neue Aufgabe eingesetzt – beispielsweise, indem durch den Systemaufruf `exec()` ein anderes Programm darin gestartet wird.
- ▶ Jeder Prozess besitzt einen Parent-Prozess. Dabei handelt es sich um denjenigen Prozess, der ihn aufgerufen hat. Wenn der Parent-Prozess vor dem Child-Prozess beendet wird, dann erhält das Child automatisch `init` als neuen Parent-Prozess.
- ▶ Child-Prozesse, die ihre Arbeit beendet haben, werden nicht vollständig beendet. Sie verbleiben als sogenannte *Zombies* im Speicher, bis ihr Parent-Prozess einen Vorgang durchgeführt hat, den man als *Reaping* (deutsch etwa: Ernte) bezeichnet – unter Linux wird es durch den Systemaufruf `waitpid()` realisiert. Der Grund für diesen Vorgang besteht darin, dass der Parent-Prozess die Möglichkeit erhalten soll zu untersuchen, ob seine Children ordnungsgemäß oder mit Fehlern beendet wurden.

Ein gut ausgestatteter Rechner hat auch mit relativ vielen Zombies kein Problem. Wenn der Parent beendet wird und `init` seine Zombie-Children übernimmt, werden diese im Übrigen recht schnell ganz beendet, da `init` regelmäßig `waitpid()` aufruft. Wenn allerdings ein wochenlang laufender Daemon zur Verarbeitung von Anfragen Unmengen von Child-Prozessen startet und wieder beendet, ohne jemals das Reaping durchzuführen, kann dies irgendwann dennoch den Arbeitsspeicher überfluten. Für Programmierer ist es daher eine »heilige Pflicht«, nach dem Beenden von Child-Prozessen `waitpid()` aufzurufen.

- ▶ Jeder Prozess reagiert auf eine Reihe verschiedener *Signale*. Diese Signale sind durchnummeriert, in der Praxis werden jedoch symbolische Namen für diese Signale verwendet, die irgendwo in der Betriebssystembibliothek definiert sind. Signale werden mithilfe des Systemaufrufs `kill()` an einen Prozess gesandt. Der etwas seltsame Name rührt daher, dass das Standardsignal den Prozess auffordert, sich zu beenden, falls kein anderes Signal angegeben wird. Wichtige Signale sind etwa folgende: `SIGTERM` beendet den Prozess normal, `SIGKILL` erzwingt einen sofortigen Abbruch, `SIGHUP` (»Hangup«) weist darauf hin, dass eine Verbindung unterbrochen wurde (etwa eine Netzwerkverbindung), und `SIGALRM` zeigt an, dass ein Timer-Alarm ausgelöst wurde, den Programmierer verwenden können, um einen Prozess nach einer definierten Zeit wieder zu wecken. `SIGCHLD` wird automatisch gesendet, wenn ein Child-Prozess beendet wird, und kann abgefangen werden, um `waitpid()` aufzurufen.
- ▶ Ein Prozess kann jederzeit selbst die Kontrolle abgeben, indem er den Systemaufruf `pause()` durchführt. In diesem Fall kann er durch ein Signal wieder geweckt werden.
- ▶ Prozesse im Benutzermodus können auch von außen unterbrochen und später wieder aufgenommen werden.

Wenn ein Prozess unterbrochen wird, muss der Systemzustand, der zurzeit herrscht, gespeichert werden, um ihn bei Wiederaufnahme erneut herzustellen. Dazu gehören vor allem die Inhalte der Prozessorregister und der Flags sowie eine Liste aller geöffneten Dateien. Wenn ein Prozess weiterläuft, findet er die Systemumgebung also genau so vor, wie er sie verlassen hat.

Neben der Prozess-ID besitzt jeder Prozess auf einem Linux-System eine User-ID (UID) und eine Group-ID (GID). Diese beiden Informationen sind für die Systemicherheit wichtig: Die User-ID kennzeichnet den Benutzer, dem der Prozess gehört, die Group-ID die Benutzergruppe. Ein Benutzer ist entweder eine bestimmte Person oder eine vom Betriebssystem definierte Einheit; einer Gruppe können beliebig viele Benutzer angehören. Ein Prozess reagiert nur auf Signale, die von einem anderen Prozess mit derselben UID und GID aus versandt

wurden. Die einzige Ausnahme sind die UID und GID 0, die dem Superuser oder `root` vorbehalten sind. Dieser spezielle Benutzer darf auf einem UNIX-System alles, also auch jeden Prozess beenden, unterbrechen oder anderweitig steuern.

Mithilfe des in Kapitel 4, »Mit der Shell arbeiten«, besprochenen Befehls `ps` können Sie sich auf der Konsole anzeigen lassen, welche Prozesse gerade laufen. Angezeigt werden die PID, die UID, die GID und der Pfad des Prozesses. Der Pfad ist die genaue Ortsangabe der Programmdatei, die in dem entsprechenden Prozess ausgeführt wird.

Prozesse besitzen den Vorteil, dass sie vollkommen voneinander abgeschirmt laufen können: Sie besitzen beiderseits getrennte Speicherbereiche und können einander nicht in die Quere kommen. Manchmal kann dieser Vorteil jedoch auch ein Nachteil sein, denn mitunter müssen Prozesse miteinander kommunizieren. Eine einfache, aber auf wenige »Wörter« beschränkte Möglichkeit ist die bereits erwähnte Verwendung von Signalen.

Eine andere Option besteht in der Verwendung sogenannter *Pipes*, die die Ausgabe eines Programms und damit eines Prozesses mit der Eingabe eines anderen verknüpfen. Pipes werden beispielsweise in der Konsole häufig eingesetzt, um die Ausgabe eines Programms durch ein anderes zu filtern (siehe Kapitel 4), können aber auch aus Programmen heraus geöffnet werden. Beispiele finden Sie in den entsprechenden konkreten Abschnitten.

Die effizienteste Möglichkeit der Kommunikation zwischen Prozessen heißt *Inter-Process Communication* oder *System V IPC*. Obwohl sie mit System V eingeführt wurde und nicht zum POSIX-Standard gehört, ist sie inzwischen in fast allen UNIX-Varianten verfügbar, zum Beispiel auch unter Linux. Im Wesentlichen verwendet die IPC zwei verschiedene Mechanismen: In sogenannte Nachrichtenwarteschlangen (Message Queues) kann ein Prozess schreiben; ein anderer kann sequenziell daraus lesen. Gemeinsame Speicherbereiche (Shared Memory) sind dagegen einfacher zu handhaben: Was ein Prozess in diesem Speicherbereich ablegt, können andere beliebig oft lesen oder ändern.

Deadlocks

Eines der Probleme, die bei der Verwendung mehrerer Prozesse auftreten können, ist eine Situation, in der mehrere Prozesse im Wartezustand gefangen bleiben, weil sie aufeinander oder auf dieselben Ressourcen gewartet haben. Das Wettrennen um den Zugriff auf Ressourcen wird als *Race Condition* bezeichnet. Zu einem *Deadlock* (einer Verklemmung) kommt es, wenn eine solche *Race Condition* unentschieden ausgeht. Beispielsweise könnten zwei Prozesse in einen *Deadlock* geraten, weil sie den Zugriff auf ein und dieselbe Datei zu sperren ver-

suchen, um anderweitige Änderungen dieser Datei zu verhindern. Ein Deadlock führt mindestens zum Absturz der betroffenen Prozesse, möglicherweise sogar zum Absturz des gesamten Systems.

Ein gutes Betriebssystemdesign vermeidet Deadlocks durch eine Reihe von Verfahren. Insbesondere reicht das normale Verfahren zum Sperren von Ressourcen nicht immer aus, um damit zurechtzukommen. Das gewöhnliche Sperren einer Datei oder einer Hardwareressource überprüft zunächst, ob die Ressource nicht anderweitig gesperrt ist. Falls sie gesperrt ist, wird der Prozess blockiert und wartet, bis die andere Sperre gelöst ist. Anschließend sperrt der aktuelle Prozess selbst die Ressource, sodass andere Prozesse, die sie ihrerseits sperren möchten, wiederum warten müssen.

Statt dieses Modells sollte eine mehrstufige Anmeldung für die Verwendung von Ressourcen eingesetzt werden:

- ▶ Ein Prozess, der eine bestimmte Ressource benötigt, versucht nicht einfach, eine Sperre für diese Ressource zu errichten, sondern überprüft zunächst, ob sie nicht bereits gesperrt ist. Falls doch, gibt er die Kontrolle ab, um nicht aktiv auf das Ende der Sperre warten zu müssen, was Ressourcen kosten würde. Er sollte nach einer gewissen Zeit erneut überprüfen, ob die Ressource noch gesperrt ist.
- ▶ Wenn die Ressource frei ist, errichtet der Prozess eine Sperre, die andere Prozesse daran hindert, diese Ressource zu verwenden.
- ▶ Nachdem der Prozess die Ressource nicht mehr benötigt, löst er die Sperre und gibt die Ressource dadurch wieder frei.

Prozesse aus Programmiersicht

Wie bereits erwähnt, erzeugt das UNIX-Prozessmodell einen neuen Prozess durch den Systemaufruf `fork()`, der eine absolut identische Kopie des ursprünglichen Prozesses erzeugt. Jede Codezeile, die hinter einem Aufruf von `fork()` steht, wird in nicht vorhersagbarer Reihenfolge doppelt ausgeführt. Da der ursprüngliche Prozess (Parent-Prozess) in der Regel etwas anderes tun soll als der neu erzeugte (Child-Prozess), müssen diese beiden irgendwie voneinander unterschieden werden. In diesem Zusammenhang ist es nützlich, dass `fork()` im Parent-Prozess die Prozess-ID (PID) des Child-Prozesses zurückgibt und im Child-Prozess 0. Folglich sehen praktisch alle `fork()`-Aufrufe in UNIX-C-Programmen schematisch so aus:

```
int f;
...
if (f = fork()) {
```

```

...
/* Parent-Prozess:
   Hier werden die Parent-Aufgaben erledigt;
   f enthält PID des Childs.          */
} else {
...
/* Child-Prozess:
   Hier werden die Child-Aufgaben erledigt;
   getpid() liefert PID des Parents.  */
}

```

Um die Forking-Funktionalität in C-Programmen verwenden zu können, müssen Sie die Header-Datei *sys/types.h* einbinden:

```
#include <sys/types.h>
```

Perl übernimmt die `fork()`-Funktionalität der C-Standardbibliothek wortwörtlich. Das folgende Beispiel zeigt ein kleines Perl-Programm, das im Parent- und im Child-Prozess jeweils eine Schleife von 1 bis 10.000 durchzählt und anzeigt. Im Parent-Prozess werden die Werte zur Verdeutlichung ein wenig eingerückt:

```

#!/usr/bin/perl
if (fork()) {
    # Parent-Prozess
    for ($i = 1; $i <= 10000; $i++) {
        print "    $i\n";
    }
} else {
    # Child-Prozess
    for ($i = 1; $i <= 10000; $i++) {
        print "$i\n";
    }
}

```

Wenn Sie das Programm ausführen, werden Sie den Wechsel zwischen ein- und ausgerückten Zahlen bemerken; Sie können die Ausgabe auch genauer untersuchen, indem Sie sie mittels `>Dateiname` in eine Datei umleiten.

Natürlich ist dieses Beispiel nicht besonders sinnvoll. Zu den bedeutendsten Aufgaben von `fork()` gehört die Implementierung von Netzwerkservern, die mit mehreren Clients zur gleichen Zeit kommunizieren. Ein Beispiel finden Sie weiter unten in diesem Kapitel.

Neben `fork()` gibt es noch einige weitere Anweisungen, die im Zusammenhang mit Prozessen nützlich sind:

- ▶ `exec()`, verfügbar in C und Perl, führt das Programm aus, dessen Pfad als Argument angegeben wurde, und kehrt nicht zurück – nachdem das Programm beendet ist, wird auch der Prozess beendet. Dies ist nützlich, um einen mittels `fork()` erzeugten Prozess mit der Ausführung eines externen Programms zu beauftragen.
- ▶ `system()`, ebenfalls in C und Perl einsetzbar, führt das als Argument angegebene Programm aus und kehrt anschließend zurück. Dies wird vorzugsweise für den Aufruf von Systembefehlen genutzt. Beispielsweise können Sie sich in einem UNIX-Programm folgendermaßen den Inhalt des aktuellen Verzeichnisses anzeigen lassen:

```
system ("ls");
```

- ▶ Der bereits in Kapitel 4 beschriebene Backtick-Operator (der Pfad eines Programms in ``) ist nur in diversen Shells wie der Bash und in Perl verfügbar und liefert die Ausgabe des aufgerufenen externen Programms als Wert zurück. Backticks sind extrem nützlich, um die Ausgabe eines Befehls genau zu analysieren – besonders in Zusammenarbeit mit Perls RegExp-Fähigkeiten. Der folgende Perl-Code gibt beispielsweise sämtliche Dateien des ausführlichen Inhaltsverzeichnisses zurück, in denen der Dateiname (letzter Posten in der Zeile) den Buchstaben a enthält:

```
$dir = `ls -l`;
@dirs = split (/\\n/, $dir);
foreach $d(@dirs) {
    print "$d\\n" if $d =~ /a[^\s]+$/;
}
```

Mithilfe von `fork()` können Sie zwar beliebig viele Prozesse erzeugen, aber diese Prozesse laufen völlig beziehungslos nebeneinander. In der bisher vorgestellten Form sind sie also nicht dafür geeignet, mehrgliedrige Aufgaben kooperativ zu erledigen. Da Programme und Tasks in der Praxis jedoch auf vielfältige Weise miteinander interagieren müssen, sind Mittel zur Kommunikation zwischen den verschiedenen Prozessen erforderlich. Weiter oben wurde bereits angedeutet, dass es vielerlei Möglichkeiten dafür gibt, unter anderem Shared Memory, Signale, Semaphoren oder Pipes. Als praktisches Beispiel wird hier die Verwendung von Pipes angesprochen.

Die Verwendung von Pipes zur Kommunikation wird hier am Beispiel von Perl demonstriert, da sie hier besonders leicht und effizient vonstatten geht. Im Grunde gibt es keinen Unterschied zwischen der Anwendung einer Pipe auf der Kommandozeile und aus einem Perl-Programm heraus. In einem Perl-Programm erzeugen Sie eine Pipe zu einem anderen Programm, indem Sie mittels `open` ein

Dateihandle öffnen und statt des üblichen Dateinamens ein lauffähiges Programm angeben. Ein Pipe-Symbol vor dem Programmnamen ("|Programm") bedeutet dabei, dass Sie die Ausgabe Ihres Programms an dieses Programm weitergeben möchten, während ein dahinterstehendes Pipe-Symbol es Ihnen ermöglicht, zeilenweise aus der Ausgabe des externen Programms zu lesen.

Beispielsweise können Sie die Ausgabe Ihres Programms folgendermaßen an den Pager `less` weiterleiten, um Daten bildschirmweise auszugeben:

```
open (AUSG, "|less");
# Ausgabebefehl:
print AUSG "Text...";
```

Auch die Eingabe aus einer Pipe ist unproblematisch. Sie können zum Beispiel folgendermaßen zeilenweise aus der UNIX-Prozesstabelle lesen:

```
open (EING, "ps aux|");
while ($prozess = <EING>) {
    # ... Zeile verarbeiten
}
```

Eine bequemere Möglichkeit, eine Pipe für die Kommunikation zwischen zwei Prozessen innerhalb Ihres Programms einzusetzen, bietet der Systemaufruf `pipe()`. Die gleichnamige Perl-Anweisung erzeugt zwei durch eine Pipe verbundene Dateihandles, deren Namen Sie durch Übergabe entsprechender Argumente festlegen können. Das erste Argument gibt das lesende, das zweite das schreibende Dateihandle an:

```
pipe (READER, WRITER);
```

In dem nachfolgenden kleinen Programm erzeugt der Parent-Prozess in einer `for`-Schleife eine Reihe von Zahlen, die in das Schreibhandle geschrieben werden. Der Child-Prozess liest sie aus dem zugehörigen Lesehandle und berechnet jeweils ihr Quadrat. Besonders sinnvoll ist diese Anwendung nicht, demonstriert aber das Prinzip:

```
#!/usr/bin/perl -w
use strict;
pipe (READER, WRITER);
my $child = fork();
if ($child) {
    # Parent-Prozess liest nur; WRITER schließen
    close WRITER;
    while (my $line = <READER>) {
        chomp $line;
        print ("Das Quadrat von $line ist ").
```

```

        $line * $line. "\n");
    }
} else {
    # Child-Prozess schreibt nur; READER schließen
    close READER;
    for (my $i = 0; $i < 100; $i++) {
        print "Bin bei $i\n";
        print WRITER "$i\n";
    }
}
}

```

Wie Sie bemerken, schließt jeder der beiden Prozesse zu Beginn eines der beiden Pipe-Handles. Es würde auch nichts nützen, das nicht zu tun, da das Pipe-Handle nur in eine Richtung funktioniert, für die Sie sich von vornherein entscheiden müssen. Im vorliegenden Fall behält der Child-Prozess das schreibende Dateihandle offen, der Parent-Prozess das lesende. Wenn die Kommunikation in zwei Richtungen stattfinden soll, müssen Sie über einen zweiten `pipe()`-Aufruf ein weiteres Paar von Dateihandles erzeugen

18.2.3 Threads

Einige Prozesse müssen gemeinsam dasselbe Problem bearbeiten und ununterbrochen miteinander kommunizieren. Dies gilt insbesondere für Prozesse, die nebeneinander im gleichen Anwendungsprogramm laufen. IPC oder andere Methoden der Prozesskommunikation sind zwar möglich, verschwenden aber auf die Dauer Systemressourcen. Interessanter ist eine Prozessvariante, bei der sich mehrere Abläufe von vornherein dieselben Ressourcen teilen.

Zu diesem Zweck werden in vielen Betriebssystemen die leichtgewichtigen und schnell zu wechselnden *Threads* verwendet. Diese besitzen innerhalb desselben übergeordneten Prozesses keine voneinander getrennten Speicherbereiche, sondern greifen auf dieselbe Stelle des Speichers zu. UNIX-Systeme unterstützen Threads in der Regel durch eine Bibliothek namens `pthread` (POSIX-Threads); in Linux können diese seit der Kernel-Version 2.4 als stabil bezeichnet werden.

Threads übernehmen häufig Aufgaben, die parallel innerhalb desselben Programms ausgeführt werden müssen. Besonders anschaulich lässt sich dies anhand eines in Echtzeit laufenden 3-D-Computerspiels erläutern: Gleichzeitig müssen Eingaben zur Steuerung der eigenen Spielfigur entgegengenommen werden, die Umgebung muss ständig neu gezeichnet werden, und es müssen permanente Zustandskontrollen stattfinden. Es wäre für einen Programmierer ein Ärgernis, wenn er sich selbst Gedanken darüber machen müsste, in welcher Reihenfolge die einzelnen Schritte wann stattfinden sollen. Wenn die verschiedenen

Aufgaben dagegen in Threads verpackt werden, führt der Prozessor sie abwechselnd in kurzen Zeitintervallen aus.

18.2.4 Speicherverwaltung

Eine der wichtigsten Aufgaben eines Betriebssystems besteht in der Verwaltung des fast immer zu kleinen Arbeitsspeichers. So gut wie alle Betriebssysteme verwenden eine echte virtuelle Speicheradressierung, bei der die von Programmen angesprochenen Speicheradressen nicht identisch mit den Hardware-Adressen des RAM-Speichers sein müssen.

Genau wie Gerätetreiber und Prozessmanagement entbindet das Speichermanagement einen Programmierer von einer recht frustrierenden Aufgabe, nämlich von der Verteilung des Arbeitsspeichers an die einzelnen Prozesse beziehungsweise Programme. Da eine richtig funktionierende Speicherverwaltung jedem Programm vorgaukelt, ihm stünde der gesamte Arbeitsspeicher zur Verfügung, müssen Sie sich beim Programmieren nicht mehr viele Sorgen machen, ob der Arbeitsspeicher reicht oder nicht.

In der Regel wird der virtuelle Speicherraum vom Betriebssystem in sogenannte *Segmente* unterteilt. Bei modernen Computersystemen beherrscht bereits der Prozessor selbst die Speichersegmentierung und kann dadurch mehr Speicher adressieren, als im physikalischen RAM zur Verfügung steht. Zu diesem Zweck enthalten aktuelle Prozessoren ein Bauteil namens *Memory Management Unit* (MMU). Spricht ein Programm eine bestimmte Speicheradresse an, dann nimmt die MMU sie entgegen und rechnet sie in die aktuell zugeordnete physikalische Speicheradresse um.

Aus der Sicht des Speichermanagements im Betriebssystem wird der Speicher in einzelne *Seiten* unterteilt, die durch das sogenannte *Paging* auf die Festplatte ausgelagert werden, wenn ein Programm sie gerade nicht benötigt, und in den Arbeitsspeicher zurückgeholt werden, wenn es sie wieder braucht. Die Datei, in der sich die ausgelagerten Speicherseiten befinden, wird als *Auslagerungsdatei* (Page File) bezeichnet. UNIX-Systeme verwenden häufig keine einzelne Datei dafür, sondern eine Plattenpartition eines speziellen Typs, die als *Swap Partition* bezeichnet wird.

Die MMU unterhält zu diesem Zweck eine *Seitentabelle*, die zu jedem Zeitpunkt darüber Auskunft gibt, welche virtuelle Speicherseite sich gerade wo befindet, sei es im Arbeitsspeicher oder in der Auslagerungsdatei. Dass eine Speicherseite benötigt wird, die zurzeit ausgelagert ist, wird dabei durch einen *Page Fault* (Seitenfehler) zum Ausdruck gebracht.

Auf einem x86-System (Intel- und AMD-32-Bit Architektur) ist eine Speicheradresse 32 Bit lang. Allerdings werden nicht einfach die verfügbaren physikalischen Speicheradressen durchnummeriert. Stattdessen ist die Adresse in drei Bereiche unterteilt:

- ▶ Die zehn obersten Bits (31 bis 22) geben den Eintrag im *Page Directory* (Seitenverzeichnis) an, verweisen also auf eine Adresse in einem Speicherbereich, der eine Liste von Seitentabellen enthält.
- ▶ Die nächsten zehn Bits (21 bis 12) enthalten die Nummer des Eintrags in der genannten *Page Table* (Seitentabelle). Dieser Eintrag verweist auf eine einzelne Speicherseite.
- ▶ Die letzten zwölf Bits (11 bis 0) geben schließlich den *Offset* an, das heißt das konkrete Byte innerhalb der Speicherseite. Dies führt dazu, dass eine Speicherseite eine Größe von 212 oder 4.096 Byte besitzt.

Auf diese Struktur der Hardware baut die Speicherverwaltung des Betriebssystems auf. Jedes Programm kann dynamisch mehr Speicher anfordern und erhält ihn, indem zurzeit nicht benötigte Speicherseiten ausgelagert werden. Es kommt daher bei einem modernen System nicht oft vor, dass eine Anwendung wegen Speichermangels abgebrochen werden muss oder gar nicht erst startet. Allerdings wird ein Rechner, der zu wenig physikalischen Arbeitsspeicher besitzt, zu langsam, weil er mehr mit dem Paging beschäftigt ist als mit sinnvoller Arbeit.

Da sich allmählich 64-Bit-Architekturen durchsetzen, wird auch die Speicherverwaltung entsprechend angepasst. Linux verwendet schon seit Kernel 2.2 intern ein dreistufiges Paging-Modell: Das Page Directory zeigt nicht gleich auf eine Page Table, sondern zunächst auf ein weiteres Verzeichnis, genannt Middle Directory. Da unter 32-Bit-Architekturen keine Verwendung dafür besteht, wird der Middle-Directory-Eintrag im Page Directory dadurch stillgelegt, dass er immer gleich lautet, also immer auf dasselbe vermeintliche Middle Directory zeigt. Da der Linux-Kernel bereits auf zahlreichen 64-Bit-Prozessoren läuft, ermöglicht dieses Vorgehen die Verwendung desselben Speicherverwaltungsmodells für alle Linux-Versionen.

18.2.5 Die Implementierung des TCP/IP-Stacks

Die Grundlage der TCP/IP-Implementierung im Linux-Kernel – und in allen anderen Systemen, die Netzwerkkommunikation ermöglichen – sind die sogenannten *Sockets*. Die TCP/IP-Kommunikation zwischen zwei Programmen findet stets über zwei Sockets statt, die man als die beiden Enden einer Netzwerkverbindung verstehen kann, vergleichbar mit dem aus Hör- und Sprechmuschel bestehenden Telefonhörer: Wenn eine Netzwerkanwendung einer verbundenen Anwendung

etwas mitteilen möchte, schreibt sie in ihr Socket hinein; die Anwendung an der Gegenstelle kann die Information aus ihrem eigenen Socket lesen. Anders als die weiter oben beschriebenen Pipes ist ein Socket-Paar stets bidirektional, kann also zum Lesen und zum Schreiben gleichzeitig verwendet werden.

Erfreulicherweise ist die Socket-Implementierung schon vor langer Zeit standardisiert worden. An der University of California in Berkeley, wo auch UNIX zu seiner vollen Blüte gereift ist, wurde eine Programmierschnittstelle für die Programmierung von TCP/IP-Netzwerkanwendungen geschrieben, die *Berkeley Socket API*.

Die ursprüngliche Fassung der Berkeley Socket API wurde als Sammlung von Bibliotheken und Header-Dateien für die Programmiersprache C unter BSD-UNIX geschrieben. Genau dieser Umstand sorgt dafür, dass C, UNIX und das Internet bis heute eine untrennbare Einheit bilden. Dennoch sind in der Folgezeit Fassungen der Socket API für viele verschiedene Programmiersprachen und Betriebssysteme geschrieben worden. In Linux wurde sie in Form von Systemaufrufen direkt in den Kernel integriert.

In diesem Unterabschnitt wird die TCP/IP-Implementierung anhand einiger Perl-Skripte vorgestellt, da diese Programmiersprache eine Reihe interessanter Module besitzt, die die Entwicklung von Netzwerkanwendungen erleichtern.

Bevor Sie die Funktionen der Berkeley Socket API in Perl effizient verwenden können, sollten Sie mittels `use Socket` das Modul `Socket` importieren. Zwar sind die eigentlichen Funktionen in den Sprachkern von Perl eingebaut, aber ohne die zahlreichen Konstanten für bestimmte Einstellungen, die in diesem Modul definiert sind, ist ihre Verwendung undenkbar.

Die im Folgenden genannten Funktionen – die in C genauso lauten und ähnliche Parameter haben – dienen dem Einrichten und Konfigurieren von Sockets.

Sockets erzeugen

Der Befehl zur Erzeugung eines Sockets hat die folgende grundlegende Syntax:

```
socket (SOCK, $domain, $type, $proto);
```

Diese Anweisung erzeugt ein neues Socket, dem das Dateihandle `SOCK` für Lese- und Schreiboperationen zugeordnet ist (natürlich können Sie statt `SOCK` einen beliebigen Dateihandle-Bezeichner verwenden).

Der Parameter `$domain` enthält einen Wert für das Einsatzgebiet des Sockets. Die entsprechenden Konstanten werden im Modul `Socket` definiert: `AF_INET` bezeichnet ein Socket für eine TCP/IP-Verbindung, während `AF_UNIX` ein sogenanntes UNIX-Domain-Socket erzeugt. UNIX-Domain-Sockets sind ein beliebtes Mittel

für die Inter-Prozess-Kommunikation auf UNIX-Systemen. Sie verwenden als Kommunikationsbasis keine Netzwerkverbindung, sondern eine pipe-ähnliche Datei im Dateisystem. Im Folgenden werden nur `AF_INET`-Sockets behandelt.

In `$type` wird eine Konstante für den Socket-Typ angegeben. Es existieren die beiden grundsätzlichen Typen `SOCK_STREAM` für ein Stream-Socket und `SOCK_DGRAM` für ein Datagramm-Socket. Stream-Sockets werden in aller Regel für TCP-Verbindungen verwendet, weil sie eine dauerhafte Verbindung zwischen zwei Netzwerkknoten erzeugen. Datagramm-Sockets sind dagegen für die UDP- und ICMP-Netzwerkkommunikation vorgesehen, weil sie verbindungslos sind.

Der vierte Parameter, `$proto`, erwartet schließlich einen Integer, der das zu benutzende Kommunikationsprotokoll angibt. Zwar existieren zur Angabe des Protokolls keine Konstanten, aber dafür können Sie die Funktion `getprotobyname()` verwenden. Genau wie die gleichnamige C-Bibliotheksfunktion beziehungsweise der entsprechende UNIX-Systemaufruf wandelt sie ein durch seinen Namen angegebenes Protokoll in die entsprechende Nummer um, indem sie es in der Standarddatei `/etc/protocols` nachschlägt. Für UNIX-Domain-Sockets wird dagegen die Konstante `PF_UNSPEC` (»kein bestimmtes Protokoll«) angegeben.

Die folgenden beiden Beispielzeilen erzeugen ein TCP-Stream-Socket beziehungsweise ein UDP-Datagramm-Socket:

```
socket (TCP_SOCKET, AF_INET, SOCK_STREAM,
        getprotobyname ('tcp'));
socket (UDP_SOCKET, AF_INET, SOCK_DGRAM,
        getprotobyname ('udp'));
```

Adressen und Ports

Wie die IP-Adressierung funktioniert und was TCP- und UDP-Ports zu bedeuten haben, wurde in Kapitel 13 ausführlich erläutert. Für die Socket-Programmierung müssen Adressen und Ports in speziellen Strukturen gespeichert werden, die der C-Standardbibliothek entstammen. Zu diesem Zweck werden verschiedene Bibliotheksfunktionen zur Verfügung gestellt, die der Umwandlung von Hostnames, Adressen und Ports in das korrekte Format dienen:

- ▶ `gethostbyname ($hostname)`

Diese Funktion wandelt den angegebenen Hostname in die entsprechende IP-Adresse um. Damit dies korrekt funktioniert, benötigen Sie Zugriff auf einen Namensdienst, der in der Lage ist, den entsprechenden Hostname aufzulösen, im Internet also beispielsweise auf einen DNS-Server. Das Ergebnis

ist eine Struktur, die den Hostname und die in ASCII-Zeichen gepackte IP-Adresse enthält.

- ▶ `inet_aton ($host_oder_ip)`
Mithilfe dieser Funktion wird ein Hostname oder eine IP-Adresse in dezimaler Vierergruppenschreibweise (zum Beispiel 192.168.0.9) in vier ASCII-Zeichen verpackt. Viele Socket-Funktionen benötigen diese Schreibweise. `inet_ntoa ($pack_adr)` erledigt übrigens die umgekehrte Aufgabe und wird mitunter benötigt, um die mit einem Datenpaket empfangene Adresse des Gegenübers zu entschlüsseln.
- ▶ `sockaddr_in ($port, $pack_adr)`
Die Funktion `sockaddr_in` macht aus einer Port-Angabe und einer mittels `inet_aton` gepackten Adresse eine speziell verpackte Host/Port-Struktur, die Sie für die Angabe der Gegenstelle bei einer Netzwerkverbindung verwenden können. Das folgende Beispiel erstellt eine solche Struktur für den Zugriff auf Port 80 (den HTTP-Server) auf dem Host *www.galileocomputing.de*:

```
$webserver = sockaddr_in
    (80, inet_aton ('www.galileo-computing.de'));
```

Interessanterweise ist `sockaddr_in` gleichzeitig in der Lage, die umgekehrte Umwandlung vorzunehmen: Wenn Sie ein gepacktes Host/Port-Paar aus einem Datenpaket erhalten (siehe unten), können Sie es folgendermaßen entpacken:

```
($port, $host) = sockaddr_in ($pack_port_host);
```

- ▶ `getservbyname ($service, $proto)`
Statt der numerischen Angabe des Ports können Sie die sogenannten Well-known Ports mit Nummern bis 1023, die bei gängigen Serverdiensten eingesetzt werden, auch durch ihren Namen angeben und per `getservbyname` umwandeln. Als zweites Argument benötigen Sie dazu das Transportprotokoll ('tcp' oder 'udp'). Zum Beispiel können Sie folgendermaßen eine Verbindung zu dem FTP-Server *ftp.uni-koeln.de* herstellen:

```
$port = getservbyname ('ftp', 'tcp');
$ftpserver = sockaddr_in
    ($port, inet_aton ('ftp.uni-koeln.de'));
```

Verbindungen herstellen, Daten senden und empfangen

Was als Nächstes mit dem Socket geschieht, hängt zum einen davon ab, ob Sie ein Stream- oder ein Datagramm-Socket erzeugt haben und, falls es ein Stream-Socket ist, ob Sie gerade einen Client oder einen Server schreiben.

UDP-Datagramm-Sockets können Sie nach dem Erstellen ohne weitere Formalitäten verwenden, um einfache Nachrichten zu versenden oder von anderen UDP-Anwendungen zu empfangen. Dazu werden die Anweisungen `send()` und `recv()` verwendet:

- ▶ `send (SOCK, $data, 0, $receiver);`
Diese Anweisung sendet den Dateninhalt `$data` über das Socket `SOCK` an den Empfänger `$receiver`. Der Empfänger wird als `sockaddr_in`-Struktur angegeben (siehe oben). Der dritte Parameter ist normalerweise 0; er kann in sehr seltenen Fällen einen speziellen Wert für besondere Routing-Einstellungen enthalten.
- ▶ `recv (SOCK, $data, $maxlength, 0);`
Die Anweisung `recv()` empfängt ein UDP-Datagramm. Der gelesene Wert wird in der Variablen `$data` gespeichert, während das Funktionsergebnis den Absender des Datagramms angibt. Von den meisten UDP-Servern können Sie Daten nur dann empfangen, wenn Sie zuvor mittels `send()` eine Anfrage an sie geschickt haben.

Auf UDP-Anwendungen wird hier nicht weiter eingegangen. Bei TCP-Sockets sieht die Angelegenheit komplizierter aus: Wenn Sie einen Server programmieren, der eine Dienstleistung zur Verfügung stellt, müssen Sie das Socket zum lauschenden (listening) Socket machen. Schreiben Sie dagegen einen Client, müssen Sie als Nächstes eine Verbindung zu einem Server-Socket herstellen.

Beispielsweise können Sie folgendermaßen eine Verbindung zu dem Webserver www.galileodesign.de herstellen und ihm eine GET-Anfrage schicken, anschließend werden sämtliche zurückgegebenen Zeilen gelesen und am Bildschirm angezeigt:

```
#!/usr/bin/perl -w
use Socket;
use IO::Handle;
$port = getservbyname ('http', 'tcp');
$dest = sockaddr_in
    ($port, inet_aton ('www.galileodesign.de'));
socket (SOCK, AF_INET, SOCK_STREAM,
    getprotobyname ('tcp'));
connect (SOCK, $dest);
# Socket in den Autoflush-Modus schalten:
SOCK->autoflush (1);
# HTTP-Anfrage für Startseite senden:
print SOCK "GET / HTTP/1.0\n\n";
# Antwort des Servers zeilenweise ausgeben:
while ($line = <SOCK>) {
```

```

    chomp $line;
    print "$line\n";
}

```

Dieses Programm übernimmt tatsächlich die Hälfte der Funktionalität eines Browsers: Es richtet eine HTTP-Anfrage an einen Webserver und liest das erhaltene Dokument. Der kompliziertere Teil eines Browsers ist freilich nicht enthalten, nämlich derjenige, der den erhaltenen HTML-Code interpretiert und ein formatiertes Dokument daraus macht. Stattdessen wird die gesamte Ausgabe des Servers einfach angezeigt, und zwar nicht nur das eigentliche HTML-Dokument (die Startseite von Galileo Design), sondern auch sämtliche HTTP-Header.

In diesem Programm werden einige neue Funktionen verwendet:

- ▶ `connect (SOCK, $dest);`
Diese Anweisung stellt eine Verbindung zu einem TCP-Server her, der mithilfe der gepackten Adresse `$dest` angegeben wird. Anschließend können Sie das Socket wie ein Dateihandle verwenden, um Daten daraus zu lesen und hineinzuschreiben.
- ▶ `SOCK->autoflush(1);`
Die Funktion `autoflush` ist im Modul `IO::Handle` definiert, das aus diesem Grund mittels `use` eingebunden wird. Sie sorgt dafür, dass Schreiboperationen in das Socket (beziehungsweise ein beliebiges Dateihandle) unmittelbar abgeschickt und nicht gepuffert werden, bis mehr Daten zusammenkommen. Wenn Sie die Zeile weglassen, bleibt das Programm einfach kommentarlos stehen.

Die restlichen Socket-Funktionen wurden bereits weiter oben erläutert. Die eigentliche Kommunikation mit dem Server erfolgt über die Funktion `print` und den Operator `<>`; die Verwendung dieser Anweisungen mit Dateihandles wurde bereits in Kapitel 17, »System-Automatisierung«, erläutert.

Interessant ist noch die eigentliche GET-Anfrage, die an den Webserver geschickt wird:

```
GET / HTTP/1.0
```

Es wird also die Startseite der Website (`/`) über das Protokoll HTTP 1.0 angefordert. In diesem Zusammenhang können Sie die aktuelle Variante HTTP 1.1 nicht verwenden, weil sie einen Host-Header erwartet. Die Anfrage muss, wie hier gezeigt, durch zwei Zeilenumbrüche abgeschlossen werden. Genauer über HTTP steht im ersten Abschnitt von Kapitel 15, »LAMP«.

Für einen Server sind zusätzlich die folgenden Anweisungen wichtig:

- ▶ `bind (SOCK, $addr);`

Diese Anweisung bindet ein zuvor erzeugtes Socket an eine (lokale) `sockaddr_in`-Adresse. Sie können jede Ihrer lokalen Schnittstellenadressen, die Loop-back-Adresse (127.0.0.1) oder die Konstante `INADDR_ANY` (für jede beliebige Adresse) verwenden.

Noch wichtiger als die Adresse selbst ist übrigens der Port, den Sie mittels `sockaddr_in` in die Struktur einfügen, weil es der spezifische Port ist, auf dem der Server lauschen soll. Beachten Sie, dass Sie unter UNIX `root`-Rechte benötigen, um einen der Well-known Ports unter 1024 anzusprechen. Darüber hinaus ist es ein erhebliches Sicherheitsrisiko, ein selbstgeschriebenes und nicht sicherheitsoptimiertes Programm auf einem solchen Port laufen zu lassen, den die Schnüffel-Tools potenzieller Angreifer standardmäßig auf Angriffsmöglichkeiten überprüfen.

Das folgende Beispiel bindet ein Socket an jede beliebige Adresse des Servers und lauscht an Port 11.111:

```
bind (SOCK, sockaddr_in (11111, INADDR_ANY));
```

- ▶ `listen (SOCK, $max_queue);`

Die Anweisung `listen` benötigt ein mittels `bind` gebundenes Socket und wandelt es in ein lauschendes Socket um: Dieses Socket wartet auf den Verbindungsversuch eines Clients mittels `connect`, wie er weiter oben erläutert wurde. Das Argument `$max_queue` gibt die maximale Größe der Warteschlange dieses Sockets an, bestimmt also, wie viele Client-Verbindungen darauf warten können, akzeptiert zu werden. Normalerweise wird hierfür die Konstante `SOMAXCONN` verwendet, die den maximalen Wert enthält, den das Betriebssystem zulässt.

- ▶ `$remote_addr = accept (SOCK_CONN, SOCK);`

Sobald auf dem lauschenden Socket eine Verbindung eingeht, wird die nächste Programmzeile ausgeführt – normalerweise enthält sie einen solchen `accept`-Befehl. `SOCK_CONN` ist ein neues, mit dem Client verbundenes Socket, das durch diese Anweisung erzeugt wird. Die gepackte Adresse des Clients steht im Funktionsergebnis.

Das Modul `IO::Socket`

Eine große Erleichterung für Perl-Programmierer ist die Verwendung des Moduls `IO::Socket` statt der bisher behandelten Bibliotheksbefehle. Das Modul stellt bequeme Funktionen zur Verfügung, die die meisten komplexen Umwandlungsbefehle überflüssig machen. Beachten Sie, dass die Anweisungen, die in diesem

Unterabschnitt behandelt werden, nicht kompatibel zur C-Standardbibliothek und zur ursprünglichen Berkeley Socket API sind.

