

X . s y s t e m s . p r e s s

X.systems.press ist eine praxisorientierte
Reihe zur Entwicklung und Administration von
Betriebssystemen, Netzwerken und Datenbanken.

Wulf Alex studierte Elektrotechnik mit der Fachrichtung Nachrichtentechnik – unter anderem bei KARL STEINBUCH – an der TH Karlsruhe und promovierte dort an der Fakultät für Chemieingenieurwesen bei HANS RUMPF.

Nach einem Forschungsaufenthalt an der Universität Uppsala (Schweden) verantwortete er die gesamte EDV eines großen Hochschulinstitutes, anfangs mit einem zentralen UNIX-Rechner, zuletzt mit einem heterogenen Netz bestehend aus etwa zweihundert Maschinen unter UNIX, Linux, MacOS und MSWindows. Ferner übernahm er Lehraufträge zur Partikelmessstechnik, zu Linux/UNIX und zur Programmierung in C/C++. Er arbeitete an mehreren Büchern mit.

Programmieren hat Wulf Alex auf einer Zuse Z 22 gelernt, die er auch als Nachtoperator gefahren hat. Heute ist er, mit etwas mehr verfügbarer Zeit, vor allem Admin eines Domestic Area Networks und als Autor und Lehrbeauftragter weiterhin im Linux/UNIX-Umfeld aktiv – wenn er nicht gerade in seinen Bergstiefeln unterwegs ist, an einem Baggersee Wachdienst leistet oder auf seinem Motorrad Europa durchstreift.

Arne Alex lernte den Umgang mit Computern zu Hause bereits vor dem Grundschulalter, anfangs auf einem Schneider CPC 464. Nach Abitur und Zivildienst studiert er an der Universität Karlsruhe Physik, Mathematik und Informatik. Innerhalb seiner Familie vertritt er die BSD-Fraktion, ist aber auch Debian GNU/Linux nicht abgeneigt.

Der Gebrauch aller modernen elektronischen Hilfsmittel im Alltag und im Studium ist für ihn eine Selbstverständlichkeit. Den Ausgleich für die Stunden am Bildschirm findet er auf dem Sportplatz und in den Bergen.

Björn Alex lernte den Umgang mit Computern zu Hause bereits im Grundschulalter, anfangs auf einem Schneider CPC 464. Nach Abitur und Zivildienst studierte er an der Universität Karlsruhe Physik bis zum erfolgreich abgeschlossenen Vordiplom. Danach wandte er sich dem Studium der Medizin an der Universität Heidelberg zu. Innerhalb seiner Familie vertritt er die Macintosh-Fraktion.

Der Gebrauch aller modernen elektronischen Hilfsmittel im Alltag und im Studium ist für ihn ebenfalls eine Selbstverständlichkeit. Eine lange Reise von Karlsruhe bis Beijing hat ihn für fremde Kulturen sensibilisiert.

Wulf Alex

Debian GNU/Linux in der Praxis

Anwendungen, Konzepte, Werkzeuge

Unter Mitarbeit von Arne Alex und Björn Alex

Mit 133 Abbildungen

 Springer

Wulf Alex

Rieslingweg 14

76356 Weingarten

alex-weingarten@t-online.de

<http://www.alex-weingarten.de/debian/>

unter Mitarbeit von Arne Alex und Björn Alex

Bibliografische Information der Deutschen Bibliothek

Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen

Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über

<http://dnb.ddb.de> abrufbar.

ISSN 1614-5216

ISBN-10 3-540-23786-0 Springer Berlin Heidelberg New York

ISBN-13 978-3-540-23786-0 Springer Berlin Heidelberg New York

Dieses Werk ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere die der Übersetzung, des Nachdrucks, des Vortrags, der Entnahme von Abbildungen und Tabellen, der Funksendung, der Mikroverfilmung oder der Vervielfältigung auf anderen Wegen und der Speicherung in Datenverarbeitungsanlagen, bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten. Eine Vervielfältigung dieses Werkes oder von Teilen dieses Werkes ist auch im Einzelfall nur in den Grenzen der gesetzlichen Bestimmungen des Urheberrechtsgesetzes der Bundesrepublik Deutschland vom 9. September 1965 in der jeweils geltenden Fassung zulässig. Sie ist grundsätzlich vergütungspflichtig. Zuwiderhandlungen unterliegen den Strafbestimmungen des Urheberrechtsgesetzes.

Springer ist ein Unternehmen von Springer Science+Business Media

springer.de

© Springer-Verlag Berlin Heidelberg 2006

Printed in Germany

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften. Text und Abbildungen wurden mit größter Sorgfalt erarbeitet. Verlag und Autor können jedoch für eventuell verbliebene fehlerhafte Angaben und deren Folgen weder eine juristische Verantwortung noch irgendeine Haftung übernehmen.

Satz: Druckfertige Daten des Autors

Herstellung: LE-TEX, Jelonek, Schmidt & Vöckler GbR, Leipzig

Umschlaggestaltung: KünkelLopka Werbeagentur, Heidelberg

Gedruckt auf säurefreiem Papier 33/3142 YL – 5 4 3 2 1 0

Vorwort

Das Buch wendet sich an Benutzer von PCs unter Debian GNU/Linux, die auf einem fertig eingerichteten System mit der Arbeit beginnen möchten. Die Hobelbank stehe bereit, der Werkzeugschrank sei gut ausgestattet, es soll ans Schaffen gehen, wie man im Südwesten sagt. Dazu müssen sie wissen, welche Aufgaben mit welchen Anwendungen oder Werkzeugen zu erledigen und wie diese wirkungsvoll zu handhaben sind.

Der Gedanke zu diesem Buch reifte bei der Durchsicht des Manuskriptes zur zweiten Auflage des Debian-Buches von PETER H. GANTEN und WULF ALEX aus derselben Reihe des Springer-Verlages. Wir hatten den Eindruck, dass dort manche Themen fehlen oder zu kurz behandelt sind. Als Lösung bot sich eine Aufteilung der Themen an: Grundlagen, Einrichten und Betrieb bei GANTEN + ALEX, die Werkzeuge und Anwendungen hier. Anders ausgedrückt: Jenes Buch richtet sich an System- und Netzverwalter, dieses an Anwender. Geringe Überschneidungen sind unvermeidlich, da jeder Band für sich allein verwendbar sein soll.

Bedenkt man, dass es weit über tausend Linux/UNIX-Werkzeuge gibt und zu einzelnen Werkzeugen oder Internet-Diensten Bücher von 800 oder mehr Seiten, wird klar, dass umfassendere Arbeiten über Linux/UNIX immer Kompromisse aus Breite und Tiefe sind. Wir haben einerseits eine Auswahl aus der Werkzeugkiste treffen und andererseits auf manche Einzelheit verzichten müssen. Da wir sowohl Anwender wie Verwalter sind und darüber hinaus viele Anwender betreut haben, hegen wir die Hoffnung, einen praxistauglichen Kompromiss gefunden zu haben. *Alles über Debian GNU/Linux* ist kein Buch, sondern ein Bücherschrank.

Die Werkzeuge sind nach Aufgaben kapitelweise zusammengefasst. Innerhalb eines Kapitels ist die Vorgehensweise fast immer gleich: Wir erläutern die Aufgabe samt Grundbegriffen und Konzepten – das ist Wissen auf Dauer – und im Anschluss einige gebräuchliche Anwendungsprogramme. Je tiefer wir in die Einzelheiten dringen, desto kurzlebiger sind die Fakten, obwohl sich in den bald vierzig Lebensjahren von UNIX und fünfzehn von GNU/Linux viele Dinge eingependelt haben. Deshalb zeigen wir, wie man sich bei Bedarf weiter informiert, und verweisen vielerorts auf unsere Informationsquellen, allen voran das Web.

Bei über zehntausend Debian-Paketen und vermutlich noch einmal der gleichen Menge an Software außerhalb von Debian, aber unter der GNU GPL oder ähnlichen Bedingungen stehen für viele Aufgaben mehrere Werkzeuge zur Auswahl. Der geneigte Leser wird gebeten, sich auch selbst in den Paketlisten auf <http://www.debian.org/distrib/packages> umzusehen und verschiedene Werkzeuge auszuprobieren. Für die Mitteilung der dabei gewonnenen Erfahrungen danken wir im Voraus. Die Vielfalt ist Segen und Fluch zugleich: Wenn Sie in einer einheitlichen Rechnerwelt bleiben, haben Sie es einfacher, alles passt zusammen. Aber Sie haben kaum eine Chance, diese Welt zu verlassen. Die Linux/UNIX-Welt dagegen war, ist und bleibt offen, sowohl bezüglich der Hardware wie der Software. Trotz dieser Fülle an freier Software sind wir vereinzelt auf nicht-freie Produkte eingegangen. Das Bild von den Möglichkeiten unter Debian GNU/Linux wird vollständiger.

Das Buch behandelt *nicht* die Einrichtung und den Betrieb eines Rechners unter Debian GNU/Linux. Das sind Pflichten des Systemverwalters (Administrators, Supervisors, Superusers, Administrateur-système, root); sie sind in GANTEN + ALEX erläutert. Ebenso werden die Werkzeuge nicht behandelt, die nur von Programmierern gebraucht werden (Compiler, Debugger etc.). Das Verständnis elementarer Begriffe aus der Linux/UNIX-Welt wie Benutzer, Gruppe, Datei, Verzeichnis, Pfad, An- und Abmelden, Speicher usw. wird vorausgesetzt.

Der Text beruht auf Erfahrungen aus fünf Jahrzehnten Umgang mit elektronischen Rechenanlagen und aus Kursen über HP-UX und GNU/Linux für Auszubildende und Studierende. Wir haben auch fremde Hilfe beansprucht und danken Angehörigen der Universität Karlsruhe für Unterlagen, Anregungen und Diskussionen. Darüber hinaus haben wir fleißig das Internet angezapft und viele dort umlaufende Guides, Primers, Tutorials, HOWTOs und Sammlungen von Frequently Asked Questions ausgewertet. Ein besonderer Dank gebührt der deutschen Wikipedia und der Suchmaschine MetaGer.

Das Manuskript wurde mit dem Texteditor `vim` und teilweise dem Emacs auf einem PC der Marke *Weingartener Katzenberg Auslese* unter Debian GNU/Linux 3.1 (*sarge*) geschrieben, auf einem CVS-Server derselben Marke gespeichert und mit `LATEX 2ε` formatiert. Die meisten Abbildungen wurden mittels `xwd`, `xfig` und GIMP erstellt.

Dem Springer-Verlag in Heidelberg danken wir für die angenehme Zusammenarbeit, insbesondere den Herren HERMANN ENGESSER und DR. FRANK SCHMIDT, die sich für beide Debian-Bücher eingesetzt und zu ihrem Erfolg beigetragen haben.

So eine Arbeit wird eigentlich nie fertig, man muss sie für fertig erklären, wenn man nach Zeit und Umständen das Möglichste getan hat, um es mit JOHANN WOLFGANG VON GOETHE zu sagen (*Italienische Reise*, Caserta, den 16. März 1787, als er an der *Iphigenie* schrieb).

Weingarten (Baden), den 16. März 2005

Wulf Alex

Arne Alex

Björn Alex

Übersicht

1	Von der Hardware zur Anwendung	1
2	Kommandointerpreter	53
3	Dienstprogramme	71
4	Suchen und Finden	115
5	Textverarbeitung	155
6	Mathematische Werkzeuge	251
7	Datenbanken und andere Wissensspeicher	273
8	Das grafische Atelier	291
9	Das digitale Tonstudio	327
10	Video und Fernsehen	351
11	Kommunikation	363
12	Office-Pakete	405
13	Spiele und Spielereien	413
A	Zahlensysteme	429
B	Zeichensätze und Sondertasten	436
C	Metazeichen	454
D	Papier- und Schriftgrößen	457
E	Farben	459

VIII Übersicht

F	Datei-Merkmale	461
G	Prozess-Signale	467
H	Slang im Netz	469
I	Beispiele LaTeX	476
J	Modem-Kommandos (Hayes)	486
K	ISO 3166 Ländercodes	488
L	Requests For Comments (RFCs)	491
M	Zeittafel	497
N	Zum Weiterlesen	508
	Personenverzeichnis	519
	Sachverzeichnis	522

Inhalt

1	Von der Hardware zur Anwendung	1
1.1	Hardware und Software	1
1.2	Betriebssystem	5
1.3	Benutzeroberflächen	9
1.3.1	Kommandozeile, curses-Fenster	9
1.3.2	X Window System (X11)	11
1.3.3	Desktop Environments (KDE, GNOME)	22
1.3.4	Multimediale Oberflächen	24
1.3.5	Oberflächen für Behinderte	25
1.3.6	Lokalisierung	30
1.4	Anwendungen	33
1.4.1	Kommando-Interpreter	33
1.4.2	Dienstprogramme	34
1.4.3	Eigentliche Anwendungen	34
1.5	Hilfen (man, info, Scrollkeeper, HOWTOs, Guides, FAQs, RFCs, Wikis, LUGs)	36
1.6	Eine Sitzung	43
1.7	Sicherheit	45
1.8	Nachrüsten von Software	49
1.8.1	Debian-Pakete	49
1.8.2	RPM-Pakete	50
1.8.3	tar-Archive (Tarballs)	50
1.8.4	Entfernen von Software	51
1.8.5	Software für MS Windows oder MacOS	52
2	Kommandointerpreter	53
2.1	Bourne-again-Shell (bash)	53
2.1.1	Gesprächspartner	53
2.1.2	Eingebaute Kommandos	57

2.1.3	Wiederholen und Editieren von Kommandos	59
2.1.4	Umlenkung	60
2.1.5	Umgebung und Shellvariable	62
2.1.6	Positionsparameter	65
2.1.7	Gruppierung von Kommandos	65
2.1.8	Restricted Shell	66
2.1.9	Wechsel der Shell	66
2.1.10	Shellskripte	67
2.2	Weitere Shells	69
3	Dienstprogramme	71
3.1	Daten in Ruhe (Verzeichnisse und Dateien)	71
3.1.1	Anzeigen und Navigieren (pwd, cd, ls, file)	71
3.1.2	Ändern von Besitz und Zugriffsrechten (chown, chgrp, chmod, umask)	73
3.1.3	Kopieren, Umbenennen und Löschen von Dateien (cp, head, tail, dd, mv, rm, split)	75
3.1.4	Anlegen, Umbenennen und Löschen von Verzeichnissen (mkdir, mv, rmdir)	77
3.1.5	Verlinken (ln)	77
3.1.6	Zeitstempel (ls, touch)	79
3.1.7	Datei-Manager (lfm, mc, xfe, nautilus, konqueror)	80
3.1.8	Packen (compress, gzip, bzip2, zip)	82
3.1.9	Archivieren (ar, tar), Backup	84
3.1.10	Mitschreiben (tee, script)	85
3.1.11	Dateisysteme im Netz (NFS, Samba, Netatalk)	86
3.1.12	Arbeiten mit externen Datenträgern (tar, mount)	89
3.1.13	DOS-Dateisysteme (mtools)	91
3.1.14	Brennen von CD/DVDs (cdrecord, bashburn, gcombust, xcdroast, k3b, cdrdao)	91
3.2	Daten in Bewegung (Prozesse)	95
3.2.1	Anzeigen von Prozessen (ps, top, gnome-system-monitor)	95
3.2.2	Priorität, Load Balancing (nice, batch, queue)	97
3.2.3	Jobkontrolle (fg, bg, jobs, nohup)	98
3.2.4	Signale (kill)	100
3.2.5	Pipes	101
3.3	Drucken	102
3.3.1	Drucksysteme (lp, lpr, CUPS)	102
3.3.2	PostScript (psutils, a2ps)	105
3.4	Benutzer, Maschine (who, id, rwho, hostname)	106
3.5	Kalender, Uhr (date, leave, cal, cron, at, calendar)	107
3.6	Secure Shell (ssh)	110
3.7	Secure Sockets Layer, Transport Layer Security (openssl, gnutls)	112
3.8	Wireless Local Area Network (WLAN)	113

4	Suchen und Finden	115
4.1	Grundbegriffe	115
4.1.1	Zeichenketten (Strings)	115
4.1.2	Reguläre Ausdrücke	116
4.1.3	Logische Verknüpfungen	120
4.1.4	Indizierung	121
4.2	Suchen nach lokalen Informationen	122
4.2.1	Suchen nach Zeichenketten in Dateien (Editoren, grep)	122
4.2.2	Volltextsuche (lookup, glimpse)	123
4.2.3	Suchen in lokalen Wörterbüchern (ding, dict)	124
4.2.4	Suchen nach binären Mustern in Dateien (gnuifft, imgseek)	125
4.2.5	Suchen nach Dateien und Verzeichnissen (find, whereis, which, locate)	126
4.2.6	Suchen nach Debian-Paketen (ara, dpkg-iasearch, packagesearch, apt-file, dlocate)	128
4.2.7	Suchen nach Informationen über Benutzer (finger)	130
4.2.8	Lokale Webserver	130
4.3	Informationen im Internet	132
4.3.1	Übersicht (CUSI, Suchfibel, Suchlexikon)	132
4.3.2	Suchen nach Netzknoten (ping, arping, host, mtr)	133
4.3.3	Internet-Domänen	135
4.3.4	Email-Anschriften (MESA)	136
4.3.5	Wörterbücher, Glossare und Lexika (LEO, Wikipedia)	137
4.3.6	Suchen nach Büchern	138
4.3.7	Anonymous FTP, BitTorrent, Jigdo	140
4.3.8	Web-Browser	142
4.3.9	Suchmaschinen im Web	144
5	Textverarbeitung	155
5.1	Grundbegriffe	155
5.1.1	Zeichensätze – von ASCII bis Unicode	155
5.1.2	Fonts	162
5.1.3	Dateiformate (txt, pdf, ps, rtf)	164
5.1.4	Allgemeines über Texteditoren	167
5.2	Werkzeuge zum Lesen und Umwandeln	168
5.2.1	Einfache Werkzeuge (cat, more, pg, less, lv, bidiv, view)	168
5.2.2	Dumper (hexcat, od)	169
5.2.3	Lesen und Umwandeln von bzw. nach PostScript (gv, a2ps)	170
5.2.4	Lesen von PDF-Dateien (Adobe Reader, xpdf, gpdf, kpdf, pdftk, gv, Evince)	171
5.2.5	Lesen von Word-Dateien und Ähnlichem (catdoc, wordview, antiword, wv, word2x, xlhtml, unrtf)	172
5.2.6	Optical Character Recognition (clara, gocr)	173

	5.2.7	Umcodieren von Zeichen und Zeichensätzen (tr, recode, uniconv, iconv, convmv)	174
5.3		Der Editor vi samt Nachwuchs	174
	5.3.1	Schnellstart	174
	5.3.2	Übersicht	175
	5.3.3	Konfiguration	176
	5.3.4	Makros	178
	5.3.5	Aufrufen und Beenden	179
	5.3.6	Navigieren im Text	181
	5.3.7	Text ändern	183
	5.3.8	Pufferspeicher und Zwischendateien	184
	5.3.9	Sonstige Kommandos	185
	5.3.10	Zusammenarbeit vim – ispell	186
	5.3.11	VimOutliner – ein Outline Prozessor	187
5.4		Universalgenie: Emacs	188
	5.4.1	Schnellstart	188
	5.4.2	Übersicht	188
	5.4.3	Einrichtung und Konfiguration	189
	5.4.4	Aufrufen und Beenden	191
	5.4.5	Navigieren im Text	192
	5.4.6	Text ändern	193
	5.4.7	Cut & Paste	194
	5.4.8	Sonstiges	194
5.5		Weitere Texteditoren	195
	5.5.1	Einfachst: nano	195
	5.5.2	Joe's Own Editor (joe)	195
	5.5.3	Der Nirwana-Editor (nedit)	196
	5.5.4	Ein Editor für KDE (kate)	196
	5.5.5	Ein Editor für GNOME (gedit)	197
	5.5.6	aXe – ein einfacher Editor unter X11	198
	5.5.7	jEdit und jed – für Programmierer	198
	5.5.8	Editor für das Rich Text Format (ted)	198
5.6		WYSIWYG-Textprozessoren	199
	5.6.1	GNOME AbiWord	199
	5.6.2	KDE KWord	200
	5.6.3	Open.org writer	201
	5.6.4	LyX	201
5.7		Hex-Editoren (bvi, hexedit, hexcurse, vche)	202
5.8		Editoren für fremde Schriften (geresh, katoob, yudit)	202
5.9		Streaming Editor (sed)	205
5.10		Listenbearbeitung (awk)	206
5.11		Formatieren	208
	5.11.1	Inhalt, Struktur und Aufmachung	208

5.11.2	UNIX-Formatierer (nroff, groff)	209
5.11.3	GNU Texinfo	210
5.11.4	DocBook	213
5.11.5	L ^A T _E X	213
5.12	Schreiben für das Web	223
5.12.1	Was ist Hypertext?	223
5.12.2	Hypertext Markup Language (HTML)	224
5.12.3	Standard Generalized Markup Language (SGML)	227
5.12.4	Extensible Markup Languages (XML, XHTML)	227
5.12.5	Web-Entwicklungsumgebung (quanta)	228
5.13	Ergänzende Textwerkzeuge	229
5.13.1	Tipptrainer (ktouch, tipptrainer, tuxtype)	229
5.13.2	Vergleichen (diff, cmp, ed)	229
5.13.3	Sortieren (sort)	230
5.13.4	Rechtschreibung prüfen (ispell, aspell, acheck)	231
5.13.5	Weitere Werkzeuge (cut, paste usw.)	232
5.14	Verschlüsselung	233
5.14.1	Aufgaben der Verschlüsselung	233
5.14.2	Symmetrische Verfahren	234
5.14.3	Asymmetrische Verfahren	235
5.14.4	Angriffe (Kryptanalyse)	237
5.14.5	Einrichten des GNU Privacy Guards (GnuPG)	237
5.15	Organisation eines Textprojektes	239
5.15.1	Verzeichnisstruktur	239
5.15.2	Unentbehrlich: make	241
5.15.3	Versionsverwaltung (rcs, cvs, subversion, arch)	243
6	Mathematische Werkzeuge	251
6.1	Grundbegriffe	251
6.2	Kleine numerische Werkzeuge (seq, pi, gbase, factor)	252
6.3	Diagramme (gnuplot)	253
6.4	Taschenrechner	254
6.4.1	Rechner für die Kommandozeile (bc, dc)	254
6.4.2	Ein Taschenrechner mit GUI (KCalc)	258
6.4.3	Zwei Taschenrechner mit GUI (calculator)	258
6.4.4	Drei Taschenrechner mit GUI (gcalctool)	259
6.5	Zufallszahlen	260
6.6	Tabellenkalkulationen	262
6.6.1	Einfachst: sc	262
6.6.2	GNOME Gnumeric	263
6.6.3	KDE KSpread	265
6.6.4	OpenOffice.org Calc	265
6.7	Numerik	266

6.7.1	Octave	266
6.7.2	Euler	267
6.7.3	Scilab	267
6.8	Statistik (GNU R)	268
6.9	Computeralgebra	269
6.9.1	Was ist ein Computeralgebrasystem?	269
6.9.2	YACAS	270
6.9.3	MuPAD	270
6.9.4	Axiom	271
6.9.5	SINGULAR	272
7	Datenbanken und andere Wissenspeicher	273
7.1	Grundbegriffe	273
7.2	Einfache Datenbank mit awk, grep und sort	278
7.3	Berkeley Database	279
7.4	PostgreSQL	279
7.4.1	Pakete und Dateien	279
7.4.2	Benutzer und Privilegien	280
7.4.3	Tabellen	281
7.4.4	Data Base Connectivity	282
7.5	MySQL und MaxDB	283
7.6	Wikis	284
7.7	Verzeichnisdienste	286
7.7.1	Whois-Verzeichnisdienst	286
7.7.2	Lightweight Directory Access (LDAP)	286
7.8	Expertensysteme (CLIPS)	289
8	Das grafische Atelier	291
8.1	Grundbegriffe	291
8.1.1	Aufgaben	291
8.1.2	Rastergrafik	292
8.1.3	Vektorgrafik	292
8.1.4	Farben	293
8.1.5	Rendern (Perspektive, Beleuchtung)	294
8.2	Zeichnen (xfig, inkscape, kseg, Z.u.L)	295
8.3	CAD (qcad)	297
8.4	Schaltpläne (gEDA, EAGLE, Chipmunk)	298
8.5	Finite Elemente (FElt, TOCHNOG, FreeFem, FFEP)	300
8.6	Digitale Fotos (gphoto/gtkam, digikam, photopc)	301
8.7	Bildbetrachter (qiv, feh, xpcd, xv)	302
8.8	Bildverarbeitung	303
8.8.1	Konverter (netpbm, autotrace)	303
8.8.2	Ein Werkzeugkoffer (GIMP)	304

8.8.3	Noch ein Koffer (ImageMagick)	308
8.8.4	Scannen (scanimage, xscanimage, xsane, quiteinsane) . . .	309
8.8.5	Panoramen (panotools, hugin)	310
8.8.6	Warpen und Morphen (morph, xmorph, gtkmorph)	311
8.8.7	Modellierer, Renderer (Ayam, Aqsis, povray, blender, yafray)	313
8.9	Virtuelle Welten	325
8.9.1	Was ist Virtuelle Realität?	325
8.9.2	Ein Anfang (OpenSG)	326
9	Das digitale Tonstudio	327
9.1	Grundbegriffe	327
9.1.1	Vom Schall zum Byte und zurück	327
9.1.2	Dateiformate	328
9.1.3	Soundsysteme (ALSA, OSS)	330
9.2	Mischer	331
9.3	Konverter, Denoiser, Audioeditoren	333
9.3.1	Encoder, Konverter (sox)	333
9.3.2	Denoiser (gwc)	333
9.3.3	Audioeditoren (Audacity, ReZound, sweep usw.)	334
9.4	Tonkonserven	336
9.4.1	Dateien (Zinf, Rhythmbox, AlsaPlayer, XMMS, ogg123)	336
9.4.2	Musik-Server (mserv)	338
9.4.3	CD/DVD-Spieler (WorkBone, xmcd, gnome-cd, KsCD)	338
9.4.4	Ripper/Encoder (cdparanoia, abcde, jack, grip, eac)	339
9.4.5	Rekorder (brec, Sound Recorder, KRec, krecord)	341
9.5	Klangerzeugung (MIDI)	342
9.6	Spracherkennung (XVoice, sphinx)	346
9.7	Sprachsynthese, Screenreader	347
9.7.1	Einfache Beispiele (saydate, saytime, flite)	347
9.7.2	Sprachsynthese (Epos, Festival, FreeTTS, MBROLA)	348
9.7.3	Screenreader (emacspeak, screader, uxdots)	349
10	Video und Fernsehen	351
10.1	Grundbegriffe	351
10.2	Konverter, Videoeditoren	353
10.3	Media Player	353
10.3.1	VLC Media Player	353
10.3.2	Xine und Frontends (Xine, gxine, Kaffeine, Totem)	354
10.3.3	Mplayer, XMMS	355
10.3.4	Ogle	356
10.3.5	Kaboodle, Noatun	356
10.3.6	Helix/Real Player	357

10.4	Fernsehen mit dem PC (xawtv, motv, tvtime, alevtd)	358
10.5	Webkamas (webcam)	361
11	Kommunikation	363
11.1	Grundbegriffe	363
11.2	Direkter Dialog (write, wall, talk, mlchat)	365
11.3	Email, Listen (mail usw.)	365
11.4	Neuigkeiten: Netnews, Usenet (nn, gnus, knode, pan, tin)	372
11.5	Internet Relay Chat (ircII, xchat, ksirc, zircon)	375
11.6	Instant Messaging (micq, imcom, kopete, gaim, jabber)	377
11.7	Präsentation (kpresenter, impress)	379
11.8	Fax, Internet-Telefonie	381
11.8.1	Fax (hylafax, efax)	381
11.8.2	Internet-Telefonie (kphone, gnomemeeting, Asterisk)	382
11.9	Vom Organizer zur Groupware	385
11.9.1	Übersicht	385
11.9.2	Notizbuch, Personal Information Manager (hnb, tina, Chandler, TuxCards, Kontact, openCRX)	386
11.9.3	Termine (calendar, WebCalendar, KOrganizer)	388
11.9.4	Anschriften, Telefonnummern (XmAddressBook, KAddressBook)	389
11.9.5	Projekte (gnotime, kfocus, opensched, planner, GanttProject, gforge, maven)	390
11.9.6	Die Krönung: Groupware	393
11.10	Daten synchronisieren (unison, ksync, kitchensync, multisync, kandy)	396
11.11	Mindmapper und Jotter (FreeMind, gjots2, kjots, think, kdissert)	398
11.12	Content Management (OpenACS, Mambo, Zope, Plone, Typo3)	400
11.13	Sprachtrainer (klatin, kvoctrain)	403
12	Office-Pakete	405
12.1	Übersicht	405
12.2	OpenOffice.org	407
12.3	GNOME Office	410
12.4	KOffice	410
12.5	CrossOver Office	412
13	Spiele und Spielereien	413
13.1	GNU Chess (gnuchess, xboard)	413
13.2	Brettspiele (gtkboard), Go (gnugo, cgoban)	414
13.3	Freeciv – ein Civilization-Klon	415
13.4	TuxRacer – für Wintersportler	417
13.5	Pingus – ein Lemmings-Klon	419

13.6	Flight Gear – ein anspruchsvoller Flugsimulator	419
13.7	Das Planetarium auf dem Desktop (kstars)	420
13.8	MegaHAL – ein Gesprächs-Simulator	421
13.9	Fraktale (kfract, kaos, xfractint)	423
13.10	Türme von Hanoi	424
13.11	Game of Life	426
13.12	Wortspielereien	427
A	Zahlensysteme	429
B	Zeichensätze und Sondertasten	436
B.1	EBCDIC, ASCII, Roman8, IBM-PC	436
B.2	German-ASCII	441
B.3	ASCII-Steuerzeichen	442
B.4	Latin-1 (ISO 8859-1), Unicode	443
B.5	Latin-2 (ISO 8859-2)	449
B.6	HTML-Entities	450
B.7	Sondertasten	451
C	Metazeichen	454
C.1	Shell	454
C.2	LaTeX	455
C.3	Reguläre Ausdrücke	456
C.4	make	456
C.5	HTML	456
D	Papier- und Schriftgrößen	457
D.1	Papierformate	457
D.2	Schriftgrößen	458
E	Farben	459
E.1	X11-Farben	459
E.2	HTML-Farben	460
F	Datei-Merkmale	461
F.1	Zugriffsrechte	461
F.2	Kennungen	463
G	Prozess-Signale	467
H	Slang im Netz	469

I	Beispiele LaTeX	476
I.1	Gelatexte Formeln	476
I.2	Formeln im Quelltext	480
J	Modem-Kommandos (Hayes)	486
K	ISO 3166 Ländercodes	488
L	Requests For Comments (RFCs)	491
L.1	Alle FYIs	491
L.2	Alle BCPs	493
L.3	Einige 1.-April-RFCs	496
M	Zeittafel	497
N	Zum Weiterlesen	508
	Personenverzeichnis	519
	Sachverzeichnis	522

Abbildungen

1.1	Aufbau der Hardware	2
1.2	Arbeitsplatz	5
1.3	GNU-Kopf	6
1.4	Tux, der Pinguin	7
1.5	Debian Swirl	8
1.6	X Window System	15
1.7	Screenshot X-Uhren	15
1.8	Deutsch-arabische Tastatur	17
1.9	X11 Screen Dump	21
1.10	KDE-Logo	22
1.11	GNOME-Logo	23
1.12	Screenshot X11-Fenster	25
1.13	Screenshot xmag	27
1.14	Screenshot Bildschirmlupe kmag	28
1.15	Aufbau eines Systems	35
1.16	Screenshot KDE Hilfezentrum	39
1.17	Screenshot Yelp	40
1.18	Anmelde-Bildschirm	43
1.19	Wichtige Mail	46
3.1	Harter Link	78
3.2	Weicher Link	79
3.3	Screenshot Midnight Commander	80
3.4	Screenshot GNOME Nautilus	81
3.5	Screenshot KDE Konqueror	82
3.6	Screenshot xsmbrowser	89
3.7	Mounten externer Datenträger	90
3.8	Screenshot gcombust	93
3.9	Screenshot GNOME System Monitor	96
3.10	Screenshot CUPS	104

4.1	Screenshot Suchmaschine htDig	132
4.2	Screenshot Email-Suchmaschine MESA	136
4.3	Screenshot wget mit gwget	141
4.4	Screenshot Firefox	143
4.5	Screenshot Suchmaschine Google	145
4.6	Screenshot Suchmaschine Altavista	146
4.7	Screenshot Suchmaschine Fireball	147
4.8	Screenshot Ask Jeeves	148
4.9	Screenshot Katalog Yahoo	149
4.10	Screenshot Katalog Allesklar	150
4.11	Screenshot Meta-Suchmaschine MetaGer	152
5.1	Screenshot Evince	171
5.2	Screenshot Editor nano	196
5.3	Screenshot Editor kate	197
5.4	Screenshot Editor gedit	197
5.5	Screenshot Editor axe	198
5.6	Screenshot AbiWord	199
5.7	Screenshot kword	200
5.8	Screenshot oowriter	201
5.9	Screenshot hexedit	203
5.10	Screenshot Editor geresch	204
5.11	Screenshot Editor yudit	204
5.12	Arbeitsablauf \LaTeX	215
5.13	Screenshot kile	222
5.14	Screenshot Quanta	228
5.15	Screenshot KTouch	230
5.16	Screenshot LinCVS	247
6.1	Plot $(\sin x)/x$	254
6.2	Dreidimensionaler Plot	255
6.3	Screenshot KCalc	257
6.4	Screenshot galculator	259
6.5	Screenshot galculator	259
6.6	Screenshot gcalctool	260
6.7	Screenshot gcalctool	260
6.8	Screenshot gcalctool	261
6.9	Screenshot Gnumeric	264
6.10	Screenshot KSpread	265
6.11	Screenshot Tabellenkalkulation Calc	266
6.12	Screenshot Scilab	268
6.13	Screenshot Vulkan	269
6.14	MuPAD 2D-Plot	270