Eingebunden wird das Modul folgendermaßen:

```
use IO::Socket;
```

Beachten Sie, dass das Modul von `IO::Handle` abgeleitet ist und deshalb bereits die oben erwähnte Funktion `autoflush` beinhaltet. Die `IO::*`-Module verfolgen alle einen objektorientierten Ansatz, der hier einfach intuitiv verwendet wird. Auf die objektorientierte Programmierung mit Perl geht dieses Buch dagegen aus gutem Grund nicht ein – sie ist ziemlich komplex und umständlich und lässt sich nicht mit der Eleganz von Java-Klassen messen.³

Mithilfe der `IO::Socket`-API wird ein neues Socket folgendermaßen eingerichtet:

```
$sock = IO::Socket::INET->new (@args);
```

Beachten Sie, dass `$sock` formal kein Dateihandle ist, sondern eine Referenz auf ein Socket, die von der entsprechenden Funktion zurückgegeben wird. Wenn Sie die strenge Syntaxüberprüfung (`use strict`) verwenden, müssen Sie diese Variable also mittels `my` deklarieren. In allen anderen Belangen funktioniert die Referenz wie ein gewöhnliches Socket. Die Funktion `IO::Socket::INET->new` erzeugt übrigens nur Sockets vom Typ `AF_INET`; für UNIX-Domain-Sockets müssen Sie dagegen `IO::Socket::UNIX->new` aufrufen.

Die Argumentliste `@args` gibt die Eigenschaften des Sockets als Name/Wert-Paare an. Die wichtigsten dieser Einstellungen sind folgende:

- ▶ `PeerAddr` gibt die Adresse des entfernten Hosts an. Sie können einen Hostname oder eine IP-Adresse verwenden. Wenn Sie `PeerAddr` angeben, wird nach der Erzeugung des Sockets automatisch `connect()` aufgerufen, es handelt sich also um eine Eigenschaft für ein Client-Socket.
- ▶ `PeerPort` ist der entfernte Port. Alternativ können Sie ihn auch, durch Doppelpunkt getrennt, bei `PeerAddr` angeben.
- ▶ `LocalAddr` gibt die lokale Adresse an.
- ▶ `LocalPort` ist der lokale Port. Wenn Sie nur `LocalPort`, aber nicht `LocalAddr` angeben, wird für die Adresse automatisch `INADDR_ANY` eingetragen – in der Regel genau das, was Sie benötigen.
- ▶ `Proto` ist der Name oder die Nummer des zu verwendenden Protokolls, also `'tcp'` oder `'udp'`.

³ Deshalb wurde die gesamte Objektorientierung in der kommenden Perl-Version 6 grund-erneuert.

- ▶ `Listen` gibt die Größe der Lauswarteschlange an. Wenn dieses Argument vorhanden ist, wird darüber hinaus automatisch `listen()` aufgerufen, sodass das Socket zum lauschenden Socket wird.
- ▶ `Type` gibt den Socket-Typ an, also eine der Konstanten `SOCK_STREAM` oder `SOCK_DGRAM`.

Das folgende Beispiel erzeugt ein typisches TCP-Client-Socket, das eine Verbindung zum FTP-Server der Uni Köln herstellt:

```
my $sock = IO::Socket::INET->new
    (Proto    => 'tcp',
     PeerAddr => 'ftp.uni-koeln.de',
     PeerPort => 'ftp');
```

Um dagegen ein lauschendes Socket einzurichten, das für einen Server verwendet werden kann, können Sie beispielsweise folgenden Befehl benutzen:

```
my $listen = IO::Socket::INET->new
    (Proto    => 'tcp',
     LocalPort => 22222,
     Listen   => SOMAXCONN);
```

Dieses Socket lauscht also auf Port 22.222 an jeder beliebigen Adresse des Hosts und lässt so viele wartende Verbindungsversuche zu, wie das Betriebssystem erlaubt.

18.2.6 Netzwerkprogrammierung – ein praktisches Beispiel

Mit dem Wissen aus dem vorigen Unterabschnitt ist es eigentlich kein Problem, beliebige Server und Clients in Perl zu programmieren. Wenn Sie wohlbekannte Protokolle implementieren möchten, müssen Sie die entsprechenden RFCs lesen und Punkt für Punkt abarbeiten. Für das Buch »Apache 2« habe ich zur Verdeutlichung des Grundprinzips einen kleinen Webserver geschrieben, der einen winzigen Ausschnitt der HTTP-Spezifikation erfüllt – die einfache Auslieferung statischer Dokumente funktioniert unproblematisch. Beschreibung und Downloadmöglichkeit finden Sie unter <http://buecher.lingoworld.de/apache2/server.html>.

Der Server

Das folgende Beispiel implementiert einen einfachen Server, der nach einer Client-Anmeldung jede Anfrage mit der Anzahl der bisherigen Anfragen dieses Clients sowie mit Datum und Uhrzeit beantwortet. Falls der Dateninhalt der Anfrage

"reset" lautet, wird die Anzahl zurückgesetzt; bei "exit" wird die Verbindung beendet. Hier zunächst der Sourcecode:

```
#!/usr/bin/perl -w

use strict;
use IO::Socket;

# Lauschendes Socket erzeugen
my $listen = IO::Socket::INET->new
    (Proto    => 'TCP',
     LocalPort => 11111,
     Listen   => SOMAXCONN);

print "Server lauscht an Port 11111 ...\n";

# Logdatei zum Anhängen öffnen
open LOG, ">>log.txt";

# Accept-Schleife
while (1) {
    # Keine Anfrage? Nächster Durchlauf!
    next unless my $conn = $listen->accept();

    # Verbindungsaufbau - Child-Prozess erzeugen
    my $child = fork;
    if ($child == 0) {
        # Im Child-Prozess? Verbindung verarbeiten
        handle_conn ($conn);
        # Wieder da? Child beenden
        exit (0);
    }
}

sub handle_conn {

    my $anzahl = 0;

    my $sock = shift;
    my $peer = $sock->peerhost;

    print " Connection from $peer\n";

    # Anfragen lesen
    while (my $request = <$sock>) {
```

```

$request =~ s/\s*$//;
# Beenden bei "exit"
last if $request eq "exit";
# reset?
$anzahl = 0 if $request eq "reset";
# Datum und Uhrzeit
my $now = scalar localtime;
# Anzahl erhöhen
$anzahl++;
# Ausgabe
print $sock "${anzahl}. Anfrage: $now\n";
# Log-Eintrag
print LOG "${peer}: \"${request}\" ($now)\n";
}
print $sock "Bye.\n";
print " Connection closed by $peer\n";
}

```

Das lauschende Socket wird nach dem weiter oben gezeigten Schema erstellt; der Server lauscht an Port 11111. Die `accept()`-Schleife springt jeweils sofort mit `next()` zum nächsten Durchgang, falls kein Verbindungswunsch eines Clients eintrifft. Ist dies der Fall, wird zunächst mittels `fork()` ein Child-Prozess erstellt, der sich exklusiv um den neuen Client kümmert. Auf diese Weise kann der Server mehrere Clients gleichzeitig bedienen. Dieses einfache Nebenläufigkeitsmodell wird als Forking-Server bezeichnet; für Hochleistungsserver ist es nicht geeignet, da das Erstellen neuer Prozesse im laufenden Betrieb zu lange dauert.

Der Child-Prozess ruft die Subroutine `handle_conn()` auf. Sie liest die Client-Anfragen zeilenweise aus dem akzeptierten Socket; das Ersetzen von Whitespace am Ende – `s/\s*//` – ist wegen der unterschiedlichen Zeilenumbrüche sicherer als `chomp()`. Der Server reagiert je nach Befehl auf die Anfragen, wie oben dargestellt. Bei "exit" wird die Schleife mit `last()` beendet.

Wie es sich für einen Server gehört, führt er eine Logdatei über alle Client-Zugriffe. Jeder Eintrag hat das Format Client-IP: "Anfrage" (Datum/Uhrzeit). Beispiel:

```
127.0.0.1: "reset" (Sun Jan 13 23:02:16 2008)
```

Der Client

Der Client ist noch trivialer als der Server und soll vor allem demonstrieren, dass es üblich ist, hinter den Kulissen andere Befehle zu verwenden als die Benutzer-eingaben. Er akzeptiert eine Eingabe nach der anderen, sendet sie – eventuell in modifizierter Form – an den Server und gibt dessen Antworten aus. Nach der

Eingabe von 'x' – dem Server-Befehl "reset" – wird er beendet. Hier das kurze Listing:

```
#!/usr/bin/perl -w

use strict;
use IO::Socket;

# Host von Kommandozeile oder localhost
my $host = $ARGV[0] || "localhost";

# Server-Verbindung
my $sock = IO::Socket::INET->new
    (Proto    => 'tcp',
     PeerAddr => $host,
     PeerPort => 11111);

print "<0> zurücksetzen, <X> beenden, <Enter> Info.\n";

# Kommunikations-Schleife
while (1) {
    print "> ";
    my $eingabe = <>;
    chomp $eingabe;

    # Je nach Eingabe Befehle an den Server senden

    if ($eingabe eq '0') {
        # Zähler zurücksetzen
        print $sock "reset\n";
    } elsif ($eingabe =~ /^x$/i) {
        # Verbindung beenden
        print $sock "exit\n";
    } else {
        # Normal: Anzahl/Datum erfragen
        print $sock "$eingabe\n";
    }

    # Serverantwort lesen
    my $antwort = <$sock>;
    $antwort =~ s/\s*$//;

    print "Antwort von $host: $antwort\n";

    # Beenden?
```

```
last if $eingabe =~ /^x$/i;  
}
```

18.3 Zusammenfassung

In diesem Kapitel haben Sie einige der Konzepte kennengelernt, die Ihr Linux-System einsetzt, um Ihnen seine erwiesenermaßen guten und stabilen Dienste zu leisten. Dies war natürlich nur ein erster Einblick; ausführlichere Informationen erhalten Sie beispielsweise in den eingangs empfohlenen Kernel-Büchern.

Um den Kernel zu verstehen, müssen Sie zuerst die grundlegende Funktionsweise der Hardware nachvollziehen können. Im ersten Abschnitt dieses Kapitels wurden daher die Funktionsgrundlagen von Prozessor, Arbeitsspeicher und Schnittstellen vorgestellt.

Der Linux-Kernel selbst verwirklicht alle Anforderungen, die an ein modernes Betriebssystem gestellt werden: Prozessverwaltung, Speichermanagement, eine leistungsfähige Schnittstelle für Anwendungsprogramme, effiziente Systemprogramme sowie eine eingebaute, standardisierte Netzwerkschnittstelle. Alle diese Konzepte wurden hier theoretisch erläutert, einige auch aus Programmierersicht.

Wer klare Begriffe hat, kann befehlen.
– Johann Wolfgang Goethe

19 Referenz der Shell- und Systembefehle

In diesem letzten Kapitel erhalten Sie noch einmal einen Überblick über die wichtigsten Kommandos zum schnellen Nachschlagen. Die allermeisten von ihnen wurden irgendwo anders in diesem Buch bereits verwendet. Einige wenige habe ich allerdings noch hinzugefügt, weil sie sich bisher sonst nirgendwo unterbringen ließen, mir aber dennoch wichtig erschienen. Die Überschriften dieser Kommandos sind mit einem Sternchen (*) gekennzeichnet.

19.1 ls – ein Kommando im Detail

Der Platz in diesem Kapitel und in diesem Buch reicht bei Weitem nicht, um jedes einzelne Kommando mit allen verfügbaren (oder auch nur der Hälfte oder einem Drittel) seiner Optionen zu besprechen. Wenn Sie diese Detailfülle benötigen, können Sie die betreffenden Manpages oder GNU info-Seiten lesen. Alternativ steht eine recht ausführliche Kommandoreferenz auch in Buchform zur Verfügung: »Linux in a Nutshell«, 4. Auflage von Ellen Siever et al., Köln 2005, O'Reilly Verlag.

Exemplarisch möchte ich aber dennoch ein Kommando mit allen zur Verfügung stehenden Optionen vorstellen, um Ihnen einen Eindruck von der Leistungsfähigkeit und Flexibilität der Shell-Kommandos zu geben. Die Wahl fiel auf `ls`, weil es einer der nützlichsten und wahrscheinlich der am häufigsten eingetippte Befehl ist.

ls

Inhalte der angegebenen Verzeichnisse auflisten

Syntax

```
ls [Optionen] [Muster [Muster ...]]
```

Die Muster werden – je nachdem, was im entsprechenden Kontext vorhanden ist – als Datei- oder Verzeichnismuster interpretiert. Falls kein Muster angegeben wird, gibt `ls` den Inhalt des aktuellen Verzeichnisses aus.

Optionen

- ▶ `-a, --all`
Listet auch Einträge auf, die mit `.` beginnen.
- ▶ `-A, --almost-all`
Wie `-a`, aber zeigt die automatischen Verzeichnisnamen `.` und `..` nicht an.
- ▶ `--author`
Zeigt den Benutzer an, der die Datei erzeugt hat (nur mit `-l`).
- ▶ `-b, --escape`
Oktale Escape-Sequenzen für Sonderzeichen
- ▶ `--block-size=BYTES`
Blockgröße in Byte für die Anzeige der Dateigröße (nur mit `-l`). Sie können eines der Kürzel `K`, `M` oder `G` an den Wert anhängen, um ihn in Kilobyte, Megabyte beziehungsweise Gigabyte anzugeben. `--block-size=1024` oder `--block-size=1K` zeigt die Größen beispielsweise in KByte an.
- ▶ `-B, --ignore-backups`
Keine Dateien anzeigen, deren Name mit `~` beginnt (typische automatische Sicherheitskopien)
- ▶ `-c`
Bei Verwendung mit `-l` wird eine `ctime(3)`-formatierte Anzeige des Änderungsdatums hinzugefügt. Die Option allein sortiert nur nach diesem Kriterium; mit `-lt` wird die Information angezeigt, und es wird danach sortiert.
- ▶ `-C`
Mehrspaltige Anzeige, Sortierung in Spalten
- ▶ `--color[=never|always|auto]`
`ls` verwendet Farben zur Unterscheidung der Dateitypen – optional kann einer der Werte `never` (keine Farbe, zum Überschreiben eines eventuellen Standards), `always` (immer einfärben) oder `auto` (automatisch, der Standardwert) angegeben werden.
- ▶ `-d, --directory`
Zeigt nicht die Inhalte, sondern nur das aktuelle Verzeichnis selbst (`.`) an. Zusammen mit `-l` ergibt dies eine Statistik, die zum Beispiel die Anzahl der Dateien im Verzeichnis anzeigt, wobei symbolische Links nicht mitgezählt werden.

- ▶ `-D, --dired`
Zusammen mit `-l` werden die nötigen Zusatzangaben für den Emacs-Modus `dired` (siehe Kapitel 10, »Klassische Texteditoren«) erzeugt.
- ▶ `-f`
Kurzfassung für `-aU`, Weglassen von `-lst` und ohne Sortierung
- ▶ `-F, --classify`
Hinter den Dateinamen Typsymbole einblenden: `*` (ausführbar), `/` (Verzeichnis), `=` (Socket), `@` (Symlink) oder `|` (Named Pipe); normale Dateien werden nicht gekennzeichnet.
- ▶ `--format=FORMATNAME`
Ausführliche Schreibweise für diverse Formatierungen: `across` oder `horizontal` erzeugt eine mehrspaltige Anzeige wie `-x`, `commas` zeigt die Dateinamen wie `-m` als komma-separierte Liste an, `long` oder `verbose` entsprechen `-l` (detaillierte Anzeige), `single-column` stellt die Einträge untereinander dar (Kürzel `-l`), und `vertical` sortiert die Dateien mehrspaltig untereinander wie `-C`.
- ▶ `--full-time`
Ausführliche Zeitangaben, entspricht `-l --time-style=full-iso`.
- ▶ `-g`
Ausführlich (wie `-l`) ohne Besitzerangabe
- ▶ `-G, --no-group`
Ausführlich (wie `-l`) ohne Gruppenangabe
- ▶ `-h, --human-readable`
Zusammen mit `-l` werden Größenangaben nicht in Byte angezeigt, sondern je nach Größe mit den Kürzeln `K` (Kilobyte), `M` (Megabyte) beziehungsweise `G` (Gigabyte) versehen. Die Variante `--si` (steht für Systeme International, die internationale Norm der physikalischen Maßeinheiten) verwendet die Basis 1.000 statt 1.024.
- ▶ `-H, --dereference-command-line`
Die auf der Kommandozeile angegebenen symbolischen Links auflösen
- ▶ `--dereference-command-line-symlink-to-dir`
Die auf der Kommandozeile angegebenen symbolischen Links auf Verzeichnisse auflösen
- ▶ `--hide=Datei-|Verzeichnismuster`
Alle Dateien ausblenden, die dem angegebenen Shell-Dateimuster entsprechen (wird durch `-a` oder `-A` aufgehoben)

- ▶ `--indicator-style=none|classify|file-type`
Andere Schreibweise für Dateitypsymbole: `classify` entspricht der Option `-F`, `file-type` wirkt wie `-p` (siehe unten); `none` blendet die Symbole aus.
- ▶ `-i, --inode`
Die `inode`-Nummer der Dateien wird zusätzlich ausgegeben.
- ▶ `-I, --ignore=Shell-Muster`
Dateien, deren Name dem angegebenen Muster entspricht, werden nicht angezeigt; im Gegensatz zu `--hide` wird diese Option nicht von `-a` oder `-A` überschrieben.
- ▶ `-k`
Kurzfassung für `--block-size=1K`
- ▶ `-l`
Ausführliche Ausgabe
- ▶ `-L, --dereference`
Informationen über die Zieldateien symbolischer Links statt über die Links selbst anzeigen
- ▶ `-m`
Die Dateinamen in einer durch Komma getrennten Liste ausgeben
- ▶ `-n, --numeric-uid-gid`
Ausführlich wie `-l`, aber Benutzer- und Gruppen-IDs werden numerisch ausgegeben.
- ▶ `-N, --literal`
Namen ohne Berücksichtigung von Steuer- oder Sonderzeichen ausgeben
- ▶ `-o`
Wie `-l` ohne Angabe der Group-IDs (entspricht `-lG`)
- ▶ `-p, --file-type`
Dateitypsymbole an Dateinamen anhängen: `/` (Verzeichnis), `=` (Socket), `@` (Symlink) oder `|` (Pipe)
- ▶ `-q, --hide-control-chars`
Nicht druckbare Zeichen in Dateinamen durch `?` ersetzen
- ▶ `--show-control-chars`
Nicht druckbare Zeichen normal anzeigen (Standard)
- ▶ `-Q, --quote-name`
Dateinamen in doppelte Anführungszeichen setzen - praktisch, wenn Sie den Verdacht haben, dass ein Dateiname Leerzeichen enthält.
- ▶ `--quoting-style=Zeichenoptionen`
Stellt die Dateinamen im angegebenen Format dar: `literal` (siehe `-N`), `locale`

- ▶ `--indicator-style=none|classify|file-type`
Andere Schreibweise für Dateitypsymbole: `classify` entspricht der Option `-F`, `file-type` wirkt wie `-p` (siehe unten); `none` blendet die Symbole aus.
- ▶ `-i, --inode`
Die `inode`-Nummer der Dateien wird zusätzlich ausgegeben.
- ▶ `-I, --ignore=Shell-Muster`
Dateien, deren Name dem angegebenen Muster entspricht, werden nicht angezeigt; im Gegensatz zu `--hide` wird diese Option nicht von `-a` oder `-A` überschrieben.
- ▶ `-k`
Kurzfassung für `--block-size=1K`
- ▶ `-l`
Ausführliche Ausgabe
- ▶ `-L, --dereference`
Informationen über die Zieldateien symbolischer Links statt über die Links selbst anzeigen
- ▶ `-m`
Die Dateinamen in einer durch Komma getrennten Liste ausgeben
- ▶ `-n, --numeric-uid-gid`
Ausführlich wie `-l`, aber Benutzer- und Gruppen-IDs werden numerisch ausgegeben.
- ▶ `-N, --literal`
Namen ohne Berücksichtigung von Steuer- oder Sonderzeichen ausgeben
- ▶ `-o`
Wie `-l` ohne Angabe der Group-IDs (entspricht `-lG`)
- ▶ `-p, --file-type`
Dateitypsymbole an Dateinamen anhängen: `/` (Verzeichnis), `=` (Socket), `@` (Symlink) oder `|` (Pipe)
- ▶ `-q, --hide-control-chars`
Nicht druckbare Zeichen in Dateinamen durch `?` ersetzen
- ▶ `--show-control-chars`
Nicht druckbare Zeichen normal anzeigen (Standard)
- ▶ `-Q, --quote-name`
Dateinamen in doppelte Anführungszeichen setzen - praktisch, wenn Sie den Verdacht haben, dass ein Dateiname Leerzeichen enthält.
- ▶ `--quoting-style=Zeichenoptionen`
Stellt die Dateinamen im angegebenen Format dar: `literal` (siehe `-N`), `locale`

- ▶ -x
Einträge zeilen- statt spaltenweise anordnen
- ▶ -X
Alphabetisch nach Dateinamenerweiterung sortieren
- ▶ -l
Nur eine Datei pro Zeile anzeigen

19.2 Alphabetische Kommandoreferenz

Hier nun die eigentliche Referenz der wichtigsten Kommandos, jeweils mit Syntaxübersicht und den gängigsten Optionen. Nicht separat aufgeführt sind die bei fast allen Kommandos verfügbaren Optionen `--help` (kurze Verwendungsübersicht ausgeben, dann sofort beenden) und `--version` (Versionsinformationen ausgeben und beenden). Bei einigen Kommandos werden gar keine Optionen angegeben, obwohl es laut Syntaxschema `welche gibt` – in diesen Fällen sind sie nicht ganz so wichtig.

apropos

Sucht in den Beschreibungen der Manpages nach regulären Ausdrücken, und listet sie mit Titel, Sektion und Kurzbeschreibung auf.

Syntax

```
apropos [Optionen] Regexp [Regexp ...]
```

Entspricht einem Aufruf von `→ man` mit der Option `-k`.

Optionen

- ▶ -e, --exact
Suche nach den angegebenen Wörtern statt nach regulären Ausdrücken
- ▶ -w, --wildcard
Suche mit Shell-Wildcards
- ▶ -M Pfad, --manpath=Pfad
Durchsucht die Manpages unter dem angegebenen Pfad; mehrere Pfade können durch Doppelpunkte getrennt werden. Standardmäßig wird die Umgebungsvariable `$MANPATH` zugrunde gelegt.

bc

Interaktive Taschenrechner-Shell mit beliebiger Fließkommagenauigkeit und Programmierfunktionen

Syntax

```
bc [Optionen] [Datei]
```

Falls die angegebene Datei nicht mit dem Befehl `quit` endet, liest `bc` nach ihrer Verarbeitung von der Standardeingabe weiter; wird keine Datei angegeben, dann geschieht dies sofort. Um `bc` nichtinteraktiv – beispielsweise in einem Shell-Skript – einzusetzen, können Sie über eine Pipe einen `echo`-Befehl an das Kommando weiterleiten. Beispiel:

```
echo "2+2" |bc
```

Optionen

- ▶ `-q, --quiet`
Keinen Begrüßungsbanner ausgeben (praktisch für die nichtinteraktive Dateiverarbeitung)
- ▶ `-s, --standard`
Nur POSIX-`bc` ohne GNU-Erweiterungen verarbeiten
- ▶ `-w, --warn`
Warnungen bei Befehlen ausgeben, die nicht konform zu POSIX-`bc` sind

Wichtige Befehle

- ▶ `Rechenausdruck`
Beliebige Terme wie $2+40$ oder $46/2$ werden berechnet, und das Ergebnis wird angezeigt. Erlaubt sind unter anderem folgende Operatoren:
 - ▶ Arithmetische Operatoren: `+`, `-`, `*`, `/`, `^` (Potenz) und `%` (Modulo, also Rest der Division).
 - ▶ Vergleichsoperatoren: `==`, `!=`, `<`, `>`, `<=`, `>=` (als Ergebnis wird jeweils 1 für wahr oder 0 für falsch angezeigt).
 - ▶ Logische Operatoren: `&&` (Und), `||` (Oder), `!` (vorangestelltes Nicht).
- ▶ `"String"`
Der angegebene String wird ausgegeben.

- ▶ `Variable=Ausdruck`

Weist der angegebenen Variablen das Ergebnis des Ausdrucks zu. Danach kann die Variable sofort in Rechenausdrücken verwendet werden. Beispiel:

```
a=16
2^a
65536
```

Noch nicht definierte Variablen besitzen automatisch den Wert 0.

Zusätzlich sind Kurzfassungen wie `a+=2` für `a=a+2` oder `n++` für `n=n+1` möglich.

- ▶ `if (Testausdruck) Wenn_Ausdruck [else Sonst_Ausdruck]`
Wenn der Testausdruck wahr ist, wird `Wenn_Ausdruck` ausgewertet, ansonsten `Sonst_Ausdruck`.
- ▶ `while (Testausdruck) Ausdruck`
Wertet den Ausdruck immer wieder aus, solange der Testausdruck wahr ist.
- ▶ `for (Init;Test;Änderung) Ausdruck`
Funktioniert wie eine `for`-Schleife in C oder anderen Sprachen: Vor dem ersten Durchlauf wird `Init` ausgeführt, bei zutreffendem Test erfolgt ein Schleifendurchlauf und nach jedem Durchlauf die Änderung. Das folgende Beispiel gibt die Reihe $2n$ für $n = 1$ bis 16 aus:

```
for (n=1; n<=16; n++) 2^n
```
- ▶ `read ()`
Liest einen Wert von der Standardeingabe. Nützlich bei nichtinteraktiver Verwendung in Shell-Skripten. Einfaches Beispiel:

```
"Eine Zahl, bitte: "  
a=read ( )  
if (a == 42) "Genau!" else "Leider nicht!"
```
- ▶ `quit`
bc beenden.

bzip2

Dateien effizient komprimieren (und wieder extrahieren)

Syntax

```
bzip2 [Optionen] [Datei|Muster ...]
bunzip2 [Optionen] [Datei|Muster ...]
```

Die angegebenen Dateien werden standardmäßig durch je eine komprimierte Datei mit der zusätzlichen Erweiterung `.bz2` ersetzt.

bzip2 kann auch implizit über `tar` aufgerufen werden.

Optionen

- ▶ `-c, --stdout`
Ergebnis auf die Standardausgabe statt in eine Datei schreiben
- ▶ `-z, --compress`
Komprimieren (kann beim Aufrufnamen `bzip2` weggelassen werden, da Standard)
- ▶ `-d, --decompress`
Dekomprimieren (Standard beim Aufruf als `bunzip2`)
- ▶ `-f, --force`
Überschreiben existierender Dateien erzwingen
- ▶ `-k, --keep`
Die Originaldateien nach dem Komprimieren nicht löschen

cd

Das aktuelle Arbeitsverzeichnis wechseln

Syntax

```
cd [Verzeichnis]
```

Legt das als relativer oder absoluter Pfad angegebene Verzeichnis als aktuelles Arbeitsverzeichnis fest. Wird `~` oder gar kein Verzeichnis angegeben, dann wird das Home-Verzeichnis des aktuellen Benutzers gewählt.

cdparanoia

Audio-CDs digital auslesen

Syntax

```
cdparanoia [Optionen]
```

Optionen

- ▶ `-B`
Jeden Track als einzelne Datei speichern (andernfalls wird die gesamte CD beziehungsweise die einzelnen Tracks als einzelne, große Sounddatei gespeichert).

- ▶ -w
Die ausgelesenen Tracks im WAV-Format speichern

chgrp

Gruppenzugehörigkeit von Dateien ändern

Syntax

```
chgrp [Optionen] Gruppe Datei|Muster [Datei|Muster ...]
```

Optionen

- ▶ -R, --recursive
Unterverzeichnisse rekursiv abarbeiten
- ▶ -v, --verbose
Ausführliche Ausgabe (nicht nur Fehler, sondern auch Erfolge werden gemeldet)

chkconfig

Den automatischen Start von Diensten in */etc/init.d* einrichten.

Syntax

```
chkconfig Option [Dienst ...]
```

Besonderheit: Es *muss* (genau) eine Option angegeben werden, damit `chkconfig` weiß, was zu tun ist.

Optionen

- ▶ -a, --add
Den Start des angegebenen Dienstes in seinen Standard-Runlevels aktivieren
- ▶ -d, --del
Den Dienst in allen Runlevels deaktivieren
- ▶ -l, --list
Ausführliche Ausgabe der aktuellen Konfiguration (mit Runlevel-Liste). Falls kein Dienst angegeben wird, erscheint die gesamte Liste.
- ▶ -t, -- terse
Kurzausgabe (nur `on`, `off` oder `xinetd`)

chmod

Dateizugriffsrechte ändern

Syntax

```
chmod [Optionen] Modus Datei|Muster [Datei|Muster ...]
```

Der `Modus` kann entweder als String oder als Oktalzahl angegeben werden. Der String besteht aus folgenden Elementen:

- ▶ *Benutzerart*: u (Eigentümer), g (Gruppe), o (Sonstige), a (alle genannten)
- ▶ *Modifikation*: + (Hinzufügen), - (Abziehen), = (exklusiv setzen)
- ▶ *Recht*: r (Lesen), w (Schreiben), x (Ausführen), s (Setuid bzw. Setgid), t (Sticky)

Mehrere Rechte-Strings können durch Komma, aber ohne Leerzeichen getrennt werden. Beispiel: `a+rx,u+w` (alle dürfen lesen und ausführen, der Eigentümer darf auch ändern).

Alternativ lässt sich der `Modus` als vierstellige Oktalzahl angeben. Die zweite bis vierte Stelle sind die Rechte von Eigentümer, Gruppe und sonstigen Users, jeweils als Summe aus 4 (Lesen), 2 (Schreiben) und 1 (Ausführen). Die vorderste Stelle ist die Summe aus 4 (Setuid), 2 (Setgid) und 1 (Sticky).

Optionen

- ▶ `-R, --recursive`
Unterverzeichnisse rekursiv durchwandern
- ▶ `-v, --verbose`
Ausführliche Ausgabe

chown

Eigentümer von Dateien ändern

Syntax

```
chown [Optionen] Benutzer[:Gruppe] Datei|Muster  
[Datei|Muster ...]
```

Optionen

- ▶ `-R, --recursive`
Unterverzeichnisse rekursiv durchwandern

- ▶ `-v, --verbose`
Ausführliche Ausgabe

clear

Bildschirm löschen

Syntax

```
clear
```

Kurzfassung: `[Strg]+[L]`

cp

Dateien kopieren

Syntax

```
cp [Optionen] Quelldatei|-muster Ziel
```

Falls das Ziel ein existierendes Verzeichnis ist, werden die Dateien unter ihrem bisherigen Namen hineinkopiert, andernfalls wird das Ziel als Dateiname interpretiert.

Optionen

- ▶ `-f, --force`
Überschreiben erzwingen
- ▶ `-i, --interactive`
Vor eventuellem Überschreiben nachfragen
- ▶ `-r, -R, --recursive`
Unterverzeichnisse rekursiv durchwandern
- ▶ `-v, --verbose`
Ausführliche Ausgabe

crontab

Die crontab eines Benutzers bearbeiten

Syntax

```
crontab [-u Benutzer] [-e | -l | -r]
```

Optionen

- ▶ -u Benutzer
Nicht die eigene crontab bearbeiten, sondern diejenige des angegebenen Users (dies darf nur root).
- ▶ -e
Öffnet die crontab in vi zum Bearbeiten. Jeder Eintrag besitzt das folgende Format:

```
Min Std Tag Mon Wochentag Kommando
```

- ▶ -l
Die aktuelle crontab anzeigen
- ▶ -r
Die crontab des besagten Benutzers löschen

curl *

Daten per HTTP, HTTPS oder FTP von einer URL empfangen oder an eine URL senden

Syntax

```
curl [Optionen] [URL]
```

Bei einem Aufruf ohne jegliche Optionen werden die Inhalte der angegebenen URL ausgelesen und auf die Standardausgabe geschrieben.

Optionen

curl besitzt zahlreiche Optionen; hier nur einige besonders wichtige für Downloads:

- ▶ -A String, --user-agent String
Den angegebenen User-Agent-String (Client-Selbstidentifikation) an den Server senden (nur HTTP)
- ▶ -I, --head
Nur die Header der Server-Antwort lesen und anzeigen; sehr praktisch zur Diagnose von HTTP- oder FTP-Problemen

- ▶ `-o Dateiname, --output Dateiname`
Ausgabe in die angegebene Datei statt nach `stdout`

date

Systemdatum und -uhrzeit einstellen oder anzeigen

Syntax

```
date [Optionen] [+Format]
date MMDDhhmm[CC]YY[.ss]
```

Die erste Schreibweise zeigt Datum und Uhrzeit an, die zweite stellt die Systemuhr um (Letzteres darf nur `root`). Dieses Beispiel zeigt ein deutsches (numerisches) Standarddatum sowie die durch Komma von diesem getrennte Uhrzeit an:

```
$ date "+%d.%m.%Y, %H:%M"
10.05.2008, 19:38
```

Dieses Beispiel stellt Datum und Uhrzeit auf den 01.01.1970, 00:00:00 Uhr UTC, also EPOCH ein (die Option `-u` für UTC ist notwendig, falls auf Ihrem System die mitteleuropäische Zeit eingestellt ist):

```
# date -u 010100001970.00
# date "+%d.%m.%Y, %H:%M:%S"
01.01.1970, 01:00:07
```

Tabelle 19.1 zeigt eine Liste aller gültigen Formatangaben. »Locale« bedeutet dabei, dass die aktuellen Landes- und Regionaleinstellungen beachtet werden. Wenn Sie Ihr openSUSE-System deutschsprachig installiert haben, können Sie also Daten wie 10.05.2008 (DD.MM.YY) und Uhrzeiten wie 19:38:41 (hh:mm:ss im 24-Stunden-Format) erwarten.

Format-Platzhalter	Bedeutung
%%	literales %
%a	abgekürzter Wochentagsname (Locale)
%A	vollständiger Wochentagsname (Locale)
%b	abgekürzter Monatsname (Locale)
%B	vollständiger Monatsname (Locale)
%c	Datum und Uhrzeit (Locale)

Tabelle 19.1 Alle gültigen Formatplatzhalter für das Kommando »date«

Format-Platzhalter	Bedeutung
%d	Tag des Monats, gegebenenfalls mit führender Null (01–31)
%D	Datum (MM/DD/YY)
%e	Tag des Monats, gegebenenfalls mit führendem Leerzeichen (1–31)
%h	abgekürzter Monatsname (Locale)
%H	Stunde, 24-Stunden-Format, gegebenenfalls mit führender Null (00–23)
%I	Stunde, 12-Stunden-Format, gegebenenfalls mit führender Null (01–12)
%j	Tag des Jahres, gegebenenfalls mit führenden Nullen (001–366)
%k	Stunde, 24-Stunden-Format, gegebenenfalls mit führendem Leerzeichen (0–23)
%k	Stunde, 12-Stunden-Format, gegebenenfalls mit führendem Leerzeichen (1–12)
%M	Minute, gegebenenfalls mit führender Null (00–59)
%n	Zeilenumbruch
%p	AM oder PM für 12-Stunden-Format
%r	Uhrzeit, 12-Stunden-Format (hh:mm:ss AM PM)
%s	Sekunden seit EPOCH
%S	Sekunde, gegebenenfalls mit führender Null (00–60)
%t	Tab
%T	Uhrzeit, 24-Stunden-Format (hh:mm:ss)
%U	Wochennummer des Jahres (beginnend mit Sonntag)
%V	Wochennummer des Jahres (beginnend mit Montag)
%w	numerischer Wochentag (0 = Sonntag, 1 = Montag, ... 6 = Samstag)
%W	Wochennummer des Jahres (beginnend mit Montag; kann im Gegensatz zu %V null sein)
%x	Datum (Locale)
%X	Uhrzeit (Locale)

Tabelle 19.1 Alle gültigen Formatplatzhalter für das Kommando »date« (Forts.)

Format-Platzhalter	Bedeutung
%y	zweistellige Jahreszahl (00–99)
%Y	vierstellige Jahreszahl
%z	numerische Zeitzone (Differenz zur UTC)
%Z	Name der Zeitzone

Tabelle 19.1 Alle gültigen Formatplatzhalter für das Kommando »date« (Forts.)

dd

Dateien konvertieren; Blöcke direkt von Gerätedateien lesen oder auf diese schreiben

Syntax

```
dd Schlüssel[=Wert] [Schlüssel[=Wert]]
```

Optionen

- ▶ `if=Eingabedatei`
Angabe der Datei oder des Geräts, aus dem gelesen werden soll (wenn Sie diese Option weglassen, wird von `stdin` gelesen)
- ▶ `of=Ausgabedatei`
Angabe der Datei oder des Geräts, auf das geschrieben werden soll (ansonsten `stdout`)
- ▶ `bs=Bytes`
Größe der jeweils zu verarbeitenden Blöcke
- ▶ `conv=Format`
Format, in das die Eingabe konvertiert werden soll. Beispiele: `ascii` (den klassischen IBM-Großrechner-Zeichensatz EBCDIC nach ASCII konvertieren), `ebcdic` (umgekehrt), `lcase` (Großbuchstaben in Kleinbuchstaben umwandeln)

diff

Dateien vergleichen

Syntax

```
diff [Optionen] Datei1 Datei2
```

Optionen

- ▶ `-i, --ignore-case`
Unterschiedliche Groß- und Kleinschreibung ignorieren
- ▶ `-b, --ignore-space-change`
Unterschiedliche Anzahlen von Leerzeichen ignorieren
- ▶ `-w, --ignore-all-space`
Sämtlichen Whitespace ignorieren
- ▶ `-B, --ignore-blank-lines`
Unterschiedliche Anzahlen völlig leerer Zeilen ignorieren

dnsdomainname *

Siehe → `hostname`

domainname

Siehe → `hostname`

dos2unix *

Konvertiert DOS/Windows- oder Mac-Textdateien (Zeilenumbrüche) in das UNIX-Format.

Syntax

```
dos2unix [Optionen] Datei|Muster [Neue_Datei]
```

```
mac2unix [Optionen] Datei|Muster [Neue_Datei]
```

Optionen

- ▶ `-o, --oldfile`
Konvertiert die angegebene Datei an Ort und Stelle (ersetzt sie) – dies ist der Standardmodus, in dem Sie auch Dateimuster angeben können.
- ▶ `-n, --newfile`
Speichert die angegebene Quelldatei und konvertiert sie in die als Zweites angegebene Zieldatei. In diesem Modus dürfen Sie auf keinen Fall Muster verwenden, weil der erste Treffer ansonsten den zweiten überschreibt.

du

Zeigt den Speicherverbrauch durch Dateien oder Verzeichnisse an.

Syntax

```
du [Optionen] Datei|Muster [Datei|Muster ...]
```

Optionen

- ▶ `-a, --all`
Ergebnis auch für alle Dateien, nicht nur für Verzeichnisse anzeigen
- ▶ `-c, --total`
Zum Schluss Gesamtsumme anzeigen
- ▶ `-h, --human-readable`
Je nach Größe die vernünftigste Maßeinheit anzeigen (K = Kilobyte, M = Megabyte, G = Gigabyte)

echo

Ausgabe einer Textzeile

Syntax

```
echo [Optionen] String
```

Im String werden Variablen aufgelöst, falls Sie diesen ein Dollarzeichen voranstellen (zum Beispiel `$PATH`). In vielen Fällen ist es sicherer, den String in doppelte Anführungszeichen zu setzen.