7.1	Screenshot Datenbankmaske	275
7.2	Screenshot Miniwiki	285
8.1	Skizze mittels xfig	296
8.2	Screenshot inkscape	297
8.3	Screenshot QCad	298
8.4	Screenshot Eagle	299
8.5	Screenshot GIMP	304
8.6	Zeichnung, aus Foto erzeugt	306
8.7	Altes Foto	307
8.8	Screenshot ImageMagick	308
8.9	Gewarpptes Porträt	311
8.10	Gemorphptes Porträt	312
8.11	Screenshot Ayam	314
8.12	Screenshot Aya-Main	315
8.13	Screenshot Aqsis, paralleles Licht	316
8.14	Screenshot Aqsis, Spotlight	319
8.15	Screenshot Blender	320
8.16	Mit Blender gerendertes Bild	325
9.1	Screenshot kmix	332
9.2	Screenshot Audioeditor sweep	334
9.3	Screenshot zinf	336
9.4	Screenshot X MultiMedia System	337
9.5	Screenshot des CD-Spielers xmcd	339
9.6	Screenshot des CD-Spielers KsCD	339
9.7	Screenshot grip	341
9.8	Screenshot Krec	342
9.9	Screenshot TiMidity	343
9.10	Screenshot NoteEdit	344
9.11	Screenshot Rosegarden	346
10.1	Screenshot Media-Player VLC	354
10.2	Schnappschuss Media-Player Totem	355
10.3	Screenshot noatun	357
10.4	Schnappschuss xawtv	359
10.5	Schnappschuss tvtime	360
11.1	Screenshot sylpheed	367
11.2	Screenshot IM-Client Psi	378
11.3	Screenshot Präsentation Impress	380
11.4	Telefone	382
11.5	Screenshot gnomemeeting	384

11.6	Screenshot TuxCards	387
11.7	Screenshot KOrganizer	389
11.8	Screenshot XmAddressBook	390
11.9	Screenshot KAddressBook	391
11.10	Screenshot planner	392
11.11	Screenshot Evolution	395
11.12	Screenshot FreeMind	399
11.13	Screenshot kjots	400
11.14	Screenshot Plone	402
12.1	Screenshot OpenOffice.org	408
12.2	Screenshot StarOffice	409
12.3	Screenshot GNOME-DB	411
12.4	Screenshot KDE kexi	412
13.1	Screenshot GNU Chess	414
13.2	Screenshot Puzzle aus gtkboard	415
13.3	Screenshot Mastermind aus gtkboard	416
13.4	Screenshot GO-Spiel	417
13.5	Screenshot FreeCiv	418
13.6	Screenshot TuxRacer	419
13.7	Screenshot Pingus	420
13.8	Screenshot Flugsimulator	421
13.9	Screenshot Kstars	422
13.10	Screenshot Fraktal	423
13.11	Screenshot Game of Life	427

Tabellen

3.1	Gepackte ps-Dateien	83
3.2	Gepackte pdf-Dateien	83
3.3	Druckkommandos	103
5.1	Zeichendarstellung	159
12.1	Übersicht Office-Pakete	406
F.1	Zugriffsrechte von Dateien und Verzeichnissen	462

Programme und andere Quellen

2.1	Shellskript mkindex	68
3.1	Shellskript brennen	94
3.2	Shellskript frequenz	101
5.1	awk-Skript Sachregister	207
5.2	L ^A T _E X-Datei dinbrief.tex	217
5.3	Makefile makebrief	217
5.4	HTML-Dokument	225
5.5	Shellscript Stilanalyse	233
5.6	Makefile Buch	242
5.7	Makefile Türme von Hanoi	243
6.1	gnuplot-Skript $y = (\sin x)/x$	253
6.2	bc-Script Türme von Hanoi	257
13.1	Shellskript Türme von Hanoi	425

Zum Gebrauch

- Das erste Kapitel enthält Grundlagen, die im weiteren Verlauf als bekannt vorausgesetzt werden. Die restlichen Kapitel sind weitgehend unabhängig voneinander.
- Hervorhebungen im Text werden *kursiv* dargestellt.
- Titel von Veröffentlichungen oder Abschnitten, Namen von Debian-Paketen oder -Distributionen, Menüpunkte, kurze Zitate oder wörtliche Rede werden im Text *kursiv* markiert.
- In Aussagen über Wörter werden diese *kursiv* abgesetzt.
- Namen von Personen stehen in KAPITÄLCHEN.
- Eingaben in die Tastatur oder Ausgaben auf den Bildschirm werden in Schreibmaschinenschrift wiedergegeben, ebenso Dateinamen, Ausschnitte aus Dateien (Text, Code) oder URLs.
- Ist eine buchstäbliche Eingabe durch einen gewöhnlichen Benutzer gemeint, so wird der Eingabe der Systemprompt vorangestellt, erforderlichenfalls mit der Angabe des jeweiligen Arbeitsverzeichnisses:

```
joe@debian:~$ ls -l /usr/bin
```

```
joe@debian:/etc$ less passwd
```

Um die Eingabe wirksam werden zu lassen, ist abschließend die Eingabetaste <cr> (Return, Enter, Amen) zu drücken. Darauf wird nicht jedes Mal hingewiesen. Lange Eingabezeilen müssen gelegentlich wegen der begrenzten Papierbreite in mehrere Druckzeilen aufgeteilt werden; auf der Tastatur sind sie als eine durchgehende Zeile ohne Unterbrechung einzugeben.

- Bei einer buchstäblichen Eingabe durch den Systemverwalter (root) sieht der Prompt anders aus:

```
debian:~# init s
```

```
debian:/var/log# more auth.log | less
```

- Tastenbezeichnungen werden in Spitzmarken eingerahmt, beispielsweise <a>, <A> oder <tab>. Sind Tastenkombinationen (keycords) einzugeben, werden die Tasten mit einem Bindestrich verbunden: <ctrl>-<c>. Den Tasten aus dem Nummernblock (keypad) wird KP vorangestellt: <KP+>. Eine Übersicht über Sondertasten wie <ctrl> findet sich im Anhang B.7 auf Seite 451.
- Wenn nichts anderes gesagt ist, stammt ein Programm oder Werkzeug aus einem gleichnamigen Debian-Paket aus der *sarge*-Distribution.
- Im Sachverzeichnis werden Umlaute wie die zugehörigen Grundlaute einsortiert (wie im Duden).
- Suchen Sie die englische oder französische Übersetzung eines deutschen Fachwortes, so finden Sie diese bei der erstmaligen Erläuterung des deutschen Wortes.
- Suchen Sie die deutsche Übersetzung eines englischen oder französischen Fachwortes, so finden Sie einen Verweis im Sachverzeichnis.
- UNIX wird hier immer als die Gattung der aus dem bei AT&T um 1970 entwickelten UNIX ähnlichen Betriebssysteme verstanden, nicht als geschützter Name eines bestimmten Produktes.
- Wir geben möglichst genaue Hinweise auf weiterführende Dokumente im Netz. Der Leser denke aber daran, dass sich sowohl Inhalte wie Adressen (URLs) ändern. Im Lauf einiger Jahre werden viele Hyperlinks von der Linkfäule befallen; das ist bei der Dynamik des Internets unvermeidlich. Wir haben jeden Hyperlink geprüft, lehnen aber eine Haftung für die Inhalte nicht von uns verfasster Webseiten oder sonstiger Dokumente ab.
- Wir bemühen uns um die neue deutsche Rechtschreibung, aber Gewohnheit ist ein eisern Hemd. Und dann hat noch unser Rechtschreib-Prüfprogramm seine Ansichten.
- Unter *Benutzer*, *Programmierer*, *Verwalter* usw. werden sowohl männliche wie weibliche Erscheinungsformen verstanden.
- Irren ist menschlich. Wir machen da keine Ausnahme und sind für freundliche Hinweise dankbar.

Von der Hardware zur Anwendung

Das Kapitel erläutert die Zusammenarbeit der Hardware und der einzelnen Software-Schichten bis hin zum Anwendungsprogramm bei der Lösung einer Aufgabe mittels eines Linux/UNIX-Systems.

1.1 Hardware und Software

Ein Computer, ein Rechner (E: computer, F: ordinateur) ist eine Maschine, die aus Hardware (E: hardware, F: matériel) und Software (E: software, F: logiciel) besteht. Die Hardware ohne die Software ist ein aufwendiger Heizlüfter, mehr nicht. Hardware besteht aus Kupfer, Eisen, Glas, Kunststoff, Pappe; man kann sie anfassen. Software besteht aus Gedanken, Informationen, Daten. Die Software ohne die Hardware vermag nichts. Es ist sogar fraglich, ob sie ohne Hardware existieren könnte. Ein Buch, eine CD/DVD, der graue Kasten auf oder neben Ihrem Schreibtisch sind Hardware, der Inhalt des Buches, der CD/DVD oder des Speichers in dem grauen Kasten ist Software. Dichterfürst FRIEDRICH VON SCHILLER sagt treffend:

Leicht beieinander wohnen die Gedanken,
doch hart im Raume stoßen sich die Sachen.

Die Verse stehen in *Wallensteins Tod* im 2. Aufzug, 2. Auftritt. Was sich hart im Raum stößt, gehört zur Hardware, was leicht beieinander wohnt, die Gedanken, ist Software.

Besagte Maschine ist ein Automat, der auf bestimmte Eingaben mit bestimmten Tätigkeiten und Ausgaben antwortet. Dieselbe Eingabe führt immer zu derselben Ausgabe, darauf verlassen wir uns. Deshalb ist es im Grundsatz unmöglich, mit Rechnern Zufallszahlen zu erzeugen (zu würfeln). Zwischen einem Briefmarken-Automaten (Postwertzeichengeber) und einem Rechner besteht jedoch ein wesentlicher Unterschied. Ein Briefmarken-Automat nimmt nur Münzen entgegen und gibt nur Briefmarken aus. Sein Verhalten ist durch seine Mechanik unveränderlich vor-

gegeben. Es hat auch mechanische Rechenautomaten für spezielle Aufgaben wie die Berechnung von Geschosshbahnen (Ballistik) oder Gezeiten¹ gegeben.

Bei einem Rechner hingegen wird das Verhalten durch die Software bestimmt, die im Gerät gespeichert ist und einfach ausgewechselt werden kann. Dieselbe Hardware kann sich wie eine Schreibmaschine, eine Rechenmaschine, eine Zeichenmaschine, wie ein CD-Spieler, ein Telefon, ein Schachspieler oder wie ein Lexikon verhalten, je nach Programm.

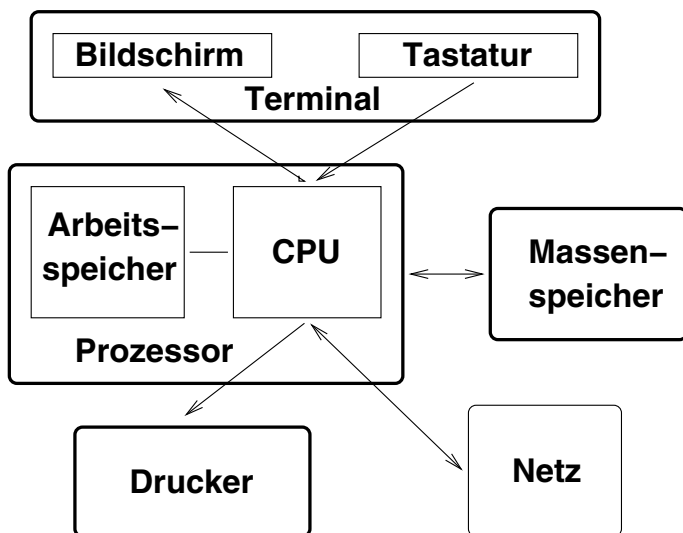


Abbildung 1.1. Aufbau der Hardware eines Rechners

Der Benutzer (E: user, F: utilisateur) sieht von einem Rechner vor allem den Bildschirm (E: screen, F: écran) und die Tastatur (E: keyboard, F: clavier), auch Hackbrett genannt. Bildschirm und Tastatur zusammen werden als Terminal (E: terminal, F: terminal) bezeichnet und stellen die Verbindung zwischen Benutzer und Maschine dar. Mittels der Tastatur spricht der Benutzer zum Rechner, auf dem Bildschirm erscheint dessen Antwort. Sofern vorhanden, zählt eine Maus oder Rollkugel (Zeigegerät, E: pointing device, mouse, trackball, F: dispositif de pointage, souris, boule de commande) auch zum Terminal.

Bei Anlagen mit mehreren Terminals wird das erste, unbedingt notwendige Terminal, das auf einer bestimmten Adresse liegt, als Konsole bezeichnet. Beim Systemstart nimmt der Rechner zuerst mit der Konsole Kontakt auf. Dorthin schickt er auch allfällige Hilferufe. Die Konsole ist der Arbeitsplatz des Verwalters. Häufig wird aber auch jedes Textterminal oder jedes Fenster, in dem ein Textterminal zur Eingabe von Kommandozeilen dargestellt wird, als Konsole bezeichnet. Für die aus ihm

¹Im Deutschen Schifffahrtsmuseum in Bremerhaven sind mechanische Gezeitenrechner ausgestellt.

heraus gestarteten Prozesse ist es das Kontrollterminal (kontrollierende Dialogstation). Auf einem Linux-Rechner gibt es bis zu 63 virtuelle Konsolen oder Terminals (`/dev/tty*`), siehe die Manualseite zu `console`.

Der eigentliche Rechner, die Prozessor- oder Zentraleinheit (E: central unit, F: unité centrale) ist in die Tastatur eingebaut wie seinerzeit beim Commodore C64 oder Schneider CPC 464, in das Bildschirmgehäuse wie beim ersten Apple Macintosh oder beim heutigen iMac, oder sie steckt in einem eigenen Gehäuse. Die wichtigsten Teile der Zentraleinheit sind der Zentralprozessor (CPU, E: central processing unit, F: processeur central) und der Arbeitsspeicher (E: memory, F: mémoire centrale, mémoire vive, mémoire secondaire).

Um recht in Freuden arbeiten zu können, braucht man noch Massenspeicher (E: storage, F: mémoire de masse), die ihren Inhalt nicht vergessen, wenn der Rechner abgeschaltet wird. Massenspeicher können fest in die Maschine eingebaut sein wie Festplatten (E: hard disk, F: disque dur) oder oder entfern- und transportierbar wie Disketten (E: floppy disk, F: disquette), Compact Disc/Digital Versatile Discs (CD/DVDs, F: disque optique compact, disque numérique polyvalent) oder Memory Sticks. In letzterem Fall benötigt der Rechner ein zu dem Medium passendes Laufwerk (E: drive, F: dérouleur) für die Scheiben bzw. einen Anschluss für die Stöpsel. Zwischen verschiedene Speicherebenen ist meist ein Cache (E: cache, F: antémémoire), ein kleiner Zwischenspeicher, eingefügt, um Geschwindigkeitsunterschiede auszugleichen und Wartezeiten zu verringern.

Da man gelegentlich etwas schwarz auf weiß besitzen möchte, gehört zu den meisten Rechnern ein Drucker (E: printer, F: imprimante). Heute ist das oft ein Laserdrucker, zunehmend mit Farbe. Aber auch Tintenstrahldrucker haben ihre Stärken, beim Ausdrucken von Fotos oder Folien beispielsweise. Bei Farbdruckern sind die Betriebskosten nicht unerheblich.

Ferner ist ein Rechner, der etwas auf sich hält, durch ein Netz (E: network, F: réseau) mit anderen Rechnern rund um die Welt verbunden. Kommunikation ist zu einer wesentlichen Aufgabe der Rechner geworden. Netze, die:

- räumlich im Prinzip unbegrenzt sind,
- einem unbegrenzten Kreis von Benutzern offen stehen,
- vielen Besitzern gehören,
- verschiedenste Techniken verwenden,

werden Wide Area Networks (WAN, F: réseau étendu) genannt. Ein bekannter Vertreter ist das Internet, der älteste das Telefonnetz. Im Gegensatz dazu stehen Local Area Networks (LAN, F: réseau locale), die

- räumlich begrenzt sind,
- nur eingetragene Benutzer zulassen.
- einem bestimmten Besitzer gehören,
- die Vielfalt der Technik möglichst überschaubar halten.

Das Campusnetz einer Hochschule ist ein größeres LAN, ebenso ein firmeninternes Netz. Bestimmte Ausprägungen von LANs sind Wireless LANs (WLAN), Domestic Area Networks (DAN, begrenzt auf ein Wohngebäude), Family Area Networks

(FAN), Storage Area Networks (SAN) usw. Virtuelle private Netze (Virtual Private Network, VPN) bauen auf der Hard- und Software eines anderen Netzes – meist des Internets – mittels Software ein LAN auf, zu dem nur die eigenen Benutzer Zugang haben. Die Grenzen sind fließend, außerdem greifen Computernetze auf andere Netze (Telefon, Kabelfernsehen) über oder umgekehrt. Ihr Internet Service Provider (ISP) bindet Ihr Netz an das große Internet an und erwartet dafür ein Entgelt.

Was um die Zentraleinheit herumsteht, wird als Peripherie bezeichnet. Die peripheren Geräte sind über Schnittstellen (Datensteckdosen, E: interface, F: interface) angeschlossen. In Abbildung 1.1 sehen wir das Ganze schematisch dargestellt. In der Mitte die CPU, eng verbunden mit dem Arbeitsspeicher. Um dieses Paar herum die Peripherie aus Terminal, Massenspeicher, Drucker und Netzanschluss. Sie können aber immer noch nichts damit anfangen, allenfalls heizen. Es fehlt noch die Intelligenz in Form eines Betriebssystems (E: operating system, F: système d'exploitation) wie Debian GNU/Linux.

Debian GNU/Linux unterstützt ein breites Spektrum von Hardware, neben PCs nach IBM-Muster auch Macintosh-Rechner und Workstations von Hewlett-Packard, IBM, Sun und anderen. Dennoch kann es in drei Bereichen zu Problemen kommen:

- Sehr neue Hardware (Grafikkarten, schnelle WLAN-Adapter),
- Laptops oder Notebooks, weil diese oft spezielle Bauteile enthalten, um Strom und/oder Platz zu sparen,
- Hardware, die eigens nur für ein bestimmtes Betriebssystem (das heißt MS Windows) entwickelt wurde.

In diesen Fällen sollten Sie sich vor größeren Anschaffungen erkundigen, ob die ins Auge gefasste Hardware mit Linux zusammenarbeitet. Dagegen stehen die Chancen gut, alte Hardware unter Linux wiederbeleben zu können.

Ein Wort noch zur Gestaltung des Arbeitsplatzes. Da man in vielen Berufen heute stundenlang am Bildschirm arbeitet, spielt die zweckmäßige Gestaltung kurzfristig für die Leistung und langfristig für die Gesundheit eine Rolle. Der Bildschirm habe eine Diagonale nicht kleiner als 17 Zoll (43 cm), 19 Zoll ist bei der üblichen Arbeitsweise mit mehreren Fenstern auf dem Schirm kein Luxus. Weder vor noch hinter dem Benutzer sollen starke Lichtquellen leuchten. Das bedeutet auch, dass die Blickrichtung möglichst parallel zu einem etwaigen Fenster verläuft. Spiegelungen auf dem Schirm sind zu vermeiden. Der Blick auf die Bildschirm-Mitte soll leicht nach unten gehen. Die Unterarme sollen, wenn die Finger auf der Tastatur aufliegen, annähernd horizontal liegen. In Sekretariaten wird die deutsche Tasten-Anordnung (qwertz) bevorzugt, von Programmierern die amerikanische (qwerty). Benutzer, die sowohl mit Mäusen wie mit Rollkugeln (trackball) gearbeitet haben, entscheiden sich meist für letztere. Der Rechner sei leise, Lärm nervt, auch wenn man es nicht sofort merkt. Lüfter kann man so konstruieren, dass man sie praktisch nicht hört (großer Durchmesser, niedrige Drehzahl). Platten sind schwieriger zu bändigen. Hohe Drehzahlen (über 7200/min) sind am Arbeitsplatz vom Übel und nur in Servern abseits der Arbeitsplätze akzeptabel. Räume mit Druckern lüften. Geräte nicht an die Heizung stellen. Pausen machen, bewegen. Es hat auch sein Gutes, wenn der Drucker in einem anderen Stockwerk steht.



Abbildung 1.2. Typischer, aber nicht vorbildlicher Arbeitsplatz. Gerade werden alte Tonbänder auf CD übertragen.

1.2 Betriebssystem

In der frühen Kindheit der Rechner – vor 1950 – kannten die Maschinen kein Betriebssystem. Die damaligen Rechner waren jedoch trotz ihrer gewaltigen räumlichen Abmessungen logisch sehr einfach und übersichtlich aufgebaut. Die wenigen Benutzer kannten sozusagen jedes Bit persönlich. Beim Programmieren musste man sich auch um die Bits einzeln kümmern. Nach heutiger Sprechweise enthielt jedes Programm sein eigenes Betriebssystem.

Die Programmierer waren damals schon so arbeitsscheu (effektivitätsbewusst) wie heute und erkannten bald, dass dieses Vorgehen nicht zweckmäßig war. Viele Programmteile wiederholten sich in jeder Anwendung. Man fasste diese Teile auf einem besonderen Lochkartenstapel oder Lochstreifen zusammen, der als Vorspann zu jeder Anwendung eingelesen wurde. Der nächste Schritt war, den Vorspann nur noch nach dem Einschalten der Maschine einzulesen und im Speicher zu belassen. Damit war das Betriebssystem geboren und die Trennung von den einzelnen Anwendungen vollzogen.

Heutige Rechner sind äußerlich nicht mehr so beeindruckend, aber logisch um Größenordnungen komplexer. Dazu kommt die Vernetzung – lokal und weltweit. Damit ein Benutzer ohne Diplom in Informatik oder Elektrotechnik mit einem Rechner arbeiten kann, muss das Betriebssystem – die Grund-Software jedes Rechners – viele Einzelheiten verbergen. Neben koordinierenden und kontrollierenden Aufgaben obliegen dem Betriebssystem die Verbindung zur Hardware mittels der Treiber und die Verwaltung der Prozesse und Daten. Genau gesprochen werden diese

Kernaufgaben vom Kern (E: kernel, F: noyau) des Betriebssystems wahrgenommen. Unter Linux ist ebenso genau gesprochen nur der Linux-Kern zu verstehen. Deshalb reden wir von Debian GNU/Linux: die Debian-Distribution eines Linux-Kerns samt GNU-Software. Da der nackte Kern ziemlich hilflos ist, zählt man zum Betriebssystem im weiteren Sinn auch notwendige Konfigurationsdateien, einen Bootloader etc. FreeBSD, NetBSD, OpenBSD, MS Windows, Solaris, HP-UX, Mac OS X bezeichnen vollständige Betriebssysteme, nicht nur Kerne.



Abbildung 1.3. Das GNU-Logo, der Kopf eines sorgengeplagten, aber dennoch hoffnungsvollen Gnus, gezeichnet von ETIENNE SUVASA

Der Urahn aller Linux/UNIX-Betriebssysteme ist das Ende der sechziger Jahre des vorigen Jahrhunderts von KENNETH THOMPSON und DENNIS RITCHIE in den Bell Laboratories des US-amerikanischen Telefonkonzerns AT&T entwickelten UNIX. Von diesem Ur-UNIX sind viele Ideen und Konzepte in andere Betriebssysteme einschließlich DOS eingeflossen. Es würde zu weit führen, hier die Entwicklung, das Auseinanderdriften und Zusammenfinden der verschiedenen Zweige zu schildern². Der *Geschichte von Unix* ist ein eigener Artikel in der deutschen Wikipedia gewidmet. Wir gebrauchen die Bezeichnung *Linux/UNIX* für die Gattung aller UNIX-ähnlichen Betriebssysteme, die sowohl freie (GNU/Linux, FreeBSD, OpenBSD, NetBSD ...) wie kommerzielle Systeme (HP-UX, AIX, Sinix, Solaris, Ultrix ...) umfasst. Viele Aussagen im Buch treffen auf die Gattung zu, einige dagegen nur auf eine bestimmte Version (z. B. *sarge*) von Debian GNU/Linux.

POSIX ist kein Betriebssystem, sondern ein umfangreicher Standard, der die Anforderungen an POSIX-konforme Betriebssysteme festlegt (<http://www.opengroup.org/>). Debian GNU/Linux entwickelt sich auf die POSIX-Vorgaben zu. Ein ähnliches Ziel verfolgt *The Single UNIX Specification* (<http://www.unix.org/online.html>) und speziell in der Linux-Welt die *Linux Standard Base* (<http://www.linuxbase.org/>).

Die Gnus (*Connochaetes*) sind eine Antilopenart in Süd- und Ostafrika, von den Buren Wildebeest genannt. Nach ALFRED BREHM sind es höchst absonderliche, gesellig lebende Tiere, in deren Wesen etwas Komisches, Feuriges, Überspanntes steckt. Das GNU-Projekt der Free Software Foundation ([---

²Eine authentische Zusammenfassung der Anfänge findet sich in *The Bell System Technical Journal* Vol. 57, July-August 1978, Nr. 6, Part 2, Seite 1897 bis 2312](http://www.</p></div><div data-bbox=)

fsf.org/) wurde 1984 von RICHARD MATTHEW STALLMAN (RMS) ins Leben gerufen mit dem Ziel, Software ohne Einschränkungen vor allem juristischer Art aller Welt zur Verfügung zu stellen. Das Projekt hat sich den Kopf eines Gnus zum Erkennungszeichen gewählt, gezeichnet von ETIENNE SUVASA, siehe Abbildung 1.3. Linux geht auf LINUS BENEDICT TORVALDS zurück, der 1991 als Student in Helsinki mit dem Intel-Prozessor 80386 herumspielte. Der bekannte gut genährte Pinguin namens TUX – siehe Abbildung 1.4 – wurde 1996 von LARRY EWING gezeichnet (<http://www.isc.tamu.edu/~lewing/linux/>). Auch <http://sjbaker.org/tux/> enthält Material zu dem Vogel.



Abbildung 1.4. Tux, der wohlgenährte Pinguin, das Linux-Logo, gezeichnet von L. EWING

Ein Benutzer könnte sich den Linux-Kern von <http://www.kernel.org/> und weitere Software vom GNU-Projekt (<http://www.gnu.org/>) sowie aus anderen Quellen herunterladen und daraus ein arbeitsfähiges Betriebssystem samt Anwendungen zusammenbasteln. Das ist sehr viel Arbeit, nicht trivial und außerdem mit dem Zwang verbunden, sich in kurzen Zeitabständen um neue Versionen der Software zu kümmern. Deshalb übernehmen Firmen oder Organisationen die Arbeit und stellen Distributionen zusammen, mit denen vergleichsweise einfach ein lauffähiges System auf einem Rechner eingerichtet werden kann. Weltweit sind einige hundert Distributionen entstanden und teilweise wieder vergangen. Das Debian-Projekt – im Jahr 1993 von IAN MURDOCK ins Leben gerufen – ist eine solche (lockere) Organisation oder Gemeinschaft von freiwilligen, unbezahlten und unbezahlbaren Mitarbeitern. Die Debian-Distribution weist unserer Meinung nach einige Vorzüge auf; sie werden bei GANTEN + ALEX ausführlich erläutert. Als Erkennungszeichen verwendet Debian den in Abbildung 1.5 dargestellten Swirl in verschiedenen Formen, einen Flaschengeist. Das englische Wort *swirl* bedeutet Wirbel, Spirale. Auf Debian bauen weitere Distributionen wie Knoppix (<http://www.knopper.net/>), Ubuntu (<http://ubuntulinux.org/>) und UserLinux (<http://www.userlinux.com/>) auf. Sie wenden sich an bestimmte Benutzerkreise und konfigurieren die Pakete zum Teil vor.

Der Erfolg von Linux, dem GNU-Projekt und anderer freier oder offener Software wie dem X Window System beeinträchtigt die Gewinnaussichten einiger Firmen oder ganzer Industriezweige. Die Bestrebungen, durch Softwarepatente, Ausweitungen des Urheberrechtes, Kriminalisierung von Kryptanalysen und Androhung kostspieliger juristischer Schritte gegen Privatpersonen oder kleine Firmen die weitere Ausbreitung freier Software zu verhindern und die Macht etablierter Firmen zu fes-

tigen, verlangen im Umgang mit Software heute eine gewisse Vorsicht. Manchen Regierungen kommt die Angst ihrer Untertanen vor Terroristen gelegen, um Informationen über alles und jeden zu sammeln. In einem bekannten deutschen Magazin für Computertechnik sind mittlerweile in jeder Ausgabe mehrere Seiten juristischen Fragen gewidmet.



Abbildung 1.5. Das Debian-Logo, der dunkelrote Swirl

Hardware plus Betriebssystem machen den Rechner aus, wobei das Betriebssystem innerhalb der durch die Hardware gezogenen Grenzen die Eigenschaften des Rechners bestimmt. PC-Hardware mit einem DOS-Betriebssystem ergibt einen DOS-Rechner. Dieselbe Hardware mit einem Linux/UNIX-Betriebssystem ergibt einen Linux/UNIX-Rechner mit erheblich erweiterten Fähigkeiten. Beispielsweise kennt ein DOS-Rechner keine verschiedenen Benutzer; wer davor sitzt, ist der Benutzer und darf alles. Linux/UNIX-Betriebssysteme hingegen sind immer Mehrbenutzersysteme. Ein Benutzer muss sich mit Benutzernamen und Passwort anmelden und darf – mit Ausnahme des Verwalters (root) – bei weitem nicht alles. Nur so können mehrere Benutzer auf einer Maschine arbeiten, ohne sich ins Gehege zu kommen, und das sogar gleichzeitig.

Zu einem Zeitpunkt kann auf einem Rechner nur ein Betriebssystem laufen. Will ich dieses wechseln – beispielsweise von Debian GNU/Linux auf MacOS – muss ich den Betrieb beenden, den Rechner herunterfahren und das neue System laden³. Anwendungsprogramme (E: application, F: application) dagegen werden im laufenden Betrieb gestartet und beendet, ohne dass der Rechner heruntergefahren wird. Üblicherweise werden auch mehrere Anwendungen gleichzeitig auf einem Rechner ausgeführt. Beendet ein Benutzer seine Sitzung, laufen alle nicht zur Sitzung gehörenden Prozesse auf dem Rechner weiter.

Auf einem Linux/UNIX-System sind beim Start und ebenso bei Feierabend drei Vorgänge zu unterscheiden:

- Strom einschalten (den Big Red Switch umlegen),
- Betriebssystem starten (booten, E: to boot, F: amorcer),
- Sitzung eröffnen (anmelden, einloggen),

³Es gibt Über-Betriebssysteme, die auf einer Hardware ausreichender Leistung mehrere virtuelle Rechner mit jeweils eigenem Betriebssystem einrichten. Suchbegriffe *User Mode Linux*, *VMware*, *Virtual Server*, *IBM Sysplex*, *Xen* (<http://www.cl.cam.ac.uk/Research/SRG/netos/xen/> und <http://xen.sf.net/>).

- +++ Arbeiten +++
- Sitzung beenden (abmelden, ausloggen),
- Rechner herunterfahren (shutdown),
- Strom ausschalten.

Die ersten und letzten beiden Punkte sind Aufgabe des Verwalters, die Sitzung ist Sache des jeweiligen Benutzers. Zieht man mitten in einer Sitzung den Stecker oder bleibt der Strom aus anderen Gründen weg, können einige Daten verloren gehen. Ein größerer Schaden sollte nicht entstehen, aber für die Gesundheit des Systems ist ein solch brutales Ende nie gut. Ein Kaltstart umfasst die ersten beiden Punkte, ein Warmstart (reboot) besteht in einem erneuten Start des Betriebssystems bei eingeschalteter Maschine. Ein sanfter Warmstart wird durch das Kommando `reboot` oder – sofern konfiguriert, siehe `/etc/inittab` – durch die Tastenkombination `<ctrl>-<alt>-` ausgelöst. Einen brutalen Warmstart ohne Sichern von Puffer-Inhalten erzwingt man durch Drücken des Reset-Knopfes am PC-Gehäuse. Wenn auch das nichts hilft (vielleicht ist der Knopf nicht angeschlossen), kann man nur noch das Stromkabel ziehen. Verfügt der Rechner über eine eingebaute Stromversorgung (USV, UPS), wehrt er sich eine Zeit lang auch dagegen.

1.3 Benutzeroberflächen

1.3.1 Kommandozeile, curses-Fenster

Unter einer Benutzer-Oberfläche (E: user interface, F: interface utilisateur) versteht man nicht die Haut, aus der man nicht heraus kann, sondern die Art, wie sich ein Terminal (Bildschirm + Tastatur + Maus oder Rollkugel) dem Benutzer darstellt, wie es aussieht (look) und wie es auf Eingaben antwortet (feel). Im einfachsten Fall besteht die Oberfläche aus einem Textterminal, oft auch Konsole genannt. Wenn Sie mit einer grafischen Benutzer-Oberfläche arbeiten, müssen Sie ein Fenster öffnen, das `xterm`, `Konsole`, `Terminal`, `Befehlsfenster`, `X-Shell` oder ähnlich heißt, um an eine Konsole zu gelangen. Linux bietet aber auch die Möglichkeit, mittels der Tastenkombination `<ctrl>-<alt>-<F1>` (unter X11, bis `<F6>`) eine virtuelle Konsole auf den Schirm zu bringen. Die Eingabe von `<alt>-<F7>` oder `<ctrl>-<alt>-<F7>` führt zur grafischen Oberfläche von X11 zurück.

Nach Anmeldung mit Benutzername (E: user name, F: nom de utilisateur) und Passwort (E: password, F: mot de passe) erscheint auf der Konsole der Systemprompt (F: invite), meist ein Dollarzeichen und rechts daneben ein blinkender Cursor. Damit teilt das System mit, dass es auf Ihre Eingaben wartet. Im einfachsten Fall tippt man seine Kommandos zeilenweise ein und schickt sie jeweils mit der Taste `<cr>` ab. Die Ausgabe des Systems erfolgt auf den Bildschirm, als Text Zeile für Zeile nacheinander. Die Grundform einer Linux/UNIX-Kommandozeile (E: command line, F: ligne de commande) sieht so aus:

```
joe@debian:~$ kommando -option argument
```

Leerzeichen trennen die Teile voneinander, dürfen also nicht innerhalb eines Kommando-Namens, einer Option oder eines Argumentes vorkommen. Optionen können, müssen aber nicht gesetzt werden. Sie modifizieren das Verhalten des Kommandos. Argumente sind Dateinamen oder dergleichen. Beendet und abgeschickt wird das Kommando mit der Taste <cr>, genauer gesagt mit einem Steuerzeichen (control operator) der Shell. Diese Art der Eingabe heißt Kommandozeile. Sie stellt die geringsten Anforderungen an Hard- und Software und ist mit Einschränkungen sogar auf druckenden Terminals (ohne Bildschirm) möglich. Für Vorleseprogramme (Screenreader) blinder Benutzer ist sie die einfachste Grundlage. Früher war die Kommandozeile die einzige Möglichkeit, eine Anwendung im Dialog mit dem Rechner zu starten. Statt Anwendung sagt man auch oft Werkzeug (E: tool, F: outil).

Vom Benutzer verlangt die Kommandozeile die Kenntnis der einzugebenden Kommandos und das zielsichere Landen auf den richtigen Tasten. Die Kommandos bieten kurze Hilfen an, die üblicherweise durch die Tasten <h> (help) oder <?> aufgerufen werden. Die Kommandozeile ist auch heute noch in Gebrauch und der einzige Weg, auf dem man jedes Kommando mit jeder Option und jedem Argument erreicht. Hinter den vielen Knöpfen und Menüs einer grafischen Benutzeroberfläche stecken oft Kommandozeilen.

Bei Linux/UNIX-Kommandos ist es eine gute Gepflogenheit, dass sie fehlerhaft aufgerufen einen Hinweis zum richtigen Gebrauch (E: usage, F: usage) geben. Probieren Sie folgende Eingaben aus, auch mit anderen Kommandos:

```
joe@debian:~$ who -x
```

```
joe@debian:~$ who -?
```

```
joe@debian:~$ who --help
```

Bei falschen Optionen oder fehlenden Argumenten reicht oft der Hinweis aus. Die Option mit zwei Minuszeichen davor stellt die Langform einer Option dar, unter Linux gebräuchlich, aber nicht überall verfügbar. Mehr erfährt man auf der zum Kommando gehörenden Manualseite, siehe Abschnitt 1.5 *Hilfen* auf Seite 36.

Ein erster Schritt in Richtung Benutzerfreundlichkeit ist die Verwendung von Menüs. Die erlaubten Eingaben werden in Form einer Liste – einem Menü – angeboten, der Benutzer wählt durch Eintippen eines Zeichens oder durch entsprechende Positionierung des Cursors die gewünschte Eingabe aus. Menüs haben zwei Vorteile. Der Benutzer sieht, was erlaubt ist, und macht bei der Eingabe kaum syntaktische Fehler. Nachteilig ist die beschränkte Größe der Menüs. Man kann nicht mehrere hundert Linux/UNIX-Kommandos mit jeweils mehreren Optionen in ein Menü packen. Ein Ausweg – in Grenzen – sind Menü-Hierarchien, die auf höchstens drei Ebenen begrenzt werden sollen, um übersichtlich zu bleiben. Einfache Menüs ohne Grafik und Mausunterstützung stellen ebenfalls nur geringe Anforderungen an Hard- und Software.

Für den ungeübten Benutzer sind Menüs eine Hilfe, für den geübten ein Hindernis und eine Bevormundung. Zu den am häufigsten ausgewählten Punkten müssen

kurze Wege führen. Es können Defaults⁴ vorgegeben sein, die durch Eingabe von `<cr>` ohne weitere Zeichen übernommen werden. Tief in Menühierarchien verbuddelte Anweisungen lassen sich teilweise unmittelbar durch Tastenkombinationen wie `<ctrl>-<c>` erreichen. Solche abgekürzten Wege werden Shortcuts genannt.

Bildschirme lassen sich in mehrere Ausschnitte aufteilen, die Fenster (E: window, F: fenêtre) genannt werden. In der oberen Bildschirmhälfte beispielsweise könnte man bei einem Dialog zwischen zwei Benutzern mittels `talk` den eigenen Text darstellen, in der unteren die Antworten des Gegenübers. Bei der Einrichtung von Debian GNU/Linux wird von solchen einfachen alphanumerischen Fenstern mehrmals Gebrauch gemacht. An die Hardware werden keine besonderen Anforderungen gestellt, ein alphanumerischer Bildschirm mit der Möglichkeit der Cursor-Positionierung mittels Pfeiltasten, Maus oder Rollkugel reicht aus. Einige Linux/UNIX-Anwendungen benutzen ebenfalls diese schmucklosen Textfenster und verlangen daher nicht als Grundlage das X Window System. Nach der bei der Programmierung solcher Anwendungen verwendeten Funktionsbibliothek (`curses` oder `ncurses`) werden solche Fenster als `curses`-Fenster bezeichnet. Die `curses`-Funktionen müssen den Terminaltyp (`xterm`, `hpterm`, `vt52`, `ansi`) kennen und werten zu diesem Behuf die Umgebungsvariable `TERM` aus. Ist sie falsch gesetzt, zeigt sich das Terminal widerspenstig.

1.3.2 X Window System (X11)

Was ist das X Window System (X11)?

Das X Window System (nicht *Windows*, das ist eine andere Baustelle) ist ein

- quelloffenes,
- grafisches,
- von der Hardware unabhängiges,
- netzfähiges

Fenstersystem, das um 1984 am Massachusetts Institute of Technology (MIT) im Athena-Projekt entwickelt wurde und frei verfügbar ist. Im Jahr 1987 wurde die Version 11 Release 1 veröffentlicht. Weitere korrekte Bezeichnungen sind X Version 11, X11 oder X, gegenwärtig im sechsten Release auch X11R6. Heute wird es von der X.Org Foundation (<http://www.x.org/>) betreut. Unter X11, so wie man es dort beziehen kann, hat man sich Funktionsbibliotheken und einige Musteranwendungen vorzustellen. Falls auf Ihrem Rechner X11 eingerichtet ist, heißt das, dass ein X-Server und X-Bibliotheken bei Ihnen gespeichert sind und Programme davon Gebrauch machen können. Auf Arbeitsplätzen unter Linux/UNIX ist X11 heute die Regel. Beim Linux Documentation Project liegen ein *X Window Overview HOWTO* und ein *X Window User HOWTO*.

Im Xerox Palo Alto Research Center wurde die Verwendung von alphanumerischen Menüs und Fenstern weiterentwickelt zu einer grafischen Benutzeroberfläche

⁴Ein Default oder Defaultwert ist ein vom System oder Programm vorgeschlagener Wert, der verwendet wird, wenn nichts anderes verlangt wird.

(E: graphical user interface, GUI, F: interface utilisateur graphique), welche die Arbeitsweise des Benutzers wesentlich bestimmt. Viele jüngere Benutzer kennen nichts anderes. Grafisch bedeutet, dass auf dem Bildschirm nicht nur die Zeichen eines Alphabets, sondern beliebige Figuren dargestellt werden können; im Wesentlichen ist das eine Frage der Elektronik hinter dem Bildschirm. Diese grafische Fenstertechnik ist von Programmen wie SMALLTALK, von MS Windows sowie von Computerherstellern wie Apple übernommen und verbreitet worden. Unter Linux/UNIX bildet X11 die Grundlage für alle grafischen Benutzeroberflächen.

Von der Hardware unabhängig bedeutet, dass X11 nicht an eine Prozessor-Familie oder einen Hersteller gebunden, sondern im Quellcode veröffentlicht ist. In den Anfangsjahren von X11 war erschwingliche Hardware nicht leistungsfähig genug, sodass sich das System nur langsam verbreitete. Mit den heutigen Prozessoren und Speicherkapazitäten sind die Ansprüche von X11 jedoch leicht zu erfüllen. Nur wer einen alten PC mit einem Prozessor-Takt von 100 MHz und einem Arbeitsspeicher von 32 MB unter Debian GNU/Linux wiederbeleben will, sollte auf X11 und alles, was darauf aufsetzt, verzichten. Typischerweise laufen Server im Netz ohne X11, weil sie keinen Bedarf dafür haben.

Die Netzfähigkeit zeichnet X11 vor allen anderen grafischen Fenstersystemen auf dem Markt aus. Netzfähig bedeutet, dass die Rechnungen (die Client-Prozesse) auf einer Maschine im Netz laufen können, während die Terminal-Ein- und -Ausgabe (der Server-Prozess) über eine andere Maschine im Netz erfolgen (Client-Server-Modell). Die gesamte Anwendung kann auf zwei Maschinen – sogar unterschiedlicher Hersteller – verteilt werden, siehe Abbildung 1.6. X11-Client und X11-Server verständigen sich mit Hilfe des X11-Protokolls (das nichts mit dem Internet und seinen Protokollen zu tun hat).

Ein Client ist ein Prozess, der Dienste verlangt, ein Server (F: serveur) ein Prozess, der Dienste leistet. Die Trennung einer Aufgabe in einen Client- und einen Server-Teil erhöht die Flexibilität und ermöglicht das Arbeiten über Netz. In der Linux-Welt ist der freie X-Server XFree86 (<http://xfree86.berlios.de/>) verbreitet. Weitere X-Server – meist nur für bestimmte Grafikkarten – findet man in den Debian-Paketlisten, indem man nach `xserver` sucht. Ein X-Server wird ergänzt durch einen X-Printer, der die Ausgabe für einen Drucker anstelle des Bildschirms vorbereitet. Möglicherweise wird die X-Printer-Funktionalität künftig in den X-Server integriert.

Man muss sich darüber klar sein, dass die X11-Daten ohne zusätzliche Maßnahmen unverschlüsselt über das Netz gehen und abgehört oder verfälscht werden können. X11 enthält nur minimale Sicherheitsvorkehrungen. Bestehen diesbezügliche Bedenken, ist der Netzverkehr zu verschlüsseln, beispielsweise über SSH, siehe Abschnitt 3.6 *Secure Shell* auf Seite 110.

Wer tiefer in X11 eindringen möchte, beginnt am besten mit man X und geht dann zu <http://x.org/> und <http://www.camb.opengroup.org/tech/desktop/x/> ins Web. Es gibt auch HOWTOs zu X11 und XFree86 im Verzeichnis `/usr/share/doc/HOWTO/`. Da an einer vollständigen Anwendung unter X11 mehrere Softwareschichten (Bibliotheken) beteiligt sind, ist es manchmal schwierig herauszufinden, wo eine Einstellung oder ein Fehler zu suchen ist. Die

Komplexität von X11 erreicht die eines Betriebssystems; man braucht einige Zeit, bis man das Meiste versteht. Zum Glück kommt ein anspruchsloser Benutzer ziemlich weit, ohne etwas von X11 zu wissen.

Start von X11

Gestartet wird der X-Server von Hand aus der Kommandozeile mit dem Kommando `startx` (dahinter verbirgt sich `xinit`) oder automatisch beim Systemstart durch einen Desktop- oder Display-Manager (Login-Manager) wie `gdm`, `kdm`, `wdm` oder `xdm`. Die Auswahl wird vom Verwalter getroffen und in `/etc/X11/default-display-manager` festgelegt. Ein Display-Manager schreibt den Login-Bildschirm (siehe Abbildung 1.18 auf Seite 43), nimmt Benutzernamen und Passwort entgegen und startet bei Erfolg einen Fenster-Manager (Window Manager). Bei diesen hat der Benutzer eine Auswahl.

X11 stellt die Funktionen bereit, um grafische Benutzeroberflächen zu gestalten, legt aber das Aussehen und das Verhalten der Oberfläche nur in Grundzügen fest. Die Einzelheiten der Oberfläche sind Sache besonderer Funktionsbibliotheken wie Motif bzw. bestimmter Programme, der Fenster-Manager, die nicht immer Bestandteil von X11 sind und teilweise auch Geld kosten. Der in X11 enthaltene Fenster-Manager ist der Tab-Window-Manager `twm`, Hewlett-Packard fügt seinen Systemen den Motif-Fenster-Manager `mwm` bei, unter Linux/UNIX findet sich der Win95-Fenster-Manager `fvwm` (F* Virtual Window Manager, <http://www.fvwm.org/>), dessen Fenster an MS Windows 95 erinnern. Das K Desktop Environment (KDE) bringt den `kwm` mit. Aus dem GNU-Projekt stammt der schlanke und stabile WindowMaker, enthalten im Debian-Paket `wmaker` (<http://www.windowmaker.org/>). Der WindowMaker kennt das Dock, eine Leiste von Icons, hinter denen sich kleine Programme, die Dock Applications (Dockapps), verbergen, die im Hintergrund laufen. Fluxbox ist ein weiterer Fenster-Manager mit einem Dock. Ein mächtiger, anpassungsfähiger Fenster-Manager ist Enlightenment (<http://www.enlightenment.org/>). Es gibt zahlreiche (zu viele) weitere Fenster-Manager, siehe die Debian-Paketlisten, Abteilung X Window System Software. Ein Fenster-Manager ist in der Regel der erste X-Client einer Sitzung und bestimmt zu einem wesentlichen Teil das Aussehen und Verhalten (look and feel) von Bildschirm, Tastatur und Maus bzw. Rollkugel.

Desktop Environments

Was der Fenster-Manager an Gestaltungsmöglichkeiten offen lässt, regelt das Desktop Environment, kurz Desktop genannt. Praktisch arbeitet man nicht ohne einen Desktop. Auch hier stehen unter Linux/UNIX mehrere zur Auswahl:

- das kommerzielle Common Desktop Environment (CDE, <http://www.opengroup.org/cde/>), das auf OSF/Motif aufsetzt,
- das freie GNU Network Object Model Environment (GNOME, <http://www.gnome.org/>, <http://www.gnome.de/>),

- das freie K Desktop Environment (KDE, <http://www.kde.org/>, <http://www.kde.org.uk/>, <http://www.kde.org/fr/>, <http://www.kde-france.org/> und weitere),
- GNUstep, ein Desktop und eine Entwicklungsumgebung mit Wurzeln in NeXT/Apple und der OpenStep-Spezifikation, <http://www.gnustep.org>,
- das Cholesterol Free Desktop Environment (XFce, <http://www.xfce.org/>).

Das Common Desktop Environment entstand aus gemeinsamer Arbeit der Firmen Hewlett-Packard, IBM, Novell und Sun. GNUstep zeigt infolge seiner Herkunft Anklänge an Mac OS X und Cocoa. Es bringt keinen eigenen Fenster-Manager mit, sondern empfiehlt den WindowMaker, siehe oben. XFce versucht, weniger gewichtig als die anderen Desktops zu sein. Die beiden für uns bedeutendsten Desktops KDE und GNOME unterscheiden sich in vielerlei Hinsicht, arbeiten aber zusammen, soweit das möglich ist. Sie gründen sich auf verschiedene höhere Funktionsbibliotheken: GNOME setzt auf GTK+ (<http://www.gtk.org/>, entstanden als GIMP Toolkit), KDE auf Qt von Trolltech (<http://www.trolltech.com/>) in Norwegen. Es bestehen Unterschiede in der Lizenzierung, die aber den Anwender nicht betreffen. Beide Environments lassen sich so konfigurieren, dass ein Benutzer kaum einen Unterschied sieht. Trotzdem hat man es einfacher, wenn man Fenster-Manager, Desktop und desktop-bezogene Anwendungen im wesentlichen nur aus einer der beiden Welten bezieht.

Die Trennung der Aufgaben zwischen Fenster-Manager, Desktop und Anwendungen wie einem Datei-Manager ist uneinheitlich, konfigurierbar und daher unübersichtlich. Ein Benutzer kann am Anfang nicht viel mehr tun, als sich an seine Einstellungen zu gewöhnen und sie Schritt für Schritt zu optimieren. Für viele Benutzer reicht das aus.

X11 über ein Netz

Zum Arbeiten mit X11 über ein Netz muss der Benutzer vor allem zwei Kommandos kennen. Auf der Maschine, vor der er sitzt (wo seine Sitzung läuft, der X-Server), gibt er mit:

```
joe@debian:~$ xhost abcd
```

der fernen Maschine namens `abcd` (wo seine Anwendung läuft, der X-Client) die Erlaubnis zum Zugriff. Das Kommando ohne Argument zeigt die augenblicklichen Einstellungen an. Die Antwort auf `xhost` sollte beginnen mit *Access control enabled*, andernfalls wäre es angebracht, mit dem Verwalter über die Sicherheit von X11 zu diskutieren. Auf der fernen Maschine `abcd` setzt man mit

```
joe@debian:~$ export DISPLAY=efgh:0.0
```

die Umgebungsvariable `DISPLAY` auf den Maschinennamen `efgh` und die Fensternummer `0.0` des X-Servers. Erst dann kann ein Client-Programm, eine Anwendung

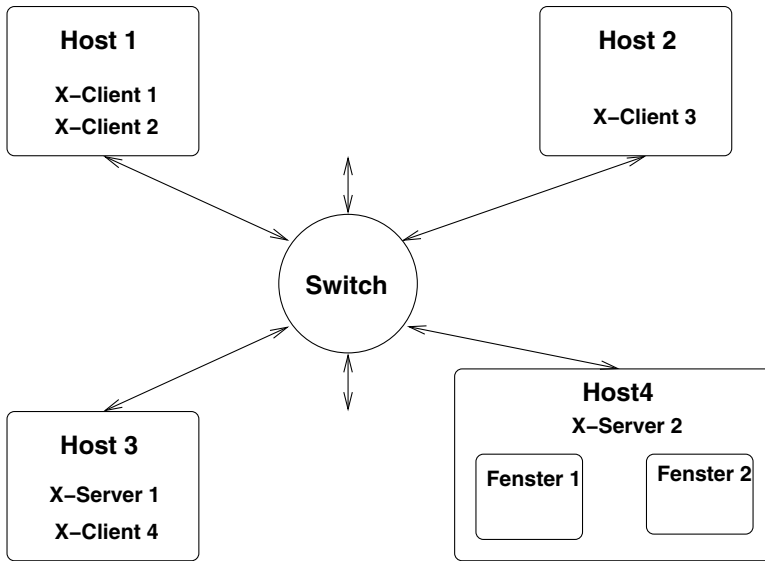


Abbildung 1.6. X-Window-Clients und -Server, durch einen Ethernet-Switch verbunden. Vier X-Clients wickeln ihre Ein/Ausgabe über zwei X-Server ab.

über das Netz den X-Server zur Ein- und Ausgabe nutzen. Die Fensternummer besteht aus Displaynummer und Screennummer und hat nur auf Maschinen mit mehreren Terminals auch Werte größer null.

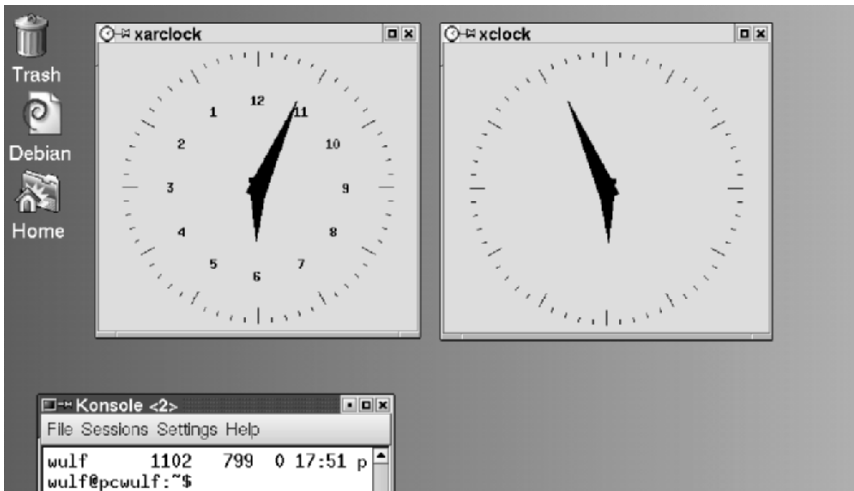


Abbildung 1.7. Screenshot (Ausschnitt) zweier Uhren, eine läuft lokal, die andere als X-Client auf einem entfernten Rechner. Leider sind bei der Netzkabelung zwei Adern vertauscht, sodass die entfernte Uhr links herum läuft. :-)

Abgesehen davon, dass die Daten unverschlüsselt über das Netz gehen und mitgelesen werden können, bestehen weitere Sicherheitslücken bei X11, die nur teilweise durch das Kommando `xhost` geschlossen werden. Einen Schritt weiter geht die Xauthority mit dem Kommando `xauth`, die zwischen Client und Server eine Art von Schlüssel (MIT Magic Cookie) austauscht. Mittels des Kommandos:

```
joe@debian:~$ xauth list
```

kann man sich die Schlüssel in der Datei `$HOME/.Xauthority` ansehen, mittels der Pipe:

```
joe@debian:~$ xauth extract - 'hostname':0.0 | rexec
clienthost xauth merge -
```

wird ein Schlüssel zur fernen Maschine `clienthost` geschickt, möglicherweise über einen noch ungeschützten Kanal. Statt `rexec` kann es auch `rsh` oder `remsh` heißen. Bei Erfolg tauscht der X11-Server nur noch Daten mit dem betreffenden X11-Client aus. Das Kommando `xhost` erübrigt sich dann, die `DISPLAY`-Variable ist beim Client nach wie vor zu setzen.

Die Secure Shell (SSH) erledigt sämtliche Schritte zum Aufbau einer sicheren Verbindung zwischen Client und Server automatisch und ist damit der einfachste Weg. In der Konfiguration der SSH (`/etc/ssh/`) ist das Weiterreichen (forwarding) von X11-Daten ausdrücklich einzuschalten. Dann kann ein Benutzer mit einem Kommando wie:

```
joe@debian:~$ ssh -f rechner2 xclock
```

auf dem entfernten Rechner namens `rechner2` einen X-Client namens `xclock` starten, der seine Ein- und Ausgabe über den lokalen X-Server tätigt, wie in Abbildung 1.7, wo die eine Uhr lokal läuft und die andere entfernt, leicht festzustellen mittels `ps`. Das Beispiel ist nicht umwerfend nützlich, aber leicht nachzuvollziehen.

Tastaturen unter X11

In Sekretariaten und im privaten Bereich bevorzugen die meisten Benutzer die deutsche Tastenanordnung (qwertz) auf der Tastatur. Programmierer dagegen nehmen lieber eine internationale (US-amerikanische) Tastatur (qwerty). In Frankreich werden auch azerty-Tastaturen verwendet, und schließlich gibt es vereinzelt die ergonomisch günstigeren DVORAK-Tastaturen. Im Handel sind auch Tastaturen mit doppelter Beschriftung erhältlich, siehe Abbildung 1.8, die einen Ausschnitt aus einer deutsch-arabischen Tastatur zeigt. Der Rechner weiß nichts von der Beschriftung der Tasten; er empfängt Scan-Codes und übersetzt sie mit Hilfe von konfigurierbaren Tabellen (keymap) in Zeichen. Muss man zwischen verschiedenen Tastenanordnungen hin- und herschalten, bieten sich mehrere Wege an. Arbeitet ein Benutzer ohne X11 auf einer Textkonsole, so lädt der Aufruf:

```
joe@debian:~$ loadkeys de-latin1
```

eine deutsche Keymap mit Umlauten etc. Umgekehrt lädt der Aufruf:

```
joe@debian:~$ loadkeys us
```

eine us-amerikanische Keymap und der Aufruf:

```
joe@debian:~$ loadkeys --default
```

die bei der Einrichtung des Systems festgelegte Keymap. Das Kommando funktioniert auch auf virtuellen Textkonsolen, wenn auf der Hauptkonsole (Nr. 7) X11 läuft, und beeinflusst diese nicht. Die Keymaps samt ihren Bezeichnungen finden sich im Verzeichnis `/usr/share/keymaps/i386/`. Die Tastatureinstellung ist unabhängig vom Locale, das in Abschnitt 1.3.6 *Lokalisierung* auf Seite 30 erläutert wird. Deshalb kann man mit amerikanischer Tastatur, aber deutschem Locale arbeiten, was beim Schreiben von Manuskripten oder Programmen manchmal sinnvoll ist.



Abbildung 1.8. Ausschnitt aus der linken oberen Ecke einer deutsch-arabischen Tastatur. Die arabischen Zeichen sind rotbraun gefärbt, die deutschen grau (Fujitsu-Siemens).

X11 kümmert sich selbst um die Übersetzung der Tastatur-Codes (scan-code) in Zeichen, die Tastatur wird im raw mode betrieben statt wie sonst üblich im xlate mode. Infolgedessen muss X11 umkonfiguriert werden. Dies erledigt ein Benutzer innerhalb seiner Sitzung mit dem Kommando (einzeilig):

```
joe@debian:~$ setxkbmap -rules xfree86 -model pc102
\ -layout "us,de,dvorak" -option
"grp:alt_shift_toggle"
```

Bei Erfolg schaltet anschließend die Tastenkombination `<alt>-<shift>` zwischen den nach `layout` genannten Tastaturen hin- und her. Die Kürzel für die Sprachen sind in `/usr/X11R6/lib/X11/xkb/symbols/` zu finden. Eine verkürzte Form des Kommandos ist (siehe Manual, einzellig):

```
joe@debian:~$ setxkbmap us,de,fr basic
\grp:alt_shift_toggle
```

Die noch kürzere Form:

```
joe@debian:~$ setxkbmap de
```

kann auch funktionieren, wenn es nur darum geht, vorübergehend die Tastenbelegung umzuschalten. Am einfachsten mit `de` und `us` ausprobieren und sich die Wirkung der `y`-Taste anschauen. Die Umschaltung gilt für den X-Server und damit für alle Desktops einer Sitzung, nicht aber für virtuelle Textkonsolen (Nr. 1 bis 6). Ähnliche Aufgaben erfüllen die beiden Werkzeuge `xkbsel` und `xkeysw`. Mit letzterem lässt sich jedem Fenster eine eigene Tastenbelegung zuordnen. Vorsicht mit exotischen Tastenbelegungen, man kommt möglicherweise nur noch durch Ausloggen zu seiner alten Belegung zurück. Mittels:

```
joe@debian:~$ xmodmap -pke | less
```

sieht man sich seine Keymap an. Der Verwalter hat weitere Möglichkeiten, die Keymaps zu gestalten. Es lassen sich auch vorgegebene Keymaps modifizieren. Wer auf einer deutschen Tastatur das Yen-Zeichen benötigt, kann sich mit `xmodmap` eine eigene Tastenbelegung basteln. Das *Keyboard and Console HOWTO* ist eine hilfreiche Lektüre, ebenso das *International Keyboard Xmodmap HOWTO*. Man ahnt gar nicht, wie verschieden Tastaturen innerlich sind.

Bildschirme unter X11

In der Datei `/etc/X11/XFree86-4` sind in der Sektion *Screen* die Bildschirmauflösungen und Farbtiefen genannt, die der X-Server anbieten soll. Auch die Darstellung nicht-lateinischer Alphabete (Griechisch, Kyrillisch, Arabisch, Hebräisch) auf der Textkonsole oder unter X11 ist möglich. Das *Unicode HOWTO* gibt Hinweise. Der erste Schritt besteht immer darin, sich passende Fonts zu beschaffen. Dabei ist die *Finding Fonts for Internationalization FAQ* behilflich. Die Eingabe:

```
joe@debian:~$ xlsfonts | less
```

listet die auf einem System verfügbaren X11-Fonts auf, einschließlich der Kurznamen (die vollständige Bezeichnung eines X11-Fonts ist länglich). Wollen wir wissen, ob kyrillische Fonts eingerichtet sind, hilft eine Pipe:

```
joe@debian:~$ xlsfonts | grep koi8
```

Mittels:

```
joe@debian:~$ xfd -fn fontname
```


erfährt man den zugehörigen Zeichensatz. Beispielsweise ist:

```
joe@debian:~$ xfd -fn koi8x16
```

ein erster Schritt in Richtung Osten und

```
joe@debian:~$ xfd -fn kanji16
```

ein zweiter.

Fenster, Buttons und Menüs

Ehe ein Fenster bzw. der mit ihm verbundene Prozess Eingaben annimmt, muss es durch Anklicken (E: to click, F: cliquer) eines beliebigen Teils mit der Select-Maustaste (links, Linksklick) aktiviert oder selektiert werden. Dabei ändert sich die Rahmenfarbe. Gibt man nun auf der Tastatur Zeichen ein, erscheinen sie im Fenster und gelangen zum Rechner. Man sagt auch, das Fenster habe den Fokus, oder besser, die Eingabe von der Tastatur sei auf das Fenster fokussiert. Es ist immer nur ein Fenster aktiv. Ein Fenster wird deaktiviert, wenn ein anderes Fenster aktiviert wird oder der Mauscursor das aktive Fenster verlässt (konfigurierbar). Der ungefährliche Aufruf:

```
joe@debian:~$ xwininfo
```

liefert einige Informationen zu einem Fenster, ebenso:

```
joe@debian:~$ xdpinfo | less
```

zu einem Display. Das ähnliche Kommando `xprop` zeigt viele Informationen an, die teilweise schwer verständlich sind.

Ein Fenster wird auf dem Bildschirm verschoben, indem man seine Kopfleiste mit der Select-Maustaste in die neue Position zieht. Nach Loslassen der Taste verharrt das Fenster an der neuen Stelle. Die Größe eines Fenster wird durch Ziehen einer Rahmenseite verändert. Zieht man eine Ecke, ändern sich die beiden angrenzenden Seiten gleichzeitig.

Gelegentlich möchte man ein Fenster vorübergehend beiseite legen, ohne es jedoch ganz zu löschen, weil mit ihm noch ein laufender Prozess verbunden ist. In diesem Fall klickt man mit der Select-Maustaste den Minimize-Knopf oder -Button an, und das Fenster verwandelt sich in ein Sinnbild, Symbol oder Icon. Das ist ein Rechteck von Briefmarkengröße am unteren Bildschirmrand. Der zugehörige Prozess läuft weiter, nimmt aber keine Eingaben von der Tastatur mehr an. Icons lassen sich auf dem Bildschirm verschieben. Um aus dem Icon wieder ein Fenster zu machen, klickt man es mit der Select-Maustaste an.

Durch Anklicken des Maximize-Buttons bringt man ein Fenster auf volle Bildschirmgröße, sodass kein weiteres Fenster mehr zu sehen ist. Das empfiehlt sich für längere Arbeiten in einem Fenster. Auf die vorherige Fenstergröße zurück kommt man durch nochmaliges Anklicken des Maximize-Buttons.

Jetzt fehlt noch der Menü-Button. Klickt man ihn an, öffnet sich unterhalb der Kopfleiste ein Menü (Pull-down-Menü) mit einigen Funktionen zur Fenstergestaltung. Eine zur Zeit nicht verfügbare oder sinnlose Funktion erscheint grau. Es ist

üblich, als untersten Punkt des Menüs eine Funktion zum Schließen des Fensters oder Verlassen des Programms einzurichten.

Terminal-Emulatoren

Auch Anhänger grafischer Benutzer-Oberflächen brauchen dann und wann eine Kommandozeile. Zu diesem Zweck stehen unter X11 Terminal-Emulatoren zur Verfügung, die in einem Fenster einen altmodischen Textbildschirm nachahmen (emulieren). Der Fortschritt gegenüber früher besteht darin, dass man zum Preis eines physikalischen Bildschirms mehrere Fenster mit jeweils einem emulierten Bildschirm bekommt. Die Tastatur lässt sich nicht per Software vervielfachen, und Rechner mit mehreren Tastaturanschlüssen sind ungewöhnlich (mehrere echte Bildschirme an einem Rechner gibt es). Aus einem solchen emulierten Terminal heraus lassen sich Programme aufrufen und benutzen, die für die Kommandozeile geschrieben sind, allen voran die Shells.

X11 bringt selbst den Emulator `xterm` mit, der ein Terminal vom Typ DEC VT102 nachahmt. Man richtet zweckmäßig auch gleich noch eines der Pakete `xtermcontrol` oder `xtermset` ein, um die Eigenschaften von `xterm` einfach ändern zu können. Näheres zu `xterm` findet sich außer im Manual auf <http://invisible-island.net/xterm/>. Um `xterm` mit einem bestimmten Font zu starten (nicht jeder Font ist mit Kurznamen erreichbar, siehe `/etc/X11/fonts/misc/xfonts-base.alias`), kann man den Wunsch als Option mitgeben:

```
joe@debian:~$ xterm -fn 10x20
```

Wer mit `xtermcontrol` arbeitet, schreibt seine Bedürfnisse in die Datei `.xtermcontrol` im Home-Verzeichnis und ruft aus `xterm` das Kommando `xtermcontrol` auf. Als Dauerlösung bietet sich eine Zeile:

```
xterm*font: 10x20
```

in der Datei `$HOME/.Xresources` an. Einzelheiten zu dieser Datei mittels `man X`, Abschnitt *RESOURCES*. Auf der Manualseite finden sich als Beispiel Eintragungen zu `Xterm`. Weitere Beispiele enthält das Verzeichnis `/usr/X11R6/lib/X11/app-defaults/`. Die aktuellen Ressourcen einer Anwendung erfährt man mit einer Eingabe wie:

```
joe@debian:~$ appres xterm
```

Das Drücken der `<ctrl>`-Taste zusammen mit einer Maustaste (Cursor im `xterm`-Fenster) bringt ein Menü für einige Einstellungen hervor, insbesondere zur Änderung der Schriftgröße.

Aus dem GNOME-Projekt stammt `gnome-terminal`, aus dem KDE-Projekt `konsole`. Beim Anklicken von *Terminal* oder *Shell* in einem Desktop kommt meist eines dieser beiden Terminals hoch.

Es gibt einige Emulatoren mit erweiterten Eigenschaften, beispielsweise ein multilinguales Terminal aus dem Debian-Paket `mlterm`, das mehrere nicht-lateinische Zeichensätze beherrscht, oder einen RDP-Client aus dem Debian-Paket `rdesktop`, der mit einem MS Windows Terminal Server kommuniziert.