Optionen

- ▶ `-n`
Den abschließenden Zeilenumbruch unterdrücken
- ▶ `-e`
Escape-Sequenzen interpretieren (zum Beispiel `\n` für Zeilenumbruch oder `\t` für Tabulator); funktioniert nur mit den erwähnten Anführungszeichen.

expr

Ganzzahlige arithmetische Operationen durchführen

Syntax

expr Ausdruck

Der Ausdruck kann aus diversen mathematischen oder logischen Operationen bestehen. Operanden und Operatoren *müssen* durch Leerzeichen voneinander getrennt werden, da sie die einzelnen Kommandozeilenargumente des Befehls bilden. Wichtig ist, dass Sie den Operatoren * (Multiplikation) und / (Division) jeweils einen Backslash voranstellen müssen, da sie für die Shell eine besondere Bedeutung haben. Hier zwei Beispiele:

```
$ expr 2 \< * 7 + 9
23
$ expr 6 + 4 \< * 9
42
```

file *

Dateitypen anhand des Datei-Inhalts ermitteln

Syntax

```
file [Optionen] [-f Eingabedatei] Datei|Muster [Datei|Muster ...]
```

Die Ausgabe besteht aus dem Namen jeder einzelnen untersuchten Datei, gefolgt von einer Typinformation. Beispiele:

```
$ file test.pdf
test.pdf: PDF document, version 1.4
$ file bild.jpg
bild.jpg: JPEG image data, JFIF standard 1.01
```

Optionen

- ▶ -b, --brief
Nur die Typinformation, nicht den Dateinamen selbst ausgeben – praktisch für die automatisierte Verarbeitung in Shell-Skripten und Ähnliches
- ▶ -f Eingabedatei, --files-from Eingabedatei
Die Liste der zu untersuchenden Dateien wird nicht auf der Kommandozeile erwartet, sondern aus der angegebenen Datei gelesen.
- ▶ -F Zeichen, --separator Zeichen
Trennzeichen, das in der Ausgabe zwischen Dateiname und -typ verwendet werden soll

- ▶ `-i, --mime`
Statt der ausführlichen Beschreibungen sollen MIME-Typen ausgegeben werden – etwa `text/html; charset=iso-8859-1` statt `HTML document text`.
- ▶ `-k, --keep-going`
Standardmäßig hört `file` beim ersten Treffer auf; diese Option sucht weiter.

find

Suche nach Dateien in einer Verzeichnishierarchie

Syntax

```
find [-H | -L | -P] Verzeichnis Suchausdruck [Suchausdruck ...]
```

Optionen

Die drei »richtigen« Optionen `-H`, `-L` oder `-P` schließen einander aus. Sie bestimmen, ob Symlinks immer (`-L`), nie (`-P`) oder nur bei expliziter Erwähnung auf der Kommandozeile (`-H`) aufgelöst werden sollen.

Suchausdrücke (Auswahl)

- ▶ `-maxdepth Anzahl`
Die maximale Tiefe zu durchsuchender Unterverzeichnisse
- ▶ `-ctime Tage`
Datei wurde mindestens vor der angegebenen Anzahl von Tagen (genauer gesagt von 24-Stunden-Perioden) zuletzt geändert.
- ▶ `-empty`
Leere Datei oder leeres Verzeichnis
- ▶ `-group Gruppe`
Datei gehört der angegebenen Gruppe. Sie können den Gruppennamen oder die numerische GID angeben.
- ▶ `-iname Muster`
Suche nach einem Muster mit Shell-Platzhaltern ohne Unterscheidung von Groß- und Kleinschreibung. Solche Muster sollten Sie bevorzugt in Anführungszeichen setzen; falls sie mit einem `*` enden, ist dies Pflicht.
- ▶ `-iregex Regexp`
Suche nach einem regulären Ausdruck ohne Unterscheidung von Groß- und Kleinschreibung

- ▶ `-name` Muster
Suche nach einem Shell-Muster mit Unterscheidung von Groß- und Kleinschreibung
- ▶ `-newer` Datei
Suche nach Dateien, die neuer sind als die angegebene Datei
- ▶ `-perm` Dateirechte
Findet Dateien, die die genannten Berechtigungen aufweisen (diese können symbolisch oder oktal angegeben werden; siehe → `chmod`). In aller Regel sollten Sie den gesuchten Berechtigungen ein `-` voranstellen (findet alle Dateien, die *mindestens* die genannten Rechte besitzen); ohne dieses Präfix muss die Angabe *genau* zutreffen.
- ▶ `-regex` Regexp
Suche nach einem regulären Ausdruck mit Unterscheidung von Groß- und Kleinschreibung
- ▶ `-type` Typ
Datei besitzt den angegebenen Typ. Beispiele: `d` = Verzeichnis, `f` = reguläre Datei, `l` = Symlink, `p` = Named Pipe (FIFO), `s` = Socket. Der Typ `l` funktioniert nicht zusammen mit der Option `-L` – außer für kaputte Symlinks.
- ▶ `-user` Benutzer
Datei gehört dem angegebenen Benutzer. Sie können den Username oder die numerische UID angeben.

grep

Suche nach regulären Ausdrücken in Dateien

Syntax

```
grep [Optionen] Regexp [Datei|Muster ...]
```

Wenn Sie keine Dateinamen angeben, wird die Standardeingabe durchsucht – in der Regel wird eine Pipe zur Verknüpfung verwendet, wie in

```
ls -l |grep "..."
```

In diesem Fall werden die gefundenen Zeilen ausgegeben. Bei Dateien lautet die Ausgabe dagegen jeweils `Datei:Fundzeile`.

Optionen

- ▶ `-f` Datei, `--file=Datei`
Die regulären Ausdrücke aus der angegebenen Datei durchprobieren

- ▶ `-F, --fixed-strings`
Nach String-Literalen statt nach regulären Ausdrücken suchen. Zu diesem Zweck können Sie das Programm alternativ als `fgrep` aufrufen.
- ▶ `-i, --ignore-case`
Nicht zwischen Groß- und Kleinschreibung unterscheiden
- ▶ `-n, --line-number`
Vor jedem Treffer zusätzlich die Zeilennummer ausgeben
- ▶ `-r, -R, --recursive`
Unterverzeichnisse rekursiv durchsuchen

Reguläre Ausdrücke

Tabelle 19.2 enthält die wichtigsten regulären Ausdrücke im Überblick. Beachten Sie, dass GNU `grep` mit der Option `-E` auch erweiterte, mit `-P` sogar Perl-kompatible reguläre Ausdrücke verarbeitet, die beide nicht Gegenstand der Tabelle sind. Eine etwas umfangreichere Einführung in Perl-RegExp finden Sie in Kapitel 17, »System-Automatisierung«.

RegExp-Konstrukt	Beschreibung	Beispiele
<code>abc</code>	die Zeichenfolge <code>abc</code>	<code>ö1</code> passt auf Köln, aber nicht auf schön oder Schnösel.
<code>[abc]</code>	eines der Zeichen <code>a</code> , <code>b</code> oder <code>c</code>	<code>B[ai]ll</code> passt auf Ball oder Bill, aber nicht auf Bell.
<code>[a-m]</code>	eines der Zeichen von <code>a</code> bis <code>m</code>	<code>10[1-5]</code> passt auf 102 und 104, aber nicht auf 107.
<code>[^abc]</code>	jedes beliebige Zeichen außer <code>a</code> , <code>b</code> , <code>c</code>	<code>B[^o]hn</code> passt auf Bahn, aber nicht auf Bohne.
<code>.</code>	ein beliebiges Zeichen	<code>R.in</code> passt auf Ruine und Rein, aber nicht auf Rind.
<code>\</code>	Escaping von Zeichen, die in RegExp eine besondere Bedeutung haben	<code>\.</code> , <code>\(</code> oder <code>\\$</code> : <code>i[a-z]+\.</code> <code>html?</code> passt auf <code>info.html</code> , <code>index.html</code> und <code>illustration.html</code> , aber nicht auf <code>inhtml</code> oder <code>info9.html</code> .
<code>?</code>	Das links stehende Element kann einmal vorkommen oder nicht.	<code>Ei?s</code> passt auf <code>Eis</code> und <code>Es</code> , aber nicht auf <code>Eins</code> .

Tabelle 19.2 Die wichtigsten regulären Ausdrücke, die »grep« versteht

RegExp-Konstrukt	Beschreibung	Beispiele
+	Das links stehende Element muss mindestens einmal vorkommen.	25+ passt auf 25 und 25555, aber nicht auf 2 oder 23.
*	Das links stehende Element kann beliebig oft vorkommen.	10* passt auf 1, 10, 1000000 und sogar auf 12 usw.
{n}	Das links stehende Element muss genau n-mal vorkommen.	test[0-9]{2} passt auf test03 und test56, aber nicht auf test2.
{m,n}	Das links stehende Element muss mindestens m- und darf höchstens n-mal vorkommen.	test[0-9]{2,4} passt auf test082 oder test25, aber nicht auf test2a oder testen.
^	Anfang des Ausdrucks	^öl passt auf ölig, aber nicht auf Köln.
\$	Ende des Ausdrucks	öl\$ passt auf Rhizinusöl, aber nicht auf Köln.
(...)	Gruppierung; dient der Anwendung von +, * u. Ä. auf Textblöcke.	T(ischt)?elefon passt auf Telefon und Tischtelefon.
	Ermöglicht Oder-Verknüpfung in (...)-Gruppen	Daten(bank verarbeitung) passt auf Datenbank und Datenverarbeitung, aber nicht auf Datenschutz.

Tabelle 19.2 Die wichtigsten regulären Ausdrücke, die »grep« versteht (Forts.)

groupadd

eine neue Gruppe erstellen

Syntax

```
groupadd [Optionen] Gruppenname
```

Optionen

- ▶ -g GID, --gid GID
Erzwingt eine bestimmte numerische Group-ID; ansonsten wird diese automatisch vergeben.

- ▶ `-o, --non-unique`
Erzeugt die Gruppe auch dann, wenn die GID bereits existiert (nur sinnvoll im Zusammenhang mit `-g`).

groupdel

Löscht eine Gruppe.

Syntax

```
groupdel [Optionen] Gruppe
```

Eine Gruppe kann nicht gelöscht werden, falls sie die primäre Gruppe irgendeines Benutzers ist.

gzip

Dateien komprimieren (und wieder extrahieren)

Syntax

```
gzip [Optionen] [Datei|Muster ...]
gunzip [Optionen] [Datei|Muster ...]
```

Die angegebenen Dateien werden durch je eine komprimierte Datei mit der zusätzlichen Erweiterung `.gz` ersetzt.

`gzip` kann auch implizit über `→ tar` aufgerufen werden.

Optionen

- ▶ `-c, --stdout`
Ergebnis auf die Standardausgabe statt in eine Datei schreiben
- ▶ `-d, --decompress`
Dekomprimieren (Standard beim Aufruf als `gunzip`)
- ▶ `-f, --force`
Überschreiben existierender Dateien erzwingen

head

Nur den Anfang von Textdateien anzeigen

Syntax

```
head [Optionen] [Datei|Muster ...]
```

Standardmäßig werden die ersten 10 Zeilen jeder Datei ausgegeben. Wird keine Datei angegeben, dann gibt `head` den Beginn der Standardeingabe aus.

Optionen

- ▶ `-c=[-]N`, `--bytes=[-]N`
Nur die ersten `N` Byte des Textes ausgeben oder – mit vorangestelltem Minuszeichen – den gesamten Text bis auf die letzten `N` Byte
- ▶ `-n=[-]N`, `--lines=[-]N`
Nur die ersten `N` Zeilen des Textes ausgeben oder – mit vorangestelltem Minuszeichen – den gesamten Text bis auf die letzten `N` Zeilen
- ▶ `-Zahl`
Kurzfassung für `-N=Zahl`; `-3` gibt beispielsweise die ersten drei Zeilen aus.

hostname

Den Hostnamen des Rechners setzen oder ändern

Syntax

```
hostname [Optionen] [Hostname]
domainname [Optionen] [NIS-Domainname]
dnsdomainname [Optionen] [DNS-Domainname]
```

Wenn Sie keinen Hostnamen angeben, wird der bisherige angezeigt, ansonsten wird der angegebene Hostname für die Dauer der aktuellen Sitzung gesetzt. Ebenso verfahren `domainname` für den NIS-Domainnamen und `dnsdomainname` für den DNS-Domainnamen.

iconv *

Den Zeichensatz eines Textes konvertieren

Syntax

```
iconv [Optionen] [Datei]
```

Wenn keine Datei angegeben wird, liest `iconv` von der Standardeingabe. Die Ausgabe erfolgt normalerweise nach `stdout`.

Optionen

- ▶ `-f Zeichensatz, --from-code=Zeichensatz`
Gibt den Zeichensatz der Quelldatei an, zum Beispiel `utf-8` oder `iso-latin-1`.
- ▶ `-t Zeichensatz, --to-code=Zeichensatz`
Gibt den Zeichensatz an, in den die ursprüngliche Datei konvertiert werden soll.
- ▶ `-o Datei, --o=Datei`
Statt auf die Standardausgabe in die angegebene Datei schreiben
- ▶ `-l, --list`
Liste aller verfügbaren Zeichensätze ausgeben

ifconfig

Netzwerkschnittstellen konfigurieren

Syntax

```
ifconfig [Schnittstelle]
ifconfig Schnittstelle [Optionen]
```

Die erste Variante zeigt die Konfiguration der angegebenen Schnittstelle beziehungsweise aller Schnittstellen an, die zweite setzt die Optionen einer bestimmten Schnittstelle.

Optionen

- ▶ `Schnittstelle`
Der Name einer Netzwerkschnittstelle wie `eth0` (erste Ethernet-Karte) oder `lo0` (Loopback)
- ▶ `up`
Die Schnittstelle in Betrieb nehmen
- ▶ `down`
Die Schnittstelle deaktivieren
- ▶ `netmask`
Eine Teilnetzmaske wie `255.255.255.0`; muss nur angegeben werden, falls sie sich nicht automatisch aus der IP-Adresse ergibt.
- ▶ `add IPv6-Adresse`
Fügt eine IPv6-Adresse hinzu
- ▶ `del IPv6-Adresse`
Entfernt eine IPv6-Adresse

- ▶ `irq` Nummer
Setzt den IRQ (nur für einige ältere Karten erforderlich, meist ISA)
- ▶ `io_addr` Adresse
Setzt die I/O-Basisadresse (wie `irq` in der Regel nur für alte Karten)
- ▶ `media` Typ
Verkabelungstyp, etwa 100BaseT oder 10Base2
- ▶ IP-Adresse
Die wichtigste Angabe: die IPv4-Adresse, falls sie nicht automatisch per DHCP zugewiesen wird

init

System initialisieren (`init` ist der allererste Prozess); Runlevel wechseln

Syntax

```
init [Optionen] [Runlevel]
```

Runlevels

- ▶ 0 Heruntergefahrener Zustand
- ▶ S Single-User-Modus (für Wartungsarbeiten)
- ▶ 1 ebenfalls Single-User-Modus
- ▶ 2 Multi-User-Modus mit lokalem Netzwerk; nur Konsole
- ▶ 3 Multi-User-Modus mit vollem Netzwerkzugriff
- ▶ 5 Multi-User-Modus mit Netzwerk und grafischer Oberfläche (KDM, GDM oder XDM)
- ▶ 6 Systemneustart (Reboot)

insmod

Kernel-Modul laden

Syntax

```
insmod Modul [Optionen]
```

Normalerweise ist → `modprobe` zu bevorzugen.

Optionen

Die Optionen sind je nach Modul recht unterschiedlich. Einige allgemeingültige sind `irq=Nummer` für den IRQ oder `io=Adresse` für die I/O-Basisadresse; die meisten Module benötigen aber gar keine Optionen.

kill

Einem Prozess ein Signal senden

Syntax

```
kill [-s Signal | -Signal] PID
kill -l [Signal]
```

Das erste Syntaxschema sendet das angegebene Signal (Standard: SIGTERM, numerisch 15) an den Prozess mit der angegebenen Prozess-ID. Dies lässt sich mit `→ ps` ermitteln. Manche Programme besitzen auch eine PID-Datei, in der ihre aktuelle PID gespeichert ist; diese Datei liegt oft unter `/var/run/Programmname`.

Die zweite Variante listet die Nummern und Namen aller verfügbaren Signale auf. Falls Sie ein bestimmtes Signal angeben, wird die zu seinem Namen passende Nummer (oder umgekehrt) angezeigt.

Wichtige Signale

- ▶ SIGTERM (15)
Prozess normal beenden
- ▶ SIGKILL (9)
Prozess gewaltsam beenden
- ▶ SIGINT (2)
Terminal-Interrupt – viele Programme verstehen dies als Aufforderung, ihre laufenden Angelegenheiten zu regeln und sich dann zu beenden.
- ▶ SIGHUP (1)
Hangup – viele Programme initialisieren sich neu (lesen zum Beispiel ihre Konfiguration neu ein), wenn sie dieses Signal erhalten.

killall

Signale an Prozesse mit bestimmten Namen senden

Syntax

```
killall [Optionen] [-s Signal | -Signal] Name
```

Das gewünschte Signal (standardmäßig auch hier SIGTERM) wird an alle Prozesse gesendet, die den angegebenen Namen haben.

Optionen

- ▶ `-I, --ignore-case`
Groß- und Kleinschreibung ignorieren
- ▶ `-i, --interactive`
Bei jedem infrage kommenden Prozess einzeln nachfragen, ob er das Signal erhalten soll
- ▶ `-r, --regex`
Nach regulären Ausdrücken statt nach vollen Namen suchen

less

Textdateien seitenweise anzeigen

Syntax

```
less [Optionen] [Dateiname]
```

Wird kein Dateiname angegeben, dann liest `less` von der Standardeingabe.

Bedienung

- ▶ `E`, `↩` oder `↓` – eine Zeile nach unten blättern
- ▶ `Y` oder `↑` – eine Zeile nach oben blättern
- ▶ `F`, `Leertaste` oder `Bild ↓` – eine Fensterhöhe nach unten scrollen
- ▶ `B`, `Bild ↑` – eine Fensterhöhe nach oben scrollen
- ▶ `Pos1` – zum Anfang des Textes springen
- ▶ `Ende` – zum Ende des Textes springen
- ▶ `/` – Suchbegriff eingeben, dann `↩`
- ▶ `?` – Suchbegriff eingeben, sucht rückwärts
- ▶ `N` – zum nächsten Suchtreffer springen
- ▶ `↕+N` – Suchrichtung umkehren und weitersuchen
- ▶ `H` – Hilfeseite anzeigen
- ▶ `Q` – `less` beenden

ln

Erstellt Hard Links oder symbolische Links.

Syntax

```
ln [Optionen] [-s] Pfad Link-Pfad
```

Standardmäßig wird ein Hard Link erstellt (beschränkt auf reguläre Dateien und auf dieselbe Partition).

Optionen

- ▶ `-b`
Falls eine Datei mit dem Namen des neuen Links existiert, wird zuerst eine Sicherungskopie erstellt.
- ▶ `-s`
Erzeugt einen Symlink. Dabei dürfen auch Verzeichnisse verknüpft werden; Link und verknüpfter Pfad können sich in unterschiedlichen Dateisystemen befinden.

logger

Eine Meldung in das Syslog schreiben

Syntax

```
logger [-p [facility:]priority] message
```

Facilities (Fehlerquellen)

- ▶ `auth`: Authentifizierung
- ▶ `authpriv`: Authentifizierung privilegierter Benutzer
- ▶ `cron`: cron-Daemon
- ▶ `daemon`: sonstige Daemons
- ▶ `ftp`: FTP-Server
- ▶ `kern`: Kernel
- ▶ `local0` bis `local7`: zur freien Verwendung durch beliebige Programme
- ▶ `lpr`: Drucker-Subsystem
- ▶ `mail`: Mail-Subsystem
- ▶ `news`: News-Subsystem

- ▶ `syslog`: interne Meldungen von `syslog` selbst
- ▶ `uucp`: UUCP-Subsystem
- ▶ `user`: Anwendungsprogramme

Priorities (Dringlichkeiten)

- ▶ `emerg`: Notfall; das System ist unbrauchbar
- ▶ `alert`: Alarm – sofortiger Eingriff erforderlich
- ▶ `crit`: kritischer Fehler
- ▶ `error`: normaler Fehler
- ▶ `warn`: Warnung
- ▶ `notice`: Hinweis
- ▶ `info`: normale Information
- ▶ `debug`: Debugging-Information

lsmod

Eine Liste aller installierten Kernel-Module ausgeben

Syntax

```
lsmod
```

man

Eine Manpage anzeigen

Syntax

```
man [Optionen] [Sektion] Manpage
man -f Titel
man -k Schlüsselwort
```

Die Standardsyntax zeigt die gewünschte Manpage an, gegebenenfalls aus der gewünschten Sektion (wichtig, falls es mehrere gleichnamige Manpages gibt).

`man -f` entspricht → `whatis`, `man -k` ist ein Synonym für → `apropos`.

Sobald eine Manpage angezeigt wird, entspricht die Bedienung → `less`.

Standardsektionen

- ▶ 0: C-Header-Dateien
- ▶ 1: Systemprogramme
- ▶ 2: Systemaufrufe
- ▶ 3: Funktionen der C-Standardbibliothek
- ▶ 4: Gerätedateien
- ▶ 5: Dateiformate (Konfigurationsdateien) u. Ä.
- ▶ 6: Spiele
- ▶ 7: Makropakete
- ▶ 8: Administrationskommandos
- ▶ 9: Kernelroutinen

mkdir

Verzeichnisse erzeugen

Syntax

```
mkdir [Optionen] Verzeichnis
```

Optionen

- ▶ `-m` Berechtigungen, `--mode` Berechtigungen
Die angegebenen Berechtigungen für das neue Verzeichnis setzen (→ `chmod`).
Die Werte der eigenen → `umask` werden davon abgezogen.
- ▶ `-p`, `--parents`
Bei der Angabe verschachtelter Pfade gegebenenfalls auch die jeweils übergeordneten Verzeichnisse neu erzeugen.

modprobe

Kernel-Modul laden oder entladen

Syntax

```
modprobe [Optionen] Kernel-Modul
```

Der wichtigste Vorteil gegenüber → `insmod` besteht darin, dass eventuelle Modul-Abhängigkeiten automatisch aufgelöst werden, indem `modprobe` vor dem angegebenen Modul zunächst die benötigten Module lädt.

Optionen

- ▶ `-n, --dry-run`
Simulationsdurchlauf; `modprobe` zeigt an, was geschähe, wenn das betreffende Modul tatsächlich geladen oder entfernt würde.
- ▶ `-r, --remove`
Das angegebene Modul wird entladen.
- ▶ `-l [Muster]`
Eine Liste der geladenen Module anzeigen. Falls ein Muster in Shell-Syntax angegeben wird, werden nur die darauf passenden Module angezeigt.

mount

Einen Datenträger oder eine Partition in den Verzeichnisbaum einbinden

Syntax

```
mount [Optionen] Gerätedatei Verzeichnis
mount
```

Das erste Syntaxschema bindet die angegebene Gerätedatei unter dem gewünschten Pfad in den Verzeichnisbaum ein. Das Verzeichnis muss existieren und leer sein. Ohne jegliche Parameter zeigt `mount` stattdessen eine Liste aller gemounteten Partitionen und Dateisysteme an.

Optionen

- ▶ `-t Dateisystem`
Angabe des konkreten Dateisystems, etwa `ext3`, `reiserfs`, `fat32`, `iso9660` oder `smbfs`.
- ▶ `-o fstab-Option[,fstab-Option,...]`
Eine komma-separierte Liste von Optionen, wie sie in `/etc/fstab` verwendet werden. Beispiele:
 - ▶ `rw|ro`: Schreib-/Lesemodus | schreibgeschützt
 - ▶ `atime|noatime`: Zugriffszeit aktualisieren | nicht aktualisieren
 - ▶ `nouser|user`: Mounten nur durch `root` | durch normale User
 - ▶ `async|sync`: Schreib-Caching | direktes Schreiben
 - ▶ `acl`: Access Control Lists verwenden

mv

Dateien verschieben

Syntax

```
mv [Optionen] Quelldatei|-muster Ziel
```

Falls das Ziel ein existierendes Verzeichnis ist, werden die Dateien unter ihrem bisherigen Namen dorthin verschoben, andernfalls wird das Ziel als Dateiname interpretiert.

Optionen

- ▶ `-f, --force`
Überschreiben erzwingen
- ▶ `-i, --interactive`
Vor eventuellem Überschreiben nachfragen
- ▶ `-v, --verbose`
Ausführliche Ausgabe

netstat

Netzwerkinformationen anzeigen

Syntax

```
netstat [Optionen]
```

Ohne Angabe von Optionen werden alle verfügbaren Informationen angezeigt. Mithilfe der Optionen können Sie dagegen festlegen, welche Daten ausgegeben werden sollen.

Optionen

- ▶ `-a`
Alle Informationen der gewählten Kategorien anzeigen. Beispiel: `-ta` zeigt TCP-Verbindungen samt Serverdiensten an.
- ▶ `-n`
Ausführlichere Ausgabe
- ▶ `-r`
Die Routing-Tabelle anzeigen

- ▶ `-t`
TCP-Verbindungen ausgeben (standardmäßig ohne Serverdienste).

nice *

Ein Programm mit veränderter Priorität starten.

Syntax

```
nice [Optionen] [Kommando [Argumente]]
```

Optionen

- ▶ `-n Wert, --adjustment=Wert`
Der Wert kann von `-20` (höchste Priorität) bis `19` (niedrigste Priorität) reichen. Wenn Sie keinen Wert angeben, wird `10` gewählt, sodass ein auf diese Art gestartetes Programm sich »netter« benimmt als mit seiner Standardpriorität `0` – daher der Name des Kommandos. Beachten Sie, dass nur `root` das Kommando `nice` mit Prioritätswerten unter `0` aufrufen darf.

nslookup

Nameserver befragen

Syntax

```
nslookup [Optionen] Abfrage [Server]
```

Die Abfrage ist in der Regel ein Domainname, zu dem Sie die IP-Adresse ermitteln möchten, oder aber die IP-Adresse, zu der Sie den Namen suchen (Reverse Lookup). Wenn Sie bei der Systemkonfiguration keine Nameserver festgelegt haben, müssen Sie als letztes Argument die IP-Adresse eines solchen angeben.

passwd

Passwörter und Benutzerinformationen ändern

Syntax

```
passwd [Optionen] [User]
```

Normale Benutzer können nur ihr eigenes Passwort ändern, während `root` einen Benutzernamen angeben kann, um das Passwort eines anderen Users zu modifi-

zieren – sogar ohne Kenntnis des bisherigen. In jedem Fall muss das neue Passwort ohne jegliches optisches Feedback zweimal identisch eingegeben werden.

Optionen

- ▶ -f
Ändert die ausführliche Benutzerinformation (GECOS).
- ▶ -g
Ändert das Passwort einer Gruppe statt eines Benutzers.
- ▶ -l
Sperrt das angegebene Benutzerkonto (nur root).
- ▶ -s
Ändert die Login-Shell eines Benutzers.
- ▶ -u
Entsperrt das angegebene Konto wieder (nur root).

ping

Die Erreichbarkeit von Hosts im Netzwerk testen

Syntax

```
ping [Optionen] Host
```

Ohne Optionen wird ein ICMP-Testpaket nach dem anderen an den gewünschten Host gesendet. Nach Betätigung von `[Strg]+[C]` wird eine Statistik angezeigt.

Optionen

- ▶ -b
ping an eine Broadcast-Adresse senden
- ▶ -c Anzahl
Nach der angegebenen Anzahl von Paketen beenden
- ▶ -t Wert
Den TTL-Wert der gesendeten IP-Datagramme auf den angegebenen Wert setzen
- ▶ -W Sekunden
Timeout in Sekunden

popd

In das zuletzt auf den `pushd`-Stack gelegte Verzeichnis wechseln

Syntax

```
popd
```

Näheres siehe → `pushd`

ps

Liste der laufenden Prozesse anzeigen

Syntax

```
ps [Optionen]
```

Die GNU-Version von `ps` unterstützt drei Arten von Optionen: POSIX (mit führendem `-`), BSD (ohne `-`) und GNU-Langformat (mit führendem `--`). In der Regel wird die BSD-Kombination `aux` verwendet: `a` = alle Prozesse (nicht nur die eigenen), `u` = benutzerorientierte Ausgabe, `x` = auch Prozesse ohne zugeordnetes Terminal (Daemons).

pstree

Baumdiagramm der laufenden Prozesse und ihrer Abhängigkeiten anzeigen

Syntax

```
pstree [Optionen] [PID]
```

Falls eine PID angegeben wird, dient diese als Wurzel, andernfalls `init`.

pushd

Verzeichnis auf einem Stack ablegen und dorthin wechseln

Syntax

```
pushd [Verzeichnis]
```

`pushd` legt Verzeichnisse auf einem speziellen Stack (Last in, first out) ab. Wird kein Verzeichnis angegeben, dann verwendet `pushd` das aktuelle Arbeitsverzeich-

nis. Ein Aufruf von `→ popd` nimmt das zuletzt auf den Stack gelegte Verzeichnis und wechselt wie `→ cd` hinein.

pwd

Das aktuelle Arbeitsverzeichnis anzeigen

Syntax

```
pwd
```

Normalerweise ist der Name des aktuellen Arbeitsverzeichnisses Teil des Prompts. Nützlich ist dieses Kommando zum Beispiel im Home-Verzeichnis des derzeitigen Benutzers, da es im Prompt nur als `~` angezeigt wird, oder in Shell-Skripten.

rm

Dateien löschen

Syntax

```
rm [Optionen] Datei|Muster [Datei|Muster ...]
```

Optionen

- ▶ `-f, --force`
Erzwingt das Löschen auch schreibgeschützter Dateien.
- ▶ `-i, --interactive`
Fragt vor jedem einzelnen Löschvorgang nach.
- ▶ `-r, -R, --recursive`
Rekursives Löschen in Unterverzeichnissen

rmmod

Kernel-Modul entfernen

Syntax

```
rmmod [Optionen] Modul
```

Normalerweise ist `→ modprobe -r` zu bevorzugen.

runlevel

Anzeige der aktuellen Runlevels

Syntax

```
runlevel
```

scp

Dateien von einem oder auf einen Remote-Rechner kopieren

Syntax

```
scp [Optionen] [[user@]host:]Quelle [[user@]host:]Ziel
```

Kopiert Dateien per → ssh-Verbindung von einem Host auf einen anderen. Für Remote-Hosts wird der Hostname vor dem durch einen Doppelpunkt davon getrennten Pfad angegeben. Kopien sind von einem entfernten auf den lokalen Rechner oder umgekehrt möglich, oder sogar zwischen zwei verschiedenen, entfernten Rechnern. Wenn auf dem Remote-Host unter einem anderen Usernamen als lokal gearbeitet werden soll, wird dieser als `user@` vor den Hostnamen geschrieben. Die Pfadangaben auf Remote-Hosts sind meist absolut, beginnen also mit `/`; relative Pfade sind nur möglich, wenn der angegebene User ein Home-Verzeichnis auf dem entsprechenden Rechner besitzt.

Optionen

- ▶ `-r`
Unterverzeichnisse rekursiv kopieren

shred *

Dateien mehrfach überschreiben und gegebenenfalls löschen

Syntax

```
shred [Optionen] Datei|Muster [Datei|Muster ...]
```

Die angegebenen Dateien werden mehrfach mit zufälligen Inhalten überschrieben, um ihre Wiederherstellung nach dem Löschen fast unmöglich zu machen.

Optionen

- ▶ `-n Anzahl, --iterations=Anzahl`
Anzahl der Durchgänge (standardmäßig 25)
- ▶ `-u, --remove`
Datei nach dem Überschreiben auch löschen
- ▶ `-z, --zero`
Nach dem Überschreiben mit Zufallswerten noch einmal mit Nullen überschreiben, um auch den Shredding-Vorgang (oberflächlich) zu verbergen.

shutdown

Das System herunterfahren oder neu starten

Syntax

```
shutdown [Optionen] now|Zeitangabe
```

Optionen

- ▶ `-h`
Herunterfahren (halt), das heißt bei neueren Systemen ausschalten
- ▶ `-k`
Nicht wirklich herunterfahren, aber die entsprechende Warnung an alle Benutzer versenden
- ▶ `-r`
Neu starten (restart)
- ▶ `now`
Gewünschte Aktion sofort durchführen
- ▶ `Zeitangabe`
Entweder `hh:mm` für eine konkrete Uhrzeit oder `+m` zum Herunterfahren nach `m` Minuten

ssh

Secure-Shell-Verbindung herstellen.

Syntax

```
ssh [Optionen] Host
```

Optionen

- ▶ -l Benutzername
Nicht mit dem aktuellen lokalen Username auf dem Fernsystem einloggen, sondern mit dem angegebenen. Alternative: `username@Host` statt des einfachen Hostnamens.
- ▶ -p Port
TCP-Port auf dem Remote-Rechner (standardmäßig 22)

sort

Zeilen von Textdateien sortieren

Syntax

```
sort [Optionen] [Datei|Muster ...]
```

Wenn keine Datei angegeben wird, dann wird von der Standardeingabe gelesen. Geschrieben wird stets auf die Standardausgabe.

Optionen

- ▶ -f, --ignore-case
Groß- und Kleinschreibung ignorieren
- ▶ -n, --numeric-sort
Numerisch statt alphabetisch sortieren
- ▶ -r, --reverse
Sortierreihenfolge umkehren

su

Als anderer Benutzer arbeiten

Syntax

```
su [Optionen] [Benutzer]
```

Falls kein Benutzername angegeben wurde, wird automatisch `root` gewählt.

Optionen

- ▶ -c Kommando, --command=Kommando
Nicht dauerhaft als anderer Benutzer arbeiten, sondern nur diesen einen Befehl unter dessen UID ausführen

- ▶ -, -l, --login
Login-Shell und Umgebung des neuen Benutzers laden

tail

Nur das Ende von Textdateien anzeigen

Syntax

```
tail [Optionen] [Datei|Muster ...]
```

Standardmäßig werden die letzten 10 Zeilen jeder Datei ausgegeben. Wird keine Datei angegeben, dann gibt `tail` den Schluss der Standardeingabe aus.

Optionen

- ▶ -c=[-]N, --bytes=[-]N
Nur die letzten N Byte des Textes ausgeben
- ▶ -f, --follow
Die Ausgabe offenhalten und neu hinzukommende Zeilen ausgeben
- ▶ -n=[-]N, --lines=[-]N
Nur die letzten N Zeilen des Textes ausgeben
- ▶ -Zahl
Kurzfassung für -N=Zahl; -4 gibt beispielsweise die letzten drei Zeilen aus

tar

Mehrere Dateien in eine Archivdatei packen

Syntax

```
tar Option[en] Archivdatei [Dateimuster ...]
```

Optionen

Bei `tar` können die Minuszeichen vor den Optionen weggelassen werden.

- ▶ -c
Die angegebenen Dateien in die Archivdatei verpacken. Schließt die Optionen -x und -t aus.
- ▶ -x
Das Archiv entpacken; in diesem Fall brauchen keine Dateien angegeben zu werden. Schließt -c und -t aus.

- ▶ -t
Nur das Inhaltsverzeichnis des Archivs auflisten. Schließt -c und -x aus.
- ▶ -z
Das Archiv nach dem Verpacken mit → `gzip` komprimieren beziehungsweise vor dem Auspacken entkomprimieren. Diese Option schließt -j aus.
- ▶ -j
Das Archiv nach dem Verpacken mit → `bzip2` komprimieren beziehungsweise vor dem Auspacken entkomprimieren. Diese Option schließt -z aus.
- ▶ -v
Ausführliche Informationen anzeigen – insbesondere die Liste der verarbeiteten Dateien.
- ▶ -f
Option für die Dateiliste; wird üblicherweise als letzte Option angegeben.

tee

Standardeingabe auf die Standardausgabe und in eine Datei schreiben

Syntax

```
tee [Optionen] Dateiname
```

Normalerweise wird `tee` über eine Pipe an ein Kommando angehängt, um dessen Ausgabe gleichzeitig lesen und protokollieren zu können.

telnet

Terminalverbindung zu einem bestimmten Host/Dienst

Syntax

```
telnet [Optionen] Host [Port]
```

Wird kein TCP-Port angegeben, so versucht `telnet`, den gleichnamigen Dienst (Port 23) zu erreichen. Dieser ist heutzutage allerdings meist deaktiviert; verwenden Sie stattdessen → `ssh`. Für alle anderen Dienste können Sie einen Portnamen aus `/etc/services` oder die entsprechende Nummer angeben. Eine entsprechende Tabelle mit Beispielen finden Sie in Kapitel 13, »Netzwerkgrundlagen«.

time *

Ausführungsdauer von Kommandos messen

Syntax

```
time [Optionen] Kommando [Parameter ...]
```

Wenn Sie zum Ausprobieren etwas messen möchten, das recht lange dauert, könnten Sie zum Beispiel dieses Kommando eingeben:

```
# time ls -R /
```

umount

Gemountete Dateisysteme abkoppeln

Syntax

```
umount [Optionen] Verzeichnis|Gerät
```

Es kann sowohl das Verzeichnis angegeben werden, unter dem das Gerät eingebunden war, als auch das Gerät selbst.

uname

Systeminformationen ausgeben

Syntax

```
uname [Optionen]
```

Optionen

- ▶ -a
Alle verfügbaren Informationen
- ▶ -i
Hardwareplattform
- ▶ -m
Maschine
- ▶ -n
Hostname

- ▶ -o
Betriebssystem
- ▶ -p
Prozessor
- ▶ -r
Kernel-Release
- ▶ -s
Kernel-Name
- ▶ -v
Kernel-Version

uniq

Aufeinanderfolgende identische Zeilen in Textdateien unterdrücken

Syntax

```
uniq [Optionen] [Eingabedatei [Ausgabedatei]]
```

Wenn keine Dateien angegeben werden, wird von `stdin` gelesen und auf `stdout` geschrieben – häufig wird die Ausgabe von `→ sort` anschließend durch `uniq` gepipet.

Optionen

- ▶ -d, --repeated
Kehrt das Verhalten von `uniq` um: Gibt *nur* die Duplikate aus.
- ▶ -f N, --skip-fields=N
Die ersten N (durch Whitespace getrennten) Felder beim Vergleich ignorieren
- ▶ -i, --ignore-case
Groß- und Kleinschreibung ignorieren
- ▶ -s N, --skip-chars=N
Die ersten N Zeichen ignorieren

unix2dos *

Konvertiert UNIX-Textdateien (Zeilenumbrüche) in das DOS/Windows-Format.

Syntax

```
unix2dos [Optionen] Datei|Muster [Neue_Datei]
```

Optionen

- ▶ `-o, --oldfile`
Konvertiert die angegebene Datei an Ort und Stelle (ersetzt sie) – dies ist der Standardmodus, in dem Sie auch Dateimuster angeben können.
- ▶ `-n, --newfile`
Speichert die angegebene Quelldatei konvertiert in die als Zweites angegebene Zieldatei. In diesem Modus dürfen Sie auf keinen Fall Muster verwenden, weil der erste Treffer ansonsten den zweiten überschreibt.

unzip

Siehe → `zip`

uptime

Statistik, wie lange das System bereits läuft und wie intensiv es zurzeit genutzt wird

Syntax

`uptime`

useradd

Erstellt neue Benutzerkonten

Syntax

`useradd [Optionen] Benutzername`

Optionen

- ▶ `-d` Verzeichnis
Home-Verzeichnis (statt `/home/Username`)
- ▶ `-g` Gruppe
Primäre Gruppe, der der Benutzer angehören soll
- ▶ `-G` Gruppe1,Gruppe2,...
Zusätzliche Gruppenmitgliedschaften
- ▶ `-m`
Das Home-Verzeichnis neu anlegen, falls es noch nicht existiert

- ▶ `-o`
Das Benutzerkonto auch dann anlegen, wenn die UID bereits existiert (nur sinnvoll in Verbindung mit `-u`)
- ▶ `-s Shell`
Die (Text-)Login-Shell des Benutzers; standardmäßig `/bin/bash`
- ▶ `-u UID`
Eine bestimmte numerische User-ID angeben (andernfalls wird sie automatisch vergeben)

userdel

Benutzerkonten löschen

Syntax

```
userdel [Optionen] Username
```

wc

Zeichen, Wörter und Zeilen zählen

Syntax

```
wc [Optionen] [Datei|Muster ...]
```

Wenn keine Dateien angegeben werden, dann wird in der Standardeingabe gezählt.