Screenshots unter X11

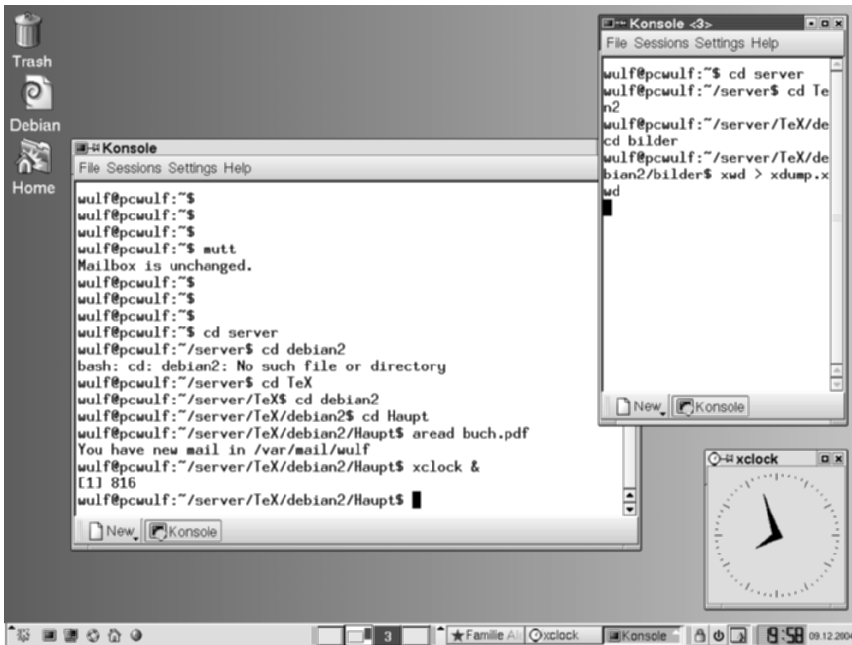


Abbildung 1.9. Screenshot eines X11-Bildschirms mittels `xwd`, hier ein KDE-Desktop mit drei Fenstern, Icons am linken und Taskbar am unteren Rand. Im Fenster rechts oben ist der Aufruf von `xwd` zu erkennen.

Die Ausgabe eines ganzen Bildschirms oder eines Fensters in eine Datei oder auf einen Drucker wird Screen Dump, Screenshot oder Bildschirmabzug genannt, siehe Abbildung 1.9. Unter X11 schreibt das Kommando `xwd` einen Dump in einem eigenen Format nach `stdout` bzw. mittels Umleitung in eine Datei:

```
joe@debian:~$ xwd > dump.xwd
```

einzugeben in ein kleines Fenster, welches das zu dumpende Fenster nicht verdecken darf. Dann klickt man mit der linken Maustaste in das zu dumpende Fenster und findet anschließend die Datei `dump.xwd` in dem zum Aufruf von `xwd` gehörenden Arbeitsverzeichnis. Mittels des Programmes `gimp` lässt sich das Dateiformat in andere Formate wie `jpg` oder `eps` umwandeln. Das Kommando `xgrab` leistet das Gleiche, ist jedoch nicht überall verfügbar. Auch Grafik-Werkzeuge wie `gimp` oder `xv` enthalten Möglichkeiten zum Dumpen des Bildschirms, ebenso das K-Desktop-Environment unter dem Menüpunkt `ksnapshot`. Einige der Werkzeuge erlauben, das Dumpen nach einer bestimmten Verzögerung auszulösen. Bilder aus dem Fernsehen oder von Video DVDs lassen sich nicht auf diese Weise festhalten.

Hierzu braucht man eine entsprechende Funktion (Schnappschuss) des Wiedergabeprogramms, siehe Abschnitt 10.4 *Fernsehen mit dem PC* auf Seite 358.

1.3.3 Desktop Environments (KDE, GNOME)

Ein Benutzer hat auf einem durchschnittlichen Arbeitsplatz folgende Möglichkeiten, Tätigkeiten im Dialog gleichzeitig laufen zu lassen:

- virtuelle Konsolen oder Terminals, meist sechs zusätzlich zum X11-Bildschirm,
- Desktops, meist vier,
- Fenster auf einem Desktop, praktisch begrenzt durch die Bildschirmgröße.

Auf dem Rechner können weitere Aktivitäten im Hintergrund laufen sowie Benutzer über das Netz (`telnet`, `rlogin`, `slogin`) eingeloggt sein. Diese Vielfalt bietet nur Linux/UNIX.

Die Desktops kann man sich als Schreibtische vorstellen, auf denen jeweils eine bestimmte Arbeit erledigt wird. Auf dem ersten Desktop kann beispielsweise ständig ein Webbrowser laufen, weil man oft auf Informationsquellen im Web zugreift. Auf dem zweiten arbeitet man in einem Terminalfenster (`xterm`, KDE Konsole, `gnome-terminal` und andere) mit einem Editor an einem Manuskript und hat dazu das Fenster maximiert. Auf dem dritten läuft ebenfalls ein Terminal, aber nicht maximiert, für Zwischenfragen (`ps`, `ls`, `mail`) an das System und auf dem vierten eine Verbindung zu einem Server. Man kann sich sehr an das Arbeiten mit Desktops gewöhnen, muss jedoch aufpassen, dass man nicht auf zwei Desktops gleichzeitig dieselbe Datei mit jeweils einem Editor öffnet, sonst gewinnt der, der als letzter zurückschreibt.



Abbildung 1.10. Das KDE-Logo

Unter Debian GNU/Linux werden vor allem zwei Desktop Environments verwendet⁵:

- das K Desktop Environment (KDE),
- das GNU Network Object Model Environment (GNOME).

⁵In der Zeitschrift c't Nr. 1/2005 erschien ein Vergleich der beiden Umgebungen KDE und GNOME

Im vorigen Abschnitt haben wir die Desktops bereits kurz erwähnt. Die Wahl zwischen beiden fällt schwer. Selbst wenn zu einem Zeitpunkt einer der beiden bei einigen Aufgaben die Nase vorn hat, kann das ein Jahr später umgekehrt aussehen. Bei KDE gab es eine Zeit lang Lizenzprobleme mit der Qt-Bibliothek; diese sind jedoch ausgeräumt. Der KDE-Datei-Manager Konqueror leistet zur Zeit mehr als der GNOME-Datei-Manager Nautilus, aber die GNOME-Entwickler schlafen nicht. Viele Anwendungen aus dem einen Desktop können auch unter dem anderen aufgerufen werden. Große Anwendungen wie der Webbrowser Mozilla oder das OpenOffice-Paket bringen eigene Bibliotheken mit und laufen auf beiden Desktops gleichermaßen. Das GNOME-Logo ist der Fußabdruck eines Gnoms – siehe Abbildung 1.11 – während KDE ein Zahnrad mit dem Buchstaben K darin verwendet (Abbildung 1.10). Der Buchstabe hat keine tiefere Bedeutung.



Abbildung 1.11. Das GNOME-Logo, der vierzehige Fußabdruck eines Gnoms

Trägt sich ein Programm bei der Einrichtung des zugehörigen Paketes nicht von sich aus in ein Menü ein oder möchte man ein selbst geschriebenes Werkzeug einem Menü hinzufügen, so kann das ein Benutzer auf folgende Weise tun:

- Im Home-Verzeichnis ein Unterverzeichnis `.menu` anlegen (Kommando `mkdir`), Zugriffsrechte 750 (Kommando `chmod`),
- in dem Unterverzeichnis mit einem Editor eine Menüdatei mit dem Namen des Programms anlegen, Zugriffsrechte 640, Inhalt siehe unten,
- Kommando `update-menus` aufrufen,
- Window-Manager neu starten (oder ausloggen, einloggen).

Näheres auf den Manualseiten zu `update-menus` und `menufile`, Beispiele für Menüdateien in `/usr/lib/menu`. In folgender Menüdatei geht es um die Einbindung eines selbst geschriebenen Shellscripts mit dem Pfad `/usr/local/bin/backup` in die Menüfolge *Haupt- oder Startmenü -> Debian -> Anwendungen -> Werkzeuge*:

```
?package(local.backup):\
needs="text"\
section="Apps/Tools"\
title="Backup"\
command="/usr/local/bin/backup"
```

Der Paketnamen muss vorhanden sein und bei lokalen Spezialitäten mit `local.` beginnen. Das Shellscript benötigt ein Textfenster, kein X11. Unter `section` werden

die englischen Bezeichnungen der Debian-Menüs verstanden. Die Gegenschrägstriche verbinden die Zeilen. Der Rest dürfte klar sein. Die Namen für Eigenerzeugnisse sollen nicht mit denen aus Debian-Paketen übereinstimmen. Mehrdeutige Namen können gut gehen, aber auch Verwirrung stiften. Auf dem KDE Desktop steht ein Kommando `kmenuedit` zur Verfügung, das die Schritte unter einer grafischen Oberfläche vereint. Weitergehende Maßnahmen sind dem Verwalter vorbehalten.

Über den Annehmlichkeiten grafischer Oberflächen mit Menüs und Knöpfchen vergesse man nicht, dass diese eine Vorauswahl der Kommandos und Optionen treffen. Nur mit der Kommandozeile erreicht man alle Kommandos und Optionen und kann noch einige Dinge mehr tun. Ein erfahrener Benutzer kennt und nutzt beide Wege.

1.3.4 Multimediale Oberflächen

Bei Spielen – siehe Abschnitt 13 *Spiele und Spielereien* auf Seite 413 – sind Maus und Tastatur manchmal suboptimale Steuergeräte. Der Wirklichkeit näher kommen Steuer- oder Freudenknüppel (E: joystick, F: manche à balais) in verschiedener Ausformung. Sie werden am USB-Port angeschlossen. Früher war ein besonderer Game-Port erforderlich, oft auf der Soundkarte zu finden.

Der Mensch hat nicht nur Augen und Finger, sondern auch noch Ohren, eine Nase, eine Zunge und eine Stimme. Es liegt also nahe, zum Gedankenaustausch mit dem Rechner nicht nur den optischen und mechanischen Übertragungsweg zu nutzen, sondern auch den akustischen und gelegentlich den Geruchssinn⁶. Letzteres wird seit altersher bei der ersten Inbetriebnahme elektronischer Geräte aller Art gemacht (smoke test), weniger während des ordnungsgemäßen Betriebes. Es gibt aber bereits Aroma-Diffusoren mit USB-Anschluss, die beim Spielen mit Autorennen-Simulatoren den Duft von Benzin und Gummi verbreiten könnten (c't Nr. 21/2004, S. 70). Der akustische Weg wird in beiden Richtungen vor allem in solchen Fällen genutzt, in denen Augen oder Finger anderweitig beschäftigt sind (Fotolabor, Operationssaal, Fahrzeuge) oder fehlen. In den nächsten Jahren wird die Akustik an Bedeutung gewinnen. Über die Nutzung des Geschmackssinnes wird noch nachgedacht (wie soll das Terminal aussehen bzw. ausschmecken?).

Im Ernst: unter einer multimedialen Oberfläche versteht man bewegte Grafiken plus Ton, digitales Kino mit Dialog sozusagen. Der Rechner gibt nicht nur eine dürre Fehlermeldung auf den Bildschirm aus, sondern lässt dazu *That ain't right* mit FATS WALLER am Piano ertönen. Umgekehrt beantworten Sie die Frage des Editors *vi*, ob er ohne Zurückschreiben aussteigen soll, nicht knapp und bündig mit einem Ausrufezeichen, sondern singen wie EDITH PIAF *Je ne regrette rien*. Die eintönige Arbeit am Terminal entwickelt sich so zu einem anspruchsvollen kulturellen Happening. Die Zukunft liegt bei multisensorischen Schnittstellen, die mit Menschen auf

⁶Nachricht in Markt & Technik vom 31. März 1994: IBM entwickelt künstliche Nase. Nachricht in der c't 21/1998: In der Ohio State University erkennt eine elektronische Nase Käsesorten am Geruch. Vielleicht fordert Sie ihr Rechner demnächst auf, den Kaffee etwas stärker anzusetzen oder die Socken zu wechseln.

zahlreichen kognitiven und physiologischen Ebenen zusammenarbeiten (Originalton aus einem Prospekt).

1.3.5 Oberflächen für Behinderte

Das Thema *Behinderte und Rechner* hat mehrere Seiten⁷. An Behinderungen kommen in Betracht:

- Behinderungen des Sehvermögens
- Behinderungen des Hörvermögens
- Behinderungen der körperlichen Beweglichkeit
- Beeinträchtigungen der Konzentrationsfähigkeit, des Gedächtnisses oder des Sprechvermögens.

Die Benutzung eines Rechners durch einen Behinderten erfordert eine Anpassung der Hardware, insbesondere des Terminals, und Rücksicht seitens der Software einschließlich der im Netz angebotenen Informationen (Web und andere Dienste). Das Stichwort lautet Zugänglichkeit oder Barrierefreiheit (E: accessibility, F: accessibilité).

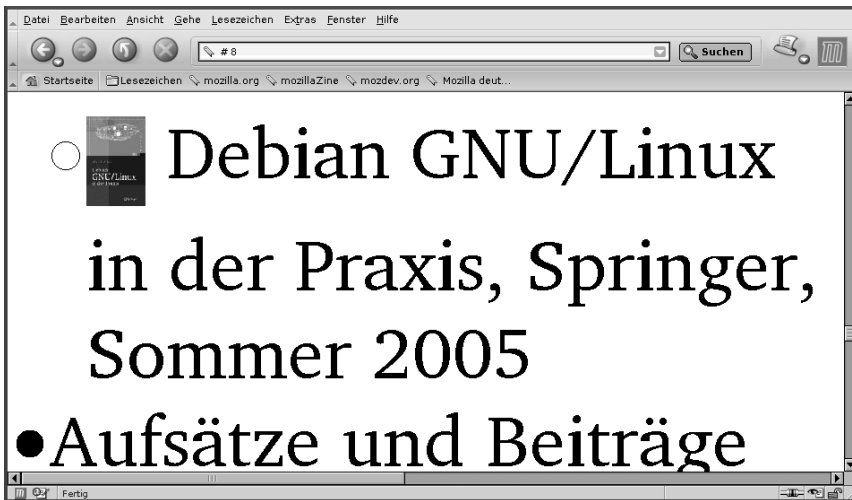


Abbildung 1.12. Screenshot eines vergrößerten X11-Fensters

Andererseits kann ein Rechner als Hilfsmittel bei der Bewältigung alltäglicher Probleme dienen, zum Beispiel beim Telefonieren. Einer der bekanntesten behinderten Benutzer ist der britische Physiker STEPHEN WILLIAM HAWKING, der sich mit

⁷Für Hilfe bei der Abfassung dieses Abschnitts danke ich MATTHIAS HÄNEL, von dessen Webseiten auf <http://www.matthias-haenel.de/> ich mit seiner Zustimmung einige Abschnitte übernommen haben. MATTHIAS HÄNEL ist Datenverarbeitungskaufmann und selbst blind, kennt sich also mit den Problemen blinder Computerbenutzer bestens aus.

seiner Umwelt per Rechner verständigt. Näheres in der deutschen Wikipedia oder auf seiner Startseite <http://www.hawking.org.uk/>.

Folgende Webseiten aus dem Debian-Umfeld widmen sich dem Thema Zugänglichkeit oder Barrierefreiheit:

- <http://www.tldp.org/HOWTO/Accessibility-HOWTO/>
- <http://www.linaccess.org/>
- <http://www.freebsoft.org/>
- <http://lars.atrc.utoronto.ca/>
- <http://www.blinux.de/>
- <http://www.de.debian.org/devel/debian-accessibility/>
- <http://developer.gnome.org/projects/gap/>
- <http://accessibility.kde.org/>

An erster Stelle ist das *Linux Accessibility HOWTO* zu erwähnen. Linaccess ist ein Projekt zur Förderung von freier Software für Behinderte. Auf dessen Website sind einige Informationen zu finden, die für behinderte Benutzer, aber auch für Entwickler von Software für behinderte Benutzer hilfreich sind. Für den Linux-Tag 2005 in Karlsruhe hatte das Projekt 15 Vorträgen organisiert, es tut sich etwas. Das Freebsoft Laboratory passt freie Software an die Möglichkeiten sehbehinderter Benutzer an. Darunter befand sich eine spezielle Debian-Version, die jedoch teilweise im Debian Accessibility Project aufgegangen ist. Eine Übersicht über die Anstrengungen der Linux-Welt auf diesem Gebiet findet sich auf der Linux Accessibility Resource Site (LARS) in der Universität Toronto. Das Blinux-Projekt bezweckt, die Benutzung von Linux durch blinde Benutzer zu verbessern. Es stellt eine FAQ-Sammlung in deutscher und englischer Sprache zur Verfügung und ist der SuSE-Distribution verbunden. Bei Debian entsteht das Projekt *Debian-Accessibility* mit demselben Ziel. Unter GNOME und KDE sind ebenfalls Bemühungen im Gange, aus den an sich günstigen Voraussetzungen von Linux fertige Hilfen für behinderte Benutzer zu entwickeln, vor allem für sehbehinderte. Viele Werkzeuge sind noch experimentell.

An drei deutschen Hochschulen bemühen sich Einrichtungen, Sehgeschädigten vor allem das Studium der Informatik, aber auch andere Tätigkeiten zu erleichtern und betreiben Forschung auf diesem Gebiet:

- Universität Karlsruhe (TH), Studienzentrum für Sehgeschädigte, <http://www.szs.uni-karlsruhe.de>,
- Fachhochschule Giessen-Friedberg, Zentrum für blinde und sehbehinderte Studierende, <http://www.fh-giessen.de/bliz/>,
- Technische Universität Dresden, ELVIS-Arbeitsgruppe Studium für Blinde und Sehbehinderte, <http://elvis.inf.tu-dresden.de/>.

Im Vereinigten Königreich (UK) befasst sich eine Organisation mit den Problemen von Benutzern, die sowohl blind wie taub (Taubblinde) sind. Diesen bleibt nur der Tastsinn, was das Rechnerterminal auf Tastatur und Braillezeile einschränkt. Die Startseite der Organisation hat den URL <http://www.deafblind.org.uk/>; unter den URLs <http://www.deafblind.com/> sowie <http://www.deafblindinternational.org/> liegt ebenfalls Material zum Thema.

Das Sehvermögen kann in vielerlei Hinsicht beeinträchtigt sein: mangelnde Sehschärfe, die nicht in jedem Fall durch Hilfsmittel (Brille, Kontaktlinsen) korrigiert werden kann, Farbsehschwächen – insbesondere die bei Männern verbreitete Rot-Grün-Schwäche – Probleme mit Lichtkontrasten oder schnell bewegten Bildern bis hin zu völliger Blindheit. Jeder, der lange genug lebt, stellt ein Nachlassen seines Seh- und Hörvermögens fest.

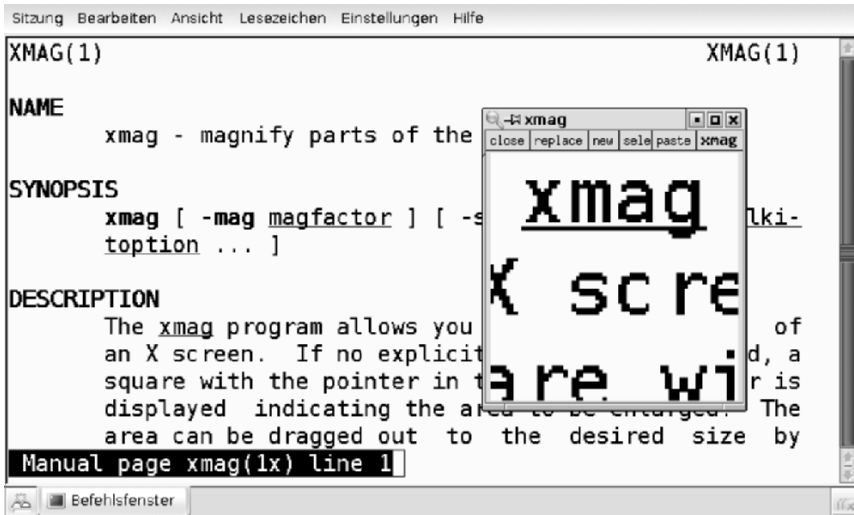


Abbildung 1.13. Screenshot der X11-Bildschirmlupe xmag

Einen größeren Maus-Cursor unter X11 bietet das einfach einzurichtende Debian-Paket *big-cursor* aus der Abteilung X Window System Software an. Dazu gibt es ein *X Big Cursor mini-HOWTO*. Will man den gesamten Bildschirm oder einen Ausschnitt daraus vergrößern – wobei man dann natürlich mehr scrollen muss – braucht man Programme, die als Bildschirmvergrößerer oder Bildschirmlupe bezeichnet werden und teilweise noch mehr können, zum Beispiel Farbkontraste verstärken oder Farben umkehren (<http://www.magnifiers.org/>). Einige sind auch mit Vorleseprogrammen (Screenreader) kombiniert.

Unter X11 lässt sich durch Konfigurieren und Auswählen von Auflösungen, die über den üblichen Werten liegen, ein virtueller Bildschirm erzeugen, der größer als der physikalische Bildschirm ist. Mit der Maus wird das sichtbare Fenster im virtuellen Bildschirm verschoben. Abbildung 1.12 zeigt ein Beispiel mit einem Ausschnitt aus einer Webseite. X11 bringt auch eine einfache Bildschirmlupe unter dem Namen *xmag* mit, die einen kleinen quadratischen Ausschnitt vergrößert darstellt, siehe Abbildung 1.13. Näheres im Manual.

Ab Debian GNU/Linux 3.1 (*sarge*) stehen die Debian-Pakete *gnome-mag* und *kmag* aus der Abteilung X Window System Software bzw. *Utilities* zur Verfügung, die beide X11 voraussetzen. Das GNOME-Werkzeug wird unter dem Namen

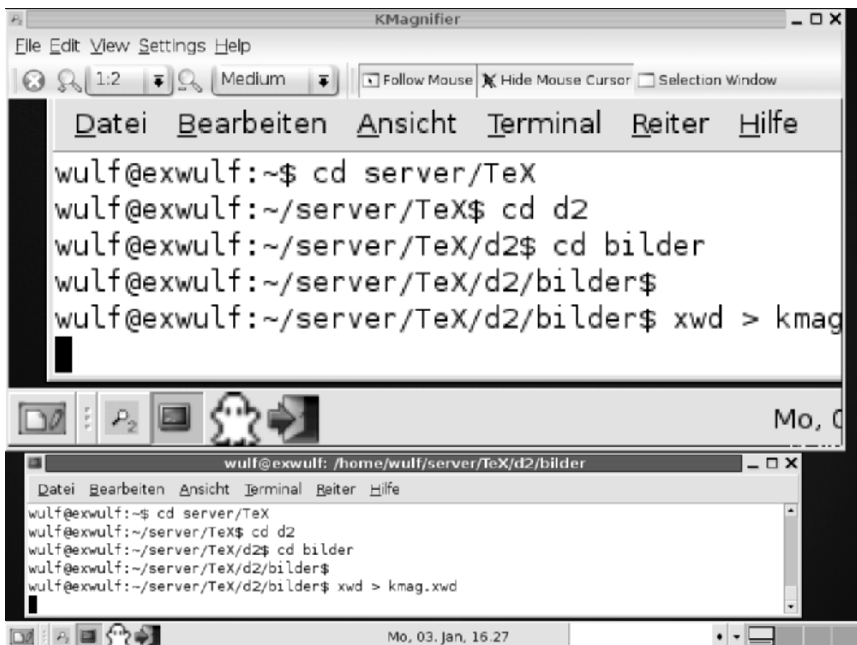


Abbildung 1.14. Screenshot mit normalem Fenster (unten) und mittels der Bildschirmlupe kmag vergrößerndem Fenster (oben). Die Vergrößerung lässt sich erhöhen, aber der dargestellte Ausschnitt wird immer kleiner. Die Anwendung ist ein Fall für einen PC mit zwei Bildschirmen (Dual Head), einen für die Übersicht, den anderen für Einzelheiten.

magnifier aufgerufen, zweckmäßig zuerst mit einer Hilfe-Option, da noch keine Manual-Seite vorhanden ist:

```
joe@debian:~$ magnifier -help
```

Der Aufruf:

```
joe@debian:~$ magnifier -target-display :0
                  -source-display :0 -horizontal
```

sollte dazu führen, dass der Bildschirm horizontal zwischen dem normalen und dem vergrößernden Fenster geteilt wird, erzeugte bei uns jedoch nichts Brauchbares. Das Debian-Paket *gnopernicus* aus der Abteilung *X Window System Software* ist zwar in *sarge* enthalten, wird aber ausdrücklich als experimentell bezeichnet. Es enthält mehrere Hilfsmittel für Sehbehinderte.

Das KDE-Werkzeug ist eine Bildschirmlupe zur beliebigen Verwendung, bringt eine Manualseite mit und zeigt auch eine kurzgefasste Hilfe an:

```
joe@debian:~$ kmag -help-all
```

Mehr erfährt man durch:

```
joe@debian:~$ khelpcenter help:/kmag
```

Der Aufruf:

```
joe@debian:~$ kmag
```

erzeugt ein Hilfsfenster, in dem die Ein- und Ausgaben des normalen Fensters vergrößert zu lesen sind, siehe Abbildung 1.14. Persönliche Einstellungen werden in `$HOME/.kde/share/config/kmagrc` gespeichert.

Ein Vorleseprogramm oder Screenreader gibt Textinhalte des Bildschirms akustisch aus. Das ist ein zweistufiger Vorgang:

- Aus der Darstellung auf dem Bildschirm wird der Text zeilenweise herausgezogen, was bei Webseiten, die alle Gestaltungstricks ausnutzen, an das Unmögliche grenzt,
- der Text wird akustisch oder auf einer Braillezeile ausgegeben.

Die beiden Stufen werden auf getrennte Software verteilt, siehe Abschnitt 9.7.3 *Screenreader* auf Seite 349. Erleichtert wird die Aufgabe durch Anwendungsprogramme, die von vornherein eine Schnittstelle für die zeichenweise Ausgabe vorsehen.

Ein Screenreader erfüllt die Aufgaben einer dem blinden Benutzer vorlesenden Person, nicht ganz mit deren Intelligenz und heute rein in Software verwirklicht. Da die Programme mit Wörterbüchern arbeiten, sind sie immer auf eine bestimmte Sprache eingestellt, die sich wechseln lässt, aber nicht so schnell, wie ein sehender Leser zwischen Sprachen umschaltet. Deshalb sind gemischtsprachige Texte unangenehm.

Die Sprachausgabe ist im Vergleich zur Braillezeile die in Anschaffung und Unterhalt preiswertere Alternative, abgesehen davon, dass Späterblindete mit der Brailleschrift Schwierigkeiten haben. Wenn es aber wie beim Programmieren auf jedes Zeichen einschließlich Leerzeichen ankommt, ist die Braillezeile unersetzlich.

Für das Hörvermögen gilt im Prinzip dasselbe wie für das Sehvermögen, nur spielt das Hören für die Arbeit am Rechner keine so bedeutende Rolle, während im Alltag Hörbehinderte andere, nicht weniger unangenehme Probleme haben als Sehbehinderte. Hörbehinderte werden im *Linux Accessibility HOWTO* mit einigen Zeilen bedacht; im Netz ist auch nicht viel zu finden. Aus eigener Erfahrung über Hörbehinderungen informiert <http://www.typolis.de/hear/>. Die Stichworte, unter denen gesprochener Text auf dem Bildschirm lesbar dargestellt wird, heißen *Telecommunications Devices for the Deaf* (Telefon) und *Closed Captioning* (Fernsehen). Unter <http://www.sourceforge.net/projects/ccdecoder/> liegt ein Werkzeug, das mit TV-Karten unter `bttv` und `video4linux` zusammenarbeitet,

Akustische Hilfen für beide Kreise werden in den Abschnitten 9.6 *Spracherkennung* auf Seite 346 und 9.7 *Sprachsynthese* auf Seite 347 besprochen.

Bei Einschränkungen der Beweglichkeit ist zu unterscheiden zwischen solchen, welche die ganze Person betreffen (Rollstuhlfahrer), und Behinderungen einzelner Gliedmaßen, vor allem der Arme und Hände. Man versuche einmal, Tastatur und Maus oder Rollkugel mit dicken Fausthandschuhen oder mit den Füßen zu betätigen.

Schließlich gibt es noch die mehr im Geistigen verankerten Einschränkungen der Konzentrationsfähigkeit, des Erinnerungs- und des Sprechvermögens. Hierzu finden

sich wenige Informationen im Netz. Diese Hindernisse sind auch nicht so einfach zu quantifizieren wie die körperlichen. Im *Linux Accessibility HOWTO* stehen ein paar Hinweise.

1.3.6 Lokalisierung

Obwohl in der Computerei vorwiegend Englisch gesprochen wird und die Fähigkeit, englischen Text zu lesen und zu verstehen, beinahe unerlässlich ist, ist die moderne Software zunehmend in der Lage, auch andere Sprachen zu bedienen. Die niedrigste Hürde stellen dabei Sprachen dar, die wie Englisch das lateinische Alphabet verwenden, eventuell um ein paar Häkchen und Pünktchen erweitert wie Deutsch oder Französisch. Die nächste Hürde sind Sprachen, die verwandte Alphabete gebrauchen wie Russisch oder Griechisch. Sprachen, die von rechts nach links schreiben wie Arabisch oder Hebräisch, werfen neue Probleme auf. Schließlich folgen Sprachen mit nicht-alphabetischen Schriften wie Chinesisch, Japanisch, Koreanisch oder Vietnamesisch, die so genannten CJK- oder CJKV-Sprachen und -Schriften (<http://www.cjk.org/> und <http://www.praxagora.com/lunde/>). Historisch interessant sind die Hieroglyphen der alten Ägypter und Mayas. Die modernen Piktogramme sind manchmal ebenso schwer zu enträtseln. An die Knotenschrift der Inkas (Quipu) scheint sich noch kein Programm gewagt zu haben.

Eine Software, die keine Annahmen bezüglich Sprache, Zeichensatz und Bräuchen wie Schreibung von Kalenderdaten trifft, wird *internationalisiert* genannt (internationalization = i18n). Sie ist hinsichtlich der Umgebung ihrer Ausführung neutral. Das erfordert einen gewissen Aufwand, der sich aber in Grenzen hält, wenn die Internationalisierung von Anfang an berücksichtigt wird. Ein Teil unserer heutigen Probleme rührt daher, dass vor dreißig Jahren kaum jemand an Benutzer gedacht hat, die eine andere Sprache als Englisch bevorzugen. Informationen zu i18n findet man unter <http://debian2.inf.tu-dresden.de/doc/manuals/introi-18n/> und bei <http://www.i18ngurus.com/>.

Unter *Lokalisierung* einer Software versteht man ihre Anpassung an eine lokale Sprache samt deren Bräuchen. *National* oder *Native Language Support* meint dasselbe. Im einzelnen betrifft eine Lokalisierung:

- die Übersetzung von Texten (Manualeiten, Menüs, Fehlermeldungen etc.) in die lokale Sprache,
- den Zeichensatz,
- die Reihenfolge beim Sortieren,
- die Schreibweise von Namen, Anschriften, Telefonnummern, Datum und Uhrzeit,
- das Dezimaltrennzeichen (Punkt oder Komma),
- Währung, Maßsystem und Papierformate.

Im POSIX-Standard wird die Menge dieser lokalen Eigenheiten *Locale* genannt und in Umgebungsvariablen niedergelegt. Der Aufruf:

```
joe@debian:~$ locale
```

zeigt die aktuellen Einstellungen des Locale für einen Benutzer an:

```
LANG=de_DE@euro
LC_CTYPE="de_DE.iso885915@euro"
LC_NUMERIC="de_DE.iso885915@euro"
LC_TIME="de_DE.iso885915@euro"
LC_COLLATE="de_DE.iso885915@euro"
LC_MONETARY="de_DE.iso885915@euro"
LC_MESSAGES="de_DE.iso885915@euro"
LC_PAPER="de_DE.iso885915@euro"
LC_NAME="de_DE.iso885915@euro"
LC_ADDRESS="de_DE.iso885915@euro"
LC_TELEPHONE="de_DE.iso885915@euro"
LC_MEASUREMENT="de_DE.iso885915@euro"
LC_IDENTIFICATION="de_DE.iso885915@euro"
LC_ALL=de_DE.iso885915@euro
```

Mit der Option `-a` (all) erfährt man alle verfügbaren Lokalisierungen, mit der Option `-m` den Zeichensatz (charmap). Manualseiten dazu in den Sektionen 1, 5 und 7. Mit der letzten sollte man beginnen.

Der POSIX-Standard (<http://www.opengroup.org/onlinepubs/007908799/xbd/locale.html>) gliedert das Locale in sechs Kategorien:

- *LC_CTYPE* Einteilung der Zeichen,
- *LC_COLLATE* Reihenfolge beim Sortieren,
- *LC_MONETARY* Darstellung von Geldbeträgen,
- *LC_NUMERIC* Darstellung von Zahlen, die keine Geldbeträge sind,
- *LC_TIME* Darstellung von Datum und Uhrzeit,
- *LC_MESSAGES* Darstellung von Nachrichten, Meldungen und Antworten.

Die Definition weiterer Komponenten unter Einhaltung der Richtlinien ist zulässig. So können auf einem Debian-PC zusätzliche Kategorien vorkommen, wie wir eben gesehen haben. Die Kategorie *LANG* gilt dabei für alle nicht ausdrücklich durch eine LC-Variable gesetzten Einstellungen, die Kategorie *LC_IDENTIFICATION* enthält Metadaten über das Locale. Die höchste Priorität hat *LC_ALL*, die niedrigste *LANG*. Ist die Variable *LC_ALL* nicht leer, ersetzt sie alle übrigen. Der Wert dieser Umgebungsvariablen könnte `de_DE@euro` oder `de_DE.UTF-8` lauten. Die unterstützten Locale sind in der Datei `/usr/share/i18n/SUPPORTED` aufgelistet. Nebenan liegen in zwei Verzeichnissen die Definitionsdateien und Zeichensätze.

Wie üblich kann der Verwalter im Verzeichnis `/etc/` systemweite Einstellungen ablegen, die von einem Benutzer mittels seiner Datei `$HOME/.bashrc` überannt werden können. In die Datei `/etc/locale.gen` sind vom Verwalter die gewünschten Zeilen einzutragen so wie sie in oben genannter Datei `SUPPORTED` stehen:

```
de_DE@euro ISO-8859-15
de_DE ISO-8859-1
de_DE.UTF-8 UTF-8
de_DE.UTF-8@euro UTF-8
fr_FR.UTF-8@euro UTF-8
ru_RU.UTF-8 UTF-8
sv_SE ISO-8859-1
```

Danach erzeugt der Verwalter die auf der Maschine auswählbaren Locale (Bindestrich beachten, oben war es ein Punkt):

```
debian:~# /usr/sbin/locale-gen
```

Das Default-Locale C bzw. POSIX braucht nicht eigens erzeugt zu werden, es steht immer zur Verfügung. Die Locale liegen dann in `/usr/lib/locale/` in einem nicht lesbaren Archiv, mittels `locale -a` überprüfbar.

Ein Benutzer fügt seiner Datei `$HOME/.bashrc` gegen Ende eine Zeile folgender Art:

```
export LC_ALL=de_DE.UTF8@euro
```

hinzu. Nach Aus- und wieder Einloggen sollte er mit dem Kommando:

```
joe@debian:~$ set | grep LC
```

den Erfolg seiner Anstrengungen begutachten können. Die letzten Zweifel beseitigt:

```
joe@debian:~$ locale
```

Während einer Sitzung setzt oder ändert dieselbe Zeile die Umgebungsvariable für die aktuelle Shell und deren Abkömmlinge, aber nicht für die ganze Sitzung (Vererbung nur vorwärts). Nach der Eingabe:

```
joe@debian:~$ export LC_ALL=fr_FR
```

zeigt das Kommando `date` das Datum in Französisch an, während der Niederschrift dieser Zeilen irgendetwas mit *ven* und *fév*. Möchte man die Manualseiten auf Französisch genießen, ist zunächst einmal das Debian-Paket *manpage-fr* einzurichten. Die Texte landen in `/usr/share/man/fr/` so wie die deutschen in `de/`. Bei der Gelegenheit sollte der Verwalter auch das Kommando `mandb` aufrufen. Mittels:

```
joe@debian:~$ man -L fr pwd
```

kann man nun die Manualseite zu `pwd` auf Französisch lesen, unabhängig vom aktuellen Locale; sie ist also vorhanden. Der Aufruf:

```
joe@debian:~$ man pwd
```

antwortete rein deutsch, trotz französischem Locale.

```
joe@debian:~$ vim --help
```

lieferte eine deutsch-französisch-englische Hilfe, nicht ganz das Erwartete. Wir lasen nicht locker:

```
joe@debian:~$ man bash
```

Voilà: *bash – Interpréteur de commandes GNU Bourne-Again-Shell*. Leider fällt gleich ein Wermutstropfen in unsere Freude: *La traduction de cette page de manuel n'est pas à jour; veuillez consulter la version anglaise pour référence*. Mehrere Tage und Systemstarts später – als das Locale längst wieder auf `de_DE` stand – meldete der Editor `vim` immer noch *Motif introuvable*. Auf einem Rechner ohne X11 kamen

die Manualseiten bei französischem Locale sofort auf Französisch und blieben dabei, auch nachdem das Locale wieder auf Deutsch gesetzt worden war. Da sind wohl noch Hausaufgaben zu erledigen.

Die Einstellungen der Tastatur sind eine Sache für sich und im Abschnitt 1.3.2 *Tastaturen unter X11* auf Seite 16 erklärt.

1.4 Anwendungen

1.4.1 Kommando-Interpreter

Wie vorhin erläutert, kommen und gehen die Anwendungen im laufenden Betrieb. Einige werden automatisch beim Systemstart gestartet und arbeiten bis zum Herunterfahren des Rechners (Dämonen), andere werden von Benutzern aufgerufen und sind aktiv, bis ihre Aufgabe erledigt ist oder der Benutzer seine Sitzung beendet. Während früher der Mangel an Anwendungsprogrammen ein Nachteil von Linux war, stehen heute für Linux – damit natürlich auch für Debian GNU/Linux – so viele Anwendungen in unterschiedlichen Entwicklungsstadien und für unterschiedliche Hardware zu Verfügung, dass es schwer fällt, den Überblick zu bewahren. Neben freier Software im strengen Sinn von Debian gibt es eingeschränkt freie Software und kommerzielle Produkte für Linux. Was *frei* im Sinne von Debian bedeutet, ist in den *Debian Free Software Guidelines* (DFSG) unter http://www.debian.org/social_contract#guidelines erklärt. Den Guidelines entsprechende Software ist in der Sektion *main* des Debian-Archivs untergebracht. Als Dienst am Benutzer bietet Debian aber auch nicht freie Software an, und zwar in der Sektion *contrib* Software, die selbst frei ist, aber von nicht-freier Software (Bibliotheken und anderes) abhängt, und in der Sektion *non-free* Software mit stark eingeschränkten Lizenzen. Den privaten Benutzer brauchen diese Unterschiede nicht zu interessieren, wer aber mittels der Software seine Brötchen verdient oder Software weiter veräußern will, sollte genau nachlesen.

Unter den Anwendungen spielen die Kommando-Interpreter (Kommandozeilen-Interpreter) – in der Linux/UNIX-Welt Shell genannt – eine herausragende Rolle. Vom Kern aus betrachtet, sind es zweifellos Anwendungen. Für den Benutzer hingegen scheinen sie die sichtbare Seite des Betriebssystems zu sein; sie sind der Gesprächspartner im Dialog am Terminal. Man findet auch die Bezeichnung *Bediener* für solche Programme. Es ist zwar denkbar, Anwendungsprogramme zu schreiben, die ohne Shell auskommen, aber praktisch beginnt jede Sitzung am Terminal mit einer Shell – der login-Shell – und endet mit dieser. Maßgebend ist der Eintrag in `/etc/passwd`. Durch X11 und die darauf aufsetzenden Login- oder Display-Manager ist die Generationenfolge der Prozesse unübersichtlich geworden. Der KDE-Display-Manager beispielsweise startet `kdeinit` als Kind von `init`. Alle weiteren Prozesse im Desktop sind Abkömmlinge von `kdeinit`. Die Prozessfolge spielt eine Rolle bei der Frage, wo welche Umgebungsvariablen gesetzt werden, aber darum soll sich der Verwalter kümmern.

Die älteste noch lebende Shell ist die Bourne-Shell `sh`, benannt nach ihrem Entwickler STEPHEN R. BOURNE, der sie um 1975 als Mitarbeiter der Bell Laboratories schrieb. Sie ließ Wünsche offen, weshalb bald darauf von BILL JOY aus Berkeley die C-Shell `csh` kam. Alle neueren Shells lassen sich auf diese beiden zurückführen. Die Korn-Shell `ksh` von DAVID G. KORN, ebenfalls Mitarbeiter der Bell Laboratories, vereinigt das Beste aus beiden Shell-Welten in sich und wurde auf verschiedene Betriebssysteme einschließlich DOS übertragen (portiert). Als freie Alternative entstand um 1990 im GNU-Projekt mit BRIAN FOX, einem Mitarbeiter der Free Software Foundation, als erstem Autor die Bourne-again-Shell `bash`, die zur am weitesten verbreiteten Shell für Linux wurde. Der Benutzer spürt erst bei fortgeschrittenen Aufgaben Unterschiede zwischen `sh`, `ksh` und `bash`. Ein Grund, von der `bash` abzuweichen, kann darin liegen, dass in einer heterogenen Umgebung alle Maschinen dieselbe Shell verwenden sollen. Näheres zu den Shells im Abschnitt 2 *Kommando-Interpreter* auf Seite 53. Ergänzend empfiehlt sich das Lesen des Eintrags unter *Shell* in der deutschen Wikipedia. Wir gehen weiter zur nächsten Gruppe von Anwendungen.

1.4.2 Dienstprogramme

Unter den Anwendungen, die vom Benutzer aufgerufen werden, gibt es einige oft gebrauchte Werkzeuge, die auf praktisch jedem System zu finden sind. Dazu rechnen Werkzeuge zum Kopieren, Verschieben und Löschen von Dateien, zur Anzeige von Datum und Zeit, zum Auflisten von Verzeichnissen, zur Anzeige von Benutzern und Prozessen und was dergleichen Kleinigkeiten mehr sind. Diese Knechte werden als Dienstprogramme (E: utility, F: utilitaire) bezeichnet. Die meisten sind in Debian-Paketen zu finden, deren Namen die Zeichenkette *utils* enthält, von den *alsa-utils* bis zu den *xutils*. Typisch für Linux/UNIX sind kleine Werkzeuge, die genau eine bestimmte Aufgabe erledigen und überschaubar sind. Komplexe Aufgaben werden mit Kombinationen aus solchen Werkzeugen bewältigt. Es gibt Ausnahmen von dieser Regel, die bekannteste ist der Emacs, siehe Abschnitt 5.4 *Universalgenie (emacs)* auf Seite 188.

Arbeitet man mit einem Desktop Environment, werden viele Aufgaben der Dienstprogramme unter einem Datei-Manager wie Konqueror (KDE) oder Nautilus (GNOME) vereinigt. Das ist bequem. Man braucht dann nur noch zu klicken und zu verschieben und von den Kommandos nichts zu wissen. Andererseits begibt man sich damit in die Gewalt des Managers: Was er nicht vorsieht oder kann, existiert für den Benutzer nicht. Dazu zählen insbesondere Kombinationen (Pipes) aus mehreren Werkzeugen.

1.4.3 Eigentliche Anwendungen

Wir kommen nun zu den eigentlichen Anwendungen, mit deren Hilfe man einen Bericht oder ein Buch schreibt, ein Haus entwirft, einen Verein oder eine Bibliothek verwaltet, die Übersicht über sein Vermögen bewahrt, Vokabeln lernt, Fotos bearbeitet, weltweit nach Informationen sucht oder sich auch nur die Zeit vertreibt. Für

Linux-Systeme stehen sehr viele Anwendungen zur Verfügung, sei es innerhalb von Distributionen, sei es unabhängig davon. Einige Benutzer finden die Vielfalt verwirrend, andere freuen sich über die Möglichkeit auszuwählen.

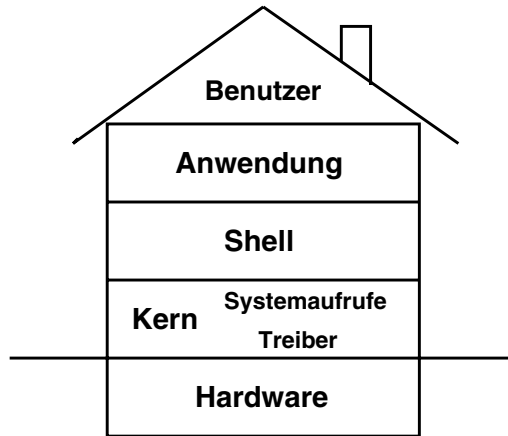


Abbildung 1.15. Aufbau eines Systems aus Hard- und Software

Je nach Umfang der Aufgabe kann ein einzelnes Werkzeug ausreichen oder ein ganzer Werkzeugkoffer erforderlich sein. Für viele Aufgaben stehen unter den zehntausend Debian-Paketen mehrere Werkzeuge zur Auswahl, oft gemäß folgender Einteilung:

- ein einfaches, text-basiertes (curses-basiertes) Werkzeug, das ohne X11 auskommt, die erste Wahl auf schwacher Hardware, für blinde Benutzer oft die einzige Wahl,
- ein grafisches, auf X11 basierendes Werkzeug, unabhängig von einem Desktop Environment,
- ein grafisches Werkzeug aus dem K Desktop Environment (KDE),
- ein grafisches Werkzeug aus dem GNU Object Modelling Environment (GNOME).

Die grafischen Werkzeuge sind teilweise nur grafische Oberflächen (Graphical User Interface, GUI, Frontend) für ein text-basiertes Werkzeug. So wie ein Schlosser oder Schreiner seinen Lieblingshammer hat, haben auch wir unsere Lieblingswerkzeuge, die wir ausführlicher darstellen. Der geneigte Leser wird daher gebeten, sich selbst in den Paketlisten auf <http://www.debian.org/> umzuschauen. Ein vergleichender Test aller Werkzeuge überschreitet unsere Möglichkeiten bei weitem und müsste jedes Jahr wiederholt werden.

Mit dieser obersten Stufe der Software ergibt sich ein Aufbau eines vollständigen Rechnersystems, den man in Form eines Häuschens nach Abbildung 1.15 veranschaulichen kann. Im Keller werkelt die Hardware, was auch tatsächlich oft der Fall ist. Wegen der gleichbleibenden, niedrigen Temperaturen ist sie dort gut aufgehoben.

Darüber kommt der Kern des Betriebssystems, der mit der Hardware über Treiber-Module (E: driver, F: pilote) und mit der höheren Software über Systemaufrufe (Kernschnittstellenfunktion, E: system call, F: primitive, appel système) verkehrt. Hardware und Kern bilden die Linux-Maschine. Auf dieser baut in den meisten Fällen eine Shell auf. Schließlich folgen die Anwendungen und im Dachstübchen der Benutzer. In Details müsste das Bild modifiziert werden, aber es vermittelt eine im Wesentlichen richtige Vorstellung.

1.5 Hilfen (man, info, Scrollkeeper, HOWTOs, Guides, FAQs, RFCs, Wikis, LUGs)

Wenn es um Einzelheiten zu den Kommandos und Werkzeugen geht, muss man nachlesen. Außer Büchern, Skripten und Fachzeitschriften stehen auf einem Linux/UNIX-System folgende Hilfen online zur Verfügung:

- das Online-Handbuch oder Manual (man-Seiten, man pages),
- das GNU-Info-System,
- verschiedene Dokumente zu einzelnen Werkzeugen unter `/usr/share/doc/`,
- die Linux HOWTOs,
- die Linux Guides.

Darüber hinaus kann man das Web und seine Suchmaschinen bemühen. Es gibt kaum eine Frage zu Linux/UNIX, auf die man auf dem eigenen Rechner oder im Netz keine Antwort findet. Man muss nur bereit sein zu lesen. Englischkenntnisse helfen.

Das Manual liegt im Dateisystem auf der Festplatte. Information wird mittels des Kommandos `man` abgerufen:

```
joe@debian:~$ man date
```

bringt die man-Seite zu dem Kommando `date` auf den Schirm. Auch zu `man` selbst findet sich eine man-Seite:

```
joe@debian:~$ man man
```

Vorausgesetzt die zugehörigen Pakete sind eingerichtet, lassen sich die Seiten in verschiedenen Sprachen ausgeben:

```
joe@debian:~$ man -L de ls
```

```
joe@debian:~$ man -L us ls
```

```
joe@debian:~$ man -L fr ls
```

Alle man-Seiten liegen auf Englisch vor, einige sind ins Deutsche oder andere Sprachen übersetzt. Das Manual ist in nummerierte Sektionen unterteilt:

1. Kommandos, ausführbare Programme, z. B. `pwd` oder `gimp`,
2. Systemaufrufe, z. B. `open()` oder `chmod()`,

3. Standardfunktionen, z. B. `fopen()` oder `tan()`,
4. Gerädateien, z. B. `tty` oder `null`,
5. Dateiformate, z. B. `inittab` oder `passwd`,
6. Spiele, z. B. `tuxracer` oder `gnugo`,
7. Makro-Pakete, z. B. `ascii` oder `signal`,
8. Kommandos zur Systemverwaltung, z. B. `ifconfig` oder `apt-get`,
9. Kernfunktionen (nicht überall vorhanden).

Manche Begriffe kommen in mehreren Sektionen vor; dann ist beim Aufruf der Manualseite die gewünschte Sektionsnummer mit anzugeben:

```
joe@debian:~$ man 5 crontab
```

Alle Manualseiten sind einheitlich gegliedert, wobei auf den wenigsten Seiten alle der folgenden Abschnitte vorkommen:

- Name (Name des Kommandos, Sektionsnummer, Zweck),
- Synopsis, Syntax (Gebrauch des Kommandos),
- Copyright der Manualseite,
- Remarks (Anmerkungen, selten),
- Description (Beschreibung),
- Options (Optionen),
- Arguments (etwaige Kommando-Argumente),
- Return Value, Exit Status (Rückgabewert von Programmen, für Programmierer),
- Examples (Beispiele),
- Overview (Übersicht),
- Defaults (Defaultwerte, vorgeschlagene Werte),
- Hardware Dependencies (hardware-abhängige Eigenheiten),
- Files (vom Kommando betroffene Dateien),
- Environment (Umgebungsvariable),
- See also (ähnliche oder verwandte Kommandos),
- Diagnosis (Fehlermeldungen),
- Bugs (Fehler, soweit bekannt),
- Caveats, Warnings (Warnungen),
- History (Entwicklungsgeschichte des Kommandos),
- Author (Urheber der Manualseite und/oder des Kommandos).

Auf umfangreichen Manualseiten wie der der Bash kommen weitere Abschnitte hinzu. Am Fuß einer Manualseite steht ihr Veröffentlichungsdatum. Die Information im Manual ist knapp gehalten und geht davon aus, dass der Benutzer bereits weiß, welches Werkzeug er braucht. Das vorliegende Buch ersetzt das Manual nicht, andernfalls wäre es ein mehrbändiges Werk. Außerdem wäre es witzlos, das Manual wiederzuzukäuen.

Weiß man nur ungefähr, wie ein Kommando lautet, hilft `whichman` weiter:

```
joe@debian:~$ whichman wo
```

bietet folgende Auswahl an:

```

/usr/share/man/man1/wc.1.gz
/usr/share/man/man1/who.1.gz
/usr/share/man/man1/w.1.gz
/usr/share/man/man1/co.1.gz
/usr/share/man/de/man1/wc.1.gz
/usr/share/man/de/man1/who.1.gz

```

während man wo nichts findet. Nach allgemeinen Begriffen lässt sich nur eingeschränkt suchen:

```
joe@debian:~$ apropos editor
```

```
joe@debian:~$ man -k editor
```

suchen nach allen man-Seiten, die mit dem Begriff *editor* etwas zu tun haben. Zum Ausdrucken einer Manualseite verwendet man eine Pipe:

```
joe@debian:~$ man who | lpr
```

Bei Schwierigkeiten fragen sie Ihren Verwalter, für das Drucken gibt es viele Wege. Drucken Sie nicht das ganze Manual aus, die meisten Informationen braucht man nie.

Gelegentlich sind die man-Seiten kurz und verweisen auf eine ausführliche Dokumentation als GNU-Info oder unter `/usr/share/doc/` in Form von HTML-Seiten oder pdf-Texten. Existiert beispielsweise zu dem Werkzeug *ent* eine Hilfe in Form einer HTML-Datei mit dem Pfad `/usr/share/doc/ent/ent.html`, so liest man diese mittels eines Web-Browsers, indem man folgenden URL eingibt:

```
file:///usr/share/doc/ent/ent.html
```

mit drei Schrägstrichen nach dem Doppelpunkt. pdf-Texte liest man mittels *gv*, *xpdf*, *evince* oder dem Adobe Reader. Manche Web-Browser rufen für Nicht-HTML-Dateien ein passendes Plugin auf.

Das Manual kennt keine Verweise oder Hyperlinks. Diesen Mangel behebt das GNU-TeXinfo-System:

```
joe@debian:~$ info date
```

Falls *info* keinen Eintrag findet, sucht es nach einer man-Seite. Die GNU-Manuals liegen größtenteils im *info*-Format vor. Sie können bei <http://www.gnu.org/manual/manual.html> oder <http://ftp.uni-bremen.de/pub/GNU/Manuals/> bezogen werden. Weiteres zu *info* im Abschnitt 5.11.3 *GNU Texinfo* auf Seite 210.

Wir kennen jetzt zwei Formate für Hilfetexte; das sind noch nicht alle. Als Vermittlungsschicht zwischen den Hilfedokumenten einerseits und benutzerseitigen Help Browsern wie dem KDE Hilfezentrum (*khelpcenter*, Symbol im KDE Hauptmenü oder auf der Taskbar ein Rettungsring) oder GNOME Yelp andererseits sieht sich der *ScrollKeeper*, ein *Electronic Cataloging System for Documentation*, zu Hause unter <http://scrollkeeper.sourceforge.net/>. Für Werkzeuge, die mit ihm zusammenarbeiten, legt er im Verzeichnis `/usr/share/omf/` Metadaten zur Dokumentation in einem lesbaren Format (*omf*) ab. Dabei handelt es

sich vorwiegend um Werkzeuge aus dem GNOME-Projekt. Abbildung 1.16 zeigt das KDE Hilfezentrum, Abbildung 1.17 Yelp mit jeweils einem vom ScrollKeeper vermittelten Hilfedokument. Mit Hilfe von ScrollServer, einem in Python geschriebenen HTTP-Server, lässt sich lokal oder in einem lokalen Netz auf die Datenbasis von ScrollKeeper mit jedem Web-Browser zugreifen. Näheres siehe <http://scrollserver.sourceforge.net/>, noch kein Debian-Paket verfügbar.

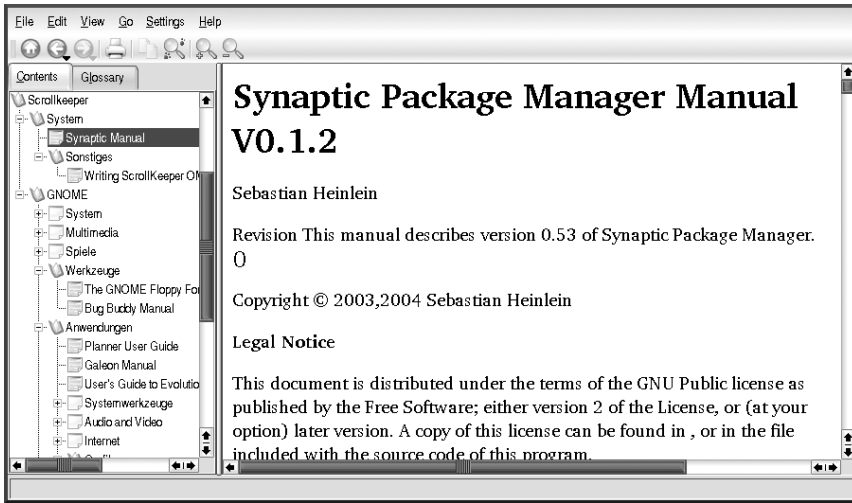


Abbildung 1.16. Screenshot des KDE Hilfezentrums mit einem über ScrollKeeper zur Verfügung gestellten Hilfedokument zur Paketverwaltung Synaptic

Die Linux-HOWTOs sind kürzere oder auch längere praxisnahe Anleitungen und Erläuterungen zu zahlreichen Themen aus der Linux-Welt. Diese und weitere Dokumente werden von The Linux Documentation Project (TLDP) erstellt und gepflegt, das unter <http://www.tldp.org/> zu erreichen ist. Deutsche Ableger sind unter <http://www.linuxhaven.de/dlhp/> und <http://www.debianhowto.de/> zu finden. Die Debian-Pakete *doc-linux-text* und *doc-linux.html* enthalten die wichtigsten HOWTOs und FAQs. Vermutlich sind auf Ihrem Rechner im Verzeichnis `/usr/share/doc/HOWTO/` einige englische oder deutsche HOWTOs gespeichert, im HTML-Format, also mit einem Web-Browser zu lesen, oder als gepackte Textdatei, mit `zless` zu lesen. Die meisten HOWTOs richten sich an Verwalter; einige auch für Anwender bedeutsame HOWTOs sind:

- 3D-Modelling,
- Astronomy,
- CD-Writing,
- Emacs Beginner,
- German,
- Linux Gamers,

- MP3,
- Scientific Computing with GNU Linux,
- Unix and Internet Fundamentals.

An entsprechenden Stellen im Buch verweisen wir auf das jeweils dazu gehörende HOWTO.

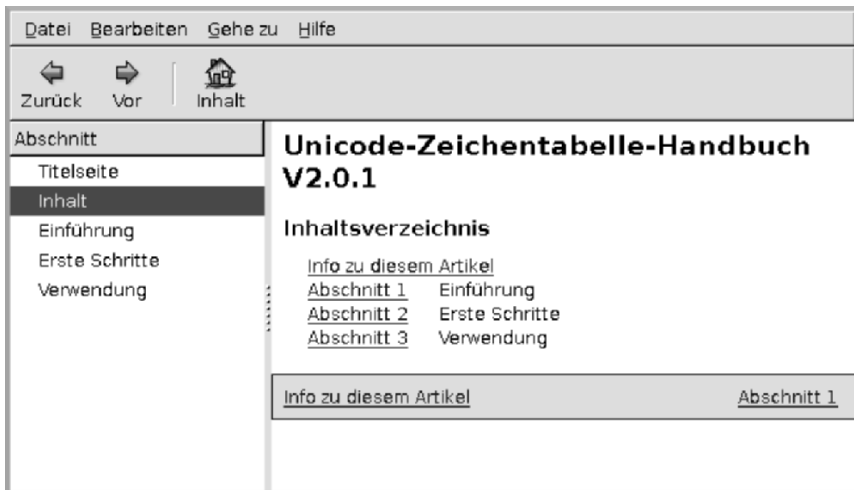


Abbildung 1.17. Screenshot von GNOME Yelp mit einem über ScrollKeeper zur Verfügung gestellten Hilfedokuments zur Zeichentabelle Unicode

Wenn ein HOWTO an Umfang und Reife zunimmt, wird manchmal ein Linux Guide daraus, der sich vor keinem Buch zu verstecken braucht. Einige für Anwender interessante Guides sind:

- Bash Guide for Beginners (2005),
- Introduction to Linux – A Hands on Guide (2005),
- Linux Dictionary (2004).

Die Guides sind in verschiedenen Formaten verfügbar, meist als HTML und PDF.

Zu vielen Werkzeugen stehen auch eigene Webseiten der Entwickler im Netz. Wo wir eine derartige Seite gefunden haben, erwähnen wir sie, da man dort – unabhängig von Debian – den Stand der Entwicklung erfährt und meist auch die neueste Version herunterladen kann. Diese wird nicht von der Debian-Paketverwaltung erfasst, der Benutzer ist für die Erfüllung von Abhängigkeiten selbst verantwortlich. Auf solchen Webseiten finden sich oft auch Links zu weiteren Informationen (FAQs usw.).

Ein noch junges Projekt im Web ist *SelfLinux*, gestaltet nach dem erfolgreichen Werk *SelfHTML* von STEFAN MÜNZ und unter dem URL <http://www.selflinux.org/> zu erreichen. Es bezeichnet sich als Hypertext-Tutorial zu GNU/Linux und hat den Vorzug, Deutsch zu sprechen.

Die Sammlungen von *Frequently Asked Questions* (FAQ) stammen aus den Netnews, einem Internet-Dienst. Weil dort in vielen Newsgruppen immer wieder dieselben Fragen auftauchen, machen sich nette Menschen die Arbeit und stellen die Fragen samt Antworten in Dokumenten zusammen, die regelmäßig in die jeweilige Newsgruppe gepostet und zusätzlich auf einem Webserver wie <http://www.faqs.org/> abgelegt werden, beispielsweise die siebenteilige UNIX-FAQ unter <http://www.faqs.org/faqs/unix-faq/faq/part1/index.html>. Manche FAQs umfassen mehrere hundert Seiten; sie sind dann nach Themen untergliedert. Die FAQs helfen in vielen Situationen, aber sie wollen kein Lehrbuch ersetzen.

Das Internet wird in Requests For Comments (RFC) beschrieben, zur Zeit etwa 4000 an der Zahl. Das sind überwiegend technisch gehaltene und nur Fachleuten verständliche Textdokumente. Einige der RFCs wenden sich jedoch an eine breite Leserschaft. Im Anhang haben wir eine Auswahl aufgelistet. Eine neue Fassung oder Version eines RFC bekommt eine neue Nummer – anders als DIN-Normen – weshalb viele der frühen RFCs heute nicht mehr gelten und durch neue RFCs abgelöst sind. Die RFCs bekommt man unter anderem bei <ftp://ftp.nic.de/pub/rfc/> oder <http://www.faqs.org/rfcs/>. Wenn man einer Frage betreffend das Internet auf den Grund gehen will, sind die RFCs eine unerlässliche Lektüre. Der erste RFC stammt von 1969, das Internet ist so alt wie UNIX.

Will man schnell wissen, was ein Fachbegriff aus der Informatik bedeutet, so ist die deutsche oder die internationale (englische) Wikipedia eine hervorragende Wissensquelle geworden:

- <http://de.wikipedia.org/wiki/>
- <http://en.wikipedia.org/wiki/>
- <http://fr.wikipedia.org/wiki/>

Die Wikipedien sind eine Art von allgemeinem enzyklopädischem Lexikon im Netz, das von vielen freiwilligen Autoren in lockerer Weise zusammengestellt wird. Jeder Leser darf auch schreiben, das heißt bestehende Einträge ändern oder neue Einträge verfassen. Infolgedessen schwankt die Qualität der Beiträge erheblich, auch werden manche Wissensgebiete bis jetzt nur lückenhaft abgedeckt. Trotzdem lohnt sich der Versuch, dort nachzulesen. Eine Wikipedia ist aktueller als gedruckte Werke – gelegentlich sogar tagesaktuell – und nicht wie diese vom Platz her beschränkt. Die englische Wikipedia umfasst über 600.000 Begriffe, die deutschsprachige rund 300.000, die französischsprachige nähert sich den 150.000. Von Zeit zu Zeit (etwa jährlich) wird die deutsche Wikipedia als DVD herausgegeben. Die Ausgabe vom Frühjahr 2005 ist unter der ISBN 3-89853-020-5 im Buchhandel preiswert erhältlich. Man lese aber vor dem Kauf online die technischen Rahmenbedingungen nach, die Scheibe ist nicht auf jedem Rechner unter jedem Betriebssystem benutzbar. Der Inhalt kann auch kostenfrei heruntergeladen werden, was wegen des Umfangs einiger Vorüberlegungen bedarf. Mit der DVD oder dem Download erspart man sich beim Nachschlagen die Verbindung zu den häufig überlasteten Wikipedia-Servern. Gelegentlich stellt die Wikipedia aus ihren Beiträgen Bücher zusammen, die man herunterladen oder in altmodischer Buchform gegen

ein billiges Entgelt erwerben kann. Für uns interessant ist der *WikiReader Internet* (2 MB, 200 Seiten) mit dem URL <http://www.wikipedia.org/wiki/Wikipedia:WikiReader/Internet>. Zur Einrichtung eines eigenen Wikis siehe Abschnitt 7.6 *Wikis* auf Seite 284.

BerliOS OpenFacts ist eine freie Wissensdatenbank über Open Source Software in Form eines Wikis, einer Wikipedia ähnlich, und über <http://openfacts.berlios.de/> zu erreichen. Der Wiki wurde 2003 vom BerliOS-Projekt geschaffen. OpenFacts überschneidet sich teilweise mit den allgemeinen Wikipedien und auch mit The Linux Documentation Project, aber da es nicht um Geld geht, arbeiten die verschiedenen Projekte zusammen, nicht gegeneinander. Im Sommer 2005 umfasste OpenFacts 200 Einträge, ist also noch ausbaufähig.

Auf GNU/Linux spezialisiert ist der LinuxWiki mit den URLs <http://linuxwiki.org/> und <http://www.linuxwiki.de/>. Das Projekt begann 2002 in Deutschland, umfasste Mitte 2005 rund 5000 Einträge und ist dabei, Ableger in anderen Sprachen hervorzubringen. Obwohl die Einträge deutschsprachig sind, ist die Beschriftung der Seiten englisch; das ist bei den großen Wikipedien schöner gelöst. Der Wiki versteht sich auch als eine Online-Linux-User-Group und bietet mehrere Wege zur Kontaktaufnahme zwischen Linux-Benutzern an. Die Syntax ist eine große Untermenge der Wikipedia-Syntax.

Auch das Debian-Projekt betreibt einen Wiki-Server (<http://wiki.debian.net/>) mit Hilfe zu zahlreichen Themen, meist auf Englisch. Eine ähnliche Unterstützung bietet das deutschsprachige Debian-Forum (<http://www.debianforum.de/>). Wer auf einem System unter Debian GNU/Linux arbeitet, sollte sich mit den vielfältigen Hilfen vertraut machen, denn kein Buch kann alle Einzelheiten zu allen Kommandos beschreiben (und dazu noch auf dem Laufenden sein). Wenn Sie an aktuellen Nachrichten aus der Debian-Welt interessiert sind, sind die *Debian Weekly News* von <http://www.de.debian.org/News/weekly/> für Sie das Richtige. Eine ähnliche Site mit Nachrichten aus der Debian-Welt ist <http://www.debianplanet.com/>.

Schließlich gibt es Fachleute, die man fragen kann. Einen schnellen Zugang findet der ratsuchende Benutzer über die *Linux User Groups* (LUG), die es auf der ganzen Welt gibt. Eine Übersicht liegt unter <http://lugww.counter.li.org/>, der Startseite der *Linux Users Groups WorldWide*. Anfang 2005 zählte sie 675 Gruppen in 93 Ländern auf. Wenn Sie eine LUG in Westmecklenburg, auf Malta oder in den Vereinigten Arabischen Emiraten suchen, wird Ihnen hier geholfen. Einige LUGs stellen auch Dokumente oder Linksammlungen ins Web.

Viel Information kommt aus dem Netz. Es ist unsinnig, diese Informationen sofort auszudrucken und abzuheften; das sollte man sich abgewöhnen. Wichtige Dokumente lokal oder im LAN zu speichern, kann Zeit sparen. Lokal gespeicherte Informationen stellt jeder Web-Browser dar, wenn man im URL als Protokoll `file://` anstelle von `http://` angibt. Ein kleiner, lokaler Webserver kann auch praktisch sein, in dessen Webpace – eventuell unter `/usr/share/doc/` – man Dokumente kopiert, die man haben will, ohne sie sofort zu Papier zu bringen. Schauen Sie sich einmal in der Debian-Paketliste das Paket *micro-httpd* an.

1.6 Eine Sitzung

Der Rechner sei eingeschaltet, hochgefahren und damit betriebsbereit⁸. Vorausgesetzt, Ihr System verwendet eine grafische Benutzeroberfläche statt einer schlichten Kommandozeile, sehen Sie einen Schirm wie in Abb. 1.18 auf Seite 43. Das genaue Aussehen hängt vom Display-Manager (xdm, gdm, kdm, wdm) und seiner Konfiguration ab.