Optionen

- ▶ `-c, --bytes`
Nur Bytes zählen (früher identisch mit Zeichen, heute je nach Zeichensatz)
- ▶ `-m, --chars`
Nur Zeichen zählen (unter Berücksichtigung des Zeichensatzes)
- ▶ `-l, --lines`
Nur Zeilen zählen
- ▶ `-w, --words`
Nur Wörter zählen

whatis

Kurzbeschreibungen aller Manpages mit bestimmtem Titel anzeigen

```
whatis [Optionen] Schlüsselwort
```

Entspricht einem Aufruf von `→ man` mit der Option `-f`.

whereis

Sucht Pfad, Konfigurationsdateien und Manpages eines Kommandos.

Syntax

```
whereis [Optionen] Kommando
```

which

Listet den Pfad eines Kommandos auf.

Syntax

```
which Kommando
```

who

Liste der eingeloggten User

Syntax

```
who [Optionen]
```

Optionen

- ▶ `-a`
Mehr Informationen anzeigen (auch Terminals, die auf Login warten usw.)
- ▶ `-m`
Nur den Benutzer der aktuellen Shell anzeigen

whoami

Die eigene User-ID ausgeben

Syntax

Whoami

wodim

Daten oder Audio-Tracks auf CD beziehungsweise DVD brennen

Syntax

```
wodim [Optionen] [-audio|-data] [Track [Track ...]]
      [-audio|-data] [Track [Track ...]] ...
```

Optionen

- ▶ `dev=Gerätedatei`
Gerätedatei des zu verwendenden Brenners gemäß → `mount`-Konventionen
- ▶ `blank=all|fast|...`
Löschen eines wiederbeschreibbaren Mediums komplett | so schnell wie möglich
- ▶ `-v`
Ausführliche Informationen anzeigen
- ▶ `-dummy`
Brennvorgang simulieren
- ▶ `-data`
Nachfolgende Tracks als Datentracks interpretieren
- ▶ `-audio`
Nachfolgende Tracks als Audiotracks interpretieren

zip

Dateien im ZIP-Format komprimieren (und wieder extrahieren)

Syntax

```
zip [Optionen] Archivdatei [Datei|Muster ...]
unzip [Optionen] Archivdatei
```

`zip` verpackt alle angegebenen Dateien in die Archivdatei im ZIP-Format, das vor allem unter Windows verbreitet ist. `unzip` entpackt die angegebene Archivdatei; die Option `-l` zeigt nur das Inhaltsverzeichnis an, ohne das Archiv tatsächlich auszupacken.

*Da man nicht universal sein und alles wissen kann, was von allen
gewusst werden kann, muss man ein wenig von allem wissen.
– Blaise Pascal*

A Glossar

Access Point Zentraler Verteilerknoten in einem → Wireless LAN.

ADSL → DSL.

AGP Accelerated Graphics Port; spezieller Steckplatz für die Grafikkarte auf dem Mainboard.

Alphakanal Farbkanal, der den Transparenzgrad jedes einzelnen Pixels in einem Bitmap-Bild regelt.

ALU Arithmetic-Logical Unit, die Rechen-
einheit eines Mikroprozessors.

ANSI American National Standards Institute, legt Normen und Industriestandards in den USA fest und entspricht somit der → DIN in Deutschland. Im Computerbereich sind beispielsweise der ANSI-C-Standard und der ANSI-Zeichensatz, die erste standardisierte 8-Bit-Erweiterung von ASCII, relevant.

Anti-Alias Kantenglättung bei der Bildschirmdarstellung von Grafiken. Da Pixel quadratisch sind, sehen Kurven und diagonale Linien in einer bestimmten Farbe treppenförmig aus. Dem wird durch zusätzliche Pixel entgegengewirkt, deren Farbe einen Mittelwert zwischen den kontrastierenden Pixeln bildet.

Antiqua Grundform der Druckschrift seit der Renaissance; Gegenbegriff zur Fraktur. Innerhalb der Antiqua-Gruppe sind allerdings viele verschiedene Formen zu unterscheiden.

Apache Der Apache HTTP Server ist der mit Abstand verbreitetste → Webserver der Welt. Seine Entwicklungsorganisation, die Apache Software Foundation, betreut daneben unzählige andere beliebte Open-Source-Projekte, etwa die JSP/Servlet-Engine Tomcat, SpamAssassin, Ant, Xalan und viele mehr.

API Application Programming Interface; Schnittstelle, über die Programmierer auf die Funktionen eines Betriebssystems oder einer Anwendung zugreifen können.

Application Server Serversoftware, die eine Umgebung für verteilte Anwendungen bereitstellt.

ARP Address Resolution Protocol; Protokoll der TCP/IP-Familie zur Umsetzung der IP-Adressen in die Hardware-Adressen von Ethernet-Karten.

ARPA Auch DARPA: Defense Advanced Research Projects Agency. Agentur des US-Verteidigungsministeriums, die Forschungsprojekte koordiniert und finanziert. Das 1969

von dieser Institution ins Leben gerufene ARPANet war der Vorläufer des Internets.

ASCII Erster weitverbreiteter, standardisierter Computerzeichensatz mit 7 Bit Datenbreite (128 Zeichen), ausreichend für alle Buchstaben der englischen Sprache sowie Ziffern, Satzzeichen, einige Sonderzeichen und Spezialfunktionen.

ASCII-/Binärmodus Unterschiedliche Datenübertragungsverfahren, etwa bei → FTP: Beim ASCII-Modus werden Zeilenumbrüche in das passende Format für die jeweilige Plattform umgewandelt, beim Binärmodus bleiben die Daten absolut intakt.

Assembler Die Maschinensprache des jeweiligen Prozessors, dargestellt in Form von Kürzeln (Mnemonics), die von Menschen gelesen werden können. Außerdem die Bezeichnung für ein Programm, das diese Schreibweise in die eigentliche Maschinensprache umwandelt.

AT-Befehlssatz Auch Hayes-Befehlssatz; Satz ASCII-basierter Steuerbefehle für fast alle Modems, die alle mit der Zeichenfolge AT beginnen.

Ausdruck Beliebige Verknüpfung aus Konstanten, Variablen und Operatoren in einer Skript- oder Programmiersprache, die letztendlich einen konkreten Wert ergibt.

Backdoor Unerwünschte Schnittstelle, über die ein Angreifer auf einem fremden Rechner Operationen durchführen kann. Meist durch → Trojaner installiert.

Berkeley Socket API Programmierumgebung für TCP/IP-Netzwerkanwendungen. Ursprünglich eine C-Bibliothek unter BSD-UNIX, inzwischen aber unter vielen Betriebssystemen und Programmiersprachen verfügbar.

Bezeichner Name einer Variablen, Funktion, Klasse oder Methode. Je nach Programmiersprache gelten unterschiedliche Konventionen für Bezeichner.

BGP Border Gateway Protocol; sehr häufig verwendetes externes Routing-Protokoll.

Big-Endian-Architektur Spezielle Hardware-Architektur: Bei Zahlen, die mehrere Byte breit sind, wird das höchstwertige Byte am Ende gespeichert. Gegenteil: → Little-Endian-Architektur.

Bildauflösung Pixeldichte in einem Bitmap, gemessen meist in Pixeln pro cm oder pro Inch. Die Ausgabeauflösung für den Druck wird dagegen in dpi (Dots per Inch) beziehungsweise Dots pro cm angegeben.

BIOS Basic Input/Output System; ROM-Baustein mit der Firmware des Personal Computers, die die wichtigsten Hardwarekomponenten überprüft und den Start des Betriebssystems einleitet.

Bitmap Computergrafik, bei der einzelne, farbige Bildpunkte (Pixel) gespeichert werden. Besonders geeignet für digitalisierte Fotos und andere Halbtonbilder.

Blue Book Erweiterter Standard für die Mixed-Mode-CD (CD mit Audio- und Daten-Tracks).

Bridge Erweiterung eines Netzwerk-Hubs, die eine Verbindung zwischen zwei verschiedenen Verkabelungsstandards bereitstellt. Beispiel: Ein 10/100 Base T-Hub mit zusätzlichem BNC-Anschluss für 10 Base 2.

Brute-Force-Attacke »Rohe Gewalt«: Versuch, ein Passwort oder einen sonstigen Code durch einfaches Durchprobieren aller Kombinationsmöglichkeiten zu knacken.

BSD Berkeley System Distribution; eine der beiden grundsätzlichen UNIX-Entwicklungslinien (die andere ist → System V). Entstand durch die Weiterentwicklung des ursprünglichen UNIX an der University of California, Berkeley.

Bus Mastering Moderne Technologie für den direkten Austausch von Daten zwischen Arbeitsspeicher und Peripherie; entlastet die CPU.

C Imperative Programmiersprache, die 1971 geschaffen wurde, um das Betriebssystem UNIX plattformübergreifend zu implementieren.

C-Standardbibliothek Plattformübergreifend standardisierte Bibliothek von C-Funktionen. Da fast alle Betriebssysteme in C programmiert wurden, ist die Standardbibliothek der Grund dafür, dass vieles unter verschiedenen Systemen sehr ähnlich funktioniert.

Child-Prozess Ein Prozess, der durch den Systemaufruf `fork()` von einem anderen abgeleitet wurde.

CIDR-Adressierung Classless Inter-Domain Routing; Standard für die IP-Adressierung, die die Trennung zwischen Netzwerk- und Host-Teil der Adresse an einer beliebigen Stelle ermöglicht.

Cookies Informationen im Textformat, die ein Browser im Auftrag eines Webservers speichert und ihm beim nächsten Besuch wieder zurückgibt. Dienen der Speicherung von Konfigurationsdaten über eine Sitzung hinweg.

CSMA/CA Carrier Sense Multiple Access with Collision Avoidance. Netzzugangsverfahren, bei dem mehrere Stationen auf dasselbe Übertragungsmedium zugreifen. Vor

dem eigentlichen Senden erfolgt eine Sendeankündigung, um Datenkollisionen zu vermeiden. Wird zum Beispiel für Wireless LAN verwendet.

CSMA/CD Carrier Sense Multiple Access with Collision Detection. Netzzugangsverfahren, vor allem für Ethernet; vergleichbar mit CSMA/CA. Allerdings kann es zu Datenkollisionen kommen, weil eine Station einfach sendet, sobald das Medium frei ist.

CSS Cascading Stylesheets; spezielle Sprache für die Formatierung von Inhalten auf Webseiten.

CUPS Common UNIX Printing System; Druckumgebung für den standardisierten Druckerzugriff unter verschiedenen UNIX-Systemen.

Daemon Im Hintergrund laufender Prozess, der eine beliebige Dienstleistung zur Verfügung stellt, meist einen Serverdienst.

DARPA → ARPA.

Datagramm IP-Datenpaket.

Dateisystem Die spezifische Organisation von Dateien und Verzeichnissen auf einem Datenträger. Linux und andere UNIX-Systeme verwenden ein virtuelles Dateisystem für die gemeinsamen Eigenschaften aller Datenträger und dateiartigen Elemente; für die konkreten Dateisysteme kommen Kernel-Treiber zum Einsatz.

Deadlock Unauflösbare Verklemmung zweier Prozesse, die beim Wettstreit (Race Condition) um eine Ressource in einen Zustand geraten, in dem sie sich gegenseitig am Weiterkommen hindern.

Default Gateway Router, der nicht für die Verbindung zu einem speziellen Netzwerk verantwortlich ist, sondern für all diejenigen Verbindungen, für die kein anderer Router konfiguriert ist.

Denial-of-Service-Angriff Attacke auf einen Server, die diesen so stark mit (oft fehlerhaften) Anfragen überlastet, dass er nicht mehr in der Lage ist, seinen eigentlichen Dienst zu versehen.

DHCP Dynamic Host Configuration Protocol; ermöglicht es Hosts, ihre IP-Konfiguration beim Booten von einem zentralen Server zu laden.

Dicke Gesamte Breite, die ein bestimmter Buchstabe einer Schrift einnimmt.

DIN Deutsches Institut für Normung; zentrale Autorität für Normen und Industriestandards in Deutschland. Bekannt sind beispielsweise die DIN-Papierformate, die Schriftklassifizierung nach DIN oder das DIN-66001-Flussdiagramm.

Disc-at-once Technik, bei der der gesamte Brennvorgang einer Audio-CD ohne Unterbrechung durchgeführt wird. Einige ältere Audio-CD-Player bestehen auf solchen CDs.

Distribution Zusammenstellung aus dem Linux-Kernel, den GNU-Tools und zahlreichen Anwendungsprogrammen mit einem mehr oder weniger komfortablen Installer. Wird entweder zum Download angeboten oder mit Datenträgern und gedruckten Handbüchern verkauft. openSUSE, das Thema dieses Buches, ist eine der erfolgreichsten Linux-Distributionen.

DMA-Kanal Direct Memory Access ist eine ältere Technik für den direkten Datenaustausch zwischen RAM und Peripherie (→ Bus Mastering).

DNS Domain Name System; hierarchisches Namenssystem für Internet-Hosts; definiert Namen wie *www.galileocomputing.de*.

DoubleWord (DWord) Allgemeine Bezeichnung für 32 Bit (4 Byte oder 2 Words).

DSL Digital Subscriber Line; Internetzugang über serielle Leitungen mit hoher Geschwindigkeit dank hochfrequenter Signale. ADSL (Asymmetric DSL) bietet eine hohe Download-, aber niedrige Upload-Datenrate (eher für Privatkunden geeignet), während bei SDSL (Symmetric DSL) beide Geschwindigkeiten gleich sind.

DTD Document Type Definition; eine Vorlage für ein spezielles SGML- oder XML-Dokumentformat.

EIDE Enhanced Integrated Device Electronics; Standardanschluss für interne Laufwerke. Wird immer häufiger durch → S-ATA ersetzt.

Escape-Sequenz Spezielle Schreibweise eines nicht darstellbaren Zeichens in einem String. In C und verwandten Sprachen sowie in der Shell zum Beispiel durch einen Backslash realisiert: \n ist ein Zeilenumbruch, \" ein Anführungszeichen innerhalb normaler Anführungszeichen und so weiter.

Ethernet Standard für lokale Netzwerke, der das CSMA/CD-Verfahren verwendet. Hardwaretechnisch über verschiedene Kabeltypen realisiert, vor allem Twisted-Pair- oder Koaxialkabel.

Ext2 Das Linux Extended File System 2 ist das klassische → Dateisystem für Linux-Systeme. Es bietet gute Performance und Stabilität, besitzt aber kein → Journal.

Ext3 Um ein → Journal erweiterte, aber voll abwärtskompatible Neufassung von → Ext2.

Wurde von openSUSE 10.2 bis 11.1 standardmäßig für alle Festplattenpartitionen verwendet.

Ext4 Nochmals erweiterte Fassung des Linux-Standarddateisystems; wichtigste Besonderheit ist die Unterstützung sehr großer Partitionen (bis 1 Exabyte). Wird seit openSUSE 11.2 als Standard eingesetzt.

Farbaddition Mischverfahren der Lichtfarben. Die drei Primärfarben Rot, Grün und Blau beleuchten eine dunkle Fläche (zum Beispiel den Bildschirm) und vermischen sich zu allen erforderlichen Farben.

Farbkanäle Aufteilung der Grundfarben eines Farbmodells in einzelne Komponenten, die separat gespeichert werden.

Farbseparation Umrechnung von RGB-Farbe in CMYK-Farbe für die Ausgabe auf einem speziellen Gerät (Drucker, Belichter und so weiter). Je nach Verfahren können unterschiedliche Ergebnisse herauskommen.

Farbsubtraktion Mischverfahren der Körper- oder Pigmentfarben. Die Grundfarben Cyan, Magenta und Gelb sowie die Kontrastfarbe Schwarz werden durch Übereinanderdruck zu verschiedenen Farben gemischt.

Farbtiefe Bit-Breite der → Farbkanäle beziehungsweise der Farbwerte eines Bildes.

FAT File Allocation Table, Dateizuordnungstabelle. Sammelname für verschiedene Windows-Dateisysteme (FAT16, FAT32).

FIFO First in, First out; Speichermodell der Queue (Warteschlange). Daten werden am einen Ende hineingeschoben und am anderen herausgeholt.

Firewall Hard- oder Software-Filter, der zwischen zwei Netzwerken (zum Beispiel einem lokalen Netzwerk und dem Internet) arbeitet, um Datenpakete nach einstellbaren Kriterien wie Adressen, Ports oder Inhalten auszufiltern.

FireWire Auch IEEE 1394; schnelle serielle Leitung (400 MBit/s); wird für den digitalen Videoschnitt, aber auch immer häufiger für den Anschluss externer Laufwerke verwendet.

Fließkommazahl Im wissenschaftlichen Format (Exponentialschreibweise) gespeicherte Zahl, bei der der Exponent die (variable) Position des Kommas bestimmt.

Freie Software Software, die ohne Einschränkungen genutzt, weitergegeben und modifiziert werden darf. Unterliegt einem sogenannten Copyleft, das bestimmt, dass Weiterentwicklungen der Software ebenfalls wieder frei zugänglich gemacht werden müssen. Siehe auch → Open Source.

FSF Free Software Foundation; 1984 von Richard Stallman begründete Initiative zur Förderung und Verbreitung von → freier Software.

FTP File Transfer Protocol; Protokoll für die Übertragung von Dateien über ein TCP/IP-Netzwerk.

Funktion In der Mathematik eine Vorschrift, die für einen Eingabewert x einen bestimmten Ausgabewert $f(x)$ berechnet. In einem Computerprogramm ein in sich geschlossener Block von Anweisungen, der mit seinem Namen und eventuellen Parametern aufgerufen wird und einen Wert zurückgeben kann.

GNOME GNU Network Object Model Environment. Neben → KDE der andere weitver-

breitete Desktop für Linux-Systeme. Basiert auf der ursprünglich für das Bildbearbeitungsprogramm GIMP entwickelten Grafikbibliothek GTK+.

GNU Rekursives Akronym für »GNU's Not UNIX«. Projekt der → FSF zur Bereitstellung der UNIX-Standard-Tools als neu programmierte → freie Software.

Grafische Benutzeroberfläche Auch GUI (Graphical User Interface). Mit der Maus bedienbare Benutzerschnittstelle eines Betriebssystems, bei der Befehle durch Anklicken von Symbolen oder per Drag and Drop erteilt werden.

Green Book Standard für die CDi (CD-interactive), einen frühen Multimedia-Versuch von Philips.

Handshake Aushandlung der Leitungsmodalitäten zwischen zwei Modems. Drei-Wege-Handshake: Beginn einer TCP-Verbindung.

HTTP Hypertext Transfer Protocol. TCP/IP-Anwendungsprotokoll zur Übertragung von Webseiten vom Server an den Browser.

Hypertext Text mit eingebauten, automatischen Querverweisen, die durch Anklicken ein anderes Dokument öffnen oder an eine andere Stelle blättern.

I/O-Basisadresse Systemressource; die Speicheradresse, über die die CPU mit einem bestimmten Gerät kommuniziert.

IANA Internet Assigned Numbers Authority; zentrale Autorität für die Vergabe von IP-Adressen.

ICMP Internet Control Message Protocol. Einfaches Protokoll, das IP-Datagramme zu

Testzwecken an entfernte Rechner versendet, um Verbindungen zu überprüfen. Mit diesem Protokoll arbeitet das Dienstprogramm ping.

IEEE Institute of Electrical and Electronics Engineers; internationale Vereinigung von Elektronik-Ingenieuren, die bestimmte Standards beschließt. Bekannt sind etwa IEEE 802, eine Sammlung von Netzwerkstandards, oder IEEE 1394, der → FireWire-Anschluss.

IMAP Internet Message Access Protocol; modernes Protokoll für den Betrieb von E-Mail-Postfächern. In einem IMAP-Postfach können die Nachrichten beliebig in Ordner sortiert und gelagert werden.

Integer Ganze Zahl.

Intrusion Detection System (IDS) Software zur Erkennung böswilliger Angriffe oder Angriffsversuche.

IP (Internet Protocol) Das Vermittlungsprotokoll der TCP/IP-Familie. Übernimmt die Adressierung der Hosts und die Übertragung der Daten über verschiedene Netzwerke hinweg. Die klassische Version, IPv4, verwendet 32 Bit lange Adressen; da diese knapp zu werden drohen, wurde inzwischen IPv6 mit 128-Bit-Adressen entworfen.

IPC (Inter-Process Communication) Auch: System V IPC. Sammlung verschiedener Methoden zur Kommunikation zwischen → Prozessen, denen eigentlich vorgegaukelt wird, sie besäßen alle Systemressourcen exklusiv. Die Kommunikation wird beispielsweise durch gemeinsame Speicherbereiche (Shared Memory) oder durch → Pipes realisiert.

IrDA InfraRed Data Association. Standard für die Infrarot-Datenübertragung zwischen dem Computer und externen Peripheriegeräten.

IRQ Interrupt Request; Systemressource: Nummer eines Kommunikationskanals zur Ansteuerung von Peripheriegeräten.

ISA Industry Standard Architecture; historischer Steckplatz für Erweiterungskarten auf dem Mainboard. Nur 16 Bit breit und mit 8,33 MHz Taktrate ausgestattet.

ISDN Integrated Services Digital Network; europäischer Standard für digitale Telefonie. Ein ISDN-Anschluss bietet zwei separate Kanäle für Telefongespräche, Faxe oder Datenübertragung mit je 64 kBit/s (B-Kanäle) sowie einen separaten Kanal für Verwaltungsdaten (D-Kanal).

ISO International Organization for Standardization; erlässt internationale Normen und Industriestandards. Im IT-Bereich sind etwa die ISO-8859-Zeichensätze oder das CD-ROM-Dateiformat ISO 9660 relevant.

Joliet Microsoft-Erweiterung des ISO-9660-Dateiformats für die Verwendung langer Dateinamen.

Journal Einrichtung in modernen → Dateisystemen, die Änderungen protokolliert, um Datenverluste bei Abstürzen zu minimieren. Die wichtigsten Journaling-Dateisysteme unter Linux sind → Ext3 und → ReiserFS.

Kapitälchen Spezielles Alphabet, in dem speziell aufbereitete Großbuchstaben an die Stelle von Kleinbuchstaben treten.

KDE K Desktop Environment; Desktop-Arbeitsumgebung für Linux und andere UNIX-Systeme. Die zugrundeliegende Bibliothek ist Qt von der norwegischen Firma Trolltech.

Kernel Der innerste Funktionskern eines Betriebssystems. Die wichtigsten Aufgaben sind Steuerung der Hardware, Prozess- und Speichermanagement. Man unterscheidet

grundsätzlich zwei Architekturen: Die klassischen monolithischen Kernel erledigen alle Aufgaben selbst, während Mikrokern die meisten Aufgaben an Prozesse im Usermodus delegieren. Der Linux-Kernel ist monolithisch, profitiert aber von Modulen, die dynamisch geladen und wieder entfernt werden können. Die Bezeichnung »Linux« gilt eigentlich nur für den von Linus Torvalds und später zahlreichen Mitstreitern entwickelten Kernel.

Kerning Auch Unterschneidung oder Laufweitungsausgleich; Anpassung der Abstände zwischen speziellen Buchstabenkombinationen.

Kindprozess → Child-Prozess

Koaxialkabel Kabeltyp, bei dem eine innere leitende Schicht von einer Isolationsschicht und einer weiteren leitenden Schicht umgeben ist. Bekannt als Antennenkabel; wurde früher auch in der Netzwerktechnik eingesetzt.

Kommandozeile Benutzerschnittstelle im Dialogbetrieb, → Shell: Ein Benutzer gibt Befehle in Textform ein, die sofort ausgeführt werden.

Konsole → Kommandozeile.

LAMP-System Ein → Webserversystem, auf dem eine Kombination aus dem Betriebssystem Linux, dem Webserver Apache, der Datenbank MySQL und der Skriptsprache PHP beziehungsweise Perl oder Python laufen.

LAN Local Area Network; lokales Netzwerk.

LaTeX System für den professionellen Textsatz, besonders für wissenschaftliche Publikationen. Basiert auf dem Satzprogramm TeX von Donald E. Knuth.

LCD Liquid Crystal Display; elektronisches Anzeigesystem auf der Basis von Flüssigkristallen.

LDAP Lightweight Directory Access Protocol; gemeinsames Zugriffsprotokoll für verschiedene → Verzeichnisdienste. Basiert auf einer Hierarchie standardisierter Objekte und Attribute, dem LDAP-Schema.

LED-Drucker Vereinfachte Form des Laserdruckers, bei der die Belichtung durch eine Reihe von Leuchtdioden durchgeführt wird.

LIFO Last in, First out; Speichermodell des Stacks (Stapel). Daten werden »oben« aufgelegt und wieder heruntergeholt.

Linux-Distribution → Distribution.

Literal »Wörtlich« gemeinte Konstante in einem Computerprogramm.

Little-Endian-Architektur Spezielle Hardware-Architektur: Bei Zahlen, die mehrere Byte breit sind, wird das höchstwertige Byte am Anfang gespeichert. Gegenteil: → Big-Endian-Architektur.

Logdatei Auch Protokolldatei. Textdatei, in der ein Programm (vornehmlich ein → Daemon ohne regelmäßige Ausgabe) Operationen, Warnungen oder Fehler protokolliert. Auf UNIX-Systemen verwenden viele Programme den zentralisierten Dienst → Syslog dafür.

Login Benutzeranmeldung an einem UNIX-System.

MAC-Adresse Hardware-Adresse einer Ethernet-Karte und anderer Netzwerkgeräte.

MIDI Musical Instruments Digital Interface; Standard für die Kommunikation mit digitalen Musikinstrumenten (Synthesizern).

MIME Multipurpose Internet Mail Extension; Standard für die Angabe von Datentypen, der ursprünglich für E-Mail-Attachments entwickelt wurde, aber auch im Web eine wichtige Rolle spielt.

MMU Memory Management Unit; Teil der CPU, der für die Speicherverwaltung zuständig ist.

Mounten Einbinden eines Datenträgers in das Dateisystem beziehungsweise einen Verzeichnisbaum.

MP3 MPEG 1 Audio Layer 3; Format für die Audiokomprimierung, das auf hörphysiologischen Messungen basiert und so vornehmlich Bereiche weglässt, die von den meisten Menschen nicht gehört werden. Da das Format kommerziell ist, wird es in openSUSE durch mitgelieferte Nicht-Open-Source-Erweiterungen unterstützt.

MPEG Motion Picture Expert Group; diverse Formate zur effizienten Speicherung von Digitalvideo mit hoher Qualität.

MTU Maximum Transmission Unit; maximale Größe für IP-Datenpakete über eine bestimmte Leitung.

MySQL Die beliebteste → relationale Datenbank mit Open-Source-Lizenz.

NAT Network Address Translation; Umsetzung privater IP-Adressen in öffentliche zum Übergang in öffentliche Netze.

NFS Network File System; von Sun entwickeltes, klassisches Netzwerk-Dateisystem für UNIX-Systeme.

NIS Einfacher Verzeichnisdienst, ursprünglich von Sun entwickelt. Wird vor allem zur Zentralisierung der Benutzeranmeldung verwendet.

NNTP Network News Transport Protocol; Protokoll für die Übertragung von Newsgroup-Daten.

Normalisierung Herstellung der Normalform einer relationalen Datenbank, in der es keine Redundanzen und Inkonsistenzen gibt.

Nyquist-Theorem Auch Shannon-Theorem; besagt, dass für die Darstellung von Daten einer bestimmten Frequenz mindestens die doppelte Trägerfrequenz (Sampling-Rate) erforderlich ist.

O-Notation Darstellungsform für die Komplexität von Algorithmen: Eine Funktion $f(N)$ besitzt die Komplexitätsklasse $O(g(N))$, wenn sich $f(N)$ für große N der Funktion $g(N)$ annähert.

Objektorientierung Modernes Programmierverfahren, bei dem Datenstrukturen und Funktionen eine Einheit bilden (Kapselung).

OpenLDAP Freie Implementierung eines → LDAP-Verzeichnisdienstes.

Open Source Pragmatischere Bezeichnung für → freie Software. Der Quellcode muss auch hier zur Verfügung gestellt werden, zudem darf auch Open-Source-Software beliebig genutzt, weitergegeben und verändert werden. Anders als bei »echter« freier Software im Sinne der → FSF kann es allerdings veränderte Lizenzen geben, die der Herstellerfirma besondere (auch kommerzielle) Nutzungsrechte einräumen oder Ähnliches.

Orange Book Standard für die beschreibbare und die wiederbeschreibbare CD (CD-R beziehungsweise CD-RW).

OSI-Referenzmodell Standardmodell für die Netzwerkkommunikation mit sieben Schichten (Funktionsebenen).

OSPF Open Shortest Path First; Routing-Protokoll.

Parent-Prozess Ein → Prozess, aus dem ein anderer durch den Systemaufruf `fork()` abgeleitet wird.

Partitionierung Unterteilung der Festplatte in mehrere Abteilungen, die sich wie separate Datenträger verhalten.

PCI Peripheral Component Interface; aktueller Anschluss für Einsteckkarten auf dem Mainboard mit 32 Bit und 33 MHz.

PCMCIA Personal Computer Memory Card International Association; Anschluss für Memory-Cards und Erweiterungsmodule an Notebooks.

PDF Portable Document Format; Dateiformat von Adobe, das in der Druckvorstufe immer häufiger als PostScript-Ersatz verwendet wird. Auch im Web werden oft PDFs zum Download angeboten. Wichtigster Vorteil: Alle im Dokument verwendeten Schriften werden mit in die Datei gepackt.

Perl Practical Extraction and Report Language; Skriptsprache, die vor allem von Systemadministratoren und für die CGI-Programmierung verwendet wird.

PGP Pretty Good Privacy; beliebtes Tool zur Verschlüsselung und digitalen Signatur von E-Mails.

Phishing Diebstahl von persönlichen Daten, Anmeldeinformationen und Passwörtern mit Hilfe gefälschter E-Mails oder Websites.

PHP Rekursives Akronym für PHP: Hypertext Preprocessor; verbreitete Sprache für serverseitige Web-Anwendungen.

Pipe Verknüpfung der Ausgabe eines Programms mit der Eingabe eines anderen. Kann sowohl in der Shell als auch für → IPC verwendet werden.

Plug & Play Automatische Konfiguration von Hardware: Geräte werden beim Anschließen automatisch erkannt.

POP3 Post Office Protocol, Version 3. Häufigstes Protokoll für E-Mail-Postfächer.

POSIX Standardisierung der grundlegenden Eigenschaften, die ein UNIX-System besitzen muss.

PostScript Druckseitenbeschreibungssprache von Adobe; wird von hochwertigen Druckern und Belichtern direkt verstanden.

PPP Point-to-Point Protocol; Verfahren für den Netzwerkzugriff über eine serielle Stand- oder Wählleitung.

Primärschlüssel Eindeutiges Identifikationsmerkmal eines Datensatzes in einer relationalen Datenbank.

Prompt Eingabeaufforderung in einer Shell.

Proportionschrift Schrift, in der die einzelnen Zeichen je nach ihrer Breite unterschiedlich viel Platz einnehmen.

Protokolldatei → Logdatei.

Prozess Aufgabe, die von einem Betriebssystem scheinbar gleichzeitig (in Wirklich-

keit schnell abwechselnd) mit anderen Aufgaben durchgeführt wird. Jedem Prozess stehen aus seiner eigenen Sicht die vollständigen, ungeteilten Systemressourcen zur Verfügung; das System betreibt Prozessmanagement, um zwischen ihnen umzuschalten.

Punkt (Point) Maßeinheit für die Schriftgröße. Heute ist in der Regel der DTP-Punkt ($1/72$ Inch) gemeint, obwohl es eine Reihe leicht abweichender Einheiten gibt, die ebenfalls Punkt heißen.

Queue → FIFO.

Race Condition Wettstreit mehrerer → Prozesse um eine Systemressource.

RAID Redundant Array of Independent (oder Inexpensive) Disks. Logischer Zusammenschluss mehrerer Festplatten zur Steigerung der Geschwindigkeit oder der Sicherheit.

RDBMS Relational Database Management System. Programm zur Verwaltung relationaler Datenbanken.

Register Rechenzelle der CPU.

Reguläre Ausdrücke In verschiedenen Programmiersprachen und Anwendungen verfügbares Verfahren für Mustervergleiche zum Suchen und Ersetzen.

ReiserFS Ein → Dateisystem mit → Journal-Funktion; nutzt den Platz bei vielen kleinen Dateien besser aus als → Ext3. Wurde unter SUSE Linux bis 10.1 standardmäßig für alle Festplattenpartitionen verwendet.

Rekursion Problemlösung durch ineinander verschachtelte Aufrufe derselben Funktion.

Relationale Datenbank Datenbank, die aus mehreren miteinander verknüpften Tabellen besteht.

RFC Request for Comments; Dokument zur öffentlichen Darstellung von Internet-Protokollen und -Standards.

RISC Reduced Instruction Set Computing; CPU-Architektur, die durch einen stark verkleinerten Befehlssatz effizienter arbeitet.

Router Rechner, der eine Verbindung zwischen zwei verschiedenen Netzwerken vermittelt.

RPC Remote Procedure Call; Technologie zur Client-Server-Kooperation von SUN, auf der → NFS und → NIS basieren. Im weiteren Sinne werden auch andere Formen der netzwerkübergreifenden Kooperation als »RPC« bezeichnet (zum Beispiel XMLRPC als Grundlage von Web Services).

S-ATA Serieller Nachfolger des internen Massenspeicheranschlusses EIDE/ATAPI.

Samba SMB-Server für UNIX-Systeme; gewissermaßen eine freie Windows-Server-Implementierung.

Sampling Digitalisierung von Audio durch regelmäßige Messungen pro Sekunde.

SAS Serial Attached SCSI; moderner, serieller Nachfolger von → SCSI, äußerlich baugleich mit → S-ATA

Scheduler → Task Scheduler.

Schichtenmodell Modell der Netzwerkkommunikation oder sonstigen Datenübertragung, das verschiedene Funktionsebenen

unterscheidet – beim OSI-Referenzmodell zum Beispiel sieben.

SCSI Small Computer System Interface; ein paralleler Anschluss für interne und externe Peripherie, besonders wichtig für Festplatten in Server-Systemen. Wird immer öfter durch das serielle SAS ersetzt.

SDSL → DSL.

SGML Standard Generalized Markup Language; klassische Meta-Sprache zur Formulierung beliebiger Auszeichnungssprachen wie etwa das herkömmliche HTML.

Shannon-Theorem → Nyquist-Theorem.

Shell Dialogbasierte Benutzeroberfläche. Linux und andere UNIX-Systeme enthalten meist mehrere Shells wie `bash`, `tcsh` oder `pdsh`.

Signal UNIX-Mechanismus zur Benachrichtigung von Prozessen. Wird über den Systemaufruf `kill()` und das gleichnamige Shell-Kommando übermittelt.

Socket Ein Endpunkt der TCP/IP-Datenkommunikation. Neben Netzwerksokkets gibt es auch UNIX Domain Sockets, die über einen Pipe-ähnlichen Mechanismus eine Kommunikation zwischen lokalen Programmen ermöglichen.

Solid State Disk → SSD

SQL Structured Query Language; Standard-Abfragesprache für relationale Datenbanken.

SSD Moderner Festplattenersatz, der intern einen – von USB-Sticks oder Digitalkamera-Speicherkarten bekannten – Flash-Rom-Spei-

cher besitzt, nach außen aber einen Festplattenanschluss (S-ATA, SAS oder auch USB).

SSH Secure Shell; Sichere Weiterentwicklung von Telnet, deren Daten über eine verschlüsselte Verbindung übertragen werden.

Stack → LIFO.

Stack-Overflow Programmabsturz durch Überfüllung des System-Stacks, also des Stapelspeichers für Rücksprungadressen.

String Zeichenkette, also ein Textblock in einem Computerprogramm.

Subnet Mask Bitmaske, die angibt, welche Bits einer IP-Adresse das Netzwerk und welche den Host darstellen.

Subnetting Unterteilung eines IP-Netzes einer herkömmlichen Klasse in mehrere Teilnetze.

Supernetting Verbindung mehrerer IP-Netze einer herkömmlichen Klasse zu einem Gesamtnetz.

SVG Streaming Vector Graphics; XML-basiertes Format für Vektorgrafik; kann auch mit interaktiven und animierten Erweiterungen im Web verwendet werden.

Swap-Partition Spezielle Festplattenpartition in einem Linux-System, die zur temporären Auslagerung von Arbeitsspeicherinhalten verwendet wird.

Syslog Einrichtung in UNIX-Systemen für die automatische Protokollierung von Fehlern und sonstigen Ereignissen.

System V Das klassische AT&T-UNIX.

System V IPC → IPC.

Systemaufruf Aufruf einer Funktion, die der Kernel des Betriebssystems bereitstellt.

Systemvariable Auch Umgebungsvariable; eine systemweit gültige Variable, in der eine Einstellung des Betriebssystems gespeichert ist.

Task Scheduler Prozess im Kernel eines Betriebssystems, der die Rechenzeit an die unterschiedlichen Prozesse verteilt.

TCP Transmission Control Protocol; zuverlässiges Transportprotokoll der Internet-Protokollfamilie.

TCP/IP Transmission Control Protocol/Internet Protocol; die Internet-Protokollfamilie, benannt nach ihrem wichtigsten Transportprotokoll und ihrem Vermittlungsprotokoll.

Telnet Terminal-Emulation in einem TCP/IP-Netzwerk.

Terminal Endgerät eines Rechners seit den 60er Jahren: Ein- und Ausgabeeinheit; zunächst umgebauter Fernschreiber, später mit Monitor ausgestattet. Heute gibt es fast nur noch virtuelle Terminals, also Programme, die eine Terminal-Umgebung bereitstellen.

TFT Thin Flat Transistor; hochwertigstes Verfahren zur Steuerung von LCD-Displays: Jeder einzelne Flüssigkristall wird durch einen eigenen Transistor angesteuert.

Thread Leichtgewichtige Alternative zu einem Prozess; meist laufen innerhalb eines Prozesses mehrere Threads. Besonderheit: Ein Thread teilt sich einen Speicherbereich und alle Ressourcen mit den anderen Threads seiner Gruppe.

Timesharing Verfahren für den gleichzeitigen Betrieb mehrerer Terminals an einem Computer.

Top-Level-Domain Oberste Ordnungseinheit von Domainnamen im Internet. Man unterscheidet Länder-Top-Level-Domains (ccTLDs) wie *.de* für Deutschland oder *.fr* für Frankreich und generische TLDs wie *.com* oder *.org*.

Track-at-once Technik, bei der der Brennvorgang einer Audio-CD Track für Track einzeln durchgeführt wird. Gegenbegriff: Disc-at-once.

Transaktion Zusammenfassung mehrerer Operationen in einer → relationalen Datenbank zu einer in sich geschlossenen Gruppe; diese lässt sich anschließend insgesamt bestätigen (Commit) oder zurücksetzen (Rollback).

TrueType Vektorbasiertes Format für Schriften, entwickelt von Microsoft und Apple. TrueType-Fonts können gleichermaßen auf Bildschirm und Drucker ausgegeben werden.

TTL Time to Live; Anzahl der Router, die ein IP-Datagramm passieren kann: Jeder Router zieht 1 von diesem Wert ab; bei 0 wird das Paket verworfen. Dies verhindert, dass unzustellbare Datagramme ewig im Netz kreisen.

UDP User Datagram Protocol; schnelles, verbindungsloses Transportprotokoll der Internet-Protokollfamilie.

Unicode Neu entwickelter Computerzeichensatz mit dem Ziel, die Zeichen sämtlicher Schriftsysteme der Welt zu standardisieren. Die Zeichenbreite beträgt je nach Bereich 1 bis 4 Byte.

Unterschneidung → Kerning.

URL Uniform Resource Locator; eindeutige Adresse eines Dokuments im Internet. Besteht aus dem Zugriffsverfahren (Protokoll), dem Hostnamen und dem Dateipfad.

USB Universal Serial Bus; moderne serielle Schnittstelle für externe Peripheriegeräte.

UTF-8 Spezielle Schreibweise für Unicode-Zeichen, in der ASCII-Zeichen mit einem Byte auskommen, während alle anderen Zeichen zwei bis vier Byte belegen.

Vektorgrafik Computergrafik, bei der Kurven und Linien auflösungsunabhängig als mathematische Formeln gespeichert werden.

Verzeichnisdienst Verteilter Informationsdienst, der der Verbreitung von Informationen über Ressourcen und der zentralen Verwaltung von Zugriffsrechten dient. Beispiele: → OpenLDAP, → NIS, Active Directory.