Abbildung 1.18. Bildschirm zum Anmelden (login) auf einem Debian-System, hier ein Schirm des GNOME Display Managers gdm. Der Hintergrund ist blau, die Blume gelb.

Der KDE-Display-Manager kdm erlaubt unter dem Button Menu die Auswahl eines Window-Managers. Default ist der KDE-Window-Manager, aber auch der GNOME-Window-Manager wird angeboten. Ferner ist es möglich, bei Schwierigkeiten mit X11 eine schlichte Kommandozeile ohne Grafik zu starten. Falls der Verwalter bereits ein Konto (E: account, F: compte) für Sie angelegt hat, melden Sie sich nun an (loggen sich ein), das heißt, Sie geben Ihren Benutzernamen ein, schalten mit der Tabulator- oder Return-Taste weiter, geben Ihr geheim zu haltendes Passwort ein und bestätigen mit der Return-Taste. Dann fängt die Maschine mächtig zu arbeiten an und stellt schließlich einen Desktop auf dem Schirm dar, dessen Aussehen vom gewählten Window-Manager (twm, fvwm, icewm, enlightenment ...) und seiner Konfiguration abhängt. Nun sind Sie an der Reihe.

⁸Es bekommt der Elektronik nicht gut, wenn sie oft ein- und ausgeschaltet wird. Bei kürzeren Unterbrechungen den Rechner eingeschaltet lassen. Das Starten und Beenden einer Sitzung dagegen ist eine reine Software-Angelegenheit ohne Einfluss auf die Lebensdauer der Elektronik.

Wenn Sie das K Desktop Environment (KDE) gewählt haben, finden Sie links unten auf der Taskbar ein K mit einem Zahnrad dahinter. Die Taskbar kann aber auch am linken, rechten oder oberen Rand liegen, alles vom Benutzer konfigurierbar. Anklicken lässt ein langes Menü emporschnellen, in dem Sie zwischen vielfältigen Programmen oder Programmgruppen wählen können. Wichtig sind ein Web-Browser wie Mozilla, Galeon, Firefox, Konqueror oder Opera, um nur einige zu nennen, eine Shell, meist unter *Dienstprogramme* oder *Utilities* verbuddelt, und der Menüpunkt *Logout* oder *Abmelden*, um die Sitzung zu beenden. Den Rest erkundet man später.

Unter dem GNU Network Object Modelling Environment (GNOME) sieht alles etwas anders aus, aber die Funktionen sind die gleichen. Meist muss man zweimal klicken, um eine Anwendung zu starten. Auch die Maustasten sind anders belegt. Ob die Anzeige eines Datei-Managers mit einem Bildchen zu jeder Datei hilfreicher ist als die Text-Ausgabe von `ls`, wagen wir zu bezweifeln, aber das ist eine Frage der Gewohnheit. Arbeiten kann ein Benutzer mit beiden Arten der Darstellung.

Wir starten ein Fenster mit einem Textterminal entweder durch Anklicken eines Bildchens in der Taskbar, das einem Bildschirm ähnelt, durch Klicken mit der rechten Maustaste auf den GNOME-Hintergrund oder durch Aufrufen eines Menüpunktes, der *XTerm*, *Konsole*, *Bash* oder *Shell* lautet. Wir wollen einfach einen altmodischen Bildschirm zur Eingabe von Kommandozeilen bekommen, wie er vor fünfundzwanzig Jahren üblich war. Sind wir so weit, tippen wir einige harmlose Kommandos ein:

```
joe@debian:~$ id
joe@debian:~$ uname -a
joe@debian:~$ date
joe@debian:~$ pwd
joe@debian:~$ ls -la
joe@debian:~$ man ls
```

Nicht vergessen, jede Zeile mittels der Taste <cr> abzuschicken. Das Kommando `id` nennt uns unsere Identität, das Kommando `uname -a` Rechnername und Betriebssystem, das Kommando `date` Tag und Uhrzeit, das Kommando `pwd` unser augenblickliches Arbeitsverzeichnis, vermutlich mit unserem Home-Verzeichnis übereinstimmend. Das Kommando `ls -la` listet das Arbeitsverzeichnis auf, infolge der Optionen in ausführlicher Form und mit allen Dateien. Schließlich bringt uns das Kommando `man ls` die Manualseite zu `ls` auf den Schirm. Da die Manualseite länger ist als der Bildschirm, schalten wir mit der Leertaste weiter. Vielleicht funktionieren sogar die beiden Tasten <Page up> und <Page down> rechts auf der Tastatur, auf deutschen Tastaturen mit <Bild> bezeichnet. Wir verlassen die Manualseite durch Eingabe von <q> (quit). Das Terminalfenster beenden wir mit dem Kommando:

```
joe@debian:~$ exit
```

oder durch Eingabe der Tastenkombination `<ctrl>-<d>` oder durch Anklicken des Buttons rechts oben, der ein X zeigt. Unsere Sitzung ist damit noch nicht beendet. Die Vielfalt der Möglichkeiten verwirrt den Anfänger und rührt daher, dass Linux/UNIX mit vielerlei Hard- und Software zurechtkommen muss und viele Programmierer aus allen Teilen der Welt ihre Ideen eingebracht haben. Es hat niemals einen Big Boss gegeben, der entschieden hätte *So wird's gemacht*. Jeder Benutzer möbliert seinen Rechner und seinen Bildschirm anders.

Ist Ihr Rechner anderen Benutzern zugänglich und wollen Sie sich kurzzeitig von Ihrem Arbeitsplatz entfernen, ohne sie zu beenden, sperren Sie Ihre Sitzung durch Anklicken des Punktes *Sitzung sperren* im KDE-Menü oder durch den Aufruf:

```
joe@debian:~$ kdesktop_lock --forcelock
```

Der Schirm wird schwarz, jede Tastatureingabe oder Mausbewegung bringt ein kleines Fenster hervor, in das Sie Ihr Passwort eingeben müssen, um die Sperre zu lösen.

Zum Abmelden (ausloggen) ist in obengenanntem Menü der Punkt Abmelden oder logout anzuklicken. Aus der Kommandozeile heraus könnte man seinen X-Session-Manager killen, aber das tut man nicht. Arbeitet man ohne grafische Benutzeroberfläche (ohne X11), lautet das Kommando wie oben im Terminalfenster `exit` oder `<ctrl>-<d>`. Das Ausschalten des Rechners ist eine völlig andere Geschichte und Aufgabe des Verwalters. Den Bildschirm dagegen können Sie bei längeren Unterbrechungen unbesorgt ausschalten. Meistens geht er von sich aus nach einer gewissen Ruhezeit in einen stromsparenden Modus, aus dem er mit erneutem Tätigkeitsbeginn aufwacht. So genannte Bildschirmschoner schonen den Bildschirm nicht, sie sind für ihn Arbeit wie jede andere. Ihr einziger Vorteil liegt darin, dass sie von Zeit zu Zeit das Bild wechseln und so ein Einbrennen verhindern.

1.7 Sicherheit

Der Schutz der Daten auf einem Rechner vor Verlust, Verfälschung oder Missbrauch ist eine Aufgabe, die in erster Linie den Verwalter angeht, aber auch der einzelne Benutzer kann und muss im eigenen Interesse seinen Teil dazu beitragen. Sicherheit ist immer mit Unbequemlichkeiten verbunden. Sie kostet Zeit, Geld und Nerven. Unsicherheit kostet langfristig mehr. Jeder Benutzer soll:

- ein sicheres Passwort wählen, das von allen Zeichenarten des US-ASCII-Zeichensatzes Gebrauch macht, nicht zu kurz (8 Zeichen oder mehr) und kein Wort aus einem Wörterbuch ist, Das Passwort ist der Haustürschlüssel zum Rechner. Falls Sie Benutzer und Verwalter in einer Person sind, wählen Sie verschiedene Passwörter für beide Tätigkeiten. Verwenden Sie für Verbindungen zu externen Diensten (Amazon, Ebay, ZUM ...) nicht Ihr Rechnerpasswort.
- sein Passwort keinem anderen Benutzer überlassen (Das Dorf weiß, was drei wissen. Das steht schon in der Edda.),



Abbildung 1.19. Wichtige Mail einer Bank. Obwohl Text, als angehängte gif-Grafik verschickt. Der Absender ist gefälscht.

- Zugriffsrechte auf Dateien und Verzeichnisse so sparsam (restriktiv) vergeben wie möglich,
- selbst für eine aktuelle Sicherungskopie (E: backup copy, F: sauvegarde) wichtiger Dateien auf einem zweiten Rechner und/oder einem entfernbaren Medium (CD/DVD, Memory Stick) sorgen,
- zweifelhafte Mails ungelesen löschen, nicht darauf antworten oder sonst wie reagieren,
- beim Versenden von Mails oder Postings vorher darüber nachdenken, an wen die Sendung geht (eine Person, viele Personen, alle Welt?), insbesondere beim Antworten auf Mails oder Postings,
- vertrauliche Mitteilungen nur verschlüsselt über das Netz schicken,
- kritische Webseiten (online-Banking) immer direkt oder aus eigenen Bookmark-Listen aufrufen, nie durch Anklicken von URLs oder Hyperlinks in einer Mail oder einem Posting, siehe Abbildung 1.19,
- mit persönlichen Daten zurückhaltend umgehen (auch bei Preisausschreiben, Kundenbefragungen und dergleichen)⁹,
- nicht alles glauben, was auf dem Bildschirm erscheint, vor allem, wenn es um den Geldbeutel geht.

Kettenbriefe, Vorschläge zur Geldanlage in Nigeria oder Appelle an die Tränendrüse (*Little girl dying*) werden dadurch nicht seriöser, dass sie per Email daherkommen. Besonders in den auf Reisen praktischen Internet-Cafés ist Zurückhaltung geboten: Der Gast weiß nicht, was der Betreiber oder ohne dessen Wissen ein Vorgänger an der Tastatur in den Rechner eingebaut hat. Auf keinen Fall Passwörter speichern lassen und möglichst zum Schluss die History löschen. Man braucht den bösen Buben die Arbeit nicht noch zu erleichtern.

⁹Im Web gibt es Firmen wie <http://www.web-detective.biz/>, die behaupten, eine Fülle von Informationen über eine Person herausbekommen zu können.

Da man nicht nur für seinen Rechner, sondern auch für manche Anwendungen oder Netzverbindungen ein Passwort benötigt und nicht ein einziges Passwort für alle Zwecke nehmen soll, ist das Gedächtnis bald überfordert. Ein Passwort-Manager oder -Safe muss her. Für einen Debian-Rechner stehen zur Auswahl:

- GNOME fpm,
- gringotts, Heimathafen <http://devel.pluto.linux.it/projects/Gringotts/>,
- KDE kwalletmanager,
- zsafe,
- passwordsafe von dem Kryptografie-Guru BRUCE SCHNEIER, im Original nicht für Linux verfügbar, aber als Quelle und in einer Portierung, siehe <http://www.schneier.com/passsafe.html> und <http://passwordsafe.sourceforge.net/>,
- daraus abgeleitet mypasswordsafe (<http://www.semanticgap.com/myps/>).

SCHNEIERS Website ist immer einen Besuch wert, wenn es um Fragen der Sicherheit oder der Kryptografie geht.

Laptops sind handlich und geraten leicht in unrechte Hände. Der ehemalige Besitzer hat dann einen doppelten Schaden: Wenn er nicht über eine aktuelle Sicherungskopie (Backup) verfügt, fehlen ihm unter Umständen wertvolle Daten, und der unrechtmäßige Besitzer freut sich nicht nur über die Hardware, sondern auch über einen Schatz vertraulicher Daten als Dreingabe. Das Benutzerpasswort schützt nicht ausreichend vor dem Missbrauch der Daten, da ist ein verschlüsseltes Dateisystem gefordert, siehe Abteilung *Utilities* der Debian-Paketliste oder <http://www.jetico.com/>, kommerziell, aber bezahlbar.

Zur Sicherheit gehört auch das wöchentliche Aktualisieren der Software. Unter Debian GNU/Linux ist das eine einfache Angelegenheit. Wenn die Datei `/etc/apt/sources.list` in Ordnung ist, Ihr Rechner über eine nicht zu langsame Internet-Verbindung verfügt (wie ADSL) und Sie root-Rechte haben, ist mit drei Kommandos in wenigen Minuten alles erledigt:

```
debian:~# apt-get update; apt-get distupgrade; apt-get
autoclean
```

Das Upgrade-Kommando fragt, ob es neue Software einrichten soll. Augenmerk sollten Sie auf die Pakete richten, die entfernt werden sollen, was selten vorkommt. Fragen Sie Ihren Verwalter, wie er es mit dem Upgrade hält, und treten Sie ihm notfalls auf die Füße.

Viren oder allgemein Malicious Software (Malware) spielt bis jetzt in der Linux/UNIX-Welt keine nennenswerte Rolle, aber das kann sich ändern. Zwei als Debian-Paket erhältliche Virens Scanner sind:

- AMaVIS, <http://www.amavis.org/>,
- ClamAV, <http://www.clamav.net/>.

Unter Linux/UNIX reicht es, einen Virens Scanner erst bei einem konkreten Verdacht einzurichten, wenn man sich ansonsten sicherheitsbewusst verhält. Der Betrieb eines Virens Scanners erfordert, seine Datenbank aktuell zu halten, um gegen neue Viren gewappnet zu sein. Fallen Sie nicht auf Virenwarnungen herein, die Ihnen von Unbekannten per Email zugesandt werden; das sind ausnahmslos Falschmeldungen (Hoax, <http://www.tu-berlin.de/www/software/hoax.shtml>) oder Virenträger. Falls Sie den Eindruck haben, von Viren heimgesucht zu werden, benachrichtigen Sie Ihren Verwalter. Es bringt nichts, wenn ein einzelner Benutzer sich schützt; das ganze lokale Netz muss entseucht und virenfest gemacht werden.

Ein Dialer oder Einwahlprogramm ist eine Software, die Ihren Rechner über das alte analoge Telefonnetz oder über ISDN (digitales Telefonnetz) mit dem Internet verbindet. Benutzer, deren Rechner nicht mit einem Telefonnetz verbunden sind (sondern ausschließlich über DSL, da gibt es nichts zu wählen), können diesen Absatz übergehen. Bei Voice over IP (Internet-Telefonie) muss man wieder Acht geben; denn die Dialer werden diese Möglichkeit nicht ungenutzt lassen. Neben nützlichen Dialern gibt es unfreiwillig oder unbemerkt eingerichtete Dialer, die Ihren Rechner gegen Ihren Willen mit kostenpflichtigen, ja sogar extrem teuren Telefonnummern verbinden. Lassen Sie nie zu, dass sich Software automatisch auf Ihrem System einrichtet, überwachen Sie Ihre Telefonverbindungen und lesen Sie in der deutschen Wikipedia unter dem Stichwort *Dialer* nach.

Unter Spoofing wird das Verbergen der eigenen, wahren Identität verstanden. Wer das tut, hat einen Grund dafür und oftmals keinen ehrenwerten. Eine Software kann Datenpakete mit gefälschtem Absender ins Netz schicken, ein Rechner kann unter falscher MAC- oder IP-Adresse oder falschem Namen auftreten, jemand kann versuchen, einen DNS-Server zu manipulieren, der Absender einer Mail kann gefälscht sein (häufig), der URL einer Webseite kann erlogen sein. Ein Benutzer sollte ein gesundes, nicht übertriebenes Misstrauen entwickeln und in kritischen Fällen Verschlüsselung und/oder Zertifizierung verwenden, siehe Abschnitte 3.6 *Secure Shell* auf Seite 110, 3.7 *Secure Sockets Layer, Transport Layer Security* auf Seite 112 und 5.14 *Verschlüsselung* auf Seite 233. Die Einrichtung dieser Mechanismen macht Arbeit, ihr Gebrauch kaum.

Phishing ist ein anderes oft gehörtes Wort, eine Art von modernem Trickbetrug. Ein Gauner versucht, Ihnen unter Vortäuschung einer vertrauenswürdigen Identität und plausibler Umstände vertrauliche Daten abzuluchsen: Passwörter aller Art, unverbrauchte Transaktionsnummern (TAN) usw., siehe Abbildung 1.19. Es ist eine Form von Identitätsklau. Selbst bei weniger geheimen Informationen wie Geburtstag oder Kundennummern muss man sich fragen, ob der Gegenüber diese Daten wirklich braucht. Er könnte sie auch nur wissen wollen, um in anderem Zusammenhang damit den Eindruck eines guten Bekannten zu erwecken und auf dem Wege des Social Engineering an Informationen zu gelangen. Ein technisch anderer Weg des Identitätsklaus ist das Pharming, bei dem ein Websurfer auf eine gefälschte Webseite gelockt wird. Es wäre schon einiges gewonnen, wenn ein Benutzer im Internet dieselbe Vorsicht walten ließe, die er auf der Straße einem Fremden gegenüber zeigt.

Mit dem *Linux Security HOWTO* (2004) können Sie das Thema Sicherheit vertiefen. Eine gute Adresse in Fragen der IT-Sicherheit ist das Bundesamt für Sicher-

heit in der Informationstechnik (<http://www.bsi.bund.de/>). Dort finden Sie Empfehlungen, die weit über das Scannen von Viren hinaus gehen. Im Internet beschäftigen sich die *Computer Emergency Response Teams* (CERT) mit Sicherheitsfragen:

- <http://cert.uni-stuttgart.de/>
- <http://www.dfn-cert.de/>
- <http://www.bsi.bund.de/certbund/>
- <http://www.us-cert.gov/>
- <http://www.cert.org/>

Der Heise-Verlag in Hannover betreibt einen Sicherheitsserver mit Informationen und Angeboten unter <http://www.heisec.de/>.

1.8 Nachrüsten von Software

1.8.1 Debian-Pakete

Wenn Sie ein Werkzeug vermissen, aber meinen, dass Sie nicht der Erste sind, der so ein Ding gebrauchen könnte, lohnt es sich, danach zu suchen. Die wichtigste Anlaufstelle ist die Webseite mit den Debian-Paketen: <http://www.debian.org/distrib/packages>. Die Seite bietet Suchfunktionen an; man kann aber auch nach Distribution (*oldstable*, *stable*, *testing*, *unstable* bzw. *woody*, *sarge*, *etch*, *sid*) und Sachgebieten geordnete Paketlisten von Hand durchwühlen. Bei den Bezeichnungen ist zu unterscheiden zwischen:

- dem Programmnamen, beispielsweise The GIMP, nur bei großen Programmen gebräuchlich,
- dem Paketnamen, hier *gimp*, wobei zu einem Programm oft mehrere Pakete gehören,
- dem Namen der Paketdatei wie *gimp_2.2.6-1_i386.deb*,
- den Namen der in einem Paket enthaltenen Dateien, deren Anzahl in die Hunderte gehen kann,
- dem Programmaufruf, hier *gimp*.

Das Mischen von Programmen aus verschiedenen Debian-Distributionen kann gut gehen, zieht aber manchmal Schwierigkeiten nach sich. Das Nachrüsten ist Aufgabe des Systemverwalters, weil man dazu Schreibrechte in Systemverzeichnissen braucht. Wenn Sie dem Verwalter mit konkreten Vorschlägen kommen, haben Sie und er es leichter. Ausführlich wird das ganze Paketwesen bei GANTEN + ALEX erklärt.

Vorausgesetzt man hat root-Rechte, geht das Nachrüsten einfach:

```
debian:~# apt-get update
```

```
debian:~# apt-get install paketname
```

Achten Sie darauf, ob eventuell bereits eingerichtete Pakete entfernt werden müssen. In diesem seltenen Fall zweimal überlegen, was man will. Schreiben Sie auf, welche Pakete Sie nachrüsten. Falls das Paket doch nicht das Wahre war, werfen Sie es wie unten beschrieben wieder hinaus.

Es gibt Debian-Pakete außerhalb der offiziellen Debian-Server. Diese Pakete stehen nicht auf den Paketlisten und nehmen nicht an der Paketverwaltung von `apt-get` teil, sie sind vielmehr mittels `dpkg` einzurichten:

```
joe@debian:~$ dpkg -i tuxcards_1.2-1_i386.deb
```

Achtung: `dpkg` verlangt den Dateinamen als Argument.

1.8.2 RPM-Pakete

Neben dem Debian-Paketsystem ist der Red Hat Package Manager (RPM, <http://www.rpm.org/>) verbreitet. Die Pakete tragen die Kennung `rpm` und lassen sich unter Debian GNU/Linux mittels `alien` meist problemlos einrichten:

```
debian:~# alien -i paket.rpm
```

Das Werkzeug konvertiert das `rpm`-Paket in ein Debian-Paket und richtet dieses ein. Die konvertierte Fassung wird anschließend gelöscht.

1.8.3 tar-Archive (Tarballs)

Haben Sie – beispielsweise bei <http://directory.fsf.org/>, dem FSF/UNESCO Free Software Directory, bei <http://ftp.uni-bremen.de/pub/GNU/>, bei <http://www.heise.de/software/> oder mit Hilfe einer Suchmaschine – Ihr gesuchtes Werkzeug als `tar`-Archiv – auch *Tarball* genannt – entdeckt (Kennung `tar`, `tar.gz` oder `tgz`), ist Arbeit angesagt. Die Vorgehensweise nach dem Herunterladen ist meist folgende:

```
joe@debian:~$ tar xzvf archivname.tar.gz
```

Danach findet man ein Verzeichnis mit dem Namen des Archivs ohne Versionsnummer vor. Man wechselt hinein:

```
joe@debian:~$ cd archivname
```

und liest zuerst die verschiedenen READMEs. Dann konfiguriert man das Makefile:

```
joe@debian:~$ ./configure
```

sieht sich das Makefile an, versucht, es halbwegs zu verstehen und ruft `make` auf:

```
joe@debian:~$ make
```

Möglicherweise bringt das Archiv eine Test- oder Checkfunktion mit, dann wäre so etwas wie:

```
joe@debian:~$ make test
```


angebracht. Die bisherigen Schritte darf jeder Benutzer ausführen, zum Einrichten (Installieren) sind root-Rechte erforderlich:

```
debian:~# make install
```

Schließlich räumt man auf:

```
joe@debian:~$ make clean
```

Die häufigsten Überraschungen beim Nachrüsten solcher Archive sind:

- Fehlende Header-Dateien oder Funktionsbibliotheken,
- obige Dateien sind zwar vorhanden, liegen aber im falschen Verzeichnis (Sym-links anlegen),
- es werden zusätzliche Hilfsprogramme benötigt (`gmake`, `patch`), dann erst einmal diese Programme nachrüsten,
- es ist zwar alles an Ort und Stelle, aber die Typen der Argumente oder Rückgabewerte sind anders, als sie das Archiv erwartet. Dann passen irgendwelche Versionen nicht zueinander, und es wird schwierig.

Immerhin konnten wir nach diesem Rezept einige Werkzeuge aus dem GNU-Projekt auf Linux- oder UNIX-Rechnern einrichten. Wenige Male haben wir aber auch das Handtuch geworfen. Falls Sie ein Debian-Paket aus dem Quellcode übersetzen, sind die Schritte ähnlich. Auf diesem Weg können Sie auch Programme in Ihrem Home-Verzeichnis nur für den eigenen Gebrauch einrichten, wozu Sie keine root-Rechte benötigen. Es versteht sich, dass auf diesem Wege eingerichtete Software nicht an dem Debian-Upgrade-Mechanismus teilnimmt.

Beim Nachrüsten von Software achte man auf die Herkunft und richte bei Zweifeln ein Programmpaket niemals auf dem Arbeitsplatz oder auf einem vernetzten Rechner ein, bevor es nicht in einer Quarantäne-Umgebung untersucht worden ist.

1.8.4 Entfernen von Software

Wenn man Software wieder loswerden möchte, sind im Grund ähnliche Schritte wie bei der Einrichtung durchzuführen. Hat man ein Debian-Paket mittels `apt-get` eingerichtet, so gibt man als root das Kommando:

```
debian:~# apt-get remove paket
```

Dabei werden nach Rückfrage alle zu dem Paket gehörenden samt unmittelbar von dem Paket abhängigen Dateien und Verzeichnisse gelöscht außer Konfigurationsdateien. Diese beanspruchen wenig Platz, haben oft Mühe gekostet und könnten bei einer erneuten Einrichtung des Paketes willkommen sein. Sollen auch sie verschwinden, ist:

```
debian:~# dpkg --purge paket
```

aufzurufen. Danach sind nur noch benutzereigene Dateien und Verzeichnisse in den Home-Verzeichnissen (beispielsweise `.emacs` oder `.emacs.d`) sowie etwaige von außerhalb Debian nachgeladene Dateien (beispielsweise Fonts) übrig. Um sie aufzuspüren, ist die Locate-Datenbank zu aktualisieren:

```
debian:~# updatedb
```

und dann mittels:

```
debian:~# locate name
```

nach verbliebenen Dateien zu suchen, wobei `name` ein Name ist, der als Datei- oder Verzeichnisname bei der Einrichtung des Paketes erzeugt worden ist, im einfachsten Fall der Paketname.

RPM-Pakete, Tarballs und dergleichen Software, die nicht mit `apt-get` eingerichtet worden ist, kann auch nicht mit diesem Werkzeug getilgt werden. Man muss wie eben beschrieben die Dateien und Verzeichnisse mit `locate` aufspüren und mit `rm` löschen, meist mit der Option `-rf` (recursive, force).

1.8.5 Software für MS Windows oder MacOS

Ist man auf Software angewiesen, die für ein fremdes Betriebssystem geschrieben ist, gibt es grundsätzlich zwei Wege, diese zum Laufen zu bringen:

- Das fremde Betriebssystem wird als Anwendung unter dem eigenen System gestartet und darunter die Software. Diesen Weg beschreiten der Microsoft Virtual PC für Macintosh-Rechner oder die VMWare Workstation, ein kommerzielles Produkt. Freie Alternativen sind Bochs und QEMU.
- Eine Anwendung unter dem eigenen Betriebssystem stellt die Systemaufrufe des fremden Betriebssystems bereit. Sofern die Software nicht an den Systemaufrufen vorbei programmiert ist (direkter Hardware-Zugriff), sollte das funktionieren. Diesen Weg geht der Emulator `wine`.

Um unter Debian GNU/Linux Anwendungen laufen zu lassen, die für MS Windows geschrieben sind, braucht man einen Windows-Emulator wie `wine` (<http://www.winehq.org/> und <http://www.linux-wine.de/>), gegebenenfalls mit einem Zusatz wie den Wine Tools (<http://www.von-Thadden.de/Joachim/WineTools/>). Ein Wine-HOWTO aus dem Jahr 2003 von JUTTA WRAGE liegt auf <http://www.la-sorciere.de/wine/>.

Zur Emulation der Macintosh-Betriebssysteme MacOS 7.5.2 bis 9.2.2 sowie MacOS X bis 10.3.3 steht das Debian-Paket `mol` (Mac-on-Linux, <http://maconlinux.org/>) bereit, allerdings nur für PowerPC-Hardware. Mit dem Debian-Paket `pearpc` lässt sich wiederum ein PowerPC auf 80386-Hardware emulieren. Ob die Kombination aus beiden Paketen funktioniert, ist fraglich. Weitere Emulationen (Amiga, Atari ...) finden sich in der Abteilung *Othersfs* der Debian-Paketlisten. Eine Emulation ist selten hundertprozentig und kostet Prozessorzeit, was bei den heutigen schnellen Prozessoren seltener auffällt.

Kommandointerpreter

Kommandointerpreter (Shells) sind die am häufigsten benutzten Anwendungen auf einem Linux/UNIX-System. Ihre Aufgabe und ihr Gebrauch beim Dialog am Terminal werden nachfolgend erläutert.

2.1 Bourne-again-Shell (bash)

2.1.1 Gesprächspartner

Auf Rechnern, die im Dialogbetrieb arbeiten (Arbeitsplatzrechner und andere), ist ein Kommando-Interpreter das am häufigsten benutzte Programm. Unter Debian GNU/Linux handelt es sich dabei fast immer um die Bourne-again-Shell, kurz Bash genannt. Für den Anfänger ist der *Bash Guide for Beginners* eine wertvolle Hilfe, zu beziehen über das Netz von <http://www.tldp.org/guides.html>, zuletzt aktualisiert im Dezember 2004 und in mehreren Formaten zu haben. Die pdf-Datei ist 1,2 MB groß und ergibt ausgedruckt 172 Seiten. Blutige Anfänger können vorher die *Introduction to Linux* desselben Autors aus derselben Quelle lesen, von ähnlichem Umfang.

Die wichtigste Aufgabe eines Kommando-Interpreters ist es, Eingaben des Benutzers als Kommandos (Befehle, Anweisungen) zu interpretieren und auszuführen. Dabei arbeitet die Shell folgende Schritte ab:

- Zuerst prüft sie, ob die Eingabe mit einem Aliasnamen (wird bald erklärt) beginnt. Falls ja, wird der Aliasname ersetzt.
- Ist das – unter Umständen ersetzte – Wort ein shell-internes (eingebautes) Kommando, wird es von der Shell selbst ausgeführt.
- Falls nicht, wird ein externes Kommando – eine Datei – in einem der in der PATH-Variablen (wird auch bald erklärt) aufgeführten Verzeichnisse gesucht. Bleibt die Suche erfolglos, erscheint auf dem Bildschirm eine Fehlermeldung: `command not found`.

- Hat die Suche Erfolg gehabt, werden die Zugriffsrechte der Datei untersucht. Falls diese dem Benutzer das Lesen und Ausführen gestatten, geht es weiter. Andernfalls: `Permission denied`.
- Die Zugriffsrechte seien ausreichend, die Datei enthalte ein Shellskript (wird auch bald erklärt) oder ein ausführbares (compiliertes) Programm. Dann lässt die Shell es in einem Kindprozess ausführen. Das Verhalten bei Syntaxfehlern (falsche Option, fehlendes Argument), ist Sache des Shellskripts oder Programms, hängt also davon ab, was sich der jeweilige Programmierer gedacht hat.
- Die Datei sei gefunden, aber eine Textdatei (Brief, Programmquelle, Manuskript) oder eine sonstige binäre Datei (Grafik, Sound). Dann bedauert die Shell, damit nichts anfangen zu können. Sie sieht die Datei als ein Shellskript oder Programm mit entsetzlich vielen Fehlern an. Dasselbe gilt für Gerätedateien oder Verzeichnisse.

Einige Kommandos führt die Shell selbst aus; dazu gehört `cd` (change directory). Logischerweise kann es dann keine Datei dieses Namens geben. Eingebaute Kommandos werden schneller ausgeführt als Shellfunktionen oder externe Kommandos (Programme). Erklärungen zu den shell-internen Kommandos finden sich auf der Manualseite zur Shell, nicht als selbständige Seiten. Die Menge der shell-internen Kommandos ließe sich nur durch Neuschreiben der Shell vergrößern. Externe oder selbständige Kommandos hingegen sind in eigenen Dateien untergebracht; ihr Kreis lässt sich beliebig und teilweise sogar recht einfach vergrößern.

Ein Alias ist ein Zweitname für ein internes oder externes Kommando. Der Aliasname wird von der Shell buchstäblich durch seine Definition ersetzt. Er gilt in der Shell, in der er angelegt wird, und in deren Abkömmlingen. Wird der Aliasname in einer Datei wie `/etc/profile` definiert, die von jeder Login-Shell abgearbeitet wird, gilt er für jeden Benutzer. Persönliche Aliasnamen kann der Benutzer in einer Datei wie `$HOME/.profile` oder `$HOME/.bashrc` definieren; sie gelten nur für seine Sitzung. Das Kommando bzw. die Zeile für diesen Zweck sieht so aus (keine Leerzeichen um das Gleichheitszeichen):

```
alias ll="ls -l"
```

Gibt man einem Alias beim Aufruf eine Option oder ein Argument mit, so macht man sich die Wirkung klar, indem man die Ersetzung selbst nachvollzieht. Beispielsweise zeigen die Zeilen:

```
alias lx="ls -l | sort"
lx verzeichnis
```

nicht den erwarteten Erfolg, weil das Argument `verzeichnis` dem Kommando `sort` übergeben wird anstatt dem Kommando `ls`. Verwendet man als Aliasnamen den Namen eines bereits vorhandenen Kommandos, ist dieses nur noch über seinen absoluten Pfad zu erreichen. Das Kommando `alias` allein zeigt alle derzeit gültigen Aliasnamen an.

In Dateinamen, die einem Kommando als Argument mitgegeben werden, haben einige Zeichen für die Shell eine besondere Bedeutung (Metazeichen, hier speziell auch Jokerzeichen oder Wildcards genannt). So bedeuten:

- der Stern (*) eine beliebige Anzahl – auch null – beliebiger Zeichen,
- das Fragezeichen (?) genau ein beliebiges Zeichen,
- der Ausdruck [abc] genau ein Zeichen aus der Zeichenmenge in den Klammern, hier die Kleinbuchstaben a, b oder c.
- der Ausdruck [0-9] genau ein Zeichen aus der Zeichenmenge in den Klammern, hier alle Ziffern,
- der Ausdruck [!0-9] oder [^0-9] (Ausrufezeichen oder Zirkumflex vorangestellt) das Komplement der Zeichenmenge, also alle Zeichen des ASCII-Alphabets, die nicht in den Klammern aufgeführt sind.