VLSM Variable Length Subnet Mask; asymmetrische Aufteilung eines IP-Netzes in Teilnetze unterschiedlicher Größe.

W3C World Wide Web Consortium; Zusammenschluss von Herstellern und Einzelpersonen, der die Standards für World Wide Web, HTML und verwandte Technologien festlegt.

WAN Wide Area Network; Netzwerk, dessen Bereich ein größeres, überregionales Gebiet umfasst. Manchmal auch einfach der allgemeine Gegenbegriff zum LAN (lokalen Netzwerk).

Webserver Server für das Protokoll → HTTP, der Webseiten an Browser ausliefert. Wird häufig mit einer serverseitigen Programmiersprache (PHP, Perl oder andere) und einem Datenbankserver kombiniert, um

Webanwendungen auszuführen (siehe auch → LAMP).

White Book Standard für die Video-CD (nicht DVD).

Window-Manager Auf den → X Window Server aufsetzende Software zur Fensterverwaltung. Der klassische Linux-Window-Manager ist FVWM; heutzutage wird allerdings immer häufiger ein komfortabler ausgestatteter Desktop wie → KDE oder → GNOME verwendet.

Wireless LAN (WLAN) Standard für drahtlose Funknetze nach IEEE 802.11.

Wohlförmtheit Eigenschaft von XML-Dokumenten, in denen die Elemente korrekt verschachtelt sind und andere Standards eingehalten werden.

Word Allgemeine Bezeichnung für 16 Bit (2 Byte).

Wortbreite Eigenschaft eines Prozessors oder einer Datenleitung; gibt an, wie viele Bits parallel verarbeitet oder transportiert werden können.

Wurm Schädlicher Code, der sich über diverse Netzwerkprotokolle verbreitet, am häufigsten per E-Mail.

WYSIWYG What You See Is What You Get; Arbeitsweise des Desktop Publishing: Ein Dokument wird am Bildschirm so erstellt, wie es später im Druck aussehen wird.

X Window Der X Window Server oder X-Server stellt die grundlegende Funktionalität für den Betrieb grafischer Benutzeroberflächen auf UNIX-Systemen bereit. Auf den X-Server baut ein → Window-Manager auf, der die Fenster und ihre Bedienelemente darstellt; noch mehr Komfort bietet ein ausgewachsener Desktop wie → KDE oder → GNOME.

XHTML XML-basierte Neudefinition von HTML.

XML eXtensible Markup Language; auf → SGML aufbauende Meta-Sprache für die Definition beliebiger Auszeichnungssprachen mit spezieller Optimierung für den Netzwerk- und Internet-Gebrauch.

XML Schema XML-basierte Sprache für die Definition von Dokumentformaten; mächtiger als DTDs.

XSL-FO eXtensible Stylesheet Language, Formatting Objects. Definition spezieller Layout-Anweisungen für die visuelle Darstellung von XML-Dokumenten.

XSLT eXtensible Stylesheet Language Transformation. Satz spezieller Anweisungen für die Umwandlung von XML-Dokumenten in andere Formate wie HTML oder PDF.

Das Paradies habe ich mir immer als eine Art Bibliothek vorgestellt.
– Jorge Louis Borges

B Literaturtipps

In diesem Anhang finden Sie einige Tipps für zusätzliche Literatur, in der die verschiedenen Themen dieses Buches vertieft werden. Unter <http://buecher.lingo-world.de/opensuse/tipps.php> finden Sie dieselbe Liste jeweils in aktualisierter Form sowie mit Links zu Online-Shops.

B.1 Allgemeines zu Linux (openSUSE und allgemein), UNIX & Co.

- ▶ Willemer, Arnold: *UNIX – das umfassende Handbuch*. Bonn: Galileo Computing 2008.
Ein ausführliches Handbuch zu allen Aspekten moderner UNIX-Systeme für Anwender, Administratoren und Entwickler.
- ▶ Siever, Ellen et al.: *Linux in a Nutshell*. Köln: O'Reilly Verlag 2005.
Eine umfangreiche, übersichtliche Referenz zu allen wichtigen Linux-Kommandos und -Dienstprogrammen.
- ▶ Barrett, Daniel J.: *Linux kurz und gut*. Köln: O'Reilly Verlag 2004.
Sehr kompakter Begleiter für unterwegs, der schnelle Hilfe beim Umgang mit Linux-Systemen bietet.
- ▶ Schroder, Carla: *Linux Kochbuch*. Köln: O'Reilly Verlag 2005.
Dieses Buch aus der bewährten »Kochbuch«-Reihe bietet kommentierte Lösungen zu Problemstellungen aus dem Alltag.

B.2 Shell, Scripting und Programmierung

- ▶ Newham, Cameron; Rosenblatt, Bill: *Einführung in die bash-Shell*. Köln: O'Reilly Verlag 2005.

Einsteigerwerk, das sowohl die interaktive Bedienung als auch das Scripting mit der bash gründlich erläutert.

- ▶ Wolf, Jürgen: *Shell-Programmierung*. Das umfassende Handbuch. 2. Auflage. Bonn: Galileo Press 2008.

Sehr ausführliches Handbuch zum Shell-Scripting mit vielen nützlichen Praxisbeispielen.

- ▶ Wolf, Jürgen: *Linux-UNIX-Programmierung. Das umfassende Handbuch*. 3. Auflage. Bonn: Galileo Press 2009.

Wo Shell-Skripte in puncto Performance oder Leistungsumfang nicht mehr ausreichen, ist echte Anwendungs- oder gar Systemprogrammierung gefragt. Dieses umfangreiche Werk erläutert die Linux-Programmierung (mit einem Seitenblick auf andere UNIX-Varianten) in ihrer »Muttersprache« ANSI-C mit samt allerlei nützlichen Bibliotheken.

- ▶ Friedl, Jeffrey E. F.: *Reguläre Ausdrücke*. 3. Auflage. Köln: O'Reilly Verlag 2007.

Eine ausführliche Referenz zur Anwendung regulärer Ausdrücke in verschiedenen Programmiersprachen mit zahlreichen Praxisbeispielen.

- ▶ Schwartz, Randal L. et al.: *Einführung in Perl*. 5. Auflage. Köln: O'Reilly Verlag 2009.

Seit Jahren das klassische Einsteigerwerk, um die Skriptsprache Perl zu erlernen, aktualisiert auf die neueste Version.

- ▶ Wall, Larry et al.: *Programmieren mit Perl*. 2. Auflage. Köln: O'Reilly Verlag 2001.

Das »Kamel-Buch« ist die ultimative Perl-Bibel: Perl-Erfinder Larry Wall und seine beiden Co-Autoren erklären mit trockenem Humor, aber technisch äußerst exakt selbst die kleinsten Details der Perl-Programmierung. In puncto Version leider nicht auf dem neuesten Stand (Erscheinungsjahr 2001!), aber dennoch wertvoll.

- ▶ Mauerer, Wolfgang: *Linux Kernelarchitektur*. München: Hanser Fachbuchverlag 2004.

Gründliche Darstellung der Konzepte des aktuellen Linux-Kernels 2.6.

- ▶ Bovet, Daniel P.; Cesati, Marco: *Understanding the Linux Kernel*. 3. Auflage. Sebastopol: O'Reilly Media 2006.

Diese (englischsprachige) Beschreibung des Kernels geht noch etwas weiter in die Tiefe und enthält unter anderem mehr Original-Quellcode mit ausführlichen Erläuterungen.

B.3 OpenOffice.org und andere GUI-Anwendungen

- ▶ Krumbein, Thomas: *OpenOffice.org 3*. Bonn: Galileo Computing 2008.
Eine praxisorientierte Einführung in alle Teilprogramme des freien Office-Pakets.
- ▶ Rahemipour, Jacqueline: *Textverarbeitung mit OpenOffice.org 3 Writer*. 2. Auflage. Bonn: Galileo Computing 2008.
Gründliche Einführung und dauerhaft brauchbare Referenz zum OpenOffice.org-Textverarbeitungsprogramm Writer, mit besonderen Anmerkungen für Word-Umsteiger.
- ▶ Schmidt, Jörg: *Tabellenkalkulation mit OpenOffice.org 3 Calc*. Bonn: Galileo Computing 2008.
Einführung und Handbuch zur Tabellenkalkulations-Komponente von OpenOffice.org.
- ▶ Osterberg, Jürgen: *GIMP 2*. Heidelberg: dpunkt.verlag 2005.
Recht ausführliches GIMP-Handbuch; eher eine Referenz als Lehrbuch.

B.4 Editoren und Auszeichnungssprachen

- ▶ Lamb, Linda, Arnold Robbins: *Learning the vi Editor*. 6. Auflage. Sebastopol: O'Reilly Verlag 1999.
Lehrbuch, Handbuch und ausführliche Referenz zum vi-Editor. Berücksichtigt auch diverse vi-Klone wie elvis oder vim.
- ▶ Cameron, Debra et al.: *Learning GNU Emacs*. 3. Auflage. Sebastopol: O'Reilly Media 2006.
Methodisches Lehrbuch und umfassende Referenz zum Arbeiten mit dem mächtigen Texteditor GNU Emacs.
- ▶ Vonhoegen, Helmut: *Einstieg in XML*. 5. Auflage. Bonn: Galileo Press 2009.
Eine gute und methodische Einführung in das XML-Format und die praktische Arbeit mit XML-Dokumenten.
- ▶ Bongers, Frank: *XSLT 2.0 und Xpath 2.0*. 2. Auflage. Bonn: Galileo Press 2008.
Methodische Einführung und sehr umfassende Referenz zu sämtlichen Elementen der Dokumenten- und Formatttransformation mit XSLT in der neuen Version 2.0.

- ▶ Seibert, Björn; Hoffmann, Manuela: *Professionelles Webdesign mit (X)HTML und CSS*. 2. Auflage. Bonn: Galileo Press 2008.
Methodische Einführung in die Webseiten-Auszeichnungssprache HTML und das Erstellen standardkonformer, moderner Websites.
- ▶ Laborenz, Kai: *CSS-Praxis*. 5. Auflage. Bonn: Galileo Press 2008.
Eine gelungene Einführung und Übersicht über die Gestaltung von Webseiten mit Cascading Stylesheets.

B.5 Netzwerk- und Systemadministration

- ▶ Frisch, Aeleen: *UNIX System-Administration*. 2. Auflage. Köln: O'Reilly Verlag 2003.
Sehr umfangreiches Werk, das auf wirklich alle Aspekte der Systemadministration eingeht. Es ist für beliebige UNIX-Arbeitsumgebungen vom Linux-Einzelplatzrechner bis zum großen Unternehmen mit Unmengen von Servern geeignet. Die neuesten Entwicklungen können naturgemäß nicht zur Sprache kommen, aber das Buch bildet dennoch eine solide Grundlage.
- ▶ Hunt, Craig: *TCP/IP-Netzwerkadministration*. 3. Auflage. Köln: O'Reilly Verlag 2002.
Die dritte Auflage dieses Klassikers berücksichtigt neben der bewährten Mischung aus Protokolltheorie und Administrationspraxis auf UNIX-Systemen auch neuere Entwicklungen wie NAT oder IPv6.
- ▶ Liu, Cricket: *DNS & BIND Kochbuch*. Köln: O'Reilly Verlag 2003.
Kompakte Sammlung praktischer Anleitungen zur Konfiguration des DNS-Servers BIND.
- ▶ Ts, Jay et al.: *Samba*. 2. Auflage. Köln: O'Reilly Verlag 2003.
Das Buch bietet eine sehr gründliche Einführung in die Arbeitsweise des freien Windows-Dateiservers Samba sowie viele praktische Hinweise zu seiner Verwendung.

B.6 LAMP

- ▶ Kersken, Sascha: *Apache 2.2*. 3. Auflage. Bonn: Galileo Press 2009.
Umfassende Anleitung zur Installation, Konfiguration, Programmierung und praktischen Anwendung des Apache-Webservers.

- ▶ Lerdorf, Rasmus et al.: *Programmieren mit PHP*. 2. Auflage. Sebastopol: O'Reilly Media 2006.

Der PHP-Erfinder Lerdorf und seine Co-Autoren stellen ausführlich alle wichtigen Aspekte der Programmierung dynamischer Web-Anwendungen mit PHP vor. Die 2. Auflage behandelt auch die Neuerungen von PHP 5, allerdings nicht in ihren neueren Erscheinungsformen 5.2 und 5.3.

- ▶ Bergmann, Sebastian: *Professionelle Softwareentwicklung mit PHP 5*. Heidelberg: dpunkt.verlag 2005.

In diesem Buch wird der Einsatz von Methoden der professionellen objekt-orientierten Softwareentwicklung mit PHP erläutert, beispielsweise Entwurfsmuster und Unit-Tests.

- ▶ Kersken, Sascha: *Praktischer Einstieg in MySQL mit PHP*. 2. Auflage. Köln: O'Reilly Verlag 2007.

Neben einem gründlichen Einstieg in Datenbankdesign und -verwaltung mit MySQL führt dieses Buch anhand eines Praxisbeispiels in die Programmierung datenbankbasierter Webanwendungen mit PHP 5 ein. Praxiserprobt durch meine regelmäßigen MySQL-Schulungen im Linuxhotel (<http://www.linuxhotel.de>). Die gedruckte Ausgabe ist vergriffen, aber unter <http://www.oreilly.de/german/freebooks/einmysql2ger/> können Sie sich das gesamte Buch als kostenloses PDF-Open-Book herunterladen.

- ▶ Kofler, Michael: *MySQL 5*. 3. Auflage. München: Addison-Wesley 2005.

Sehr umfangreiches Anwendungs- und Referenzhandbuch zum Datenbankserver MySQL.

Index

- !~, Perl-Operator 830
- #! (Shebang) 109, 791
- \$, Operator
 - in *RegExp* 829
- \$, Perl-Variablen 814, 816
- \$_, Perl-Variablen 817
- \$0, Systemvariable 159
- %, Perl-Variablen 816
- & (im Hintergrund starten) 162
- && (Befehlsverknüpfung) 175
- &&, Operator
 - als *Perl-Fallentscheidung* 822
 - Perl* 822
- (Befehlsverknüpfung) 174
- (LaTeX) 514
- *, Operator
 - RegExp* 828
- +, Operator
 - RegExp* 828
- . (vor Dateinamen) 154
- ., Perl-Operator 821
- .bashrc, Konfigurationsdatei 158
- .htaccess-Dateien, Apache 2 731
- / als Partition 66
- /dev/null 168
- /etc/cron.daily, Konfigurationsdatei 785
- /etc/cron.hourly, Konfigurationsdatei 785
- /etc/cron.monthly, Konfigurationsdatei 785
- /etc/cron.weekly, Konfigurationsdatei 785
- /etc/cups/cupsd.conf 262
- /etc/dhcpd.conf, Konfigurationsdatei 681
- /etc/exports, NFS-Konfigurationsdatei 656
- /etc/fstab, Konfigurationsdatei 73, 380
 - atime* 74
 - auto* 75
 - defaults* 75
 - journal* 75
 - NFS-Volumes* 657
 - noatime* 74
 - noauto* 75
 - nouser* 75
 - ordered* 75
 - quota* 75
 - ro* 74
 - rw* 74
 - user* 75
 - writeback* 75
- /etc/hosts, Datei 621
- /etc/init.d, Verzeichnis 144
- /etc/init.d/rcN.d, Verzeichnisse 143
- /etc/openldap/slapd.conf, Konfigurationsdatei 676
- /etc/passwd, Konfigurationsdatei 127
- /etc/profile, Konfigurationsdatei 158
- /etc/shadow, Konfigurationsdatei 129
- /etc/smb.conf, Samba-Konfigurationsdatei 661
- /etc/squid/squid.conf, Konfigurationsdatei 682
- /etc/sudoers, Konfigurationsdatei 790
- /etc/sysconfig/apache2 716
- /etc/sysconfig/displaymanager, Konfigurationsdatei 216
- /etc/sysconfig-Editor, YaST-Modul 217
- /etc/vsftpd.conf, Konfigurationsdatei 771
- /etc/xinetd.conf, Konfigurationsdatei 763
- /etc/xinetd.d/*, Konfigurationsdateien 764
- /home als Partition 66
- /var/log/messages, Logdatei 789
- <>, Perl-Operator 814
- <a>, HTML-Tag 482
- <area>, HTML-Tag 488
- <body>, HTML-Tag 473
-
, HTML-Tag 474
- <caption>, HTML-Tag 490
- <dd>, HTML-Tag 481
- <Directory>, Apache-Direktive 733
- <dl>, HTML-Tag 481
- <dt>, HTML-Tag 481
- <form>, HTML-Tag 493
- <h1> bis <h6>, HTML-Tags 477
- <head>, HTML-Tag 473
- <html>, HTML-Tag 473
- , HTML-Tag 486
- <input>, HTML-Tag 493
- , HTML-Tag 479
- <Location>, Apache-Direktive 734
- <map>, HTML-Tag 488

- <meta>, HTML-Tag 474, 496
 - , HTML-Tag 479
 - <p>, HTML-Tag 476
 - <pre>, HTML-Tag 478
 - <select>, HTML-Tag 495
 - <style>, HTML-Tag 502
 - <table>, HTML-Tag 489
 - <td>, HTML-Tag 490
 - <textarea>, HTML-Tag 496
 - <th>, HTML-Tag 490
 - <title>, HTML-Tag 473
 - <tr>, HTML-Tag 490
 - , HTML-Tag 478
 - <VirtualHost>, Apache-Direktive 737
 - =~, Operator 830
 - ?, Operator
 - RegExp* 827
 - @, Perl-Variablen 816
 - @_, Perl 824
 - ^, Operator
 - in RegExp* 827, 829
 - `` (Ausgabe von Befehlen verwenden) 176
 - `, Operatoren 872
 - | (Pipe) 172
 - |, Operator
 - in RegExp* 830
 - || (Befehlsverknüpfung) 175
 - ||, Operator
 - als Perl-Fallentscheidung* 822
 - Perl* 822
 - ~ (Home-Verzeichnis) 153
 - 0x0C Win95 FAT32, Dateisystem-ID 71
 - 0x82 swap, Dateisystem-ID 71
 - 0x83 Linux, Dateisystem-ID 71
 - 0x8E Linux LVM, Dateisystem-ID 71
 - 0xA0 Hibernation, Dateisystem-ID 71
 - 0xFD RAID, Dateisystem-ID 71
 - 1:1-Beziehung, RDBMS 526
 - 1:n-Beziehung, RDBMS 527
 - 10 Base T, Ethernet-Standard 595
 - 100 Base T, Ethernet-Standard 595
 - 1000 Base FL, Ethernet-Standard 595
 - 1000 Base TX, Ethernet-Standard 595
 - 48-Bit-LBA 58
- A**
-
- Abbilder, Installation von 94
 - Abfrage, RDBMS
 - Auswahlabfrage* 529
 - Abhängigkeit (Softwarepakete) 92
 - AbiWord 247
 - Absatz
 - Ausrichtung* 287
 - Blocksatz* 287
 - Hurenkind* 289
 - linksbündig* 287
 - rechtsbündig* 287
 - Schusterjunge* 289
 - Zeilenfall* 288
 - zentriert* 287
 - Absatz, Typografie 286
 - Absätze
 - HTML* 476
 - Absatzformate, Writer 299
 - absoluter Pfad 153
 - Absorption, Farbe 324
 - Abwedeln, GIMP-Werkzeug 355
 - accept(), Perl-Funktion 882
 - ACCEPT, iptables-Ziel 648
 - Access Control List (ACL) 130
 - Access Control List → ACL
 - Access Point 939
 - ACL 75
 - ACL → Access Control List
 - Adams, Douglas 47
 - Addblock Plus, Firefox-Addon 367
 - Administration
 - rc-Skripte* 146
 - Admin-Rechte 641
 - Adobe PostScript 260
 - Adressbus 842
 - Wortbreite* 844
 - ADSL 608
 - Aggregatfunktionen, MySQL 553
 - AGP 856, 939
 - aide, Intrusion Detection System 642
 - AIX, Betriebssystem 28
 - Aktualisierung
 - des Systems* 64
 - Alias, Apache-Direktive 731
 - alias, Shell-Built-in 783
 - alias, Shell-Kurzbefehl 158
 - Allow, Apache-Direktive 732
 - AllowOverride, Apache-Direktive 732
 - Alphabet
 - Geschichte* 275
 - griechisches* 276

- lateinisches* 276
- phönizisches* 275
- ALRM, Signal 868
- ALSA 266
- ALTER TABLE, MySQL-Anweisung 556
- ALU 841, 939
- amaroK, KDE-Audio-Player 238
- Änderungsabfragen, MySQL 555
- Anführungszeichen
 - verschiedene, Bedeutung in Perl* 820
- Anmeldung
 - grafische* 104
- Anonymous FTP
 - Konfiguration* 772
 - Konzept* 769
- Anschnitt, Bilder im 293
- ANSI 939
- Anti-Alias 939
- Antiqua 282, 939
- Antiqua-Varianten 282
- Antivirenprogramme 640
- Anwendungsschicht, TCP/IP 577
- Apache 939
 - als Proxyserver* 684
 - Authentifizierung* 738
 - virtuelle Hosts* 737
 - Xalan* 467
- Apache 1.3 706
- Apache 2 706
 - .htaccess-Dateien* 731
 - <Directory>, Direktive* 733
 - <Location>, Direktive* 734
 - <VirtualHost>, Direktive* 737
 - Alias, Direktive* 731
 - Allow, Direktive* 732
 - AllowOverride, Direktive* 732
 - AuthBasicProvider, Direktive* 732
 - Authentifizierung in Version 2.2* 739
 - AuthName, Direktive* 732
 - AuthType, Direktive* 732
 - AuthUserFile, Direktive* 733
 - CGI-Konfiguration* 715
 - Deny, Direktive* 733
 - DirectoryIndex, Direktive* 733
 - Direktiven* 731
 - DocumentRoot, Direktive* 734
 - Eigenschaften* 707
 - Grundausrüstung* 709
 - Grundkonfiguration* 738
 - Haupt-Server* 713
 - htpasswd-Dienstprogramm* 733
 - Installation mit YaST* 708
 - Konfiguration* 730
 - Konfigurationsdateien* 709
 - Konfigurationskontexte* 731
 - Listen, Direktive* 734
 - LoadModule, Direktive* 734
 - manuelle Installation* 718
 - Module* 712
 - NameVirtualHost, Direktive* 735
 - openSUSE-Konfigurationsdatei* 716
 - Options, Direktive* 735
 - Order, Direktive* 735
 - Redirect, Direktive* 736
 - Require, Direktive* 736
 - Satisfy, Direktive* 736
 - ScriptAlias, Direktive* 736
 - ServerAdmin, Direktive* 736
 - ServerName, Direktive* 737
 - ServerRoot, Direktive* 737
 - ServerSignature, Direktive* 737
 - ServerTokens, Direktive* 737
 - SSL-Konfiguration* 715
 - Startseite festlegen* 733
 - TCP-Port* 712
 - Verzeichnisschutz* 738
 - virtuelle Hosts* 713
 - Website-Verzeichnis* 734
 - Weiterleitungen* 736
 - YaST-Konfigurationsmodul* 711
- Apache-Lizenz 31
- Apache-Logdateien auswerten 833
- API 939
- AppArmor 127
- Application Server 939
- apropos, Kommando 120, 894
- Arbeitsspeicher 840
- Arbeitsverzeichnis 185
 - ändern* 185
 - ermitteln* 185
- Archivdateien 393
- Archive
 - tar* 141
- A-Record (DNS) 628
- Arithmetische Operationen, Calc 320
- ARP 593, 939
- ARPA 939
- ARPANet 575

array_rand(), PHP-Funktion 759
 Arrays
 aus Strings, Perl 817
 Perl 816
 zusammenfassen, Perl 817
 AS, MySQL-Klausel 551
 ASCII 940
 ASCII-Art 478
 ASCII-Modus 940
 Assembler 940
 Nicht-x86 849
 x86-Beispiele 848
 Asymmetrische Verschlüsselung 645
 AT&T 26
 UNIX System V 27
 AT-Befehlssatz 940
 Athena, ursprüngl. X-Server-Projekt 208
 atime, Partitionsmodus 74
 Audio-CD 385
 über Soundkarte abspielen 267
 Auflicht-Scanner 268
 Auflösung
 der Grafikkarte 209
 Ausdruck 940
 Ausgabe
 Perl 814
 Ausgabesteuerung
 durch das Betriebssystem 24
 Ausgabeumleitung 166
 Auslagerungsdatei 875
 Auslagerungsdatei (Windows), verschieben
 56
 Auswahl (GIMP)
 Ellipse 343
 Farbbereichswähler 343
 Lasso 343
 Rechteck 342
 Zauberstab 343
 Auswahlabfrage 529
 Auswahlabfragen, MySQL 549
 Auswahlwerkzeug, Inkscape 360
 AuthBasicProvider, Apache-Direktive 732
 Authentifizierung
 Apache 2 738
 MySQL 559
 AuthName, Apache-Direktive 732
 AuthType, Apache-Direktive 732
 AuthUserFile, Apache-Direktive 733
 auto, Partitionsmodus 75

Autoconf 140
 Autoflush-Modus 881
 Automake 140
 configure-Skript 141
 Automatisch starten 143
 chkconfig 145
 Runlevel-Editor 145
 Automatische Partitionierung 67
 AVG(), MySQL-Funktion 554
 awk 811, 827
 Axmark, David 530

B

Backdoor 940
 background-attachment, CSS-Angabe 507
 background-color, CSS-Angabe 507
 background-image, CSS-Angabe 507
 background-repeat, CSS-Angabe 507
 Backtick, Operator 872
 Backticks 176
 Backups 641
 Barock-Antiqua 282
 bash 159
 .bashrc, Konfigurationsdatei 158
 /etc/profile, Konfigurationsdatei 158
 alias 783
 Skripte 790
 unalias 785
 bc, Kommando 796, 895
 Beagle Search 244
 Bedingter Sprung 847
 Befehlsmodus, Vim 406
 Befehlstabelle 842
 der CPU 842
 Befehlsverknüpfung
 logisches Oder 175
 logisches Und 175
 Befehlszeiger (CPU-Register) 842
 Bell Laboratories 26
 Benutzer
 /etc/passwd, Datei 127
 Details (YaST) 132
 einrichten (bei Installation) 80
 root 869
 verwalten (Shell) 187
 verwalten (YaST) 127, 131
 Benutzeranmeldung
 LDAP 678

- NIS 671
- Benutzerkonten 128
 - konfigurieren (bei Installation)* 80
- Benutzermodus 863, 867
- Benutzeroberfläche 24
 - grafische* 25
 - Konsole* 25
- Benutzerverwaltung
 - grafische (bei Installation)* 80
- Bereiche löschen, Vim 412
- Berkeley Socket API 876, 940
- Berkeley, Universität 27, 877
 - UNIX-Version* 27
- Beschreibbare DVDs 388
- Betriebssystem
 - Benutzermodus* 863
 - Benutzeroberfläche* 24
 - Bibliothek* 866
 - booten* 862
 - BSD-UNIX* 27
 - Dateiverwaltung* 24
 - Definition* 23
 - Dialogverarbeitung* 26
 - Ein- und Ausgabesteuerung* 24
 - FreeBSD* 28
 - Gerätetreiber* 864
 - Hardwaresteuerung* 24
 - HP UX* 28
 - IBM AIX* 28
 - ITS (Incompatible Timesharing System)* 26
 - Kernel* 861
 - Kernelmodus* 863
 - Linux* 28, 37
 - Mac OS X* 28
 - MULTICS* 26
 - Multitasking* 863, 866
 - Prozessmanagement* 24
 - Speichermanagement* 24
 - Stapelverarbeitung* 26
 - Sun Solaris* 28
 - Systemaufrufe* 849, 863
 - Systemprogramme* 864
 - Task Scheduler* 863
 - Threads* 862, 874
 - Timesharing* 26
 - UNIX* 27
 - UNIX System V* 27
- Bezeichner 940
- Beziehungen (RDBMS) 526
- Bézier, Pierre 362
- Bézier-Kurven-Werkzeug, Inkscape 362
- BGP 940
- Bibliothek
 - des Betriebssystems* 866
- Big-Endian-Architektur 940
- BIGINT, MySQL-Datentyp 546
- Bild erstellen (GIMP) 336
- Bildauflösung 328, 940
 - Druckauflösung* 328
 - Scannen* 330
- Bildbearbeitung 323
 - Bildauflösung* 328
 - Bildgröße* 328
 - Bildschirmauflösung* 329
 - Farbmodelle* 325
 - Farbtiefe* 327
 - GIMP* 336
 - Krita* 237
 - Scan-Auflösung* 330
- Bilddateiformate 332
 - GIF* 334
 - JPEG* 334
 - Photoshop* 333
 - PNG* 335
 - PSD* 333
 - TIFF* 333
- Bilder
 - in HTML einbetten* 486
- Bildgröße 328
- Bildgröße ändern, GIMP 346
- Bildschirmauflösung 329
- Bildwiederholrate (Monitor) 210
- bin, Systemverzeichnis 150
- Binärmodus 940
- BIND
 - manuell kompilieren* 142
- bind(), Perl-Funktion 882
- BIND-Nameserver 624
 - A-Record* 628
 - CNAME-Record* 629
 - Konfiguration* 625
 - MX-Record* 630
 - NS-Record* 629
 - PTR-Record* 628
 - Reverse-Lookup-Zone* 625
 - SOA-Record* 628
 - Zonendaten-Dateien* 627
 - Zonendefinition* 625

- BIOS 840, 940
 - BIOS-Setup
 - Boot-Datenträger* 59
 - Bitmap 940
 - Bitmap-Grafik 323
 - BitTorrent 50
 - BLOB, MySQL-Datentyp 548
 - Blockgeräte (Block Devices) 864
 - Blocksatz 287
 - in HTML* 477
 - Silbentrennung* 287
 - Blowfish, Verschlüsselungsalgorithmus 134
 - Blue Book 940
 - Blue Book (Mixed-Mode-CD) 386
 - Bluetooth 861
 - Booten 103, 862
 - Konfiguration* 398
 - Optionen* 104
 - Rettungskonsole* 401
 - Bootloader
 - GRUB* 400
 - LILO* 401
 - Bootmanager
 - Konfigurieren (Installation)* 94
 - Bootmenü 103
 - Boot-Reihenfolge (BIOS) 59
 - Bootsektor 372
 - border, CSS-Angabe 506
 - Börries, Marco 273
 - Bourne Again Shell (bash) 159
 - Bourne-Shell (sh) 159
 - Branch Prediction → Sprungvorhersage
 - Brennen (CDs/DVDs) 389
 - Bridge 940
 - Broadcast-Adresse (IP) 580
 - Browser
 - als FTP-Client* 767
 - Firefox* 364
 - Brute-Force-Attacke 940
 - BSD 941
 - BSD-Lizenz 31
 - BSD-UNIX 27
 - FreeBSD* 28
 - btfs, Dateisystem 71
 - Buchstaben
 - Bestandteile* 278
 - Geschichte* 275
 - Großbuchstaben* 278
 - Kapitälchen* 279
 - Kleinbuchstaben* 279
 - Buffer, Emacs 428
 - Build Service (openSUSE) 51
 - Bunte Bücher (CD-Standards) 385
 - bunzip2, Kommando 395
 - Bus 851
 - Bluetooth* 861
 - Definition* 841
 - der CPU* 842
 - drahtloser* 860
 - EIDE* 857
 - FireWire* 859
 - Funk-Schnittstelle* 861
 - Hot-Plugging* 859
 - Infrarot* 860
 - IrDA* 861
 - Kartensteckplätze* 856
 - Laufwerksanschlüsse* 856
 - paralleler* 860
 - PS/2* 860
 - RS-232* 860
 - SAS* 859
 - S-ATA* 857
 - SCSI* 858
 - serieller* 860
 - USB* 859
 - Bus Mastering 855, 941
 - bzip2, Kommando 395, 896
- ## C
-
- C, Programmiersprache 27, 941
 - exec()-Funktion* 872
 - fork()-Funktion* 870
 - Standardbibliothek* 866
 - system()-Funktion* 872
 - Cache 842
 - bei Festplatten* 371
 - Level 1* 843
 - Level 2* 843
 - Calc 313
 - arithmetische Operationen* 320
 - Diagramme* 317
 - Formeln* 316
 - Funktionen* 316
 - ISTLEER(), Funktion* 321
 - ISTTEXT(), Funktion* 321
 - ISTZAHL(), Funktion* 321
 - MAX(), Funktion* 320

- MIN(), Funktion 320
- MITTELWERT(), Funktion 316
- ODER(), Funktion 321
- PRODUKT(), Funktion 320
- QUOTIENT(), Funktion 320
- REST(), Funktion 320
- SUMME(), Funktion 316
- Tabellen erstellen 313
- UND(), Funktion 321
- VERWEIS(), Funktion 321
- WENN(), Funktion 320
- Zellbezüge 318
- Zellenformate 315
- Captive NTFS 377
- Cäsar-Code 643
- case, Shell-Built-in 801
- cat, Kommando 194
- CAV → Konstante Winkelgeschwindigkeit
- CCD (Scanner) 268
- CD
 - beschreibbare 386
 - Disc-at-once 387
 - Lead-in-Area 387
 - Lead-out-Area 387
 - Multisession 387
 - Track-at-once 387
- cd, Kommando 185, 897
- CDi 386
- cdparanoia, Kommando 390, 897
- CDR_DEVICE, Umgebungsvariable 389
- cdrecord 389
- cdrkit 389
- CD-ROM 385
 - brennen 386, 389
 - Geschwindigkeit 386
- cdrtools 389
- CD-RW 386
- CD-Standards 385
 - Blue Book 386
 - Green Book 386
 - Orange Book 386
 - Red Book 385
 - White Book 386
 - Yellow Book 386
- CD-Text 385
- Centronics-Anschluss 860
- CGI 715
- CGI-Sicherheitsprobleme 639
- Chamäleon (Geeko) 107
- Chamäleon (openSUSE) 107
- CHAP 607
- CHAR, MySQL-Datentyp 547
- chargen, Serverdienst 766
- chgrp, Kommando 190, 898
- Child-Prozess 867, 941
- Chipsatz 841
- chkconfig, Kommando 145, 898
- CHLD, Signal 868
- chmod, Kommando 190, 899
- chomp, Perl-Funktion 814
- chown, Kommando 190, 899
- chroot
 - für FTP 772
- chroot-Umgebung 642
- CHS (Festplattenadressierung) 369
- CIDR (Classless Inter-Domain Routing) 581
- CIDR-Adressierung 941
- CISC-Prozessor 846
 - Beispiele 846
- clear, Kommando 900
- Client-Anfrage (HTTP) 695
- close, Perl-Anweisung 826
- closedir, Perl-Anweisung 826
- CLV → Konstante lineare Geschwindigkeit
- CMYK-Farbe 325
- cn, LDAP-Element 674
- CNAME-Record (DNS) 629
- Codd, Edgar F. 538
- color, CSS-Angabe 507
- Color-Management 325
- comp.os.minix, Newsgroup
 - Posting über Linux 33
- Compact Disc 385
- Compiler 849
- Composite-Window-Manager 221
- Computergrafik 323
 - Bitmaps 323
 - Farbmodelle 325
 - Farbtiefe 327
 - Vektorgrafik 323
- configure, Automake-Skript 141
- connect(), Perl-Funktion 881
- console, Display-Manager 218
- Constant Linear Velocity → Konstante lineare Geschwindigkeit
- Cookies 941
 - in PHP 750
- Copyleft 30

- cp, Kommando 179, 900
- CPU 840
- crack (Passwort-Knackprogramm) 129
- Crackerangriffe 638
- Crackertools 642
- CREATE DATABASE, MySQL-Anweisung 539
- CREATE TABLE, MySQL-Anweisung 540
- CREATE USER, MySQL-Anweisung 560
- Cronjobs 785
 - /etc/cron.*-Dateien 785
 - crontab 786
- crontab, Kommando 786, 900
- Crosslink-Ethernet-Kabel 594
- Cross-Site-Scripting 639
- csh 159
- C-Shell (csh) 159
- CSMA/CA 941
- CSMA/CA, Netzzugangsverfahren 603
- CSMA/CD 941
- CSMA/CD, Netzzugangsverfahren 592
- CSS 499, 941
 - <style>, HTML-Tag 502
 - Absatzformatierung 505
 - Abstand vom linken Rand 508
 - Abstand vom oberen Rand 508
 - Anzeigeart 505
 - Aufgabe 500
 - Ausrichtung 505
 - Außenrand 505
 - background-attachment 507
 - background-color 507
 - background-image 507
 - background-repeat 507
 - Bilder 507
 - border 506
 - color 507
 - display 505
 - Einzug 505
 - Element (Tag) formatieren 500
 - externe Dateien 502
 - Farbangaben 504
 - Farben 507
 - feste Werte 503
 - fett 504
 - font-family 504
 - font-size 504
 - font-style 504
 - font-weight 504
 - für XML-Dokumente 467
 - Hintergrund befestigen 507
 - Hintergrund kacheln 507
 - Hintergrundbild 507
 - Hintergrundfarbe 507
 - Innenabstand 505
 - Klassen 501
 - kursiv 504
 - Laufweite 505
 - Layer 507
 - Layer, Beispiele 508
 - left 508
 - letter-spacing 505
 - line-height 505
 - Linien 505
 - margin 505
 - padding 505
 - position 508
 - Positionsart 508
 - Pseudoformate 501
 - Rahmen 505, 506
 - Schriftart 504
 - Schriftgröße 504
 - Selektor 500
 - Stapelreihenfolge 508
 - Struktur 500
 - style, HTML-Attribut 502
 - text-align 505
 - text-decoration 504
 - Textfarbe 507
 - Textformatierung 504
 - text-indent 505
 - top 508
 - unabhängige Stile 501
 - unterstrichen 504
 - vertical-align 505
 - vertikale Ausrichtung 505
 - Vorteile 500
 - Wertangaben 503
 - Zeilenhöhe 505
 - z-index 508
- C-Standardbibliothek 866, 941
- CUPS 941
- CUPS, Drucksystem 260
 - Konfigurationsdatei 262
 - Scheduler 261
 - Starten 262
- curl, Kommando 901
- Cyrus IMAP Server 781

D

-
- Daemon 142, 941
 - DARPA 941
 - DATABASE(), MySQL-Funktion 538
 - Datagramm 941
 - Datagramme
 - IPv4 582
 - IPv6 585
 - Datagramm-Sockets 878
 - date, Kommando 204, 902
 - DATE, MySQL-Datentyp 548
 - Datei 24
 - Dateien
 - /etc/hosts* 621
 - bearbeiten, Perl* 825
 - Eigentümer ändern (Shell)* 190
 - finden (Shell)* 187
 - Gruppe ändern (Shell)* 190
 - kopieren (Shell)* 179
 - löschen (GNOME)* 117
 - löschen (KDE)* 112
 - löschen (Shell)* 181
 - umbenennen* 181
 - umbenennen (Shell)* 181
 - vergleichen (Shell)* 198
 - verschieben* 181
 - verschieben (Shell)* 181
 - Dateiformate
 - Bilder* 332
 - GIF* 334
 - JPEG* 334
 - PNG* 335
 - TIFF* 333
 - Dateihandles, Perl 825
 - Dateimanager
 - Dolphin* 107
 - Konqueror* 110
 - Nautilus (GNOME)* 115
 - Dateimuster 178
 - Unterschied zu RegExp* 179
 - Unterschied zu Windows* 179
 - Dateinamen
 - Groß- und Kleinschreibung* 153
 - mit '.' beginnende* 154
 - Sonderzeichen in* 153
 - Dateirechte → Zugriffsrechte
 - Dateiserver
 - NFS 655
 - Samba 661
 - Dateisysteme 375, 941
 - btrfs* 71
 - erzeugen* 382
 - Ext2* 71, 375, 942
 - Ext3* 72, 376, 942
 - Ext4* 376
 - FAT* 72, 943
 - FAT12* 377
 - FAT16* 377
 - FAT32* 377
 - ISO 9660* 378
 - JFS* 72
 - Journal* 376, 945
 - Journaling* 72
 - Loop-Devices* 396
 - NTFS* 377
 - Optionen* 73
 - proc* 397
 - ReiserFS* 72, 376, 948
 - smbfs* 663
 - verschlüsseltes* 73
 - virtuelle* 392
 - virtuelles* 150
 - wählen (Installation)* 71
 - XFS* 72, 376
 - Dateisystem-ID 71
 - FAT 32 (0x0C)* 71
 - Hibernation (0xA0)* 71
 - Linux (0x83)* 71
 - Linux LVM (0x8E)* 71
 - RAID (0xFD)* 71
 - swap (0x82)* 71
 - Dateitest-Operatoren, Perl 823
 - Dateiverwaltung 24
 - Datenbank
 - 1:1-Beziehung* 526
 - 1:n-Beziehung* 527
 - Auswahlabfrage* 529
 - Beziehungen* 526
 - Datenfeld* 525
 - Datensatz* 525
 - Entity* 525
 - Fremdschlüssel* 526
 - Index* 526
 - Konsistenz* 525
 - m:n-Beziehung* 527
 - Primärschlüssel* 526
 - relationale* 525