Die Interpretation der Jokerzeichen wird auch Pfadnamen-Erweiterung, Expansion oder Globbing genannt. Das betroffene Kommando merkt davon nichts, es bekommt von der Shell eine Liste der Dateinamen, auf die das Muster zutrifft. Ausgenommen von der Liste sind Dateinamen, die mit einem Punkt beginnen (verborgene Dateien). Findet die Shell im Arbeitsverzeichnis keinen passenden Namen, übergibt sie das Muster buchstäblich (ohne Globbing). Der Aufruf:

```
joe@debian:~$ ls abc*z
```

listet alle Dateien des Arbeitsverzeichnisses auf, deren Namen mit `abc` beginnt und mit `z` endet, beispielsweise:

```
abcz  abc1z  abc123z  abc.z  abc.gz  abcache.fiz  abc-heute_fuzz
```

Will man Dateinamen erwischen, die mit einem Punkt beginnen, muss dieser ausdrücklich genannt werden:

```
joe@debian:~$ grep muster .*rc
```

Der Aufruf sucht im Arbeitsverzeichnis in allen Dateien, deren Name mit einem Punkt beginnt und auf `rc` endet. Berührt ist die Eingabe:

```
joe@debian:~$ rm abc*
```

Gerät dem Benutzer aus Versehen ein Leerzeichen zwischen `abc` und den Stern, so werden im Arbeitsverzeichnis erst die Datei `abc` und dann alle Dateien gelöscht. Besonders wirkungsvoll wird das Kommando mit der Option `-rf` (rekursiv und ohne Rückfragen). Auch Leerzeichen und Tabulatorzeichen haben eine besondere Bedeutung; sie trennen die Glieder der Kommandozeile voneinander. Ebenso trennen ein Semikolon oder ein Zeilenwechsel Kommandos voneinander. Trotz der Ähnlichkeit sind diese Muster keine regulären Ausdrücke (siehe Abschnitt 4.1.2 *Reguläre Ausdrücke* auf Seite 116). Man kann für Dateinamen, die die Shell interpretieren soll, keine regulären Ausdrücke gebrauchen.

In der Bash bedeutet die allein stehende oder einem Pfad vorangestellte Tilde das Home-Verzeichnis des Benutzers, ist also ebenfalls ein Metazeichen. Die Eingabe:

```
joe@debian:~$ ls -ld ~/.*
```

in irgendeinem Arbeitsverzeichnis bewirkt die Auflistung aller Punktdateien und -verzeichnisse im Home-Verzeichnis des Benutzers, ohne in die Verzeichnisse hinabzusteigen. Ähnlich bedeutet ein allein stehender oder einem Pfad vorangestellter

Punkt das Arbeitsverzeichnis. Zwei Punkte bezeichnen das dem Arbeitsverzeichnis übergeordnete Verzeichnis. Der Aufruf:

```
joe@debian:~$ cd ..
```

befördert uns im Dateisystem ein Stockwerk nach oben.

Braucht man ein Metazeichen der Shell buchstäblich, muss man es quoten, das heißt ihm seine Sonderbedeutung nehmen. Für *quoten* wird auch sperren, zitieren, entwerten oder maskieren gesagt. Es gibt drei Stufen des Quotens:

- ein Gegenschrägstrich (backslash) quotet genau das nachfolgende Zeichen, auch einen zweiten Gegenschrägstrich oder ein einzelnes Hochkomma,
- ein Paar von Gänsefüßchen (double quote) quotet die eingeschlossenen Zeichen mit Ausnahme von \$ (Dollar), Accent grave (back quote), Gegenschrägstrich und Gänsefüßchen,
- ein Paar von Hochkommas oder Apostrophen (single quote) quotet alle eingeschlossenen Metazeichen außer dem Hochkomma (warum wohl?).

Am häufigsten benutzt man die Hochkommas, auch wenn vielleicht eine schwächere Quotung ausreichen würde. Ist ein Dateiname oder eine andere Zeichenkette mit Leerzeichen der Shell zu übergeben, so ist sie in Gänsefüßchen oder Hochkommas einzuschließen, damit die Shell sie nicht aufgliedert:

```
joe@debian:~$ mv 'dummer name' besserer-name
```

Unter Linux/UNIX sind in Dateinamen alle ASCII-Zeichen¹ erlaubt außer dem ASCII-Zeichen Nr. 0 (das intern eine Zeichenkette abschließt) und dem Schrägstrich, der die Teile eines Pfades (Verzeichnisnamen, Dateinamen) voneinander abgrenzt. Es wäre aber ungeschickt, in Dateinamen Leerzeichen, Jokerzeichen, Zeilenwechsel, Steuerzeichen (<ctrl> oder <esc>) oder Umlaute zu verwenden. Man handelt sich nur Schwierigkeiten ein.

Einen Ausdruck in der Kommandozeile, der zwischen rückwärts gerichteten Hochkommas (Accent grave, back quote) steht, ersetzt die Shell durch das Ergebnis dessen, was eingerahmt ist, sinnvollerweise ein Kommando. Der Mechanismus wird Kommando-Substitution genannt. Im Druck sind diese Hochkommas oft nur undeutlich zu erkennen, man muss genau hinschauen oder sich überlegen, was die Zeile bedeuten soll. Ein Beispiel:

```
joe@debian:~$ echo "Dateien: \n 'cat *debian*'"
```

Das eingebaute Shell-Kommando `echo` schreibt sein Argument nach `stdout` (Bildschirm). Das Argument ist hier zusammengesetzt und steht deshalb in Gänsefüßchen. Als erstes wird die Zeichenkette `Dateien: \n` geschrieben, in der `\n` ein Newline-Zeichen (Zeilenwechsel) bedeutet. Dann wird das Kommando `cat *debian*` ausgeführt, das alle Dateien, in deren Namen die Zeichenkette `debian` vorkommt, nach seinem `stdout` schreibt, hier also an die Stelle des Ausdrucks in

¹Genau genommen alle Zeichen aller Zeichensätze mit den beiden genannten Ausnahmen, aber viele Anwendungsprogramme bekommen Bauchweh mit exotischen Zeichensätzen.

den Backquotes einschließlich der Backquotes. Das `echo`-Kommando hat also unter Umständen viel Text auszugeben. Dass sich das Ergebnis auch anders erzielen lässt, steht außer Frage.

Die Bash kann als Kommando-Argument eingetippte, unvollständige Dateinamen vervollständigen. Veranlasst wird dies durch die `<tab>`-Taste. Folgender Aufruf im entsprechenden Arbeitsverzeichnis:

```
joe@debian:~$ acroread DE<tab>
```

führt zur Anzeige:

```
joe@debian:~$ acroread  
DE-Mehrere-Ethernetkarten-HOWTO.pdf
```

die dann mit `<cr>` wie üblich abgeschickt wird. Die Ergänzung funktioniert auch für Kommando-Namen. Bei mehreren Möglichkeiten der Ergänzung schreibt die Shell den Namen so weit, bis eine Entscheidung des Benutzers gefordert ist. Zweimaliges Drücken von `<tab>` an einer solchen Stelle zeigt alle Möglichkeiten an. Eine angenehme Hilfe bei länglichen Namen.

2.1.2 Eingebaute Kommandos

Welche Kommandos in die Shell eingebaut und welche durch externe Programme in eigenen Dateien verwirklicht werden, ist bis zu einem gewissen Maß eine Frage der Zweckmäßigkeit. Auf eingebaute (interne, builtin) Kommandos kann ein Alias, aber kein Link gesetzt werden. Sie werden im Prozess der jeweiligen Shell ausgeführt, nicht in einem eigenen Prozess. Die Bash kennt folgende eingebauten Kommandos:

- `#` (Doppelkreuz) leitet Kommentar bis zum Zeilenwechsel ein,
- `:` (Doppelpunkt, gefolgt von einem Leerzeichen) leeres Kommando,
- `. cmd` (Punkt - Leerzeichen - Kommando) liest die Kommandos im Shellskript `cmd` und führt sie selbst aus. Damit lassen sich beispielsweise in einem Skript Variable für die aktuelle Shell definieren. Ohne den Punkt würde das Skript in einem Kindprozess ausgeführt. Der Vorgang wird auch als *sourcen* bezeichnet; der Punkt darf durch das auffälligeres Wort *source* ersetzt werden,
- `alias name="definition"` definiert einen neuen Namen für ein Kommando,
- `bg` (background) Job in den Hintergrund schicken, siehe Abschnitt 3.2.3 *Jobkontrolle* auf Seite 98,
- `bind` bindet Tastenkombinationen an Funktionen (Einzelheiten im Manual),
- `break` verlässt eine Kommandoschleife (kommt in Shellskripten vor),
- `builtin` dient zur Unterscheidung eingebauter Kommandos von gleichnamigen externen Kommandos oder Funktionen,
- `cd` (change directory) wechselt das Verzeichnis,
- `command` nur eingebaute Kommandos oder solche, die über die `PATH`-Variable gefunden werden, werden ausgeführt,
- `compgen` beeinflusst die Ergänzung von Namen in der Kommandozeile,

- `complete` beeinflusst die Ergänzung von Argumenten in der Kommandozeile, Einzelheiten im Manual,
- `continue` springt an den Anfang einer Kommandoschleife zurück (kommt in Shellskripten vor),
- `declare, typeset` Deklaration von Variablen in Shellskripten,
- `dirs` (directories) Liste gespeicherter Verzeichnisnamen anzeigen,
- `disown` Jobkontrolle, siehe Abschnitt 3.2.3 *Jobkontrolle* auf Seite 98,
- `echo` schreibt seine Argumente nach `stdout`, gefolgt von einem Zeilenwechsel,
- `enable` ein eingebautes Kommando ein- oder ausschalten,
- `eval` Kommandos zu einem einzigen Kommando verbinden und dann ausführen,
- `exec` die Shell durch ein Kommando ersetzen und dann ausführen; die Shell ist damit weg,
- `exit` Shell beenden,
- `export` eine Shellvariable an Kindprozesse vererben,
- `fc` (fix command) Kommandos aus dem Kommandoverlauf (history) auswählen,
- `fg` (foreground) Job in den Vordergrund holen, siehe Abschnitt 3.2.3 *Jobkontrolle* auf Seite 98,
- `getopts` Optionen aus der Kommandozeile herausfischen,
- `hash` Namen von Kommandos speichern, um sie schneller wiederzufinden,
- `help` kurze Hilfe zu einem eingebauten Kommando anzeigen,
- `jobs` aktive Jobs anzeigen,
- `kill` Signal an Prozess schicken, siehe Abschnitt 3.2.4 *Signale* auf Seite 100,
- `let` arithmetischen Ausdruck auswerten.
- `local` lokale Variable in einer Funktion erzeugen,
- `logout` login-Shell beenden, geht auch mit `exit`,
- `popd` Eintrag von Verzeichnis-Speicher (stack) entfernen,
- `printf` (print formatted) ähnlich wie `echo` mit mehr Möglichkeiten zur Formatierung der Ausgabe (C-Programmierer kennen die Anweisung),
- `pushd` Verzeichnis auf Verzeichnis-Speicher (stack) legen,
- `pwd` (print working directory) Arbeitsverzeichnis nennen (es gibt auch ein externes Kommando `/bin/pwd`),
- `read` von `stdin` lesen (kommt in Shellskripten vor),
- `readonly` Variable als nur lesbar erklären,
- `return` eine Funktion verlassen,
- `set` Shellvariable auflisten, Optionen setzen,
- `shift` Positionsparameter nach links schieben, wird in Shellskripten gebraucht,
- `shopt` Shell-Optionen anzeigen, ein- und ausschalten,
- `suspend` Ausführung der Shell unterbrechen (suspendieren),
- `test, []` auf Bedingung testen, meist nach einem `if`,
- `times` von der Shell verbrauchte Prozessor-Zeit ausgeben,
- `trap` Verhalten der Shell nach Eintreffen eines Signals definieren,
- `type` Kommando untersuchen, aber nicht ausführen (`type pwd`),
- `ulimit` Obergrenze für Größe von Dateien setzen,

- `umask` Defaultwert für die Zugriffsrechte zu erzeugender Dateien oder Verzeichnisse setzen, und zwar als oktaler Komplementärwert (Rechte = 750 erfordert `umask=027`),
- `unalias` Alias löschen,
- `unset` Variable oder Funktion löschen,
- `wait` auf die Beendigung eines Prozesses warten.

Auf der Manualseite zur Bash findet man die Beschreibung der internen Kommandos im Abschnitt *SHELL BUILTIN COMMANDS*; sie haben keine eigenen Manualseiten. Man kann jahrzehntelang mit Shells arbeiten und hat vielleicht die Hälfte der internen Kommandos gebraucht. Einige werden praktisch nur in Shellskripten verwendet, nicht im Dialog.

Das Kommando `echo` ist ein eingebautes Shellkommando, aber auch ein externes Kommando `/bin/echo` mit eigener Manualseite. Mittels:

```
joe@debian:~$ builtin echo Hallo
```

erzwingt man den Aufruf des eingebauten Shellkommandos, mittels:

```
joe@debian:~$ /bin/echo Hallo
```

den Aufruf des externen Kommandos. Beide unterscheiden sich geringfügig. Beispielsweise lässt sich nur mit dem eingebauten Shellkommando die Farbe der Ausgabe beeinflussen:

```
joe@debian:~$ builtin echo -e "\e[31mHallo\e[m"
```

Das Vorkommando kann möglicherweise weggelassen werden. Das Argument enthält je eine Escape-Sequenz zur Umschaltung der Farbe und am Ende zum Zurücksetzen. Die Escape-Sequenzen (XTerm Control Sequences) werden in `/usr/share/doc/xtermcontrol/ctlsqs.txt`, Stichwort *Character Attributes*, erläutert.

2.1.3 Wiederholen und Editieren von Kommandos

Die alte C-Shell und alle modernen Shells beherrschen einen History-Mechanismus (Kommando-Verlauf), der die eingegebenen Kommandos in chronologischer Folge zwischenspeichert und beim Beenden der Sitzung in einer Datei `.bash_history` im Home-Verzeichnis des Benutzers ablegt. Wer über die nötigen Leserechte verfügt, kann dort nachlesen, was ein Benutzer in der jüngeren Vergangenheit getrieben hat. Mausclicks werden nicht gespeichert, nur Eingaben in die Kommando-Zeile. Web-Browser führen eine eigene History, das ist eine andere Geschichte.

Die gespeicherten Kommandos lassen sich erneut aufrufen. Von Bedeutung sind jedoch nur die jüngsten Kommandos, vielleicht drei. Man erreicht sie mittels der Cursortaste `↑`. Das jüngste Kommando lässt sich auch durch zwei Ausrufezeichen wiederholen (schließt `<cr>` mit ein).

Eine Kommandozeile kann vor dem Abschicken mit `<cr>` editiert werden. Das braucht man, wenn man sich bei der Eingabe eines Kommandos vertippt und dieses

mit der Cursortaste zurück geholt hat oder wenn man ein Kommando mit geänderten Argumenten wiederholen möchte. Durch Setzen der Umgebungsvariablen `EDITOR` oder `FCEDIT` legt man fest, welcher Editor verwendet werden soll. Ist keine der Umgebungsvariablen gesetzt, funktionieren wenigstens die Cursortasten und die Rücktaste (backspace), was oft ausreicht.

2.1.4 Umlenkung

Wenn wir mit Dateien arbeiten, müssen unsere Werkzeuge und Programme wissen, mit welchen. Dafür tragen die Dateien Namen, die innerhalb eines Dateisystems eindeutig sind, zumindest, wenn man den absoluten Pfad von der Wurzel des Dateisystems bis zur Datei berücksichtigt. Es ist unmöglich, innerhalb eines Verzeichnisses einen Dateinamen zweimal zu vergeben².

Will ein Programm (genauer: ein Prozess) auf eine Datei zugreifen, muss es diese als erstes öffnen, geradeso wie Sie ein Buch aufschlagen, wenn Sie darin lesen wollen. Der Name der zu öffnenden Datei wird in einem Datei-Pointer gespeichert, das ist ein Pointer auf eine C-Struktur von Typ `FILE`, was nur Programmierer etwas angeht. Daneben vergibt ein Programm fortlaufende, bei Null beginnende Nummern für die von ihm geöffneten Dateien, Datei-Bezeichner, -Deskriptoren oder Handles genannt.

Unsere Sitzungsshell öffnet für jeden von ihr gestarteten Prozess drei Dateien mit den Datei-Pointern `stdin` (Standard-Eingabe), `stdout` (Standard-Ausgabe) und `stderr` (Standard-Fehlerausgabe) und den Datei-Bezeichnern 0, 1 und 2. Andere Systeme kennen noch `stdaux` (Standard Auxiliary Device, erste serielle Schnittstelle) und `stdprn` (Standard-Printer, erste parallele Schnittstelle), braucht Linux/UNIX nicht. Hinter der Datei `stdin` steckt zunächst die Tastatur, hinter `stdout` der Bildschirm und hinter `stderr` ebenfalls der Bildschirm. Das entspricht unserer gewöhnlichen Arbeitsweise im Dialog mit Prozessen: wir geben auf der Tastatur ein und lesen vom Bildschirm. Die Trennung von `stdout` und `stderr` hat den Vorteil, dass wir die Ergebnisse einer Rechnung (Nutzdaten) von den Fehlermeldungen unterscheiden können. Viele Programme sind so geschrieben, dass sie von `stdin` lesen und nach `stdout` bzw. `stderr` schreiben; sie werden Filter (in einem allgemeinen Sinn) genannt. Ein Beispiel ist das Dienstprogramm `cat`.

Mit Hilfe der Shell lassen sich die hinter den drei Pointern oder Bezeichnern stehenden Geräte für einen Prozess umdefinieren. Der Mechanismus wird Umlenkung (E: redirection) genannt. Der Prozess (das Kommando) erhält von der Shell die bereits umgelenkten Pointer oder Bezeichner. Das hat den Vorteil, dass man sich beim Schreiben eigener Kommandos nicht um den Umlenkungsmechanismus zu kümmern braucht. Das Kommando `cat` schreibt nach `stdout`. Ein Aufruf wie:

```
joe@debian:~$ cat textdatei
```

schreibt den Inhalt der Datei `textdatei` in einem Rutsch auf den Bildschirm. Lenken wir `stdout` um:

²Der Verwalter könnte es hinbekommen, aber der wird sich hüten.

```
joe@debian:~$ cat textdatei > neue_datei
```

so schreibt `cat` immer noch nach `stdout`, aber dahinter steckt jetzt die Datei `neue_datei`. Falls sie noch nicht existiert, wird sie angelegt. Existiert sie schon, wird sie von Anfang an überschrieben, ein etwaiger Inhalt geht verloren. Das lässt sich mit folgendem Aufruf vermeiden:

```
joe@debian:~$ cat textdatei >> neue_datei
```

der den Inhalt von `textdatei` an einen etwaigen Inhalt von `neue_datei` anhängt. Auch hier wird eine noch nicht existierende Datei neu angelegt. Für die Umlenkung von `stdin` wird der entgegengesetzte Pfeil verwendet:

```
joe@debian:~$ sort < datei1 > datei2
```

Das Kommando `sort` liest die Datei `datei1`, sortiert sie zeilenweise und schreibt das Ergebnis in die Datei `datei2`. Auf die Reihenfolge der beiden Umlenkungen kommt es dabei nicht an. Eine Umlenkung der Ausgabe nach `/dev/null`:

```
joe@debian:~$ date > /dev/null
```

führt zur rückstandlosen Entsorgung der ausgegebenen Bytes. Bei einer Umlenkung nach `/dev/recycl` lassen sich die Bytes wiederverwenden, nicht auf jedem System eingerichtet. Gelegentlich benötigt man eine Umlenkung von `stderr` nach `/dev/null`. Lesen aus `/dev/null` liefert ein EOF-Zeichen (End Of File).

Die Pfeile lassen sich auch zur Verbindung von Datei-Bezeichnern verwenden. Beispielsweise verbindet:

```
joe@debian:~$ command 2>&1
```

den Datei-Bezeichner 2 (`stderr`) des Kommandos `command` mit dem Datei-Bezeichner 1 (`stdout`). Die Fehlermeldungen von `command` landen in derselben Datei wie die eigentliche Ausgabe. Lenkt man noch `stdout` um, so spielt die Reihenfolge der Umlenkungen eine Rolle. Die Eingabe:

```
joe@debian:~$ command 1>output 2>&1
```

lenkt zunächst `stdout` (Datei-Bezeichner 1) in die Datei `output`. Anschließend wird `stderr` (Datei-Bezeichner 2) in die Datei umgelenkt, die mit dem Datei-Bezeichner 1 verbunden ist, also nach `output`. Vertauscht man die Reihenfolge der beiden Umlenkungen, so wird zunächst `stderr` nach `stdout` (Bildschirm) umgelenkt (was wenig Sinn macht, weil `stderr` ohnehin der Bildschirm ist) und anschließend `stdout` in die Datei `output`. In der Datei `output` findet sich nur die eigentliche Ausgabe. Sind Quelle und Ziel einer Umlenkung identisch:

```
joe@debian:~$ command >filename <filename
```

so hat das unabhängig von der Reihenfolge in der Kommandozeile ohne Warnung die meist unerwünschte Wirkung, dass die Datei geleert wird (nicht gelöscht). Eine Datei leert man üblicherweise anders.

2.1.5 Umgebung und Shellvariable

Eine Shell stellt für jede Sitzung eine Umgebung (E: environment, F: environnement) bereit, die aus Zeichenketten der Form:

```
TERM=vt100
PATH=/usr/local/bin:/usr/bin:/bin:/usr/X11R6/bin:/usr/games
```

besteht, die von Programmen abgefragt werden können³. Eine Reihe von Umgebungsvariablen erzeugt die Sitzungshell, weitere werden von Programmen angelegt, und schließlich kann der Benutzer ausdrücklich solche Variable erzeugen oder verändern. Unter der Bash listet die Eingabe:

```
joe@debian:~$ set
```

alle aktuellen Umgebungsvariablen auf. In der Sitzung, in der diese Zeilen geschrieben wurden, waren es 47 an der Zahl. Die Eingabe:

```
joe@debian:~$ TERM=vt100
```

weist der bereits vorhandenen und wichtigen Umgebungsvariablen TERM den Wert vt100 zu. Falls TERM noch nicht existierte, würde sie angelegt. Die Variable nennt den Terminaltyp. Eine Eingabe wie:

```
joe@debian:~$ PATH=" "
```

weist der ebenfalls wichtigen Umgebungsvariablen PATH die leere Zeichenkette als Wert zu, selten ein sinnvoller Wert. Die Variable gibt den Suchpfad für Kommandos an. Bleibt die rechte Seite der Zuweisung frei:

```
joe@debian:~$ PS4=
```

wird die Variable gelöscht, sie ist nicht mehr definiert. Sehen wir uns eine aus dem Leben gegriffene Liste von Umgebungsvariablen an:

- BASH=/bin/bash absoluter Pfad der Bash,
- BASH_VERSINFO=([0]="2" [1]="05b") (gekürzt) Version der Bash, aufgelöst in einzelne Zeichenketten,
- BASH_VERSION='2.05b.0(1)-release' Version der Bash, am Stück,
- COLORTERM=
- COLUMNS=82 Spalten des aktuellen Textfensters (xterm),
- DESKTOP_SESSION=kde
- DIRSTACK=()
- DISPLAY=:0 von X11 gesetzte Variable, siehe Abschnitt 1.3.2 *X Window System (X11)* auf Seite 11,
- EDITOR=/usr/bin/vi Lieblingseditor,
- EUID=1000 Effektive Benutzer-ID,

³Programmiertechnisch ist die Umgebung ein Array of Strings (Array of Array of Character) mit dem Pointer `envp`.

- `EXINIT='set exrc'` Anweisung an den Editor `vi`, etwaige `.exrc`-Dateien in den Verzeichnissen zu lesen,
- `GDMSESSION=kde` Variable des GNOME Display Managers, siehe Abschnitt 1.3.2 *Start von X11* auf Seite 13,
- `GDM_XSERVER_LOCATION=local` dito,
- `GROUPS=()`
- `GS_LIB=/home/wulf/.fonts`
- `GTK2_RC_FILES=/etc/gtk-2.0/gtkrc:/home/wulf/.gtkrc-2.0` (gekürzt) Variable des GIMP Tool Kit, einer Grafikbibliothek,
- `HISTFILE=/home/wulf/.bash_history` Kommandoverlauf,
- `HISTFILESIZE=500` Kommandoverlauf,
- `HISTSIZE=500` Kommandoverlauf,
- `HOME=/home/wulf` Home-Verzeichnis,
- `HOSTNAME=pcwulf` Rechnername ohne Domäne,
- `HOSTTYPE=i386` Rechnertyp (Prozessortyp),
- `IFS=$' \t\n'` Inter Field Separator, Trennzeichen in der Kommandozeile: Zwischenraum (space), Tabulator, Zeilenwechsel (newline),
- `KDE_FULL_SESSION=true` Variable des K Desktop Environments,
- `KDE_MULTIHEAD=false` dito (mehrere reale Bildschirme),
- `KONSOLE_DCOP='DCOPref(konsole-1404,konsole)'` dito,
- `KONSOLE_DCOP_SESSION='DCOPref(konsole-1404,session-2)'` dito,
- `LANG=de_DE@euro` Variable aus dem Locale, siehe Abschnitt 1.3.6 *Lokalisierung* auf Seite 30,
- `LANGUAGE=de_DE:de:en_GB:en:fr_FR:fr`
- `LC_ALL=de_DE.iso885915@euro` Variable aus dem Locale,
- `LINES=26` Zeilen des aktuellen Textfensters (xterm),
- `LOGNAME=wulf` Benutzername beim Einloggen, ändert sich nicht während einer Sitzung,
- `LS_COLORS='no=00:fi=00:di=01;34:ln=01;36'` (gekürzt) Farben für das Kommando `ls` zur Unterscheidung von Dateitypen,
- `MACHTYPE=i386-pc-linux-gnu` Maschinentyp,
- `MAILCHECK=600` Periode in Sekunden zur Abfrage auf neue Email,
- `OLDPWD=/home/wulf/TeX/d2/text` vorheriges Arbeitsverzeichnis,
- `OPTERR=1`
- `OPTIND=1`
- `OSTYPE=linux-gnu` Typ des Betriebssystems,
- `PATH=/usr/local/bin:/usr/bin:/bin:/usr/bin/X11` Suchpfad für Kommandos,
- `PIPESTATUS=([0]="0 ")`
- `PPID=1404` Prozess-ID des Elternprozesses der Shell (`kdeinit`),
- `PS1='\u@\h:\w\$ '` erster Prompt,
- `PS2='> '` zweiter Prompt, für Mehrteilige Kommandos,
- `PS4='+ '` vierter Prompt, der dritte scheint abhanden gekommen zu sein,
- `PWD=/home/wulf/TeX/d2/bash` Arbeitsverzeichnis,

- `REPLYTO=alex-weingarten@t-online.de` Kopfzeile für ausgehende Email, die die Anschrift für etwaige Antworten nennt,
- `SESSION_MANAGER=local/pcwulf:/tmp/.ICE-unix/1391` Variable des K Desktop Environments,
- `SHELL=/bin/bash` aktuelle Shell,
- `SHELLOPTS=braceexpand:emacs:hashall:histexpand` (gekürzt) Optionen der Shell,
- `SHLVL=2` Level (Version) der Secure Shell,
- `SSH_AGENT_PID=1334` Prozess-ID des SSH-Agenten, der die privaten Schlüssel verwaltet,
- `SSH_AUTH_SOCK=/tmp/ssh-BrEBuq1283/agent.1283` Variable der Secure Shell,
- `TERM=xterm` Terminaltyp,
- `UID=1000` augenblickliche User-ID des Benutzers, kann in einer Sitzung wechseln,
- `USER=wulf` zugehöriger Benutzername,
- `USERNAME=wulf` wie `LOGNAME`,
- `WINDOWID=27262983` von X11 gesetzte Variable.
- `XAUTHORITY=/home/wulf/.Xauthority` von X11 gesetzte Variable,
- `XCURSOR_SIZE=`
- `XCURSOR_THEME=default`
- `XPSEVERLIST=`
- `_`=set jüngstes ausgeführtes Kommando, hier das Kommando zur Anzeige dieser Liste.

Eine Umgebungsvariable gilt zunächst nur für die Shell, in der sie definiert wurde. Sie wird nicht an Kindprozesse vererbt und niemals rückwärts von einem Kindprozess auf den Elternprozess. Soll eine Umgebungsvariable von der Shell auf ihre Kindprozesse vererbt werden – was oft gewünscht ist – muss sie exportiert werden. Die Bash exportiert die Variablen der Sitzungs-Shell automatisch, für die übrigen muss der Benutzer sorgen, zweckmäßig gleich bei ihrer Definition:

```
joe@debian:~$ export JETZT="Sonntag abend"
```

Das `export`-Kommando ohne Argument listet die aktuellen exportierten Umgebungsvariablen auf. Greift man auf die Umgebungsvariablen zu, ist deren Namen ein Dollarzeichen voranzustellen:

```
joe@debian:~$ echo $TERM
```

Die Prompts (Eingabeaufforderung, Systemanfrage, Bereitzeichen) mit den Namen `PS1` ff. werden als Umgebungsvariable definiert. Klassisch sind das Dollarzeichen für Normalbenutzer der Bourne-Shell, das Prozentzeichen für Normalbenutzer der C-Shell und das Doppelkreuz für den Verwalter. In modernen Shells verfügt der Prompt über eine Spur von Intelligenz. Setzt man (nach dem Dollarzeichen ein Leerzeichen):

```
joe@debian:~$ PS1="\u@\h:\w\$ "
```

so werden im ersten Prompt vor dem Dollarzeichen der Benutzername (user), der Rechnername (host) und das Arbeitsverzeichnis (working directory) angezeigt, was in einer vernetzten Welt die Orientierung erleichtert. Die Zeile steht in der Datei `$HOME/.bashrc`. Kennt man die Steuersequenzen (Escape-Sequenzen) seines Terminals, lässt sich ein farbiger Prompt erzeugen:

```
joe@debian:~$ PS1="\033[31m\u@\h:\w$\033[37m "
```

`\033` ist das unsichtbare Escape-Zeichen, vertreten durch seine oktale ASCII-Nummer, `[31m` stellt die Farbe Rot ein, `[37m` die Farbe Weiß, nachzulesen mittels `man terminfo` im Abschnitt *Color Handling*.

2.1.6 Positionsparameter

Neben den beliebig benannten Umgebungsvariablen, denen Werte zugewiesen werden können, kennt die Shell Positions- oder Aufrufparameter, auch Parametervariable genannt, denen ohne unser Zutun die Bestandteile der Kommandozeile zugewiesen werden. Auf diese Weise lässt sich die Zeile analysieren und manipulieren. Die Namen der Parameter beginnen mit einem Dollarzeichen. Es bedeuten:

- `$0` das erste Glied der Kommandozeile, das heißt das Kommando selbst,
- `$1` das zweite Glied der Kommandozeile usw. bis `$9`,
- `$10` das elfte Glied der Kommandozeile usw.
- `$*` (Dollar Stern) die ganze Kommandozeile ohne `$0`, als eine einzige Zeichenkette,
- `$@` (Dollar Klammeraffe) die ganze Kommandozeile ohne `$0`, jedes Glied in einer einzelnen Zeichenkette,
- `$#` (Dollar Doppelkreuz) die Anzahl der Glieder der Kommandozeile ohne `$0`,
- `$?` (Dollar Fragezeichen) der Rückgabewert (Returnwert, Exit Code) des zuletzt ausgeführten Kommandos,
- `$$` (Dollar Dollar) die Prozess-ID der Shell.

Will man eine temporäre Datei mit einem eindeutigen Namen erzeugen, kann man dazu die Prozess-ID der jeweiligen Shell heranziehen:

```
joe@debian:~$ cp datei datei.$$
```

Die Hauptanwendung der Positionsparameter liegt in Shellskripten, siehe Abschnitt 2.1.10 *Shellskripte* auf Seite 67.

2.1.7 Gruppierung von Kommandos

Der Zeilenwechsel (`<cr>`) beendet eine Kommando-Eingabe und schickt sie auf die Reise zum Prozessor. Will man mehrere Kommandos in einer Zeile eingeben, sind sie durch Semikolons zu trennen:

```
joe@debian:~$ date; hostname; pwd; id
```

Die Kommandofolge stellt mit einem Schlag die verlorene Orientierung in Zeit und Raum wieder her. Soll die gemeinsame Ausgabe in eine Datei umgelenkt werden, sind die Kommandos zu klammern (Leerzeichen zwischen Klammer und Kommando beachten):

```
joe@debian:~$ { date; hostname; pwd; id; } > datei
```

2.1.8 Restricted Shell

Manchen Benutzern möchte man nicht die volle Freizügigkeit auf dem System gewähren, die ein gewöhnlicher Benutzer genießt. Dazu trägt man in der Datei `/etc/passwd` für diese Benutzer eine restricted Shell `rbash` ein. Das ist eine um folgende Möglichkeiten reduzierte Bash:

- Wechsel von Verzeichnissen mittels `cd`,
- Verändern der Variablen `SHELL`, `PATH`, `ENV` oder `BASH_ENV`,
- Aufrufen von Kommandos mit absolutem Pfad.
- Übergeben von Dateinamen mit absolutem Pfad an das eingebaute `source`-Kommando (einzelner Punkt),
- Umlenkung der Ausgabe,
- Verwendung des `exec`-Kommandos,
- Ausbrechen aus der restricted Shell

und noch ein paar Kleinigkeiten, siehe Abschnitt *RESTRICTED SHELL* auf der Manualseite zur Bash.

Ein anderer Weg, die Bewegungsfreiheit eines Benutzers einzuschränken, ist das Einsperren in einen Zweig der Datei-Hierarchie, indem man für ihn mittels `chroot` sein Wurzelverzeichnis auf das obere Ende des Zweiges legt. Beispielsweise wird so der Verzeichnisbaum für anonyme Benutzer (FTP) auf die öffentlich zugänglichen Verzeichnisse und Dateien begrenzt. Man muss dann einige Werkzeuge, die dem anonymen Benutzer zugänglich sein sollen (`ls`), in dieses Gefängnis kopieren (nicht linken).

2.1.9 Wechsel der Shell

Wie wir im nächsten Abschnitt sehen werden, gibt es Alternativen zur Bash. Eine andere Shell als Login-Shell ist in `/etc/passwd` einzutragen. Als Login-Shells kommen alle in `/etc/shells` aufgeführten Shells in Betracht. Mit dem Kommando `chsh` ändert ein Benutzer dauerhaft seine Login-Shell. Da unter Umständen weitere Dateien (NIS, LDAP) von dem Wechsel erfahren müssen, sollte ein Benutzer solche Änderungen nur in Abstimmung mit seinem Verwalter vornehmen.