- Schlüssel 526
- Datenbus 842
 - Wortbreite 844
- Datenfeld 525
- Datenkomprimierung 331
- Datenpakete
 - Frames 593
- Datensatz 525
- Datensicherung 641
- Datenträger
 - Mounten 154
- Datentypen, MySQL 545
- Datenübertragung
 - Geschwindigkeitsmessung 853
 - parallele 851
 - serielle 851
- DATETIME, MySQL-Datentyp 548
- Datum und Uhrzeit
 - einstellen (*Installation*) 64
 - EPOCH 866
 - Shell 204
- daytime, Serverdienst 764
- dc, LDAP-Element 674
- dd (Didot-Punkt) 276
- dd, Kommando 374, 904
- DDN-Schichtenmodell 576
- DDR-RAM 851
- de Icaza, Miguel 219
- Deadlock 869, 941
- Debian GNU/Linux 39
- DECIMAL, MySQL-Datentyp 547
- Default Gateway 942
- defaults, Partitionsmodus 75
- defined, Perl 823
- Defragmentieren (Windows) 57
- DELETE, MySQL-Anweisung 557
- Denial of Service 638
- Denial-of-Service-Angriff 942
- Deny, Apache-Direktive 733
- Dereferenzierung
 - Perl 818
- DES, Verschlüsselungsalgorithmus 133
- DESCRIBE, MySQL-Anweisung 543
- Desktops
 - auswählen (*Installation*) 65
 - GNOME 239
 - KDE 220
 - virtueller (KDE) 107
- dev, Systemverzeichnis 151
- DHCP 679, 942
 - /etc/dhcpd.conf, Konfigurationsdatei* 681
 - Client einrichten (YaST) 597
 - Client-Konfiguration 679
 - Leasing 680
 - Netzwerkkonfiguration beziehen über 98
 - Range 681
 - Server-Konfiguration mit YaST 679
- Diagramme, Calc 317
- Dialogverarbeitung 26
- Dicke 942
- Dicke, Typografie 278
- Didot, François 276
- Didot-Punkt, Schriftgröße 276
- die, Perl-Anweisung 822
- diff, Kommando 198, 904
- dig, Kommando 633
- Digitale Signatur 645
- DIMM-Module (RAM) 850
- DIN 942
 - Papierformat 289
 - Schriftfamilien 280
- Directory Node (info-Startseite) 121
- DirectoryIndex, Apache-Direktive 733
- direcd, Emacs-Modus 433
- discard, Serverdienst 767
- Disc-at-once 387, 942
- Diskettenlaufwerk 378
- display, CSS-Angabe 505
- Display-Manager 217
- Distribution 942
- Distributionen
 - Fedora Core, Distribution 39
 - Mandriva 40
 - openSUSE 39
 - RedHat 39
 - Slackware 39
 - Ubuntu 39
- DMA-Kanal 855, 942
 - Direct Memory Access 855
- DNS 942
- dnsdomainname, Kommando 905
- DNS-Server 623
- DocBook 440
- DOCTYPE, Verarbeitungsanweisung
 - HTML 473
- DOCTYPE-Angabe
 - XML 451
- DocumentRoot, Apache-Direktive 734

- Dokumentstruktur
 - HTML 472
 - Dolphin, KDE-Dateimanager 234
 - Domain Name System (DNS) 621
 - BIND-Nameserver 624
 - Funktionsweise 622
 - Nameserver 623
 - Round-Robin-Verfahren 629
 - Domainname
 - bei Installation festlegen 95
 - domainname, Kommando 905
 - Doppelte Anführungszeichen, Perl 820
 - dos2unix, Kommando 905
 - DOUBLE, MySQL-Datentyp 547
 - DoubleWord (DWord) 942
 - Download
 - HTML-Hyperlinks 483
 - Download von openSUSE
 - Developer-Version 50
 - stabile Version 48
 - Drag and Drop 865
 - Drahtlose Schnittstellen 860
 - DRAM 850
 - Drehwerkzeug, GIMP 347
 - DROP DATABASE, MySQL-Anweisung 557
 - DROP TABLE, MySQL-Anweisung 557
 - DROP USER, MySQL-Anweisung 563
 - DROP, iptables-Ziel 648
 - Druckauflösung 328
 - Drucker 259
 - CUPS 260
 - Freigabe über Samba 663
 - HPGL 260
 - Laserdrucker 260
 - lpd 261
 - lpr, Konsolen-Druckbefehl 264
 - PostScript 260
 - PPD-Datei 264
 - Tintenstrahldrucker 259
 - verwenden 264
 - Warteschlange 263
 - YaST-Konfiguration 262
 - Druckraster 328
 - Rasterwinkel 328
 - DSL 608, 942
 - ADSL 608
 - anschließen 609
 - bei der Installation konfigurieren 100
 - Konfiguration (YaST) 609
 - SDSL 608
 - über Fernsehkabel 608
 - über Satellit 609
 - DSO 254
 - DSSS, WLAN-Technik 602
 - DTD 450, 942
 - Alternativen angeben 454
 - Attributdeklaration 457
 - Attributnotwendigkeit 460
 - Attributtypen 458
 - definieren 451
 - Elementdeklaration 453
 - Entities aus externen Dateien 461
 - Entities deklarieren 461
 - externe Entity-Deklarationen 462
 - Häufigkeitsangabe 455
 - Klammern 454
 - DTP-Punkt, Schriftgröße 276
 - du, Kommando 382, 906
 - Durchlicht-Scanner 268
 - Durchschuss, Typografie 278
 - DVD 387
 - beschreibbare 388
 - DVD+R 388
 - DVD+RW 388
 - DVD-R 388
 - brennen 389
 - DVD-RAM 388
 - DVD-ROM 388
 - Dateiformat 388
 - Geschwindigkeit 388
 - DVD-RW 388
 - DVI, Dateiformat 513
 - dvipdf, Kommando 513
 - dvips, Kommando 513
 - Dyroff, Roland 47
- ## E
-
- Easy Software Products 261
 - Ebenen, GIMP 357
 - echo, Kommando 193, 906
 - echo, Serverdienst 766
 - Editoren 405
 - elvis 406
 - Kate 237
 - vi 405
 - Effektive Group-ID 192
 - Effektive User-ID 191

- Effizienz der CPU 846
- EIDE 857, 942
 - anschließen* 857
 - im Vergleich zu SCSI* 857
- Ein-/Ausgabe 25
 - Dialogverarbeitung* 26
 - Lochkarten* 25
 - Perl* 814
 - Stapelverarbeitung* 26
 - Terminal* 26
 - Timesharing* 26
- Ein-/Ausgabekanäle 166
- Ein-/Ausgabumleitung 166
 - Testprogramm* 168
- Einfache Anführungszeichen, Perl 820
- Einfügeabfragen, MySQL 544
- Einfügen, Vim 413
- Eingabe
 - Perl* 814
- Eingabeaufforderung 111, 157
- Eingabemodus, Vim 406
- Eingabesteuerung
 - durch das Betriebssystem* 24
- Eingabeumleitung 167
- Eingabevervollständigung 163
- Einsteckkarten 856
 - einbauen* 856
- Einwegverschlüsselung 645
- Eliza 434
- Ellipsenauswahl, GIMP-Werkzeug 343
- Ellipsenwerkzeug, Inkscape 361
- elsif, Perl 823
- elvis, Texteditor 406
- Emacs, Texteditor 425
 - auto-fill* 429
 - Befehlsübersicht* 435
 - Buffer* 428
 - Buffer wechseln* 428
 - Bufferliste* 428
 - C-Mode* 429
 - Dateien öffnen* 428
 - Datum und Uhrzeit* 434
 - dired Mode* 433
 - Eliza* 434
 - Ersetzen* 431
 - Fundamental Mode* 428
 - gnus* 434
 - gnus, Emacs-Newsreader* 434
 - Hilfe* 426
 - HTML-Mode* 429
 - Inkrementelle Suche* 431
 - Kalender* 434
 - Mail-Funktionen* 434
 - Markieren* 430
 - Meta-Taste, Entsprechung* 426
 - Minibuffer* 429
 - Modi* 428
 - Navigation* 430
 - News* 434
 - Rückgängig machen* 431
 - Scratch-Buffer* 428
 - Shell Mode* 432
 - Shell-Befehle ausführen* 432
 - Suchen* 431
 - Tastenkürzel* 426
 - Tastenkürzel-Schreibweise* 426
 - Tetris* 434
 - Text Mode* 429
 - Varianten* 425
 - Verzeichnisse bearbeiten* 433
 - Wortumbruch* 429
 - Zähler* 430
 - Zusatzfunktionen* 432
- E-Mail
 - Attachment* 776
 - Evolution, GNOME-Client* 247
 - HTML-Hyperlinks auf* 484
 - IMAP-Protokoll* 779
 - KMail, Client* 238
 - Multipart-Nachrichten* 776
 - POP3-Protokoll* 777
 - RFC-822-Nachricht* 775
 - SMTP-Protokoll* 773
 - Verschlüsselung* 645
 - YaST-Konfiguration* 779
- E-Mail-Protokolle 772
- Embedded Systems 840
- Entity
 - Datenbank* 525
- Entity-Referenzen
 - HTML* 474
 - XML* 446, 461
- ENUM, MySQL-Datentyp 548
- Ephemeral Ports 590
- EPOCH 866
- eq, Perl-Operator 821
- Escape-Sequenz 197, 942
- etc, Systemverzeichnis 151

- Ethernet 942
 - 10 Base T-Standard 595
 - 100 Base T-Standard 595
 - 1000 Base FL-Standard 595
 - 1000 Base TX-Standard 595
 - Allgemeines 592
 - ARP 593
 - Crosslink-Kabel 594
 - CSMA/CD 592
 - Frames 593
 - Hardware-Adresse 593
 - Hub 594
 - ifconfig*, Kommando 595
 - Konfiguration mit YaST 596
 - MAC-Adresse 593
 - Switch 594
 - Twisted-Pair-Kabel 593
 - Ettrich, Matthias 219
 - Euro-ISDN (EDSS1) 618
 - Evolution, GNOME Information Manager 247
 - Ewing, Larry 34
 - ex, Zeileneditor 405
 - exec(), C-Funktion 872
 - exec(), Systemaufruf 867
 - Exit-Code 175
 - Expert-Zeichensatz 280
 - Exploits 637
 - Exponentialschreibweise, Perl 819
 - export, Shell-Built-in 794
 - expr, Kommando 797, 906
 - Ext2, Dateisystem 71, 375, 942
 - ext3
 - als Standard* 67
 - Ext3, Dateisystem 72, 376, 942
 - Ext4, Dateisystem 376
- F**
-
- facility (Syslog) 788
 - Falcon, MySQL-Tabellentyp 532
 - Fallentscheidungen
 - durch logische Operatoren*, Perl 822
 - Shell-Skripte* 798
 - Farbaddition 325, 943
 - Farbbereichswähler, GIMP-Werkzeug 343
 - Farbe 324
 - CMYK 325
 - Color-Management 325
 - Farbaddition 325
 - Farbmodelle 325
 - Farbsubtraktion 326
 - Farbtiefe 327
 - HSB-Modell 326
 - Komplementärfarben 325
 - Körperfarben 325
 - Lab-Modell 326
 - Pigmentfarben 325
 - Primärfarben 325
 - Reflektion und Absorption 324
 - RGB 325
 - Sättigung 326
 - Farbe wählen, GIMP 356
 - Farbkanäle 943
 - Farblaserdrucker 260
 - Farbmodelle 325
 - Farbseparation 943
 - Farbsubtraktion 326, 943
 - Farbtiefe 327, 943
 - der Grafikkarte* 209
 - Farbton/Sättigung, GIMP 359
 - Farbverläufe, GIMP 352
 - Farbverlaufswerkzeug, Inkscape 363
 - Fast Ethernet 595
 - FAT 32 (0x0C), Dateisystem-ID 71
 - FAT, Dateisystem 72, 943
 - FAT12, Dateisystem 377
 - FAT16, Dateisystem 377
 - FAT32, Dateisystem 377
 - Fedora Core, Linux-Distribution 39
 - Fehr, Thomas 47
 - Festbreitenschrift 279
 - Festplatte
 - defragmentieren (Windows)* 57
 - größer als 137 GB* 58
 - neue einbauen für OpenSUSE* 56
 - Partitionierung* 66
 - Festplatte vorbereiten, Installationsschritt 67
 - Festplatten
 - Anschlüsse 856
 - Cache 371
 - CHS-Adressierung 369
 - EIDE 857
 - formatieren* 382
 - Geschwindigkeit* 370
 - konstante Winkelgeschwindigkeit* 370
 - LBA-Adressierung 369

- Mittlere Zugriffszeit* 371
- RAID 383
- SAS 859
- S-ATA 857
- SCSI 858
- fetchmail 781
- FFII 46
- fg, Kommando 162
- fgrep, Kommando 910
- FHSS, WLAN-Technik 602
- FIFO 943
- file, Kommando 907
- filter, iptables-Tabelle 647
- find, Kommando 187, 908
- Firebug, Firefox-Addon 367
- Firefox 364
 - Adblock Plus, Addon* 367
 - Addons* 367
 - als FTP Client* 767
 - Firebug, Addon* 367
 - Web Developer, Addon* 367
- Firewall 641, 943
 - aktivieren (bei Installation)* 96
 - iptables* 646
 - Konfiguration (YaST)* 650
 - netfilter* 646
- FireWire 859, 943
- Flachbettscanner 268
- Fließkomma-Literale, Perl 819
- Fließkommazahl 943
- FLOAT, MySQL-Datentyp 547
- Floppy-Laufwerk 378
- FLOPS (CPU-Geschwindigkeit) 846
- FLUSH PRIVILEGES, MySQL-Anweisung 564
- FLUSH TABLES, MySQL-Anweisung 565
- FM-Synthese (MIDI) 267
- font-family, CSS-Angabe 504
- font-size, CSS-Angabe 504
- font-style, CSS-Angabe 504
- font-weight, CSS-Angabe 504
- for, Shell-Built-in 803
- foreach, Perl-Schleife 817
- fork(), C-Funktion 870
- fork(), Systemaufruf 867
- Formatieren 382
 - bei Installation* 71
- Format-Symboleiste, Writer 294
- Formatvorlagen, Writer 303
- Formeln, Calc 316
- Formelsatz (LaTeX) 520
- FORWARD, iptables-Kette 647
- Fraktur 283
- Frames, Datenpakete 593
- Französische Renaissance-Antiqua 282
- Freax (ursprünglicher Linux-Name) 34
- Free Software Foundation (FSF) 29
- FreeBSD 28
- Freie Software 943
 - Abgrenzung zu Open Source* 32
 - Apache-Lizenz* 31
 - BSD-Lizenz* 31
 - GPL, Lizenz* 29
 - Motivation* 28
- Freihandwerkzeug, Inkscape 362
- Fremdschlüssel, RDBMS 526
- Frequency-Hopping 602
- Front Side Bus 844
- fsck, Kommando 382
- FSF 943
- Fstab-Optionen 73
 - Daten-Journaling-Modus* 75
 - durch Benutzer mountbar* 75
 - keine Zugriffszeit* 74
 - Kontingentunterstützung (Quota)* 75
 - nicht beim Systemstart einhängen* 75
 - schreibgeschützt* 74
 - Zugriffskontrolllisten (ACL)* 75
- FTP 767, 943
 - Anonymous, Konfiguration* 772
 - Anonymous, Konzept* 769
 - ASCII-/Binärmodus* 770
 - Befehle* 770
 - Browser als Client* 767
 - chroot* 772
 - Clients* 767
 - ftp, Konsolen-Client* 769
 - gFTP, Client* 768
 - HTML-Hyperlinks auf* 484
 - Server* 771
 - TFTP, Vergleich mit* 767
 - vsftpd, Server* 771
- ftp, Kommando 769
- ftp_proxy, Umgebungsvariable 684
- Füllwerkzeug, GIMP 349
- Füllwerkzeug, Inkscape 362
- Funk-Schnittstelle 861
- Funktion 943

Funktionen

Shell-Skripte 808

Funktionen, Calc 316

FVWM, Window-Manager 247

G

gcc

eigene Programme kompilieren 170

gdm, Display-Manager 217

ge, Perl-Operator 821

Gebrochene Schriften 283

gedit, GNOME-Texteditor 114

Geeko (openSUSE-Chamäleon) 107

Gerätedateien 151

Gerätetreiber 24, 864

Blockgeräte 864*Zeichengeräte* 864

Gestaltung

Anschnitt 293*Elemente anordnen* 291*Goldener Schnitt* 291*Grundregeln* 289*Layout* 291*Linien* 294*optische Mitte* 291*Ornamente* 294*Papierformat* 289*Rahmen* 294*Schmuckelemente* 294

GET, HTTP-Befehl

zum HTML-Formularversand 493

gethostbyname(), Systemaufruf 878

getprotobyname(), Systemaufruf 878

getservbyname(), Systemaufruf 879

Geviert, Typografie 278

gFTP, FTP-Client 768

GhostScript 513

GID (Group-ID) 128, 190

von Prozessen 868

GIF, Bilddateiformat 334

Gigabit Ethernet 595

GIMP 336

Abwedeln 355*Bildgröße ändern* 346*Datei öffnen* 338*Dokumentfenster* 340*Drehen* 347*Ebenen* 357*Ellipsenauswahl* 343*Farbbereichswähler* 343*Farbe wählen* 356*Farbton/Sättigung* 359*Farbverläufe* 352*Füllwerkzeug* 349*Helligkeit/Kontrast* 359*Hilfslinien* 341*Kanäle* 358*Klonen* 354*Lasso* 343*Lineale* 340*Lupe* 345*Nachbelichten* 355*neues Bild erstellen* 336*Perspektive ändern* 348*Pfadwerkzeug* 344*Pinsel* 354*Pipette* 344*Radiergummi* 354*Rechteckauswahl* 342*scannen (XSane)* 339*schärfen* 355*Scherwerkzeug* 348*Screenshot* 338*skalieren* 348*speichern* 339*spiegeln* 348*Sprühpistole* 354*Stempel* 354*Stift* 353*Textwerkzeug* 348*Tinte* 354*Umrisse wählen* 343*Vektorgrafik* 344*Verlaufswerkzeug* 352*Verschiebewerkzeug* 345*verschmieren* 355*weichzeichnen* 355*Werkzeugpalette* 341*Winkelmesser* 345*Zauberstab* 343*zuschneiden* 346*Zwischenablage einfügen* 338

GNOME 239, 943

als Haupt-Desktop (Installation) 65*als root arbeiten* 124*Anwendungen (Menü)* 113*Banshee, Audio-Player* 247

- Dateien löschen* 117
- Dokumente (Menü)* 113
- E-Mail* 247
- erste Schritte* 113
- Evolution* 247
- gedit, Texteditor* 114
- GtK+-Bibliothek* 219
- Hardware konfigurieren* 240
- KDE, Vergleich mit* 219
- Konfiguration* 239
- Look & Feel einrichten* 239
- Nautilus, Dateimanager* 115, 244
- Orte (Menü)* 113
- Panel* 113
- System (Menü)* 114
- Systemeinstellungen* 242
- Terminal* 157
- YaST starten* 124
- Zugriffsrechte setzen* 115
- GNOME Office 247
 - AbiWord* 247
 - Gnumeric* 247
- GNOME Terminal 117, 246
 - Shell-Skript ausführen* 117
- GNU 944
 - Autoconf* 140
 - Automake* 140
 - Libtool* 140
- GNU Emacs 425
- GNU General Public License 29
- GNU info 121
 - Bedienung* 121
 - blättern* 122
 - Directory Node* 121
 - Navigation* 121
- GNU/Linux 31
- Gnumeric 247
- GNU-Projekt 29
- gnus, Emacs-Newsreader 434
- Goldener Schnitt 291
- GPL 29
 - Copyleft* 30
 - für MySQL* 530
 - Linux unter* 34
 - Programme unter* 30
 - Version 3.0* 30
- Grafik 323
 - Inkscape* 359
- Grafikkarte 209
- AGP 210
- Auflösung* 209
- Farbtiefe* 209
- Geschwindigkeit* 210
- Konfiguration, SaX* 213
- PCI* 210
- RAMDAC* 210
- Grafiktablett 214
- Grafische Benutzeroberfläche 25, 207, 944
 - Drag and Drop* 865
 - X Window Server* 207
- GRANT, MySQL-Anweisung 561
- Green Book 944
- Green Book (CDI) 386
- grep, Kommando 196, 909
- Groß- und Kleinschreibung
 - Dateinamen* 153
 - ignorieren in RegExp* 830
- groupadd, Kommando 188, 911
- groupdel, Kommando 912
- Group-ID 128, 190
 - Effektive Group-ID (EGID)* 192
 - per Setgid ändern* 192
 - von Prozessen* 868
- GRUB, Bootloader 400
- Grundkonfiguration 123
- Grundlinie, Typografie 277
- Gruppen
 - verwalten (YaST)* 134
- Gruppieren, Inkscape 363
- gt, Perl-Operator 821
- GtK+, Grafikbibliothek 219
- GUI → Grafische Benutzeroberfläche
- gunzip, Kommando 395
- gzip, Kommando 395, 912

H

- Handbücher 118
- Handschriftliche Antiqua 283
- Handshake 944
- Handshake, Modem-Kommunikation 614
- Hard Link 154
 - erzeugen (Shell)* 182
- Hardware
 - bei Installation konfigurieren* 101
 - BIOS* 840
 - Bus* 841, 851
 - Bus Mastering* 855

- Chipsatz 841
- Diskettenlaufwerk 378
- DMA-Kanal 855
- Drucker 259
- Grafikkarte 209
- Hot-Plugging 253
- I/O-Basisadresse 854
- IRQ 854
- Konfiguration (YaST) 265
- Konfiguration, GNOME 240
- Monitor 210
- Prozessor 840
- RAM 850
- RAM-Speicher 840
- Ressourcen 854
- ROM-Speicher 840
- Scanner 268
- Schnittstellen 841
- Soundkarte 266
- Steuerung durch Betriebssystem 24
- Unterstützung unter Linux 42
- Hardware-Interrupt 848
- Hashes
 - Schlüssel 818
- Hashes, Perl-Variablen 816
- Hayes, Modem-Befehlssatz 613
- Hayes-Befehlssatz 940
- head, Kommando 195, 912
- Helligkeit/Kontrast, GIMP 359
- Hewlett-Packard
 - HP UX, Betriebssystem 28
- Hexadezimalzahlen in Perl 819
- Hibernation (0xA0), Dateisystem-ID 71
- HIER-Dokumente 167
 - Perl 821
 - Variablensubstitution 822
- Hilfe 118
 - GNU info 121
 - Manpages 118
- Hilfslinien (GIMP) 341
- Hines, Chuck 247
- Hintergrundprozesse 162
- History 164
- Home-Verzeichnis 151
 - von root 152
- Hostname
 - bei Installation festlegen 95
 - konfigurieren (YaST) 599
- hostname, Kommando 913
- Host-zu-Host-Transportschicht, TCP/IP 577
- Hot Spots (Image-Maps) 488
- Hot-Plugging 253, 859
- HP UX, Betriebssystem 28
- HPGL 260
- HR/DSSS, WLAN-Technik 602
- HSB-Farbmodell 326
- HTML
 - <a>-Tag 482
 - <area>-Tag 488
 - <body>-Tag 473
 -
-Tag 474
 - <caption>-Tag 490
 - <dd>-Tag 481
 - <dl>-Tag 481
 - <dt>-Tag 481
 - <form>-Tag 493
 - <h1> bis <h6>, Tags 477
 - <head>-Tag 473
 - <html>-Tag 473
 - -Tag 486
 - <input>-Tag 493
 - -Tag 479
 - <map>-Tag 488
 - <meta>, Tag 496
 - <meta>-Tag 474
 - -Tag 479
 - <p>-Tag 476
 - <pre>-Tag 478
 - <select>-Tag 495
 - <style>-Tag 502
 - <table>-Tag 489
 - <td>-Tag 490
 - <textarea>-Tag 496
 - <th>-Tag 490
 - <title>-Tag 473
 - <tr>-Tag 490
 - -Tag 478
- Absatzausrichtung 477
- Absätze 476
- Absendeknopf 494
- Anker 485
- Aufzählungen 478
- Aufzählungszeichen wählen 479
- Auswahlmenü, Formulare 495
- Beschreibung für Suchmaschinen 497
- Bilder als Hyperlinks 487
- Bilder einbetten 486
- Blocksatz 477

- Body 472
- Button 494
- Checkboxen 494
- Client-Side-Image-Maps 488
- CSS 499
- Definitionsliste 481
- DOCTYPE-Angabe 473
- Dokumentkopf 472
- Dokumentkörper 472
- Dokumentstruktur 472
- Dokumenttitel 473
- Dokumenttyp 473
- Download-Hyperlinks 483
- E-Mail-Hyperlinks 484
- Entity-Referenzen 474
- Formulardaten-Codierung 493
- Formulare 492
- Formulare, Absendeknopf 494
- Formulare, Button 494
- Formulare, Checkboxen 494
- Formulare, Hidden-Felder 495
- Formulare, Löschknopf 494
- Formulare, Passwortfelder 494
- Formulare, Radiobuttons 494
- Formulare, Reset-Button 494
- Formulare, Schaltfläche 494
- Formulare, Submit-Button 494
- Formulare, Textbereich 496
- Formulare, Textfelder 494
- Formulare, Versandmethode 493
- Formularelemente 493
- Formular-URL 493
- FTP-Hyperlinks 484
- geeignete Titel 473
- geschütztes Leerzeichen 476
- GET, Formularversandmethode 493
- Glossarliste 481
- Head 472
- Hidden-Formularfelder 495
- Hyperlinks 481
- Hyperlinks ins Web 483
- Image-Maps, clientseitige 488
- Image-Maps, serverseitige 487
- Kommentare 489
- Layer 507
- Links 481
- Listen 478
- Löschknopf 494
- Meta-Tags 496
- nicht nummerierte Liste 478
- nummerierte Liste 479
- Nummerierungsart wählen 479
- Passwortfelder 494
- Pfadangaben in Links 482
- POST, Formularversandmethode 493
- Quanta Plus, Editor 510
- Radiobuttons 494
- Refresh 497
- Reset-Button 494
- robots.txt-Datei 499
- Schaltfläche 494
- Schlüsselwörter für Suchmaschinen 498
- Seite neu laden 497
- seiteninterne Links 485
- Server-Side-Image-Maps 487
- Sonderzeichen 474
- style-Attribut 502
- Stylesheets 499
- Submit-Button 494
- Suchmaschinen-Informationen 497
- Tabellen 489
- Tabellen ausrichten 491
- Tabellen-Attribute 491
- Tabellenbeschriftung 490
- Tabellenbreite 491
- Tabellenhöhe 491
- Tabellenrahmen 491
- Tabellenzeilen 490
- Tabellen-Zellabstand 491
- Tabellenzellen 490
- Tabellenzellen verbinden 492
- Tabellenzellen-Attribute 491
- Tabellenzellen-Ausrichtung 491
- Textbereich 496
- Textfelder 494
- Textformatierung 474
- Textmarken 485
- Überschriften 477
- verschachtelte Listen 480
- vorformatierter Text 478
- XHTML 472
- Zeichensatz angeben 474
- Zeilenumbruch 474
- htpasswd, Apache-Dienstprogramm 733
- HTTP 944
- http_proxy, Umgebungsvariable 684
- HTTP-Anfrage 695

- httpd.conf, Apache-Konfigurationsdatei 730
 - HTTP-Header 701
 - HTTP-Protokoll → Hypertext Transfer Protocol
 - HTTP-Proxy (squid) 682
 - HTTPS 715
 - HTTP-Server 706
 - Hub 594
 - Hughes, James 35
 - HUP, Signal 868
 - Hurenkind, Absatzfehler 289
 - Hyperlinks, HTML 481
 - Hypertext 944
 - Hypertext Transfer Protocol (HTTP) 691
 - Anfrage 695
 - Header 701
 - Kommunikationsablauf 692
 - Statuscodes 696
-
- I/O-Basisadresse 854, 944
 - IANA (Internet Assigned Numbers Authority) 582, 944
 - IBM
 - AIX, Betriebssystem 28
 - IceWM, Window-Manager 249
 - ICMP 944
 - iconv, Kommando 913
 - IEEE 944
 - IEEE 802.11 (WLAN) 602
 - IEEE 802.3 (CSMA/CD) 592
 - IEEE-1394 943
 - if, Perl 822
 - nachgestellt, Perl 823
 - if, Shell-Built-in 798
 - ifconfig, Kommando 595, 914
 - IFS, Umgebungsvariable 806
 - IMAP 779, 944
 - Cyrus-Server 781
 - Index
 - im RDBMS 526
 - Indizes, MySQL 542
 - inet_aton, Netzwerkprogrammierung 879
 - inetd 762
 - info, Kommando 121
 - Bedienung 121
 - blättern 122
 - Navigation 121
 - Infrarot-Anschluss 860
 - init, Kommando 143, 915
 - init-Prozess 867
 - Init-Verzeichnisse 143
 - Inkrementelle Suche, Emacs 431
 - Inkscape 359
 - Auswahlwerkzeug 360
 - Bézier-Kurven-Werkzeug 362
 - Ellipsenwerkzeug 361
 - Farbverlaufswerkzeug 363
 - Freihandwerkzeug 362
 - Füllwerkzeug 362
 - Gruppieren 363
 - Kalligraphie-Werkzeug 362
 - Kurvenauswahlwerkzeug 360
 - Polygonwerkzeug 361
 - Rechteckwerkzeug 361
 - Sternwerkzeug 361
 - SVG 363
 - Textwerkzeug 362
 - Werkzeugeleiste 360
 - InnoDB, MySQL-Tabellentyp 531
 - inode 154
 - INPUT, iptables-Kette 647
 - INSERT, MySQL-Anweisung 544
 - insmod, Kommando 254, 915
 - Installation 53
 - abschließen 101
 - Booten von CD/DVD aktivieren 59
 - Bootmanager konfigurieren 94
 - Boot-Optionen 59
 - Dateisysteme wählen 71
 - Datum und Uhrzeit einstellen 64
 - Desktop auswählen 65
 - Domainnamen festlegen 95
 - DSL einrichten 100
 - Erstkonfiguration 94
 - Experteneinstellungen 94
 - Festplatte vorbereiten 67
 - Firewall aktivieren 96
 - GNOME 65
 - große Festplatten 58
 - Hardware-Konfiguration 101
 - Hostnamen festlegen 95
 - IP-Adresse einstellen 99
 - ISDN einrichten 100
 - KDE 65
 - Lizenzvereinbarung 63
 - Medien überprüfen 63
 - Modems konfigurieren 100

- neben bestehendem Windows* 54
- Netzwerkkonfiguration* 96
- Netzwerkschnittstellen* 97
- Neuinstallation oder Update* 64
- Partition reservieren* 55
- Partitionierung* 66
- Proxy einrichten* 100
- root-Passwort festlegen* 83
- RPM-Pakete* 137
- selbst kompilieren* 140
- Software auswählen* 84
- Sprachen einstellen* 92
- Standardbenutzer einrichten* 80
- Standard-Runlevel* 94
- starten* 59
- Subnetzmaske festlegen* 99
- Systemstart* 94
- Tastaturbelegung* 93
- Textmodus* 65
- VNC aktivieren* 100
- Zeitzone einstellen* 64
- Installation von Abbildern 94
- Installationspakete
 - für MySQL* 530
- Installationsquellen (Softwareauswahl) 90
- INT, MySQL-Datentyp 546
- Integer 944
- Integer-Literale, Perl 819
- Intel 841
- Intel-Assembler 848
- Internet
 - Zugang per Modem* 613
 - Zugang über DSL* 608
 - Zugang über ISDN* 616
 - Zugang über Wählverbindungen* 606
- Internet Protocol 944
- Internet Protocol (IP) 578
 - IP-Adressen* 579
 - IPv4* 579
 - IPv4 und IPv6* 579
 - Routing* 586
 - TTL* 587
- Internet Service Provider
 - einrichten (YaST)* 610
- Internetschicht, TCP/IP 577
- Internet-Schichtenmodell 576
- Inter-Prozess-Kommunikation 869
 - System V IPC* 869
 - über Pipes* 869
 - über Signale* 869
- Interprozesskommunikation
 - Named Pipes* 173
 - Signale* 202
- Interrupts
 - Hardware* 848
- Intrusion Detection System 642, 944
- IO 881, 882, 883, 884
- IP-Adressen 579
 - Broadcast-Adresse* 580
 - CIDR* 581
 - DHCP* 679
 - für Sockets* 878
 - IPv4* 579
 - IPv6* 584
 - Klassen* 579
 - Loopback* 582
 - manuell festlegen (bei Installation)* 99
 - mehrere für eine Netzwerkkarte* 598
 - private* 582
 - spezielle* 582
 - Subnet Mask* 581
 - VLSM* 582
 - zuweisen, Linux* 596
- IP-basierter virtueller Host 714
- IPC 944
- IPC → Inter-Prozess-Kommunikation
- IP-Header
 - IPv4* 582
 - IPv6* 585
- IP-Routing 586
- iptables, Kommando 646
 - ACCEPT, Ziel* 648
 - Beispiele* 650
 - DROP, Ziel* 648
 - filter, Tabelle* 647
 - FORWARD, Kette* 647
 - INPUT, Kette* 647
 - Ketten* 647
 - mangle, Tabelle* 647
 - nat, Tabelle* 647
 - OUTPUT, Kette* 647
 - Parameter* 648
 - POSTROUTING, Kette* 647
 - PREROUTING, Kette* 647
 - QUEUE, Ziel* 648
 - RETURN, Ziel* 648
 - Tabellen* 647
 - Ziele* 647

- IPv4 579
 - Datagramme* 582
 - IPv6 584
 - Datagramme* 585
 - Tunnelung* 585
 - IrDA 861, 944
 - IRIX 72
 - IRQ 854, 945
 - reserviert* 854
 - ISA 945
 - ISDN 616, 945
 - Allgemeines* 616
 - anschießen* 617
 - bei der Installation konfigurieren* 100
 - einrichten (YaST)* 618
 - Euro-ISDN-Protokoll* 618
 - Kanäle* 617
 - SyncPPP* 619
 - ISDN-Karte 617
 - ISO 945
 - ISO 9660, Dateisystem 378
 - ISP → Internet Service Provider
 - ISTLEER(), Calc-Funktion 321
 - ISTTEXT(), Calc-Funktion 321
 - ISTZAHL(), Calc-Funktion 321
 - ITS, Betriebssystem 26
 - iwconfig, Kommando 606
 - iwlist, Kommando 606
- J**
-
- JFS, Dateisystem 72
 - jobs, Shell-Builtin 163
 - JOIN, MySQL-Klausel 553
 - join, Perl-Anweisung 817
 - Joliet 945
 - Journaling
 - Modi* 75
 - Journaling-Dateisystem 72, 376, 945
 - JPEG, Bilddateiformat 334
- K**
-
- K3b, CD-/DVD-Brennsoftware 390
 - Kabelanschluss als Internetzugang 608
 - Kalligraphie-Werkzeug, Inkscape 362
 - Kanäle, GIMP 358
 - Kapitalchen 279, 945
 - Kartensteckplätze
 - AGP* 856
 - PCI* 856
 - PCI-Express* 856
 - Kate, KDE-Texteditor 108, 237
 - KDE 220, 945
 - als Haupt-Desktop (Installation)* 65
 - als root arbeiten* 124
 - amaroK, Audio-Player* 238
 - Dateien löschen* 112
 - Desktop wechseln* 107
 - Dolphin, Dateimanager* 107, 234
 - drucken* 264
 - erste Schritte mit* 106
 - GNOME, Vergleich mit* 219
 - Kate, Texteditor* 108, 237
 - KMail* 238
 - KOffice* 237
 - Konfiguration* 221
 - Konqueror* 110
 - Konsole (Terminalfenster)* 157
 - Kontakt* 238
 - Kooka, Scan-Programm* 270
 - KRename* 238
 - Krita, Bildbearbeitung* 237
 - Krusader* 238
 - KSpread* 237
 - KWord* 237
 - NetworkManager* 601
 - Panel* 106
 - Qt-Bibliothek* 219
 - Quanta Plus, Webeditor* 510
 - Taskwechsel* 107
 - Terminalfenster* 157
 - YaST starten* 124
 - Zugriffsrechte setzen* 110
 - kdewebdev → Quanta Plus, Webeditor
 - KDM, Display-Manager 104, 217
 - Kegelhöhe, Schriftgröße 276
 - Kernel 861, 945
 - Child-Prozess* 867
 - exec(), Systemaufruf* 867
 - fork(), Systemaufruf* 867
 - init-Prozess* 867
 - kill(), Systemaufruf* 868
 - kompilieren* 255, 257
 - Mikrokern* 34, 862
 - monolithischer* 34, 862

- neuen installieren* 257
- Parent-Prozess* 867
- pause()*, *Systemaufruf* 868
- Swap-Partition* 875
- Systemaufrufe* 863, 865
- Task Scheduler* 863
- Kernel-Module 253
 - Abhängigkeiten auflösen* 254
 - entladen* 255
 - laden* 254
 - Liste anzeigen* 253
 - Optionen* 254
- Kernelmodus 863, 867
- Kernighan, Brian 27
- Kerning 945
- Kerning, Typografie 278
- Ketten, iptables 647
- keys, Perl 818
- kill(), *Systemaufruf* 868
- kill, Kommando 202, 916
- KILL, Signal 868
- killall, Kommando 203, 916
- Klammern
 - in RegExp* 829
 - korrespondierende, Vim* 409
- Klassen, IP-Adressen 579
- Klassizistische Antiqua 282
- Klonen, GIMP-Werkzeug 354
- KMail, KDE-Mail-Client 238
- Knoppix 53
- Knuth, Donald E. 511
- Koaxialkabel 945
- KOffice 237
 - KPresenter* 237
 - KSpread* 237
 - KWord* 237
- Kommandos
 - alias, Shell-Built-in* 783
 - apropos* 120, 894
 - Art ermitteln* 161
 - awk* 811
 - bc* 796, 895
 - bunzip2* 395
 - bzip2* 395, 896
 - case, Shell-Built-in* 801
 - cat* 194
 - cd* 185, 897
 - cdparanoia* 390, 897
 - chgrp* 190, 898
 - chkconfig* 145, 898
 - chmod* 190, 899
 - chown* 190, 899
 - clear* 900
 - cp* 179, 900
 - crontab* 786, 900
 - curl* 901
 - date* 204, 902
 - dd* 374, 904
 - diff* 198, 904
 - dig* 633
 - dnsdomainname* 905
 - domainname* 905
 - dos2unix* 905
 - du* 382, 906
 - dvipdf* 513
 - dvips* 513
 - echo* 193, 906
 - export, Shell-Built-in* 794
 - expr* 797, 906
 - fg* 162
 - fgrep* 910
 - file* 907
 - find* 187, 908
 - for, Shell-Built-in* 803
 - fsck* 382
 - ftp* 769
 - grep* 196, 909
 - groupadd* 188, 911
 - groupdel* 912
 - gunzip* 395
 - gzip* 395, 912
 - head* 195, 912
 - hostname* 913
 - iconv* 913
 - if, Shell-Built-in* 798
 - ifconfig* 595, 914
 - info* 121
 - init* 143, 915
 - insmod* 254, 915
 - iptables* 646
 - iwconfig* 606
 - iwlist* 606
 - jobs (Shell-Builtin)* 163
 - kill* 202, 916
 - killall* 203, 916
 - latex* 513
 - ldapsearch* 677
 - less* 194, 917

- ln* 182, 918
- logger* 789, 918
- losetup* 396
- lpr* 264
- ls* 183, 889
- lsmmod* 253, 919
- mac2unix* 905
- make* 141
- man* 118, 919
- mkdir* 186, 920
- mkfs* 382
- modprobe* 254, 920
- mount* 155, 379, 657, 663, 921
- mv* 181, 922
- netstat* 632, 922
- nice* 923
- nslookup* 633, 923
- passwd* 189, 923
- pdflatex* 514
- ping* 630, 924
- popd* 186, 925
- ps* 201, 925
- pstree* 202, 925
- pushd* 186, 925
- pwd* 185, 926
- rm* 181, 926
- rmmod* 255, 926
- rpm* 137
- runlevel* 143, 927
- sax2* 212
- scp* 687
- sed* 810
- sendmail* 779
- shred* 927
- shutdown* 203, 928
- slapadd* 677
- sort* 199, 929
- ssh* 686, 928
- startx* 216
- su* 124, 188, 929
- sudo* 789
- tail* 195, 930
- tar* 141, 393, 930
- tcpdump* 635
- tee* 931
- telnet* 634, 931
- test* 798
- time* 932
- top* 202
- traceroute* 631
- type* 161
- typeset, Shell-Built-in* 796
- umount* 380, 932
- unalias, Shell-built-in* 785
- uname* 200, 932
- uniq* 199, 933
- unix2dos* 933
- unzip* 396, 937
- uptime* 200, 934
- useradd* 188, 934
- userdel* 935
- visudo* 790
- wc* 199, 935
- whatis* 120, 936
- whereis* 936
- which* 936
- who* 200, 936
- whoami* 176, 936
- wodim* 389, 937
- zip* 396, 937
- zypper* 140
- Kommandos, umask 192
- Kommandozeile 945
- Kommentare
 - in HTML* 489
- Kommunikation zwischen Prozessen 869
- Kompilieren
 - make, Kommando* 141
 - Makefiles* 141
 - Open-Source-Projekte* 140
- Kompilieren (Kernel) 255
- Komplementärfarben 325
- kompress, Typografie 278
- Komprimierung 331
 - verlustbehaftete* 331
 - verlustfreie* 331
- Konfiguration
 - Grundeinstellungen bei Installation* 94
- Konfigurationsdateien
 - .bashrc* 158
 - /etc/cups/cupsd.conf* 262
 - /etc/fstab* 380
 - /etc/hosts* 621
 - /etc/profile* 158
 - /etc/sysconfig/displaymanager* 216
 - /etc/vsftpd.conf* 771
 - /etc/xinetd.conf* 763
 - /etc/xinetd.d/** 764
 - Apache 2* 709

- Konqueror
 - als Dateimanager* 110
 - Dateien löschen* 112
 - Zugriffsrechte setzen* 110
- Konsole 25, 945
- Konsole (KDE-Terminal)
 - Shell-Skript ausführen* 111
- Konsole, KDE-Terminal 157, 235
- Konstante lineare Geschwindigkeit 386
- Konstante Winkelgeschwindigkeit 370
- Kontakt, KDE-Informationsmanager 238
- Kontrollstrukturen
 - foreach, Perl* 817
 - Shell-Skripte* 798
- Kooka, Scan-Programm 270
- Kooperatives Multitasking 863
- Kopieren (Dateien) 179
- Koprozessor 841
- Korn Shell (ksh) 159
- Körperfarben 325
- KPresenter 237
- KRename 238
- Krita, KDE-Bildbearbeitung 237
- Krusader 238
- Kryptoanalyse 644
- Kryptografie 643
 - asymmetrische Verschlüsselung* 645
 - Cäsar-Code* 643
 - Digitale Signatur* 645
 - Einwegverschlüsselung* 645
 - Grundbegriffe* 643
 - Message-Digest* 645
 - ROT13* 643
 - SSH* 646
 - SSL/TLS* 646
 - symmetrische Verschlüsselung* 644
- ksh 159
- KSpread 237
- KTorrent, BitTorrent-Client 50
- Kubuntu, Linux-Distribution → Ubuntu
- Kursiv, Schriftschnitt 279
- Kurvenuswahlwerkzeug, Inkscape 360
- KWord 237

- L**

- Lab, Farbmodell 326
- Lamarr, Hedy 602
- Lampport, Leslie 512
- LAMP-System 945
- LAN 945
- Laserdrucker 260
 - Farbe* 260
- Lasso, GIMP-Werkzeug 343
- LaTeX 514, 945
- latex, Kommando 513
- LaTeX, Satzsprache 511
 - Aufzählungen* 518
 - Autor angeben* 515
 - Beispieldokument* 512, 521
 - deutsche Texte* 516
 - Dokumentklassen* 514
 - fett und kursiv* 517
 - Formelsatz* 520
 - Gliederung* 515
 - Listen* 518
 - Makropakete* 514
 - Mathematikmodus* 520
 - PDF erzeugen* 513
 - Schriftfamilien* 517
 - Schriftgrößen* 517
 - Sonderzeichen* 516
 - Tabellen* 519
 - Titel angeben* 515
 - Umlaute* 516
- Laufweite, Typografie 278
- Laufwerke
 - Anschlüsse* 856
- Lauschende Sockets 882
- Layout
 - Anschnitt* 293
 - Broschüren* 292
 - Fachbücher* 291
 - Kataloge* 292
 - Linien* 294
 - Prospekte* 292
 - Rahmen* 294
 - Romanseiten* 291
 - Tageszeitungen* 292
 - umfließen* 292
 - Werbeanzeigen* 292
 - Zeitschriften* 292
- LBA
 - 48 Bit* 58
- LBA (Festplattenadressierung) 369
- LCD 946

- LCD-Monitor 211
 - Funktionsprinzip* 211
 - TFT 211
 - Vorteile 211
- LDAP 673, 946
 - Attribute 674
 - Filter 677
 - Grundlagen 673
 - LDIF-Format 673
 - OpenLDAP-Server einrichten 676
 - phpLDAPadmin 677
 - Schema 674
 - UNIX-Benutzerkonten 675
- LDAP-Elemente
 - cn 674
 - dc 674
 - ou 674
- ldapsearch, Kommando 677
- LDIF 673
- le, Perl-Operator 821
- Lead-in-Area (CD) 387
- Lead-out-Area (CD) 387
- Leasing (DHCP) 680
- LED-Drucker 946
- Leerzeichen
 - in Dateinamen* 154
- left, CSS-Angabe 508
- less, Kommando 194, 917
 - zur Manpage-Anzeige* 119
- letter-spacing, CSS-Angabe 505
- Level-1-Cache 843
- Level-2-Cache 843
- Libtool 140
- Licht 324
- Lichtwellen 324
- LIFO 946
- Ligaturen 279
- LIKE, MySQL-Klausel 550
- LILO, Bootloader 401
- LIMIT, MySQL-Klausel 552
- Lineale (GIMP) 340
- Linear-Antiqua
 - serifenbetonte* 282
 - serifenlose* 282
- Lineare Geschwindigkeit, konstante 386
- line-height, CSS-Angabe 505
- Linien, Gestaltungselement 294
- Link
 - Hard Link* 154
 - Symbolischer Link* 154
- Linux 28, 37
 - als Markenname* 43
 - Aufbau 861
 - Debian, Distribution* 39
 - Distributionen* 38, 946
 - Fedora Core, Distribution* 39
 - Hardware-Plattformen* 37
 - Hardware-Unterstützung* 42
 - IP-Adresse zuweisen* 596
 - Kampf der SCO-Group gegen* 45
 - Kernel-Versionen* 35
 - Mandriva, Distribution* 40
 - Namensstreit* 31
 - openSUSE, Distribution* 39
 - Prozessmodell* 867
 - Rechtliche Probleme* 42
 - RedHat-Distribution* 39
 - Slackware, Distribution* 39
 - Softwarepatente, Bedrohung durch* 46
 - Softwareunterstützung* 40
 - Tux (Maskottchen)* 34
 - Ubuntu, Distribution* 39
 - Ursprünge* 33
 - vs. Microsoft* 44
 - Waschmittel* 43
- Linux (0x83), Dateisystem-ID 71
- Linux LVM (0x8E), Dateisystem-ID 71
- Linux Mark Institute 43
- Listen
 - Perl* 816
 - qw//, Perl* 820
- listen(), Perl-Funktion 882
- Listen, Apache-Direktive 734
- Literal 819, 946
- Little-Endian-Architektur 946
- Live-CD
 - Knoppix* 53
 - openSUSE* 53
- LiveCD 49
- Lizenz
 - der Distribution* 63
- Lizenzen
 - Apache-Lizenz* 31
 - BSD* 31
 - GPL* 29
- Lizenzen, MySQL 530
- ln, Kommando 182, 918
- LOAD DATA INFILE, MySQL-Anweisung 567
- LoadModule, Apache-Direktive 734

- localtime, Perl-Anweisung 814
 - Lochkarten 25
 - Logdatei 788, 946
 - /var/log/messages* 789
 - facility* 788
 - logger, Kommando* 789
 - MySQL* 568
 - priority* 789
 - Syslog* 788
 - logger, Kommando 789, 918
 - Login 946
 - Benutzernamen angeben* 105
 - grafisches* 104
 - KDM* 104
 - Passwort eingeben* 105
 - root, grafisch als* 105
 - Logische Operatoren
 - Short-Circuit-Logik* 822
 - Lokale Zeit
 - im Vergleich zu UTC* 65
 - Loopback-Adresse 582
 - Loop-Devices 396
 - Löschabfragen, MySQL 557
 - Löschen
 - Dateien, GNOME* 117
 - Dateien, KDE* 112
 - Löschen (Dateien) 181
 - Löschen, Vim 412
 - losetup, Kommando 396
 - lpd 261
 - lpr, Kommando 264
 - ls, Kommando 183, 889
 - Elemente der ausführlichen Liste* 184
 - lsm, Kommando 253, 919
 - lt, Perl-Operator 821
 - Lupe, GIMP-Werkzeug 345
 - LVM 384
 - Dateisystem-ID für* 71
- ## M
-
- m
 - n-Beziehung, RDBMS* 527
 - m//, Perl-Operator 827, 830
 - Mac OS X 28
 - mac2unix, Kommando 905
 - MAC-Adresse 593, 946
 - Macek, Marko 249
 - Mail Transfer Agent 779
 - mailto
 - HTML-Hyperlink* 484
 - MAILTO, Umgebungsvariable 787
 - Mainboard
 - Chipsatz* 841
 - Kartensteckplätze* 856
 - Majuskelziffern 280
 - make, Kommando 141
 - Makefiles 141
 - man, Kommando 118, 919
 - Steuerung* 119
 - Mandriva, Linux-Distribution 40
 - mangle, iptables-Tabelle 647
 - Man-in-the-Middle-Angriff 639
 - Manpages 118
 - apropos, Kommando* 120
 - Kategorien* 119
 - man, Kommando* 118
 - Stichwörter suchen* 120
 - Titel suchen* 120
 - whatis, Kommando* 120
 - Mantel, Hubert 47
 - Manuelle Partitionierung 67
 - Maps (NIS) 666
 - margin, CSS-Angabe 505
 - Markenname Linux 43
 - Maschinenbefehle 848
 - Massenspeicher
 - DVD* 387
 - Master Boot Record 372
 - sichern* 374
 - Matching
 - reguläre Ausdrücke* 830
 - Mathematik (LaTeX) 520
 - Maus
 - Einsatz in der Shell* 165
 - MAX(), Calc-Funktion 320
 - MAX(), MySQL-Funktion 554
 - MD5, Verschlüsselungsalgorithmus 134
 - md5sum, Kommando 60
 - für Windows* 61
 - Mediävalziffern 280
 - MEDIUMINT, MySQL-Datentyp 546
 - Mehrkern-Prozessoren 845
 - Mena, Federico 219
 - menuconfig, Kernel-Konfigurator 256
 - Message-Digest 645
 - METAFONT
 - TeX-Zeichensätze* 511

- Microsoft
 - Angriffe gegen Linux* 44
- MIDI 267, 946
 - FM-Synthese* 267
 - Wavetable-Synthese* 267
- Midnight Commander 237
- Mikrokern 34, 862
- Mikroprozessor 840
- Mikrotypografie 284
 - Satzzeichen* 284
 - Zahlen* 285
- MIME 946
 - E-Mail-Nachricht* 775
 - Multipart-Nachrichten* 776
 - Nachrichtenheader* 775
 - Types* 776
 - XML-Dokumente* 441
- MIN(), Calc-Funktion 320
- MIN(), MySQL-Funktion 554
- Minix 32
- Minuskelziffern 280
- MIPS (CPU-Geschwindigkeit) 846
- Mirroring (RAID) 383
- Mittellänge, Schriftgröße 277
- MITTELWERT(), Calc-Funktion 316
- Mixed-Mode-CD 386
- mkdir, Kommando 186, 920
- mkfs, Kommando 382
- mlterm, Terminalemulation 249
- MMU 843, 875, 946
 - Seitentabelle* 875
- Mnemonics 940
- Modem 613
 - AT-Befehlssatz* 613
 - bei der Installation konfigurieren* 100
 - einrichten (YaST)* 614
 - Handshake* 614
 - Hayes-Befehlssatz* 613
 - Pulswahl* 614
 - Tonwahl* 614
- modprobe, Kommando 254, 920
- Modularität
 - von UNIX* 27
- Moirée-Effekt
 - verhindern* 328
- Molenaar, Bram 406
- Monitor 210
 - Bildwiederholrate* 210
 - Konfiguration, SaX* 213
 - LCD* 211
 - Röhrenmonitor* 210
 - Zeilenfrequenz* 211
- Monolithischer Kernel 34, 862
- Monty (Michael Widenius) 530
- Moore's Gesetz 843
- mount, Kommando 155, 379, 921
 - NFS-Volumes* 657
 - SMB-Volumes* 663
- Mounten 154, 946
 - NFS-Freigaben* 657
- Mountpoint 69
 - einstellen (Installation)* 76
- MOV-Befehl, x86-Assembler 848
- MP3
 - Audio-Dateiformat* 946
 - Wiedergabe mit Banshee* 247
- MPEG 946
- MTU 598, 946
- MULTICS 26
- Multipart-E-Mails 776
- Multisession-CD 387
- Multitasking 863, 866
 - kooperatives* 863
 - präemptives* 864
 - Unterstützung durch CPU* 842
- Mustervergleich 196
- mv, Kommando 181, 922
- MX-Record
 - BIND-Nameserver* 630
- my, Perl-Variablendeklaration 815
- my.cnf, MySQL-Konfigurationsdatei 567
- MyISAM, MySQL-Tabellentyp 531
- MySQL 530, 946
 - absichern* 535
 - Aggregatfunktionen* 553
 - Aliasnamen in Abfragen* 551
 - ALTER TABLE, Anweisung* 556
 - Änderungsabfragen* 555
 - AS, Klausel* 551
 - Aufzählungstypen* 548
 - Auswahlabfragen* 549
 - Auswahlkriterien* 549
 - Authentifizierung* 559
 - automatisch starten* 533
 - AVG(), Funktion* 554
 - Backups* 564
 - Backups automatisieren* 566
 - Benutzerrechte* 561

- Benutzerverwaltung* 558
- BIGINT, Datentyp* 546
- Binärblöcke* 548
- BLOB, Datentyp* 548
- CHAR, Datentyp* 547
- CREATE DATABASE, Anweisung* 539
- CREATE TABLE, Anweisung* 540
- CREATE USER, Anweisung* 560
- DATABASE(), Funktion* 538
- Datenbanken erstellen* 539
- Datenbanken löschen* 557
- Datentypen* 545
- DATETIME, Datentyp* 548
- Datums- und Uhrzeittypen* 548
- DECIMAL, Datentyp* 547
- DELETE, Anweisung* 557
- DESCRIBE, Anweisung* 543
- DOUBLE, Datentyp* 547
- DROP DATABASE, Anweisung* 557
- DROP TABLE, Anweisung* 557
- DROP USER, Anweisung* 563
- Eigenschaften* 531
- Einfügeabfragen* 544
- ENUM, Datentyp* 548
- Export in Textdateien* 567
- Falcon-Tabellen* 532
- Fließkommadatentypen* 547
- FLOAT, Datentyp* 547
- FLUSH PRIVILEGES, Anweisung* 564
- FLUSH TABLES, Anweisung* 565
- Funktionen* 550
- GRANT, Anweisung* 561
- Hilfsprogramme* 532
- Import aus Textdateien* 567
- Import/Export* 564
- Indizes* 542
- InnoDB-Tabellen* 531
- INSERT, Anweisung* 544
- Installationspakete* 530
- INT, Datentyp* 546
- JOIN, Klausel* 553
- Joins* 552
- Kollationen (Sortierfolgen)* 539
- Konfigurationsdateien* 567
- LIKE, Klausel* 550
- LIMIT, Klausel* 552
- Lizenz* 530
- LOAD DATA INFILE, Anweisung* 567
- Logdateien* 568
 - lesen* 569
- Löschabfragen* 557
- manuell installieren* 532
- MAX(), Funktion* 554
- MEDIUMINT, Datentyp* 546
- MIN(), Funktion* 554
- Mustervergleiche* 550
- my.cnf* 567
- MyISAM-Tabellen* 531
- mysqladmin, Hilfsprogramm* 557
- mysqlbinlog, Hilfsprogramm* 569
- mysql-Client* 534
- mysqldump, Hilfsprogramm* 564
- mysqli, PHP-Schnittstelle* 757
- numerische Datentypen* 545
- Operationen* 550
- ORDER BY, Klausel* 555
- REAL, Datentyp* 547
- REGEXP, Klausel* 550
- Replikation* 570
- Replikation nutzen (PHP)* 759
- REVOKE, Anweisung* 563
- SELECT, Anweisung* 549
- SET PASSWORD, Anweisung* 561
- SET, Datentyp* 549
- SHOW CHARACTER SET, Anweisung* 539
- SHOW COLLATION, Anweisung* 540
- SHOW CREATE TABLE, Anweisung* 543
- SHOW DATABASES, Anweisung* 538
- SHOW TABLES, Anweisung* 538
- SMALLINT, Datentyp* 546
- sortieren* 555
- Sortierfolgen* 539
- Storage Engines* 531
- String-Datentypen* 547
- SUM(), Funktion* 554
- Tabellen erstellen* 540
- Tabellen löschen* 557
- Testdatenbank für PHP* 752
- TEXT, Datentyp* 548
- Textblöcke* 548
- TIME, Datentyp* 548
- TIMESTAMP, Datentyp* 548
- TINYINT, Datentyp* 546
- UPDATE, Anweisung* 555
- VARCHAR, Datentyp* 547
- Versionen in openSUSE* 530

- WHERE, Klausel 549
 Zeichensätze 539
 Zufällige Serverwahl 759
 Zugriff über PHP 754
 mysqladmin, Hilfsprogramm 557
 mysqlbinlog, Hilfsprogramm 569
 mysql-Client 534
 beenden 536
 interne Befehle 536
 mit Passwort starten 536
 nicht-interaktiver Betrieb 566
 mysqld 532
 mysqldump, Hilfsprogramm 564
 mysql-Schnittstelle 757
- ## N
-
- Nachbelichten, GIMP-Werkzeug 355
 Namensbasierter virtueller Host 714
 Namensräume, XML 462
 Nameserver 623
 BIND 624
 NameVirtualHost, Apache-Direktive 735
 NAT (Network Address Translation) 580, 946
 nat, iptables-Tabelle 647
 Nation, Robert 247
 Nautilus 115
 Dateien löschen 117
 GNOME-Dateimanager 244
 Zugriffsrechte setzen 115
 Navigation in Vim 407
 ndiswrapper 606
 ne, Perl-Operator 821
 Nessus 642
 netfilter 646
 netstat, Kommando 632, 922
 Network Address Translation (NAT) 580
 NetworkManager 601
 Netzwerke
 Ethernet 592
 inetd 762
 NetworkManager 601
 OSI-Referenzmodell 576
 Schichtenmodell 576
 Sicherheit 637
 Wählverbindungen 606
 xinetd 763
 Zugang per Modem 613
 Zugang über DSL 608
 Zugang über ISDN 616
 Netzwerkkarte
 Konfiguration mit YaST 596
 Netzwerkkonfiguration
 bei Installation 96
 Netzwerkprogrammierung
 Berkeley Socket API 876
 IO::Socket 882
 sockaddr_in 879
 Sockets 876
 Netzwerkschnittstellen
 IP-Adresse festlegen 99
 konfigurieren (Installation) 97
 über DHCP konfigurieren 98
 Zeroconf 99
 Netzzugangsschicht, TCP/IP 577
 Netzzugangsverfahren
 CSMA/CA 603
 CSMA/CD 592
 Neuinstallation 64
 NFS 655, 666, 946
 /etc/exports, Konfigurationsdatei 656
 /etc/fstab-Eintrag 657
 auf andere Server zugreifen 657
 Konfiguration mit YaST 658
 Starten 656
 Verzeichnisse freigeben 656
 nfsd 655
 nice, Kommando 923
 NIS 666, 947
 Client-Konfiguration mit YaST 671
 Maps 666
 Server-Konfiguration mit YaST 667
 NNTP 947
 noatime, Partitionsmodus 74
 noauto, Partitionsmodus 75
 Normalisierung 947
 nouser, Partitionsmodus 75
 Novell 48
 Novell AppArmor 127
 nslookup, Kommando 633, 923
 NS-Record (DNS) 629
 NTFS, Dateisystem 377
 Linux-Zugriff 377
 Nulldevice 168
 Nyquist-Theorem 947

O

-
- OASIS 273
 - Oberlänge, Typografie 278
 - Objektorientierung 947
 - Perl* 883
 - ODER(), Calc-Funktion 321
 - OFDM, WLAN-Technik 602
 - Oktalzahlen in Perl 819
 - Online-Updates 137
 - O-Notation 947
 - Open Source 947
 - open, Perl-Anweisung 825
 - opendir, Perl-Anweisung 826
 - OpenDocument-Format 273
 - OpenLDAP 673, 947
 - Client-Konfiguration mit YaST* 678
 - LDAP-Filter* 677
 - ldapsearch, Kommando* 677
 - phpLDAPadmin* 677
 - Server einrichten* 676
 - slapadd, Kommando* 677
 - slapd.conf, Konfigurationsdatei* 676
 - OpenOffice.org 273
 - Calc* 313
 - Neues in Version 3.0* 274
 - Writer* 274
 - Open-Source-Software 32
 - kompilieren* 140
 - openSUSE
 - Build Service* 51
 - Download, Developer-Version* 50
 - Download, stabile Version* 48
 - Geeko (Chamäleon)* 107
 - Grundkonfiguration* 123
 - installieren* 53
 - Linux-Distribution* 39
 - LiveCD* 49
 - Lizenzvereinbarung* 63
 - Partition reservieren für* 55
 - Wiki* 50
 - openSUSE-Firewall 646
 - konfigurieren (YaST)* 650
 - openSUSE-Live-CD 53
 - Operatoren
 - !~* 830
 - \$, RegExp* 829
 - *, RegExp* 828
 - +, RegExp* 828
 - ., Perl* 821
 - =~* 830
 - ?, RegExp* 827
 - ^, RegExp* 827, 829
 - ``* 872
 - |, RegExp* 830
 - Backticks* 872
 - eq, Perl* 821
 - ge, Perl* 821
 - gt, Perl* 821
 - le, Perl* 821
 - lt, Perl* 821
 - m//* 830
 - ne, Perl* 821
 - s///* 831
 - Stringvergleiche, Perl* 820
 - tr///* 832
 - opt, Systemverzeichnis 151
 - Options, Apache-Direktive 735
 - Optische Mitte 291
 - Orange Book (CD-R, CD-RW) 386, 947
 - ORDER BY, MySQL-Klausel 555
 - Order, Apache-Direktive 735
 - Ornamente, Gestaltungselement 294
 - OSI-Referenzmodell 576, 947
 - OSPF 947
 - ou, LDAP-Element 674
 - OUTPUT, iptables-Kette 647

P

-
- Packet-Sniffer
 - tcpdump* 635
 - Wireshark* 636
 - padding, CSS-Angabe 505
 - Page Fault (Speicher) 875
 - Page File (Auslagerungsdatei) 875
 - Pager 194
 - Paging (Speicher) 843, 875
 - Pakete (Software) 84
 - Abhängigkeiten überprüfen* 92
 - detaillierte Optionen* 91
 - Installationsquellen* 90
 - Paketgruppen* 89
 - Schemata* 84
 - Sprachen* 89
 - Suchfunktion* 90
 - Paketgruppen (Softwareauswahl) 89
 - Panel (GNOME) 113

- Panel (KDE) 106
- PAP 607
- Papierformat 289
 - DIN* 289
 - Nicht-DIN* 290
 - Tageszeitungen* 290
- Parallele Datenübertragung 851
- Parallelport 860
- Parent-Prozess 867, 947
- Partition
 - ändern 70
 - atime, Modus* 74
 - auto, Modus* 75
 - BtrFS-Dateisystem* 71
 - Dateisysteme* 71
 - Dateisystem-ID* 71
 - defaults, Modus* 75
 - Ext2-Dateisystem* 71
 - Ext3-Dateisystem* 72
 - FAT-Dateisystem* 72
 - formatieren* 71
 - Fstab-Optionen* 73
 - für OpenSUSE reservieren* 55
 - Größe* 73
 - Größe ändern* 78
 - JFS-Dateisystem* 72
 - löschen* 78
 - Mountpoint* 69
 - Mountpoint wählen* 76
 - neue erstellen* 77
 - noatime, Modus* 74
 - noauto, Modus* 75
 - nouser, Modus* 75
 - quota, Modus* 75
 - ReiserFS-Dateisystem* 72
 - ro, Modus* 74
 - rw, Modus* 74
 - user, Modus* 75
 - XFS-Dateisystem* 72
- Partitionierung 947
 - /home-Partition* 66
 - /-Partition* 66
 - automatische oder manuelle* 67
 - bei der Installation* 66
 - Dateisysteme* 71
 - Dateisystem-ID* 71
 - Einstellungen ändern* 70
 - Empfehlungen* 80
 - erweiterte Optionen* 79
 - Experteneinstellungen* 79
 - formatieren* 71
 - Fstab-Optionen* 73
 - Größe festlegen* 73
 - Liste vorhandener Partitionen* 68
 - neue Partition erstellen* 77
 - Partition löschen* 78
 - Partitionsgröße ändern* 78
 - root-Partition* 66
- Partitionstabelle 372
- Partitionstypen 373
- passwd, Kommando 189, 923
- Password Authentication Protocol → PAP
- Passwort
 - /etc/passwd-Datei* 127
 - /etc/shadow, Datei* 129
 - ändern (Shell)* 189
 - crack (Knackprogramm)* 129
 - gutes, leicht zu merkendes* 84
 - Optionen (YaST)* 133
 - Verschlüsselung* 133
 - von root zurücksetzen* 403
- Patches, Sicherheit 640
- PATH, Umgebungsvariable 161
- pause(), Systemaufruf 868
- PCDATA, Text in XML 447
- PCI 856, 947
- PCI-Express 856
- PCMCIA 947
- PDF 947
 - aus LaTeX-Dokumenten* 513
- pdflatex, Kommando 514
- PDP, Kleincomputerserie von DEC 27
- Pentium, Prozessorfamilie 846
- Peripherie
 - Einsteckkarten* 856
- Perl 947
 - !~-Operator* 830
 - \$* 814, 816
 - \$_* 817
 - \$_-Operator, RegExp* 829
 - %* 816
 - &&-Operator* 822
 - *-Operator, RegExp* 828
 - +Operator, RegExp* 828
 - .-Operator* 821
 - <>-Operator* 814
 - =~-Operator* 830
 - ?-Operator* 827

@ 816
 @_ 824
 ^-Operator, RegExp 827, 829
 || zur Fallentscheidung 822
 ||-Operator 822
 |-Operator, RegExp 830
 alternative Zeichen in RegExp 827
 Anführungszeichen 820
 Arrays 816
 Arrays zusammenfassen 817
 Ausgabe 814
 chomp-Funktion 814
 close 826
 closedir 826
 Dateien bearbeiten 825
 Dateihandles 825
 Dateitest-Operatoren 823
 defined 823
 Dereferenzierung 818
 die-Anweisung 822
 Eingabe 814
 elsif 823
 eq-Operator 821
 Ersetzen durch RegExp 831
 exec()-Funktion 872
 Exponentialschreibweise 819
 Fließkomma-Literale 819
 foreach-Schleife 817
 ge-Operator 821
 gt-Operator 821
 Hashes 816
 Hexadezimalzahlen 819
 HIER-Dokumente 821
 if 822
 if, nachgestelltes 823
 Integer-Literale 819
 join 817
 keys 818
 le-Operator 821
 Listen 816
 Listenfunktionen 816
 Literale 819
 localtime-Anweisung 814
 lt-Operator 821
 m//, Operator 827, 830
 my 815
 ne-Operator 821
 Oktalzahlen 819
 open-Anweisung 825
 opendir 826

Parameter von Subroutinen 824
 pop 816
 print-Anweisung 814
 Programmabbruch 822
 Programme starten 813
 push 816
 q// 820
 qq// 820
 qw// 820
 readdir 826
 Referenzen 818
 reguläre Ausdrücke 826
 s///, Operator 827
 s///-Operator 831
 scalar-Anweisung 814
 Shebang 813
 shift 816
 Short-Circuit-Logik 822
 Skalare 814, 816
 split 817
 strenge Wertüberprüfung 815
 strict 815
 String-Literale 819
 Strings zerlegen 817
 String-Vergleiche 820
 String-Verkettung 821
 sub 824
 Subroutinen 824
 system()-Funktion 872
 tr///, Operator 832
 unless 823
 unshift 816
 use-Direktive 815
 Variablen 815
 Variablenexistenz prüfen 823
 Variablensubstitution 814
 Verzeichnisse verarbeiten 826
 -w, Modifikator 813
 Warnungen aktivieren 813
 Zeichengruppen in RegExp 827
 Zeilenumbrüche entfernen 814
 Perl, Programmiersprache
 accept()-Funktion 882
 Autoflush-Modus 881
 Backticks 872
 bind()-Funktion 882
 connect()-Funktion 881
 gethostbyname()-Funktion 878
 getprotobyname()-Funktion 878
 getservbyname()-Funktion 879

- IO 881, 882
- listen()*-Funktion 882
- Objektorientierung 883
- pipe()*, Funktion 873
- Pipes verwenden* 872
- Socket-Modul* 877
- Personal Information Manager
 - Evolution* 247
 - Kontakt* 238
- Perspektive ändern, GIMP 348
- Peterson, Christine 32
- Pfad
 - in HTML-Hyperlinks* 482
- Pfad (Verzeichnis) 153
- Pfadwerkzeug, GIMP 344
- PGP 645, 947
- Phishing 948
- PHP 948
 - array_rand()*, Funktion 759
 - Cookies* 750
 - Datei-Uploads* 748
 - Formulardaten auslesen* 747
 - Grundsyntax* 746
 - Installation über YaST* 709
 - manuelle Installation* 739
 - mysql-Schnittstelle* 757
 - MySQL-Replikation nutzen* 759
 - mysql-Schnittstelle* 754
 - MySQL-Server zufällig wählen* 759
 - php.ini, Konfigurationsdatei* 741
 - Programmierung* 745
 - Sessions* 750
- php.ini, Konfigurationsdatei 741
- phpLDAPAdmin, LDAP-Verwaltungstool 677
- phpMyAdmin
 - Grundkonfiguration* 745
 - Installation* 744
- Pica-Point, Schriftgröße 276
- PID (Prozess-ID) 867
- Pigmentfarben 325
- ping, Kommando 630, 924
- Pinsel, GIMP-Werkzeug 354
- Pipe 172, 948
 - Anwendung* 872
 - Named Pipe* 173
 - Perl* 872
 - zur Inter-Prozess-Kommunikation* 869
- pipe()*, Perl-Funktion 873
- Pipeline (CPU-Warteschlange) 846
- Pipette, GIMP-Werkzeug 344
- Pixelgrafik 323
- Plug & Play 948
- PNG, Bilddateiformat 335
- Polygonwerkzeug, Inkscape 361
- pop, Perl-Anweisung 816
- POP3 777, 948
 - Befehle* 778
 - Sitzung* 777
- popd, Kommando 186, 925
- position, CSS-Angabe 508
- POSIX 948
- POSIX-Standard 28
- POST, HTTP-Befehl
 - zum HTML-Formularversand* 493
- Postfix
 - Sendmail-Kompatibilität* 779
 - YaST-Konfiguration* 779
- Postfix, MTA 779
- POSTROUTING, iptables-Kette 647
- PostScript 260, 948
- PPD 264
- PPP 607, 948
 - CHAP* 607
 - PAP* 607
- Präemptives Multitasking 864
- PREROUTING, iptables-Kette 647
- Primärfarben 325
- Primärschlüssel 526, 948
- print, Perl-Anweisung 814
- priority (Syslog) 789
- Prisma 324
- Private IP-Adressen 582
- proc-Dateisystem 397
- Procmail 781
- PRODUKT(), Calc-Funktion 320
- Programmiersprachen
 - C* 27
 - Compiler* 849
 - Systemaufrufe* 849
- Prompt 111, 948
- Proportionalschrift 279, 948
- Provider (Internet) 610
- Proxy
 - bei der Installation konfigurieren* 100
- Proxy-Client 684
- Proxyserver 682
 - Apache* 684
 - squid* 682

- Prozess 142, 948
 Benutzermodus 867
 Child-Prozess 867
 Deadlock 869
 Definition 866
 duplizierte 870
 Exit-Code 175
 Group-ID 868
 im Benutzermodus 863
 im Kernelmodus 863
 init 867
 Kernelmodus 867
 Kommunikation 869
 Management durch Betriebssystem 24
 Multitasking 866
 Parent-Prozess 867
 Prozess-ID (PID) 867
 Race Condition 869
 Reaping 867
 Signalverarbeitung 868
 Threads als Alternative 874
 Unterstützung durch CPU 842, 848
 User-ID 868
 verwalten (Shell) 201
 Verwaltung durch Betriebssysteme 866
 Zombies 867
- Prozessmanagement 24
 Prozessor 840, 843
 Adressbus-Wortbreite 844
 als Bauteil 841
 ALU 841
 Arbeitsweise 847
 Architektur 846
 Aufbau 841
 Bedingter Sprung 847
 Befehlstabelle 842
 Befehlszeiger 842
 Bestandteile 841
 Bus 842
 Cache 842
 CISC 846
 Datenbus-Wortbreite 844
 der Grafikkarte 209
 Effizienz 846
 FLOPS 846
 Intel 841
 Maschinenbefehle 848
 Mehrkern 845
 MIPS 846
 Pentium-Familie 846
 Pipeline 846
 Prozesse 848
 Register 842
 Registerwortbreite 844
 RISC 846
 Sprungbefehle 847
 Stack 847
 Steuerbus-Wortbreite 844
 Steuerwerk 842
 Taktfrequenz 844
 unbedingter Sprung 847
 Unterprogramm-Aufrufe 847
 Wortbreite 844
- Prozessverwaltung 866
 Prüfbit 852
 ps, Kommando 201, 925
 PS/2-Anschluss 860
 PSD, Bilddateiformat 333
 pstree, Kommando 202, 925
 PTR-Record (DNS) 628
 Public-Key-Verschlüsselung 645
 Pulswahlverfahren 614
 Punkt (Typographie) 948
 Punkt, Schriftgröße 276
 push, Perl-Anweisung 816
 pushd, Kommando 186, 925
 pwd, Kommando 185, 926
- ## Q
-
- q//, Perl-Quoting 820
 qq//, Perl-Quoting 820
 Qt, Grafikbibliothek 219
 Qualitätsfaktor, Scan-Auflösung 330
 Quanta Plus, Webeditor 510
 Queue 948
 QUEUE, iptables-Ziel 648
 Quota (Festplattenkontingente) 75
 quota, Partitionsmodus 75
 QUOTIENT(), Calc-Funktion 320
 qw//, Perl-Quoting 820
- ## R
-
- Race Condition 869, 948
 Radiergummi, GIMP-Werkzeug 354

- Rahmen, Gestaltungselement 294
- RAID 383, 948
 - Dateisystem-ID für* 71
 - Levels* 383
 - Mirroring* 383
 - Stripe Set* 383
 - Stripe Set mit Parity* 383
- RAID (OxFD), Dateisystem-ID 71
- RAM 840, 849
 - als Bauteil* 850
 - Auslagerungsdatei* 875
 - Bedeutung in der Speicherhierarchie* 843
 - DDR-RAM* 851
 - der Grafikkarte* 209
 - DIMM-Module* 850
 - Dynamic* 850
 - einbauen* 850
 - empfohlene Menge* 851
 - Paging* 875
 - Segmentierung* 875
 - Seitenfehler* 875
 - Static* 850
 - Verwaltung durch das Betriebssystem* 875
 - virtuelle Adressierung* 875
- RAMDAC (Grafikkarte) 210
- Rasterweite 328
- Rasterwinkel 328
- Raymond, Eric S. 31
- rc-Skripte 146
- RDBMS 525, 948
- readdir, Perl-Anweisung 826
- REAL, MySQL-Datentyp 547
- Reaping 867
- Rechteckauswahl, GIMP-Werkzeug 342
- Rechteckwerkzeug, Inkscape 361
- Record (Datensatz) 525
- Red Book (Audio-CD) 385
- RedHat, Linux-Distribution 39
- Redirect, Apache-Direktive 736
- Referenzen
 - auflösen, Perl* 818
- Referenzen, Perl 818
- Reflektion, Farbe 324
- Regexp
 - Vim, Texteditor* 409
- RegExp → Reguläre Ausdrücke
- REGEXP, MySQL-Klausel 550
- Register 948
 - Wortbreite* 844
- Register der CPU 842
- Reguläre Ausdrücke 196, 948
 - !~, Operator* 830
 - =~, Perl-Operator* 830
 - alternative Textteile* 830
 - alternative Zeichen* 827
 - beliebig viele Zeichen* 828
 - ein oder mehr Zeichen* 828
 - ersetzen* 831
 - Groß- und Kleinschreibung ignorieren* 830
 - in MySQL* 550
 - Klammern* 829
 - Leerzeichen* 829
 - m//, Operator* 830
 - Matching* 830
 - mehrere Treffer finden* 830
 - Modifikatoren* 830
 - Muster* 827
 - optionale Zeichen* 827
 - Perl* 826
 - s///, Operator* 831
 - Teilausdrücke* 829
 - Vim, Texteditor* 409
 - Whitespace* 829
 - Wortgrenzen* 829
 - Wortzeichen* 829
 - Writer* 306
 - Zeichen ausschließen* 827
 - Zeichenanzahl* 828
 - Zeichengruppen* 827
 - Zeilenanfang* 829
 - Zeilenende* 829
 - Ziffern* 829
- Reguläre Ausdrücke (Perl)
 - m//, Operator* 827
 - s///, Operator* 827
- ReiserFS, Dateisystem 72, 376, 948
- Rekursion 948
- Relationale Datenbank 525, 949
 - 1:1-Beziehung* 526
 - 1:n-Beziehung* 527
 - Auswahlabfrage* 529
 - Beziehungen* 526
 - Fremdschlüssel* 526
 - Index* 526
 - Konsistenz* 525
 - m:n-Beziehung* 527
 - Primärschlüssel* 526
 - Schlüssel* 526

- relativer Pfad 153
 - Remote-Verwaltung (VNC) 687
 - Renaissance-Antiqua
 - Französische* 282
 - Venezianische* 282
 - Replikation
 - MySQL 570
 - Require, Apache-Direktive 736
 - Ressourcen
 - Hardware-* 854
 - Zuweisung* 855
 - REST(), Calc-Funktion 320
 - Rettungskonsole 401
 - RETURN, iptables-Ziel 648
 - REVOKE, MySQL-Anweisung 563
 - RFC 579, 949
 - 1035 (DNS) 621
 - 1078, *TCP-Multiplexing* 765
 - 172, *FTP-Urfassung* 767
 - 2045 bis 2049, *MIME* 775
 - 2060, *IMAP* 779
 - 2460 (*IPv6*) 579
 - 2616 (*HTTP/1.1*) 691
 - 2821, *SMTP* 773
 - 2822 (*Textnachricht, aktuell*) 693
 - 2822, *Textnachricht* 775
 - 768 (*UDP*) 591
 - 791 (*IPv4*) 579
 - 793 (*TCP*) 588
 - 821, *SMTP-Urfassung* 773
 - 822 (*Textnachricht*) 693
 - 822, *Textnachricht* 775
 - 867, *daytime* 764
 - 959, *FTP* 767
 - RGB-Farbe 325
 - RISC 949
 - RISC-Prozessor 846
 - Ritchie, Dennis 27
 - rm, Kommando 181, 926
 - rmmod, Kommando 255, 926
 - ro, Partitionsmodus 74
 - robots.txt, Suchmaschinen-Info 499
 - Röhrenmonitor 210
 - ROM 840
 - Bedeutung* 840
 - root 83, 152
 - Benutzer* 869
 - Home-Verzeichnis von* 152
 - in GNOME arbeiten als* 124
 - in KDE arbeiten als* 124
 - Kommandos ausführen als* 789
 - Login als (grafisch)* 105
 - Passwort festlegen* 83
 - root werden* 188
 - Systemverzeichnis* 152
 - Rootkits 638
 - root-Partition 66
 - root-Passwort 83
 - Root-Passwort zurücksetzen 403
 - ROT13 643
 - Round-Robin-DNS 629
 - Router 949
 - Definition* 586
 - Routing 586
 - Konfiguration (YaST)* 600
 - Standardgateway* 600
 - RPC 949
 - rpm, Kommando 137
 - RPM-Pakete 137
 - RS-232 860
 - Runlevel 143, 927
 - init, Kommando* 143
 - Standard bei Installation* 94
 - Runlevel-Editor 145
 - rw, Partitionsmodus 74
 - rxvt, Terminalemulation 248
- ## S
-
- s///, Perl-Operator 827, 831
 - Samba 661, 949
 - /etc/smb.conf, Konfigurationsdatei* 661
 - als Client für Windows-Server* 663
 - Drucker freigeben* 663
 - globale Parameter* 662
 - Konfiguration mit YaST* 664
 - starten* 661
 - Verzeichnisse freigeben* 663
 - Windows-Freigabeart* 663
 - samba-vscan 640
 - Sampling 949
 - SANE 269
 - SAS 859
 - sash 159
 - S-ATA 857, 949
 - Satellit, DSL-Verbindung 609

- Satisfy, Apache-Direktive 736
- Satz
 - umfließen* 292
- Satzspiegel 285
- SaX, X-Konfigurations-Tool 212
 - Grafikkarte* 213
 - Grafiktablett* 214
 - Konfiguration abschließen* 215
 - Monitor* 213
 - Touchscreen* 214
 - XFine2* 215
- sax2, Kommando 212
- sbin, Systemverzeichnis 151
- scalar, Perl-Anweisung 814
- Scanner 268
 - Auflicht* 268
 - Bildauflösung* 330
 - Durchlicht* 268
 - Flachbettscanner* 268
 - Konfiguration (YaST)* 269
 - Kooka, Software* 270
 - Module/Treiber* 270
 - Qualitätsfaktor* 330
 - SANE* 269
 - xsane, Software* 270
- Schärfen, GIMP 355
- Scheduler 949
- Schema (LDAP) 674
- Schemata (Softwarepakete) 84
- Scherwerkzeug, GIMP 348
- Schichtenmodell 576, 949
 - Alltagsbeispiel* 576
 - OSI-Referenzmodell* 576
 - TCP/IP* 576
- Schleifen
 - foreach, Perl* 817
 - Shell-Skripte* 803
- Schlüssel
 - eines Hashes* 818
 - im RDBMS* 526
- Schnittstellen
 - Hardware* 841
- Schreibschriften 283
- Schriftgröße
 - Dicke* 278
 - Didot-Punkt* 276
 - DTP-Punkt* 276
 - Kegelhöhe* 276
 - Mittellänge* 277
 - Oberlänge* 278
 - Pica-Point* 276
 - Punkt* 276
 - Untertlänge* 278
 - Versalhöhe* 277
- Schusterjunge, Absatzfehler 289
- SCO Group, Kampf gegen Linux 45
- scp, Kommando 687
- Screenshot
 - GIMP* 338
- ScriptAlias, Apache-Direktive 736
- Scrollen, Vim 408
- SCSI 858, 949
 - anschließen* 858
 - ID* 858
 - Terminator* 858
- SCSI-ID 858
- SDSL 608, 949
- sed 810, 827
- Segmentierung (Speicher) 875
- Seiten (Speicher) 875
- Seitenfehler (Speicher) 875
- Seitenformate, Writer 301
- Seitentabelle (Speicher) 875
- SELECT, MySQL-Anweisung 549
- sendmail, Kommando 779
- Serial ATA → S-ATA
- Serielle Datenübertragung 851
 - Bedeutung* 853
 - Kontrollbit* 852
 - Leitungskonventionen* 852
 - Prüfbit* 852
 - Startbit* 852
 - Stoppbit* 852
- Serifenbetonte Linear-Antiqua 282
- Serifenlose Linear-Antiqua 282
- ServerAdmin, Apache-Direktive 736
- ServerName, Apache-Direktive 737
- ServerRoot, Apache-Direktive 737
- ServerSignature, Apache-Direktive 737
- ServerTokens, Apache-Direktive 737
- Session-Hijacking 639
- Session-Tracking (PHP) 750
- SET PASSWORD, MySQL-Anweisung 561
- SET, MySQL-Datentyp 549
- Setgid-Bit 192
- Setuid-Bit 191

- SGI 72
- SGML 949
 - HTML-DTD 472
 - XML als moderne Version 440
- sh 159
- Shannon-Theorem 949
- Share Level Security, Samba 663
- Shebang 109
 - Perl 813
 - Shell-Skripte 791
- Shell 149, 949
 - /etc/profile, Konfigurationsdatei 158
 - alias 158
 - als Kommandointerpreter 159
 - Anfang einer Textdatei anzeigen 195
 - Arbeitsverzeichnis 185
 - Arbeitsverzeichnis ändern 185
 - Arbeitsverzeichnis ermitteln 185
 - Ausgabeumleitung 166
 - Backticks 176
 - bash 159
 - Befehlseingabe 160
 - Befehlsverknüpfungen 174
 - Benutzer verwalten 187
 - csh 159
 - Dateieigentümer ändern 190
 - Dateien finden 187
 - Dateien kopieren 179
 - Dateien löschen 181
 - Dateien umbenennen 181
 - Dateien vergleichen 198
 - Dateien verschieben 181
 - Dateigruppe ändern 190
 - Dateimuster 178
 - Datum und Uhrzeit 204
 - Ein-/Ausgabeumleitung 166
 - Eingabeumleitung 167
 - Eingabevervollständigung 163
 - Ende einer Textdatei anzeigen 195
 - Ermitteln der aktuellen 159
 - Escape-Sequenz 197
 - Grundbegriffe 150
 - Grundfunktionen 160
 - Gruppen verwalten 188
 - Hintergrundprozesse 162
 - History 164
 - Kopieren mit der Maus 165
 - ksh 159
 - Links erzeugen 182
 - Motivation zum Einsatz 149
 - Pager 194
 - Passwörter ändern 189
 - Pipes 172
 - Prozessverwaltung 201
 - reguläre Ausdrücke suchen 196
 - root werden 188
 - sash 159
 - sh 159
 - Systemdetails ermitteln 200
 - Systeminformationen 200
 - Tastenkürzel 164
 - Text ausgeben 193
 - Text seitenweise anzeigen 194
 - Textdateien ausgeben 194
 - Textinhalte sortieren 199
 - Textstatistik 199
 - Umgebung 158
 - Umgebungsvariablen 158
 - Verhalten der Kommandos 176
 - verschiedene im Vergleich 159
 - Verzeichnisinhalt anzeigen 183
 - Verzeichnisse erstellen 186
 - Virtuelles Terminal 156
 - Zugriffsrechte ändern 190
- Shell-Built-ins
 - alias 783
 - case 801
 - export 794
 - for 803
 - if 798
 - typeset 796
 - unalias 785
 - while 807
- Shell-Skript
 - ausführen 111
 - Shebang 109
- Shell-Skripte 790
 - awk 811
 - case 801
 - Einführungsbeispiel 791
 - Fallentscheidungen 798
 - for 803
 - Funktionen 808
 - if 798
 - Kontrollstrukturen 798
 - Schleifen 803

- sed* 810
- Variablen* 794
- while* 807
- shift, Perl-Anweisung 816
- Short-Circuit-Logik 822
- SHOW CHARACTER SET, MySQL-Anweisung 539
- SHOW COLLATION, MySQL-Anweisung 540
- SHOW CREATE TABLE, MySQL-Anweisung 543
- SHOW DATABASES, MySQL-Anweisung 538
- SHOW TABLES, MySQL-Anweisung 538
- shred, Kommando 927
- shutdown, Kommando 203, 928
- Shuttleworth, Mark 39
- Sicherheit
 - Admin-Rechte* 641
 - Backups* 641
 - CGI u. Webanwendungen* 639
 - chroot-Umgebung* 642
 - Crackerangriffe* 638
 - Crackertools* 642
 - Cross-Site-Scripting (XSS)* 639
 - DoS-Attacken* 638
 - Exploits* 637
 - Firewall* 641, 646
 - Intrusion Detection Systems* 642
 - Kryptografie* 643
 - Man-in-the-Middle-Angriff* 639
 - Menschliches Versagen* 642
 - netfilter/iptables* 646
 - Netzwerke* 637
 - Patches installieren* 640
 - Rootkits* 638
 - Schutzmaßnahmen* 640
 - Session-Hijacking* 639
 - Social Engineering* 642
 - SQL-Injection* 639
- SIGALRM, Signal 868
- SIGCHLD, Signal 868
- SIGHUP, Signal 868
- SIGKILL
 - Signal* 868
- Signal 949
 - SIGALRM* 868
 - SIGCHLD* 868
 - SIGHUP* 868
- SIGKILL* 868
- SIGTERM* 868
- Verarbeitung durch Prozesse* 868
- zur Inter-Prozess-Kommunikation* 869
- Signale (IPC) 202
- Signatur, digitale 645
- SIGTERM, Signal 868
- Silbentrennung
 - für Blocksatz* 287
- Single-User-Modus 372
- Skalare, Perl-Variablen 814, 816
- Skalieren, GIMP-Werkzeug 348
- Slackware, Linux-Distribution 39
- slappadd, Kommando 677
- SMALLINT, MySQL-Datentyp 546
- smbclient, Samba-Dienst 663
- smbd 661
- smbfs, Dateisystem 663
- SMTP 773
 - Befehle* 774
 - Sitzung* 774
 - Zugriffsbeschränkung* 773
- SOA-Record (DNS) 628
- Social Engineering 642
- sockaddr_in, Netzwerkprogrammierung 879
- Socket 949
- Socket, Perl-Modul 877
- Sockets 876
 - accept(), Funktion* 882
 - Adressen* 878
 - bind(), Funktion* 882
 - connect(), Funktion* 881
 - Datagramme senden und empfangen* 880
 - Domain* 877
 - erzeugen* 877
 - IO* 882
 - IP-Adresse* 878
 - lauschende* 882
 - listen(), Funktion* 882
 - Protokoll* 878
 - TCP* 880
 - TCP-Client* 881
 - TCP-Port* 878
 - TCP-Server* 882
 - Typ* 878
 - UDP* 880
 - Verbindung aufnehmen* 882
- SoftRAID 384

- Software
 - Angebot unter Linux* 40
 - installieren* 136
 - installieren (RPM)* 137
 - mit YaST installieren* 136
 - Verteilung auf Verzeichnisse* 152
- Softwareauswahl
 - detaillierte Optionen* 91
 - Installationsquellen* 90
 - Paketabhängigkeiten* 92
 - Paketgruppen* 89
 - Schemata* 84
 - Sprachen* 89
 - Suchfunktion* 90
- Softwareauswahl (Installation) 84
- Softwarepatente, Bedrohung durch 46
- Solaris, Betriebssystem 28
- Solid State Disk → SSD
- Sonderzeichen
 - in Dateinamen* 153
- sort, Kommando 199, 929
- Sound
 - amaroK, Player* 238
 - Banshee, GNOME-Player* 247
- Soundkarte 266
 - ALSA* 266
 - Anschlüsse* 267
 - Audio-CDs abspielen* 267
 - Konfiguration (YaST)* 268
 - MIDI* 267
 - SP-DIF-Anschlüsse* 267
- SP-DIF-Anschlüsse 267
- Speicher
 - Management durch Betriebssystem* 24
 - RAM* 840
 - ROM* 840
 - virtueller* 843
- Speichermanagement 24
- Speicherseiten 875
- Speicherverwaltung 875
 - x86-Systeme* 876
- Spektrum 324
- Spiegeln, GIMP 348
- splendid, Typografie 278
- split, Perl-Anweisung 817
- Splitter (DSL) 609
- Sprachen
 - einstellen (Installation)* 92
- Sprachen (Softwareauswahl) 89
- Sprühpistole, GIMP-Werkzeug 354
- Sprungbefehle
 - bedingte* 847
 - der CPU* 847
 - unbedingte* 847
- Sprungmarken, Vim 411
- Sprungvorhersage 843
- Sprungvorhersage → Prozessor
- SQL 538, 949
- SQL-Injection 639
- squid, Proxyserver 682
 - /etc/squid/squid.conf, Konfigurationsdatei* 682
- SRAM 850
- SSD 371
- SSH 646, 685, 950
 - Dateien kopieren (scp)* 687
- ssh, Kommando 686, 928
- sshd 685
- SSL 646
- SSL (Apache 2) 715
- Stack 950
 - der CPU* 842, 847
- Stack-Overflow 950
- Stack-Zeiger 842, 848
- Stallman, Richard M. 26, 29, 31, 425
- Stand-Alone Shell (sash) 159
- Standardausgabe 166
- Standardeingabe 166
- Standardfehlerausgabe 166
- Standardgateway 600
- Standard-Runlevel, Installationsoption 94
- Stapelverarbeitung 26
- Star Division 273
- Startbit 852
- startx, Kommando 216
- Statuscodes (HTTP) 696
 - 100 Continue* 697
 - 101 Switching Protocols* 697
 - 1xx (Informationen)* 696
 - 200 OK* 697
 - 201 Created* 698
 - 2xx (Erfolg)* 697
 - 301 Moved Permanently* 698
 - 302 Found* 698
 - 303 See Other* 698
 - 304 Not Modified* 699
 - 307 Temporary Redirect* 698
 - 3xx (Umleitung)* 698

- 401 *Unauthorized* 700
- 403 *Forbidden* 700
- 404 *Not Found* 700
- 4xx (*Client-Fehler*) 699
- 500 *Internal Server Error* 701
- 5xx (*Server-Fehler*) 701
- Typen* 696
- stderr umleiten 167
- stderr, I/O-Kanal 166
- stdin, I/O-Kanal 166
- stdout umleiten 166
- stdout, I/O-Kanal 166
- Steinbild, Burchard 47
- Stempel, GIMP-Werkzeug 354
- Sternwerkzeug, Inkscape 361
- Steuerbus 842
 - Wortbreite* 844
- Steuerwerk der CPU 842
- Sticky-Bit 192
- Stift, GIMP-Werkzeug 353
- Stoppbit 852
- Stream-Sockets 878
- strict, Perl 815
- String 950
- String-Literale, Perl 819
- Strings
 - aus Arrays, Perl* 817
 - vergleichen, Perl* 820
 - Verkettung, Perl* 821
 - zerlegen, Perl* 817
- String-Verkettung
 - Perl* 821
- Stripe Set (RAID) 383
- Stripe Set mit Parity (RAID) 383
- su, Kommando 124, 188, 929
- sub, Perl-Subroutine 824
- Subnet Mask 581, 950
- Subnetting 950
- Subnetzmaske 99
- Subroutinen
 - Parameter* 824
 - Wertrückgabe* 825
- Subroutinen, Perl 824
- Suchbefehl
 - Vim, Texteditor* 409
- Suchen/Ersetzen
 - Writer* 305
- Suchmaschinen
 - Anmeldung bei* 499
 - HTML aufbereiten für robots.txt-Datei* 497
- Suchpfad 161
- sudo, Kommando 789
- sudoers 790
- SUM(), MySQL-Funktion 554
- SUMME(), Calc-Funktion 316
- Sun Microsystems 273
 - Solaris, Betriebssystem* 28
- Supernetting 950
- Superuser 869
- SUSE
 - Distributions-Geschichte* 47
 - Firmengeschichte* 47
 - openSUSE-Projekt* 48
 - Übernahme durch Novell* 48
- SUSE Linux 48
- SUSE Linux Enterprise Server (SLES) 48
- SUSE, Linux-Distribution → openSUSE
- SVG (Scalable Vector Graphics) 363, 440, 950, 951
- swap (0x82), Dateisystem-ID 71
- Swap-Partition 73, 875, 950
 - Dateisystem-ID für* 71
- Swapping (Speicher) 843
- Switch 594
- Symbolischer Link 154
 - erzeugen (Shell)* 182
- Symmetrische Verschlüsselung 644
- SyncPPP (ISDN) 619
- Syntaxhervorhebung
 - Vim, Texteditor* 419
- sys/types.h, C-Bibliothek 871
- Syslog 788, 950
 - facility* 788
 - logger, Kommando* 789
 - priority* 789
- syslogd 788
- System V 950
 - Init* 143
 - IPC* 869, 950
 - UNIX* 27
- system(), C-Funktion 872
- Systemaufrufe 849, 863, 865, 950
 - exec()* 867
 - fork()* 867
 - kill()* 868
 - pause()* 868
 - waitpid()* 867

Systemprogramme 864
 Systemstart 103
 Einstellungen 94
 im Textmodus 218
 Installationseinstellung 94
 mit grafischer Oberfläche 217
 Systemvariable 950
 Systemverzeichnisse
 bin 150
 dev 151
 etc 151
 home 151
 opt 151
 root 152
 sbin 151
 usr 151
 var 151

T

Tabellen
 in HTML 489
 Tabellen erstellen, Calc 313
 Tabellen, iptables 647
 Tabellen, Writer 307
 Tabellenkalkulation 312
 Gnumeric 247
 KSpread 237
 Tags
 XML 441
 tail, Kommando 195, 930
 Taktfrequenz
 der CPU 844
 des Mainboards 844
 praktische Bedeutung 845
 Tanenbaum, Andrew S. 32
 tar, Kommando 141, 393, 930
 Task Scheduler 863, 950
 Taskwechsel
 in KDE 107
 Tastaturbelegung einstellen 93
 Tastaturbelegung, Installationsoption 93
 Tastenkürzel, Shell 164
 TCP 950
 Paket-Header 588
 TCP Urgent Data 591
 TCP/IP 950
 Anwendungsschicht 577
 ARP 593
 chargen 766
 daytime 764
 Diagnose und Fehlersuche 630
 dig 633
 discard 767
 echo 766
 Host-zu-Host-Transportschicht 577
 HTTP 691
 ifconfig, Kommando 595
 IMAP, Anwendungsprotokoll 779
 inetd 762
 Internetschicht 577
 IP-Broadcasting 580
 IP-Protokoll 578
 iptables 646
 IPv4 579
 IPv4-Datagramme 582
 IPv6 584
 IPv6-Datagramme 585
 IPv6-Tunnelung 585
 Kommunikationsverfahren 578
 mehrere IP-Adressen 598
 MTU 598
 NAT (Network Address Translation) 580
 netfilter 646
 netstat 632
 Netzzugangsschicht 577
 nslookup 633
 ping 630
 POP3, Anwendungsprotokoll 777
 Routing 586
 Schichtenmodell 576
 SMTP, Anwendungsprotokoll 773
 TCP 588
 tcpdump 635
 telnet 634
 time 766
 traceroute 631
 Transportprotokolle 588
 Transportschicht 577
 TTL 587
 UDP 591
 xinetd 763
 TCP-Client-Socket 881
 mit IO 884
 tcpd 763
 tcpdump, Kommando 635
 TCP-Header 588
 TCP-Port
 für Sockets 878

- TCP-Protokoll → Transmission Control Protocol
- TCP-Server-Socket 882
 - mit IO* 884
- tee, Kommando 931
- Teilnetzmaske 581
- Telefonleitung
 - Pulswahl* 614
- Telnet 950
- telnet, Kommando 634, 931
- TERM, Signal 868
- Terminal 26, 950
 - GNOME Terminal* 117
- Terminalemulation
 - GNOME Terminal* 246
 - KDE-Konsole* 235
 - mlterm* 249
 - rxvt* 248
- Terminalfenster 157
- Terminator
 - SCSI* 858
- test, Kommando 798
- TeX, Textsatzsystem 511
- TEXT, MySQL-Datentyp 548
- text-align, CSS-Angabe 505
- Textdateiformate
 - Vorteile* 439
- text-decoration, CSS-Angabe 504
- Texteditor 405
 - ex* 405
 - für XML verwenden* 440
- text-indent, CSS-Angabe 505
- Textmodus (System ohne grafische Oberfläche) 65
- Textmodus, System starten im 218
- Textverarbeitung
 - AbiWord* 247
 - KWord* 237
- Textwerkzeug, GIMP 348
- Textwerkzeug, Inkscape 362
- TFT 950
- TFT-Monitor 211
- TFTP 767
- Thompson, Ken 27
- Thread 862, 874, 950
- TIFF, Bilddateiformat 333
- time, Kommando 932
- TIME, MySQL-Datentyp 548
- time, Serverdienst 766
- Timesharing 26, 951
- TIMESTAMP, MySQL-Datentyp 548
- Tinte, GIMP-Werkzeug 354
- Tintenstrahldrucker 259
 - Bubble-Technik* 260
 - Piezo-Technik* 260
- TINYINT, MySQL-Datentyp 546
- Titel, HTML-Dokument 473
- Tonwahlverfahren 614
- top, CSS-Angabe 508
- top, Kommando 202
- Top-Level-Domain 951
- Torvalds, Linus 31, 32
- Touchscreen 214
- tr//, Perl-Operator 832
- traceroute, Kommando 631
- Track-at-once 387, 951
- Transaktion 951
- Transistor
 - TFT* 211
- Transmission Control Protocol (TCP) 588
 - Ephemeral Ports* 590
 - Ports* 590
 - Urgent Data* 591
 - Well-Known Ports* 590
- Transportprotokolle, TCP/IP 588
- Transportschicht → Host-zu-Host-Transportschicht
- Treiber 24
- Trolltech 219
- TrueType 951
- TTL (Time To Live) 587
- Tunnelung, IPv6 585
- Tux (Linux-Maskottchen) 34
- Twisted-Pair-Kabel 593
- type, Kommando 161
- typeset, Shell-Built-in 796
- Typografie
 - Absatz* 286
 - Antiqua* 282
 - Antiqua-Varianten* 282
 - Barock-Antiqua* 282
 - Buchstabenentwicklung* 275
 - Dicke* 278
 - Durchschuss* 278
 - Expert-Zeichensatz* 280
 - Festbreitenschrift* 279
 - Fraktur* 283
 - Französische Renaissance-Antiqua* 282

gebrochene Schriften 283
gemeine 279
Gestaltung 289
Geviert 278
Großbuchstaben 278
Grundlinie 277
Handschriftliche Antiqua 283
Hurenkind 289
Kapitälchen 279
Kapitale 278
Kegelhöhe 276
Kerning 278
Klassizistische Antiqua 282
Kleinbuchstaben 279
kompress 278
kursiv 279
Laufweite 278
Ligaturen 279
Majuskeln 278
Majuskelschiffchen 280
Mediävalschrift 280
Mikrotypografie 284
Minuskeln 279
Minuskelschiffchen 280
Mittellänge 277
Oberlänge 278
Proportionalschrift 279
Satzspiegel 285
Satzzeichen 284
Schreibschriften 283
Schriftfamilien 280
Schriftwahl 283
Schusterjunge 289
Serifenbetonte Linear-Antiqua 282
Serifenlose Linear-Antiqua 282
splendid 278
Unterbildung 278
Unterschneidung 278
Venezianische Renaissance-Antiqua 282
Versalhöhe 277
Versalien 278
Zahlen 285
Zeile 285
Zeilenabstand 278
Zeilenfall 288
Ziffern 280

U

Ubuntu, Linux-Distribution 39
 UDF 388
 UDP 951
 UDP-Protokoll → User Datagram Protocol (UDP)
 UID (User-ID) 128
 von Prozessen 868
 umask 192
 umask, Kommando 192
 Umbenennen (Dateien) 181
 Umfließen, Elemente durch Text 292
 Umgebung 158
 Umgebungsvariablen 158
 CDR_DEVICE 389
 ftp_proxy 684
 http_proxy 684
 IFS 806
 MAILTO 787
 PATH 161
 umount, Kommando 380, 932
 unalias, Shell-built-in 785
 uname, Kommando 200, 932
 Unbedingter Sprung 847
 UND(), Calc-Funktion 321
 Unicode 951
 uniq, Kommando 199, 933
 University of California, Berkeley 27, 877
 UNIX 27
 Berkeley System Distribution (BSD) 27
 BSD 27
 HP UX 28
 IBM AIX 28
 IP-Adresse zuweisen 596
 Linux 28, 37
 Mac OS X 28
 Modularität 27
 POSIX-Standard 28
 Sun Solaris 28
 System V 27
 unix2dos, Kommando 933
 unless, Perl-Fallentscheidung 823
 unshift, Perl-Anweisung 816
 Unterlänge, Typografie 278
 Unterprogramm
 Aufruf durch CPU 847
 Unterschneidung 951
 Unterschneidung, Typografie 278

unzip, Kommando 396, 937
 Update
 des Systems 64
 UPDATE, MySQL-Anweisung 555
 uptime, Kommando 200, 934
 Urgent Data, TCP 591
 URL 951
 USB 859, 951
 use, Perl-Direktive 815
 User Datagram Protocol (UDP) 591
 User Level Security, Samba 663
 user, Partitionsmodus 75
 useradd, Kommando 188, 934
 userdel, Kommando 935
 User-ID 128, 190
 Effektive User-ID (EUID) 191
 per Setuid ändern 191
 von Prozessen 868
 usr, Systemverzeichnis 151
 UTC
 im Vergleich zur lokalen Zeit 65
 UTF-8 951

V

V.24-Schnittstelle 860
 var, Systemverzeichnis 151
 VARCHAR, MySQL-Datentyp 547
 Variablen
 Arrays, Perl 816
 deklarieren, Perl 815
 Existenz testen, Perl 823
 Hashes, Perl 816
 Perl 815
 Shell-Skripte 794
 Skalare, Perl 814, 816
 Substitution, Perl 814
 Variablensubstitution
 in Perl-HIER-Dokumenten 822
 Perl 814
 Vektorgrafik 323, 951
 GIMP 344
 Inkscape 359
 OpenOffice.org 308
 Venema, Wietse 779
 Venezianische Renaissance-Antiqua 282
 Verlaufswerkzeug, GIMP 352
 Versalhöhe, Schriftgröße 277
 Verschieben (Dateien) 181
 Verschiebewerkzeug, GIMP 345
 Verschlüsselung
 Blowfish 134
 Dateisystem 73
 DES 133
 MD5 134
 Passwörter 133
 Verschmieren, GIMP-Werkzeug 355
 Versteckte Dateien 154
 vertical-align, CSS 505
 VERWEIS(), Calc-Funktion 321
 Verzeichnisbaum 150
 klassische vs. moderne Anordnung 152
 Pfad 153
 Verzeichnisdienst 951
 Verzeichnisdienste 666
 NIS 666
 OpenLDAP 673
 Verzeichnisse
 Arbeitsverzeichnis 185
 erstellen (Shell) 186
 Inhalt anzeigen (Shell) 183
 Verzeichnisse verarbeiten, Perl 826
 Verzeichnisse, wichtige
 /etc/init.d 144
 /etc/init.d/rcN.d 143
 vi, als Verweis auf Vim 406
 vi, Texteditor 405
 Vim, Texteditor 406
 Automatisches Einrücken 419
 Befehlsmodus 406
 Befehlsübersicht 420
 Bereiche löschen 412
 Dateiverwaltung 416
 Einfügen 413
 Eingabemodus 406
 Ersetzen 410
 Hilfe 419
 Klammern, korrespondierende 409
 Löschen 412
 Mehrfenstermodus 417
 Modi 406
 Navigation 407
 Rückgängig machen 415
 Scrollen 408
 Sprungmarken 411
 Suchen 409
 Suchoptionen 410
 Syntaxhervorhebung 419

- Textobjekte* 413
 - Visueller Modus* 414
 - Weitersuchen* 410
 - Wiederholen* 415
 - Zähler* 408
 - Zeichensuche* 408
 - Zusatzfunktionen* 419
 - Viren
 - Antivirenprogramme* 640
 - Schutzmaßnahmen* 640
 - Virtueller Host 737
 - Virtueller Host (Apache 2) 713
 - namensbasierter vs. IP-basierter* 714
 - Virtueller Speicher 843, 875
 - Virtueller Speicher, Partition für 73
 - Virtuelles Terminal 156
 - visudo, Kommando 790
 - VLSM 951
 - VLSM (Variable Length Subnet Mask) 582
 - VNC 687
 - bei der Installation konfigurieren* 100
 - Volkerding, Patrick 39
 - VPN 646
 - vsftpd, FTP-Server 771
 - Anonymous FTP* 772
 - chroot* 772
- ## W
-
- w, Perl-Modifikator 813
 - W3C 951
 - Wahlverbindungen 606
 - waitpid(), Systemaufruf 867
 - Wall, Larry 812
 - WAN 951
 - Warnungen
 - in Perl aktivieren* 813
 - Warteschlange (Drucker) 263
 - Wavetable-Synthese (MIDI) 267
 - wc, Kommando 199, 935
 - wdm, Display-Manager 218
 - Web Developer, Firefox-Addon 367
 - Webanwendungen
 - Sicherheitsprobleme* 639
 - Webbrowser
 - Firefox* 364
 - Webformulare
 - mit PHP auslesen* 747
 - Webproxy (squid) 682
 - Webseiten
 - Quanta Plus, Editor* 510
 - Webserver 706, 951
 - Websites
 - robots.txt-Datei* 499
 - Wechseldatenträger
 - DVD* 387
 - Weichzeichnen, GIMP 355
 - Weiß, Mischung aller Lichtfarben 324
 - Weizenbaum, Joseph 434
 - Well-Known Ports
 - Definition* 590
 - Tabelle einiger wichtiger* 590
 - WENN(), Calc-Funktion 320
 - Werkzeugpalette (GIMP) 341
 - Wertrückgabe
 - Perl-Subroutinen* 825
 - whatis, Kommando 120, 936
 - WHERE, MySQL-Klausel 549
 - whereis, Kommando 936
 - which, Kommando 936
 - while, Shell-Built-in 807
 - White Book 952
 - White Book (Video-CD) 386
 - Whitespace
 - in RegExp* 829
 - who, Kommando 200, 936
 - whoami, Kommando 176, 936
 - Widenius, Michael 530
 - WindowMaker, Window-Manager 248
 - Window-Manager 952
 - Composite* 221
 - FVWM* 247
 - GNOME, Desktop* 239
 - IceWM* 249
 - KDE, Desktop* 220
 - WindowMaker* 248
 - Windows
 - Auslagerungsdatei verschieben* 56
 - Festplatte defragmentieren* 57
 - md5sum* 61
 - Partitionen verwalten* 55
 - Vorbehandlung für parallele Linux-Installation* 54
 - Windows-Serverdienste 661
 - Windows-Treiber verwenden 606
 - wine, Windows-Emulation 42
 - Winkelgeschwindigkeit, konstante 370
 - Winkelmesser, GIMP-Werkzeug 345

- Wireshark, Packet-Sniffer 636
 - WLAN 602, 952
 - Access Point* 604
 - Allgemeines* 602
 - Basic Service Set (BSS)* 603
 - CSMA/CA-Verfahren* 603
 - Extended Service Set (ESS)* 604
 - Frequenzbereiche* 602
 - Hardware* 603
 - iwconfig, Kommando* 606
 - iwlist, Kommando* 606
 - Konfiguration (YaST)* 605
 - Sicherheit* 605
 - Übertragungstechnik* 602
 - wodim, Kommando 389, 937
 - Wohlgeformtheit 952
 - Wohlgeformtheit, XML-Dokumente 448
 - World Wide Web Consortium → W3C
 - Wortbreite 952
 - der CPU* 844
 - Writer 274
 - Absatzformate* 299
 - Format, Symbolleiste* 294
 - Formatvorlagen* 303
 - reguläre Ausdrücke* 306
 - Seitenformate* 301
 - Suchen/Ersetzen* 305
 - Tabellen* 307
 - Zeichenformate* 297
 - Zeichenfunktionen* 308
 - Würmer
 - Schutzmaßnahmen* 640
 - WYSIWYG 952
- X**
-
- X Window 952
 - X.org Foundation 208
 - X11R6 208
 - X11R7 208
 - x86-Assembler
 - Beispiele* 848
 - Xalan, XSLT-Prozessor 467
 - xconfig, Kernel-Konfigurator 255
 - XDM, Display-Manager 217
 - XEmacs 426
 - XFine2, X-Feintuning 215
 - XFree86 208
 - XFS, Dateisystem 72, 376
 - Xgl, X-Server 208
 - XHTML 472, 952
 - xinetd 763
 - XML 439, 952
 - Attributdeklaration (DTD)* 457
 - Attribute* 441, 444
 - Attribute, falsche Verwendung* 445
 - Attribute, Verwendung* 445
 - Beispieldokument* 441
 - CDATA-Blöcke* 447
 - CSS verwenden* 467
 - Dateiendung* 441
 - DocBook* 440
 - DOCTYPE-Angabe* 451
 - Dokumenteingabe* 440
 - Dokument-Fragmente* 449
 - Dokumentstruktur* 441
 - DTD* 450
 - DTD definieren* 451
 - DTD, Attribute deklarieren* 457
 - DTD, Elemente deklarieren* 453
 - DTD, Entities deklarieren* 461
 - Elementdeklaration (DTD)* 453
 - Elemente (Tags)* 443
 - Entities deklarieren (DTD)* 461
 - Entity-Referenzen* 446, 461
 - Entity-Referenzen vermeiden* 447
 - Entity-Referenzen, numerisch* 446
 - Hierarchie* 443
 - leere Tags* 444
 - leere Tags, Kurzfassung* 444
 - mehrere Namensräume* 463
 - MIME-Types* 441
 - Namensräume* 462
 - Namensräume, mehrere im Dokument* 463
 - PCDATA* 447
 - Processing Instructions (PI)* 443
 - PUBLIC-ID* 451
 - Quanta Plus, Editor* 510
 - Schema* 464
 - Schema-Beispiel* 464
 - SGML-Erbe* 440
 - Standalone-Dokument* 443
 - Standard-Namensraum* 462
 - Steueranweisungen* 443
 - SVG* 363, 440
 - SYSTEM-ID* 451
 - Tag-Namen* 443
 - Tags* 441

- Tag-Verschachtelung 443
 - Universalität 440
 - verschachtelte Tags 443, 449
 - Verschachtelungsfehler 450
 - wichtige Dokumentformate 440
 - Wohlgeformtheit 448
 - Wurzelement 449
 - XHTML 440
 - xmlns-Angabe 463
 - xml-Steueranweisung 442
 - XSL-FO 467
 - XSLT 467
 - Zeichensätze 442
 - XML Schema 464, 952
 - Attribute deklarieren 465
 - Elemente deklarieren 465
 - verschachtelte Elemente 465
 - XPath
 - Überblick 470
 - XSane
 - aus GIMP aufrufen 339
 - xsane, Scan-Programm 270
 - X-Server 207
 - abbrechen 215
 - automatisch starten 218
 - Entwicklungsgeschichte 208
 - Funktionsbeschreibung 207
 - Konfiguration 212
 - Konfiguration abschließen 215
 - SaX, Konfigurations-Tool 212
 - startx 216
 - Systemstart ohne 218
 - X11R6 208
 - X11R7 208
 - XFree86 208
 - Xgl 208
 - XSL Formatting Objects 467
 - XSL-FO 467, 952
 - XSLT 467, 952
 - Parameter 469
 - Prozessor 467
 - wichtige Elemente 469
 - Wurzelement 469
 - Xalan, Prozessor 467
 - XPath-Ausdrücke 470
 - xsl:call-template, Tag 469
 - xsl:copy-of, Tag 470
 - xsl:for-each, Tag 469
 - xsl:if, Tag 470
 - xsl:param, Tag 469
 - xsl:text, Tag 469
 - xsl:value-of, Tag 469
 - xsl:with-param, Tag 469
 - XSLT-Prozessor 467
- ## Y
-
- YaST
 - /etc/sysconfig-Editor 217
 - als Installationsprogramm 63
 - als Konfigurationsprogramm 123
 - Apache 2 konfigurieren 711
 - Apache-Installation 708
 - Benutzer verwalten 127, 131
 - Benutzerdetails 132
 - Boot-Konfiguration 398
 - Cyrus IMAP-Konfiguration 781
 - DHCP-Client einrichten 597
 - DHCP-Server konfigurieren 679
 - Drucker konfigurieren 262
 - DSL-Konfiguration 609
 - Ethernet konfigurieren 596
 - fetchmail-Konfiguration 781
 - Gruppen verwalten 134
 - Hardware einrichten 265
 - Hauptmenü 126
 - Hostnamen konfigurieren 599
 - HTTP-Server 711
 - im Textmodus starten 124
 - in GNOME starten 124
 - in KDE starten 124
 - Internet Service Provider einrichten 610
 - ISDN einrichten 618
 - LDAP-Client konfigurieren 678
 - Mail-Konfiguration 779
 - Modem konfigurieren 614
 - Netzwerkkarte einrichten 596
 - NFS-Konfiguration 658
 - NIS-Client-Konfiguration 671
 - NIS-Server-Konfiguration 667
 - Online-Updates 137
 - openSUSE-Firewall konfigurieren 650
 - Passwortoptionen 133
 - PHP-Installation 709
 - Postfix-Konfiguration 779
 - Procmail-Konfiguration 781
 - Programme automatisch starten 145
 - Routing-Konfiguration 600

Runlevel-Editor 145
Samba-Konfiguration 664
Scanner einrichten 269
Software installieren 136
Soundkarte einrichten 268
WLAN-Konfiguration 605
 Yellow Book (CD-ROM) 386

Z

Zahlen

Typografie 285
 Zähler, Emacs 430
 Zähler, Vim 408
 Zauberstab, GIMP-Werkzeug 343

Zeichen

alternative in RegExp 827
ausschließen in RegExp 827
beliebig viele, in RegExp 828
eines oder mehr, in RegExp 828
genaue Anzahl, in RegExp 828
Gruppen in RegExp 827
optionale, in RegExp 827
Wortbestandteile in RegExp 829

Zeichenformate, Writer 297

Zeichenfunktionen, Writer 308

Zeichengeräte (char devices) 864

Zeichensatz

in HTML angeben 474
in MySQL 539

Zeichensuche, Vim 408

Zeile, Typografie 285

Zeilenabstand 278

optischer 278

Zeilenfall 288

Zeilenfrequenz (Monitor) 211

Zeilenumbrüche

entfernen, Perl 814

HTML 474

Zeitzone

einstellen (Installation) 64

Zellbezüge, Calc 318

Zellenformate, Calc 315

Zeroconf, Netzwerkkonfiguration über 99

Ziel, iptables 647

Ziffern

in RegExp 829

Majuskelziffern 280

Mediävalziffern 280

Minuskelziffern 280

Typografie 280

z-index, CSS-Angabe 508

zip, Kommando 396, 937

Zombie 867

Zugriffskontrollliste → Access Control List

Zugriffsrechte 110, 155

ändern (Shell) 190

numerische Darstellung 156

rwX-String-Darstellung 155

Setgid-Bit 192

Setuid-Bit 191

setzen (GNOME) 115

setzen (KDE) 110

Sticky-Bit 192

umask 192

Zuschneiden, GIMP-Werkzeug 346

Zwischenablage

GIMP, Inhalte einfügen 338

zypper, Kommando 140