Im laufenden Betrieb ruft man die gewünschte Shell als Kommando auf:

```
joe@debian:~$ tcsh
```

Falls sie verfügbar ist, arbeitet man dann mit der neuen Shell als einem Kindprozess der alten. Der Aufruf:


```
joe@debian:~$ exec tcsh
```

ersetzt die alte Shell durch die neue. In beiden Fällen ändern sich einige Shellvariable. Diese Änderungen sind vorübergehend. Es kann nur passieren, dass sich ein Login-Manager die letzte Shell merkt und beim Start einer neuen Sitzung wieder aufruft.

2.1.10 Shellskripte

Eine weitere Aufgabe der Shell, die jedoch in diesem Buch nur gestreift wird, ist das Übersetzen (Interpretieren) von Programmen, die als Shellskripte (Stapeldatei, Batchdatei, Kommandodatei) bezeichnet werden. Der *Advanced Bash Scripting Guide* ist im Netz von <http://www.tldp.org/guides.html> in mehreren Formaten zu beziehen und eine gute Einführung mit zahlreichen Beispielen, zuletzt aktualisiert im Februar 2005.

Im einfachsten Fall handelt es sich bei Shellskripten (E: shell script, F: script de commandes) um längere Eingaben in die Kommandozeile, die man in eine Datei schreibt und dann gemeinsam unter dem Namen der Datei aufruft. Wenn der Verwalter jeden Morgen nacheinander die Kommandos `apt-get update`, `apt-get dist-upgrade` und `apt-get autoclean` eingibt, liegt es nahe, diese mit einem Texteditor in eine Datei `apt` (der Name ist noch unbenutzt) zu schreiben, pro Zeile ein Kommando, ihr Ausführungsrechte (750) zu verleihen und fürderhin morgens nur noch `apt` zu tippen, Ersparnis 50 Anschläge. Das Shellskript `frequenz` auf Seite 101 ist ein anderes Beispiel.

```
#!/bin/sh
# Shellskript mkindex zum Erzeugen von index.html
# W. Alex, Karlsruhe, 2005-02-20

# Functions

function header()          # creates index.html
{
    echo "<HTML><HEAD><TITLE>$1</TITLE></HEAD>" > \
    index.html
    echo "<BODY><H3>Index von $1</H3>" > index.html
}

function hyperlink()       # appends hyperlinks
{
    if [ "$1" != "index.html" ]
    then
        echo "<A href=\"$1\">$1</A><BR>" > index.html
    fi
}

function footer()          # finishes index.html
{
    echo "<HR>'hostname' `pwd` `date`" > index.html
}
```

```

echo "</BODY></HTML>" > index.html

# test and beautify index.html

/usr/bin/tidy -iumq index.html &> /dev/null
}

function selection()      # selects among files etc.
{
  for name in `ls -l -p`
  do
    if [ -d $name ]      # directory
    then
      hyperlink $name
      cd $name
      header `pwd`
      selection          # recursive call
      footer
      cd ..
    elif [ -f $name ]    # regular file
    then
      hyperlink $name
    else
      :                  # do nothing
    fi
  done
}

# Main part

cd /usr/share/doc        # may be replaced

header `pwd`
selection
footer

```

Quelle 2.1 . Shellskript `mkindex` zum Erzeugen eines Inhaltsverzeichnisses mit Einträgen in Form von Hyperlinks

Die Möglichkeiten der Shell als Programmiersprache reichen viel weiter. Sie kennt Bedingungen, Schleifen, Arithmetik, Funktionen, alles, was man von einer imperativen Programmiersprache (Fortran, Pascal, C ...) verlangt außer Gleitkomma-Arithmetik. Selbst diese letzte Einschränkung stimmt heute nur noch eingeschränkt. Da Shellskripte interpretiert (bei jedem Aufruf erneut übersetzt) werden, hält sich ihre Geschwindigkeit in Grenzen. Ihre Stärke sind kleine, systemnahe Aufgaben wie folgende: Im Verzeichnis `/usr/share/doc/` und seinen Unterverzeichnissen soll jeweils ein Inhaltsverzeichnis in einer Datei `index.html` angelegt werden, um den Verzeichniszweig für den Zugriff über ein lokales Web aufzubereiten. Die Quelle 2.1 löst die Aufgabe und dürfte mit Hilfe des eingestreuten Kommentars verständlich

sein. Die erste Zeile enthält den Shebang, den Hinweis, welcher Interpreter zu verwenden ist. Hier wird die Bourne-Shell `/bin/sh` verlangt, die unter Linux ein Link auf die Bourne-again-Shell `/bin/bash` ist. In allen weiteren Zeilen leitet das Doppelkreuz einen Zeilenkommentar ein. Bemerkenswert ist der rekursive Aufruf der Funktion `selection`, der dazu führt, dass das Skript in alle untergeordneten Verzeichnisse hinabsteigt. Mit der alten Bourne-Shell war das nicht möglich, alle neueren Shells kennen rekursive Funktionen. Ein zweites Beispiel für ein ausgewachsenes Shellskript findet sich in Abschnitt 13.10 *Türme von Hanoi* auf Seite 424.

Wenn Sie ein Shellskript unter MS Windows oder Mac OS schreiben, müssen die Zeilenwechsel an Linux/UNIX angepasst werden, siehe Abschnitt 5.1.3 *Textdateien aus anderen Welten* auf Seite 164.

Das waren nur die wichtigsten Eigenschaften der Bash. Eine vollständige Beschreibung würde ein dickes Buch füllen. Viele Einzelheiten braucht man aber im Alltag nicht zu wissen, sondern schlägt sie erst bei Bedarf nach. Falls das Debian-Paket *bash-doc* eingerichtet ist, findet man das 140 Seiten umfassende *Bash Reference Manual* von CHET RAMEY und BRIAN FOX unter `/usr/share/doc/bash/bashref.pdf`. Im Netz liegt es unter anderem auf <http://www.network-theory.co.uk/docs/bashref/> (Ausgabejahr beachten).

Eine verbreitete Alternative zur Shell als Programmiersprache (nicht als Kommando-Interpreter im Dialog) ist die Sprache *Perl* (Practical Extraction and Report Language), entwickelt 1987 von LARRY WALL und nicht zu verwechseln mit der Programmiersprache *Pearl* (Process and Experiment Automation Real-time Language). Perl vereinigt Elemente der Shell, des `awk`, des `sed` und der universalen Programmiersprache C. Ihre Stärke ist die Verarbeitung von Zeichenketten. Näheres in der Wikipedia und bei <http://www.perl.org/>, <http://www.cpan.org/>, <http://www.perl.de/> sowie <http://perl-seiten.privat.t-online.de/>.

Weitere in der Linux/UNIX-Welt gebräuchliche Skriptsprachen sind Tcl (Tool command language, gesprochen tickl), meist in Verbindung mit einem Toolkit als Tcl/Tk anzutreffen (<http://www.tcl.tk/>), PHP (<http://www.php.net/>, <http://www.rrz.uni-hamburg.de/RRZ?Software/PHP/> und das PHP-HOWTO) und Python, eine objektorientierte Sprache, die relativ einfach zu lernen ist (<http://www.python.org/>). Näheres am schnellsten in der Wikipedia.

2.2 Weitere Shells

Die Bourne-Shell war anfangs als Kommando-Interpreter nicht ganz befriedigend. Deshalb wurde bald in Berkeley von BILL JOY die C-Shell `csh` geschrieben, die als Kommando-Interpreter mehr leistete, als Skriptsprache aber wegen ihrer Anlehnung an die Programmiersprache C ein Umgewöhnen erforderte. Sie enthielt auch in den ersten Jahren mehr Fehler als erträglich. Aus diesen beiden Shells haben sich im Lauf der Jahre eine Reihe weiterer Shells entwickelt, die zum Teil die beiden Lini-

en wieder zusammenführen. Hier eine Übersicht über andere Shells, soweit sie als Debian-Paket verfügbar sind (genannt sind die Paketnamen):

- *bash-minimal* eine minimale Bash,
- *csch* aktuelle C-Shell aus OpenBSD-Quellen,
- *dash* Debian-Almquist-Shell, eine beliebte POSIX-konforme Shell, die schlanker ist als die Bash, von KENNETH ALMQUIST,
- *dbishell* Spezialshell, interaktive SQL-Shell zum Gedankenaustausch mit Datenbanken,
- *dsh* Distributed Shell oder Dancers Shell, nützlich in PC-Clustern, geht in Richtung Load Balancer, siehe Abschnitt 3.2.2 *Priorität* auf Seite 97, und http://tentakel.biskalar.de/similar/dsh_home/,
- *es* eine erweiterbare (extensible) Shell mit einer Syntax ähnlich wie *rc* (siehe unten),
- *esh* eine einfache Shell mit einer LISP-ähnlichen Syntax,
- *eshell* eine Spezialshell für den Emacs-Editor,
- *kiss* KARELS interaktive einfache Shell,
- *lsh* Baby-Shell mit DOS-kompatiblen Befehlen,
- *noshell* eine Shell ohne Funktionalität, die man in der Datei */etc/passwd* Dämonen zuweisen kann,
- *osh* Operator's Shell mit einem definierbaren Zugriffsprofil für Benutzer, die administrative Teilaufgaben erfüllen sollen,
- *pdsh* Public-Domain-Version der Kornshell; da seit einigen Jahren der Quellcode der Original-Kornshell frei ist (<http://www.kornshell.com/>), überflüssig geworden,
- *posh* mit der Debian-Policy übereinstimmende Shell (was das auch immer bedeuten mag),
- *psh* Perl-Shell, interaktiv, mit den Fähigkeiten von Perl,
- *rc* Shell aus dem Plan-9-Projekt von AT&T mit Anklängen an die C-Shell (<http://plan9.bell-labs.com/plan9dist/>),
- *sash* eine statisch gelinkte Shell mit vielen eingebauten Kommandos, die in Krisenfällen helfen kann, wenn die normale Shell nicht mehr erreichbar ist,
- *tcsh* TENEX C Shell, fälschlicherweise auch als Total- oder Turbo-C-Shell gedeutet, eine Weiterentwicklung der C-Shell (<http://www.tcsh.org/Home>, TENEX war um 1972 ein Betriebssystem für die DEC PDP 10 mit einigen wegweisenden Eigenschaften),
- *zsh* Z-Shell, Alleskönner (noch nie gebraucht, aber es gibt Liebhaber der Z-Shell, siehe http://www.michael-prokop.at/computer/tools_zsh_liebhaber.html).

Die Secure Shell (SSH) wird in Abschnitt 3.6 *Secure Shell* auf Seite 110 behandelt. Sie ist keine Alternative zur Bash, sondern dient anderen Zwecken.

Dienstprogramme

Hier geht es um einfache Hilfsmittel für verschiedene Zwecke wie die Manipulation von Dateien oder Verzeichnissen oder die Anzeige von Informationen aus dem System.

3.1 Daten in Ruhe (Verzeichnisse und Dateien)

3.1.1 Anzeigen und Navigieren (`pwd`, `cd`, `ls`, `file`)

Daten werden in Dateien (E: file, F: fichier) gespeichert, die wegen ihrer Vielzahl in Verzeichnissen (E: directory, F: répertoire) zusammengefasst werden. Dateien und Verzeichnisse werden unter Linux/UNIX in einer baumartigen Hierarchie angeordnet, Wurzel oben. Das Kommando `pwd` (print working directory, shellintern und -extern) zeigt an, in welchem Verzeichnis der Benutzer gerade arbeitet. Das Verzeichnis wird Arbeitsverzeichnis oder aktuelles Verzeichnis genannt. Das Kommando ist harmlos, es kennt weder Argumente noch braucht es Optionen und kann keinen Schaden anrichten.

Mit dem shellinternen Kommando `cd` (change directory) wechselt man das Arbeitsverzeichnis. Ohne Argument aufgerufen, wechselt es immer in das Home-Verzeichnis des Benutzers. Man kann aber auch einen absoluten oder relativen Pfad als Argument benennen. Ein absoluter Pfad beginnt mit einem Schrägstrich, der das Wurzelverzeichnis an der Spitze der Dateihierarchie bezeichnet:

```
joe@debian:~$ cd /usr/local/bin
```

Ein relativer Pfad beginnt ohne Schrägstrich und geht vom Arbeitsverzeichnis aus:

```
joe@debian:~$ cd TeX/debian2
```

Hier kann `TeX` nur ein Unterverzeichnis des Arbeitsverzeichnisses sein, wenn das Kommando Erfolg haben soll. Zwei Punkte als Argument bedeuten das dem Arbeitsverzeichnis übergeordnete Verzeichnis:

```
joe@debian:~$ cd ../TeX/debian2
```

wäre in einem Parallelverzeichnis zum Verzeichnis `TeX` sinnvoll. Ein Punkt ist das Arbeitsverzeichnis selbst, was als Argument für `cd` witzlos ist, aber gelegentlich gebraucht wird um zu erzwingen, dass ein Kommando aus dem Arbeitsverzeichnis und nicht aus `/usr/bin` oder dergleichen ausgeführt wird:

```
joe@debian:~$ ./ssh restart
```

Der Punkt ist auch ein Ausweg, wenn man ein Kommando aufrufen möchte, dessen Name ungeschickterweise mit einem Bindestrich beginnt:

```
joe@debian:~$ ./-dummer_name
```

Dieser Punkt hat nichts mit dem Punkt zu tun, mit dem die Namen verborgener Dateien oder Verzeichnisse beginnen, und auch nichts mit dem Punkt als Shellkommando (`source`).

Das Kommando `ls` (`list`) entspricht dem DOS-Kommando `dir` und listet in seiner einfachsten Form das Arbeitsverzeichnis auf. Mit einem Verzeichnis- oder Dateinamen als Argument listet es das angegebene Verzeichnis bzw. die Datei auf, was bei einer Datei wenig sinnvoll ist. Mit Optionen – deren es eine Fülle gibt – wird das Kommando reizvoller. Einige Beispiele:

- `ls -l` (`long`) zeigt ausführliche Informationen,
- `ls -a` (`all`) zeigt auch verborgene (`hidden`) Dateien und Unterverzeichnisse,
- `ls -la` (`long all`) zeigt ausführlich auch verborgene Dateien und Unterverzeichnisse an,
- `ls -li` zeigt ausführliche Informationen einschließlich der Inode-Nummern,
- `ls -b` ersetzt nicht druckbare Zeichen im Dateinamen durch deren oktale Nummer,
- `ls /usr/bin` listet das Verzeichnis `/usr/bin` auf,
- `ls -lia /usr/bin` listet das Verzeichnis `/usr/bin` einschließlich verborgener Dateien und Verzeichnisse auf und teilt auch die Inode-Nummern mit.

Die restlichen neunundneunzig Optionen entnimmt man der Manualseite. Eine Ausgabe von `ls -l` sieht folgendermaßen aus:

```
total 8692
-rw-r--r-- 1 root root 408718 Apr 14 2002 System.map-2.4.18-686
-rw-r--r-- 1 root root 512 Jul 1 10:44 boot.0200
lrwxrwxrwx 1 root root 11 Jun 9 14:58 boot.b -> boot-menu.b
drwx----- 2 root root 12288 Jun 9 14:53 lost+found
-rw----- 1 root root 7168 Jun 12 17:04 map
-rw-r--r-- 1 root root 632393 Apr 14 2002 vmlinuz-2.4.18-686
```

Darin bedeuten:

- das Zeichen links außen den Typ (reguläre Datei, weicher Link, Directory) des Eintrags,
- die nächsten neun Zeichen die Zugriffsrechte (`read`, `write`, `execute`) von Besitzer, Gruppe und Rest der Welt,
- die Zahlenspalte den Linkzähler, der die Anzahl der harten Links auf die Datei bzw. das Verzeichnis angibt.

- die dritte und vierte Spalte Besitzer (root) und Gruppe (ebenfalls root),
- die nächste Zahl die Größe der Datei in Bytes,
- die Zeitangabe den Zeitpunkt des jüngsten schreibenden Zugriffs (mtime),
- die letzte Spalte den Namen der Datei oder des Verzeichnisses.

Die Namen verborgener (hidden) Dateien oder Verzeichnisse beginnen mit einem Punkt¹. Sie werden von `ls` ohne entsprechende Option nicht angezeigt, sind aber nicht geheim oder verschlüsselt. Es handelt sich meist um vom System oder von Anwendungen angelegte Hilfsdateien mit Konfigurationsdaten, um die sich der Benutzer nur selten zu kümmern braucht.

Das Kommando `file` versucht, den Typ einer Datei zu ermitteln. Bei der Vielzahl von Dateitypen gelingt das nicht immer. Erfolgreiche Beispiele:

- `file /bin` liefert `directory`,
- `file buch.pdf` liefert `PDF document, version 1.3`,
- `file buch.tex` liefert `ASCII English text`,
- `file tallinn.jpg` liefert `JPEG image data ...`,
- `file fuer_elise.mp3` liefert `mp3 file with ID3 2.0 tag`,
- `file /usr/bin/vi` liefert `symlink to /etc/alternatives/vi`,
- `file /bin/cp` liefert `ELF 32-bit LSB executable ...`,
- `file abc` liefert `gzip compressed data`

Das letzte Beispiel ist insofern interessant, als hier beim Dateinamen die übliche Kennung `gz` fehlt. Die Antwort ist richtig; `file` schaut also in die Datei hinein und verlässt sich nicht auf den Dateinamen. Ähnliches leistet das shellinterne Kommando `type`.

3.1.2 Ändern von Besitz und Zugriffsrechten (`chown`, `chgrp`, `chmod`, `umask`)

Mit den Kommandos `chown` (change owner) und `chgrp` (change group) werden Besitzer und Gruppe einer Datei oder eines Verzeichnisses geändert:

```
joe@debian:~$ chown bjoern manuskript
```

```
joe@debian:~$ chgrp www index.html
```

```
joe@debian:~$ chown wwwadm:www webdir
```

Statt der Namen können die Nummern (User-ID, UID und Group-ID, GID) verwendet werden. Das Recht zum Ändern steht nur dem (alten) Besitzer und wie immer dem Verwalter zu. Im dritten Beispiel werden Besitzer und Gruppe gemeinsam geändert. Die wichtigste Option ist `-R` (rekursiv), die oft in Verbindung mit einem Jokerzeichen im Datei- oder Verzeichnisnamen auftritt:

```
joe@debian:~$ chown -R wwwadm:www webdir
```

¹Diese Punktdateien oder -verzeichnisse können sich zu einer Plage auswachsen. Kaum hat man einmal eine neue Software ausprobiert, macht sich schon wieder ein solcher Eintrag im Home-Verzeichnis breit. Von Zeit zu Zeit sollte man vorsichtig ausmisten.

Der Aufruf ändert Besitzer und Gruppe des Verzeichnisses `webdir` samt allem, was darin oder darunter liegt, entsprechende Zugriffsrechte vorausgesetzt.

Die Zugriffsrechte an einer Datei oder einem Verzeichnis sind in dreimal drei Rechte unterteilt, die sich als Matrix darstellen lassen. Wir haben einerseits eine Einteilung nach Benutzern:

- den Besitzer (owner), wie er in der Datei `/etc/passwd` aufgeführt ist, also nicht immer eine reale Person,
- die Gruppe (group), wie sie in der Datei `/etc/group` steht; zunächst immer die aktuelle Gruppe des erzeugenden Benutzers,
- den Rest der Welt (others), nicht *jeder* oder *alle*.

und andererseits eine Einteilung nach Art der Zugriffe:

- Lesen (read), was bei Verzeichnissen einen Zugriff mittels `ls` bedeutet, Wert `r` oder `4`,
- Schreiben (write), was bei Verzeichnissen das Anlegen, Umbenennen oder Löschen von darin enthaltenen Objekten bedeutet, Wert `w` oder `2`,
- Ausführen (execute), was zunächst nur für ausführbare Programme sinnvoll ist, bei Textdateien aber nicht schadet, Wert `x` oder `1`.

Für ein Verzeichnis ist mit *Ausführen* das Hinein- oder Hindurchgehen (Durchsuchen) gemeint. Es kommt vor, dass man seinen Gruppenkollegen oder dem Rest der Welt das Hineingehen in ein Verzeichnis gestattet, aber nicht das Auflisten. Das ist zwar keine effektive Sicherheitsmaßnahme, aber doch ein kleiner Schutz vor neugierigen Blicken. Linux/UNIX-Verzeichnisse sind auch nur Dateien. Sie enthalten zweispaltige Tabellen, in denen einer eindeutigen Dateinummer (Inode-Nummer) ein Name zugeordnet wird.

Die Zugriffsrechte lassen sich auf mehrere Weisen darstellen. Wir halten die Darstellung in Form einer dreiziffrigen Zahl für praktisch. Man addiert die vorstehend genannten Zahlenwerte jeweils für den Besitzer, die Gruppe und den Rest der Welt und schreibt sie in der genannten Reihenfolge hin:

- 700 bedeutet alle Rechte (Lesen, Schreiben, Ausführen) für den Besitzer, für alle anderen Benutzer nichts,
- 750 bedeutet alles für den Besitzer, Lesen und Ausführen für die Gruppe, für den Rest der Welt nichts,
- 644 bedeutet Lesen und Schreiben für den Besitzer, für alle übrigen Benutzer nur Lesen,
- 007 wäre eine hirnrissige Kombination von Rechten, warum?

Bei der Kombination 000 verbleibt dem Besitzer nur noch ein Recht. Er darf die Zugriffsrechte ändern.

Damit ein Benutzer auf eine Datei zugreifen kann, müssen zwei Voraussetzungen erfüllt sein:

- er muss an der Datei ausreichende Rechte haben,
- der Pfad vom Wurzelverzeichnis bis zu der Datei muss ihm lückenlos das Hindurchgehen (x) gestatten.

Wenn Sie also einige Dateien in Ihrem Home-Verzeichnis für andere Benutzer lesbar machen wollen, müssen alle im Pfad darüberliegenden Verzeichnisse für die anderen das x-Recht haben. Das bringt mit sich, dass Sie die übrigen Dateien und Verzeichnisse in Ihrem Home-Verzeichnis gut abdichten sollten.

Die Kombination aus dreimal drei Rechten reicht für viele Zwecke aus. Mehr könnte unübersichtlich werden. In Datenbanken hingegen sind mehrere hundert Rechte zu verteilen, wofür der Datenbank-Verwalter ausgebildet und zuständig ist. In anderen Zusammenhängen finden sich Access Control Lists (ACL), in denen ins Einzelne gehende Rechte definiert und vergeben werden.

Wenn eine Datei oder ein Verzeichnis angelegt wird, werden die Zugriffsrechte per Default auf einen Wert gesetzt, der durch das shellinterne Kommando `umask` in der Datei `/etc/profile` systemweit oder in `$HOME/.bash_profile` benutzerspezifisch festgelegt wird. Dem Kommando wird als Argument die Zahlenfolge mitgegeben, die die Ergänzung des gewünschten Wertes zu 777 bildet. Das Argument 022 resultiert folglich in den Zugriffsrechten 755. Kommandos, die Dateien oder Verzeichnisse anlegen (`touch`), können jedoch eigene Vorstellungen haben. Es empfiehlt sich, die Zugriffsrechte neuer Dateien oder Verzeichnisse mittels `ls -l` zu überprüfen.

3.1.3 Kopieren, Umbenennen und Löschen von Dateien (`cp`, `head`, `tail`, `dd`, `mv`, `rm`, `split`)

Die Entwickler des Ur-UNIX waren schreibfaul. Deshalb lautet das Kopierkommando `cp`, nicht `copy`. Das Kommando ist einigermaßen intelligent. Wenn die Zugriffsrechte ausreichen, tut es das, was sich der Benutzer vorstellt. Man muss nur aufpassen, dass man nicht eine Datei auf sich selbst kopiert. Das kann zum Verlust der Datei führen, aber die Debian-Version von `cp` erkennt den Schwachsinn und protestiert. Sollen nur die ersten oder letzten Zeilen einer Textdatei betrachtet oder kopiert werden, helfen die Kommandos `head` oder `tail`. Der Aufruf:

```
joe@debian:~$ head -24 datei1 > datei2
```

kopiert die ersten 24 Zeilen der Datei `datei1` nach `datei2`. Ohne Option werden als Default 10 Zeilen genommen. Ohne die Umlenkung (siehe Abschnitt 2.1.4 *Umlenkung* auf Seite 60) geht die Ausgabe nach `stdout`, das heißt auf den Bildschirm.

Obige Kopierbefehle kopieren eine Datei unabhängig davon, wie die Daten auf dem Speichermedium organisiert sind. In seltenen Fällen will man jedoch eine Datei, Image genannt, unverändert Bit für Bit kopieren. Das wird als *dumpen* bezeichnet. Eine Anwendung ist das Dumpen eines Disketten- oder CD-Images von der Platte auf einen entsprechenden Datenträger. Der Aufruf (`dd` = disc dump, `if` = input file, `of` = output file):

```
joe@debian:~$ dd if=boot.img of=/dev/fd0 bs=1024
conv=sync; sync
```

kopiert die Imagedatei `boot.img` aus dem Arbeitsverzeichnis auf das Medium, das im Gerät `/dev/fd0` (Floppy-Laufwerk) liegt. Der Rest sind im Manual nachzulesende Parameter, da das Kommando beim Dumpen die Daten umorganisieren kann.

Das abschließende `sync` schreibt die Puffer auf die Platte. Die Image-Dateien zum Einrichten von Debian GNU/Linux von Floppy, CD oder DVD gibt es bei Debian.

Zum Umbenennen oder Verschieben in ein anderes Verzeichnis nimmt man das Kommando `mv` (move). Im Grunde besteht es aus einem Kopier- und einem Löschvorgang. Etwas Vorsicht ist beim Verschieben ganzer Verzeichnisse angebracht. Im Dateisystem (Verzeichnisbaum) dürfen keine geschlossenen Wege (Schleifen) entstehen. Zumindest ältere Dateisysteme waren dagegen nicht immun, wenn der Verwalter am Werk war.

Linux/UNIX-Dateinamen dürfen bis zu 255 Zeichen² lang sein und alle Zeichen eines beliebigen Zeichensatzes enthalten außer dem Schrägstrich (ASCII Nr. 47) und dem Nullbyte (ASCII Nr. 0), weil diese beiden Zeichen in Linux/UNIX-Dateinamen eine besondere Bedeutung tragen. Der Schrägstrich trennt die Teile eines Pfades voneinander, das unsichtbare Nullbyte beendet den Namen bzw. Pfad. Man handelt sich jedoch mit Dateinamen unter Umständen Probleme ein, wenn man über die druckbaren Zeichen des US-ASCII-Zeichensatzes hinausgeht. Schon die Metazeichen der Shell sind in Dateinamen kritisch, können aber durch Quoten unschädlich gemacht werden. Enthält ein Dateiname Leerzeichen (space, ASCII Nr. 32), würde er für die Shell in mehrere Namen zerfallen. Quoten verhindert dies:

```
joe@debian:~$ mv "datei 123" datei_123
```

Dateikennungen (Anhängsel, Erweiterung, E: extension) wie `txt` sind in der Linux/UNIX-Welt erlaubt – auch mehrfach und von beliebiger Länge – aber nicht so verbreitet wie in anderen Welten. Sie sind normaler Bestandteil des Namens und zählen daher zur Länge des Namens. Insbesondere haben die Namen ausführbarer Dateien nicht die Kennung `exe`, sondern gar keine. Einige Anwendungen erwarten bestimmte Kennungen.

Zum Löschen von Dateien greift man zu dem Kommando `rm` (remove). Im Original-Linux/UNIX ist beim Gebrauch Vorsicht geboten, das Kommando gehorcht gnadenlos. So etwas wie `undelete` gibt es nicht, was weg ist, ist weg. Berühmtest ist das Kommando in Verbindung mit dem Jokerzeichen `*`, das eine beliebige Anzahl beliebiger Zeichen bedeutet und auf alle Dateien im Arbeitsverzeichnis zutrifft. Am wirkungsvollsten ist das Kommando, wenn es der Verwalter im Wurzelverzeichnis mit einer Option eingibt:

```
debian:/# rm -rf *
```

Er hat dann hinterher viel Platz im Dateisystem. Heute sind auch hier die Sitten verweichlicht. Löscht man eine Datei, indem man sie in den Abfalleimer (E: trash) verschiebt, kann man sie dort wieder herausholen, solange die Müllabfuhr den Eimer noch nicht geleert hat. Will man eine Datei nur leeren, nicht löschen, kopiert man den Bit Bucket `/dev/null` auf die Datei:

```
joe@debian:~$ cp /dev/null datei
```

²Früher 14 Zeichen, heute teilweise bis 1024 Zeichen.

`/dev/null` ist in diesem Zusammenhang eine leere Datei, die beim Lesen nur das unsichtbare End-of-File-Zeichen (EOF) zurück liefert. Welches Zeichen das tatsächlich ist, wird in der jedem C-Programmierer bekannten Datei `/usr/include/stdio.h` festgelegt.

Ist man der einzige Benutzer auf dem System, bemerkt ein versehentliches Löschen sofort und verfügt über die Rechte und Kenntnisse eines Verwalters, besteht noch eine Chance, gelöschte Daten zu retten. Man muss dann in den Single-User-Modus (`init 1`) gehen und kann mit einem Dateisystem-Editor oder -Debugger (`lde, recover`) versuchen, die Daten zu finden und in eine neue Datei zu verpacken. Andernfalls markiert das System die von den Daten belegten Sektoren auf dem Massenspeicher als frei und belegt sie zu einem unvorhersehbaren Zeitpunkt neu.

Aus letzterer Bemerkung erhellt, dass Daten zunächst nur logisch gelöscht werden, das heißt, nicht mehr über einen Dateinamen ansprechbar sind. Physikalisch können sie – eventuell in Bruchstücken – weiterhin auf der Platte vorhanden sein. Will man Daten sicher löschen, überschreibt man sie mit irgendeinem Zeichenmuster (Zufallszahlen) und löscht sie anschließend. Das Kommando `wipe` erledigt das zuverlässig. Für den Normalgebrauch ist `wipe` ein zu dicker Hammer; vor der Veräußerung einer Platte sollte man persönliche Daten damit beseitigen.

Zu Zeiten, als die Dateigrößen schneller wuchsen als das Fassungsvermögen der Datenträger (Disketten), teilte man unhandliche Dateien mittels `split` in Häppchen geeigneter Größe auf, die sich mittels `cat` wieder zusammenfügen ließen. Das Werkzeug wird heute selten gebraucht.

3.1.4 Anlegen, Umbenennen und Löschen von Verzeichnissen (`mkdir`, `mv`, `rmdir`)

Verzeichnisse werden mit dem Kommando `mkdir` (make directory) angelegt. Der Benutzer braucht dazu Schreibrecht in dem übergeordneten Verzeichnis. Ein leeres Verzeichnis wird mit dem Kommando `rmdir` (remove directory) gelöscht. Enthält das Verzeichnis noch Dateien oder Unterverzeichnisse, kann man `rm` rekursiv aufrufen:

```
joe@debian:~$ rm -rf verzeichnis
```

Die Option `f` erzwingt (force) das Löschen ohne lästige Rückfragen. Zum Umbenennen oder Verschieben von Verzeichnissen dient das bereits erwähnte Kommando `mv`. Das ältere, dem Verwalter vorbehaltene Kommando `mvdir` gibt es in Debian GNU/Linux nicht.

3.1.5 Verlinken (`ln`)

Alle Informationen über eine Datei außer ihrem Namen stehen in einer so genannten Inode (information node), die durch eine im Dateisystem eindeutige Nummer (inode number) gekennzeichnet ist. Jede Datei hat genau eine Inode und damit eine Nummer. Die Nummern erfährt man, indem man das Kommando `ls` mit der Option

-i aufruft. Alle Inodes eines Dateisystems liegen in einer Inode-Liste an einem bestimmten Platz im Dateisystem. Der gewöhnliche Benutzer kann nicht direkt auf eine Inode zugreifen, mittels `cat` etwa. Er ist auf das Kommando `ls` mit verschiedenen Optionen angewiesen. Für Programmierer gibt es einen Systemaufruf `stat()`, mit dem eine Inode gelesen werden kann, siehe man `2 stat`.

Verzeichnis 1

Name	Inode
<i>abc</i>	4711

Inode-Liste

4711

Datei

Verzeichnis 2

Name	Inode
<i>xyz</i>	4711

Abbildung 3.1. Harter Link

Die Verzeichnisse enthalten nur die Zuordnung Inode-Nummer - Dateiname. Daraus folgt, dass einer Inode-Nummer ohne weiteres mehrere Namen – unter Umständen in verschiedenen Verzeichnissen, aber im selben Dateisystem – zugewiesen werden können. Eine solche Zuordnung wird als harter Link bezeichnet. Das Kommando `ls -l` nennt im Anschluss an die Zugriffsrechte die Anzahl der harten Links einer Datei. Wo die Namen verbuddelt sind, ist eine andere Frage, `find` hilft. Wenn der letzte Name gelöscht wird, geht der Linkzähler auf Null, und die Datei ist logisch aus dem Diesseits verschwunden. Harte Links können die Grenze eines Dateisystems (Partition) nicht überschreiten. Gewöhnliche Benutzer können auch keine harten Links auf Verzeichnisse anlegen, und Verwalter sollten äußerst zurückhaltend damit umgehen. Ein Benutzer könnte in seinem Home-Verzeichnis folgendermaßen einen harten Link anlegen:

```
joe@debian:~$ ln furchtbar_langer_Dateiname f1D
```

Ein Verwalter darf das natürlich auch in System-Verzeichnissen, um einigen Kommandos schönere Zweitnamen zu geben. Man mache sich den Unterschied zwischen einem harten Link und einer Kopie klar. Eine Kopie ist eine neue Datei mit neuer Inode-Nummer und eigenem Lebenslauf. Es ist unsinnig, bei harten Links von einem Original zu sprechen, alle Namen zu einer Inode-Nummer sind gleichberechtigt. Zwangsläufig gehören zu allen Namen dieselben Zugriffsrechte und sonstigen Attribute.

Wegen der Einschränkungen für harte Links hat man weiche oder symbolische Links erfunden. Sie bestehen aus einer kleinen Datei, die einen Verweis auf die verlinkte Datei (Zieldatei) enthält. Der Aufwand ist geringfügig höher als bei harten Links, dafür dürfen die weichen Links auf Ziele außerhalb ihres Dateisystems zeigen und auch von gewöhnlichen Benutzern für Verzeichnisse angelegt werden (s = symbolic):

```
joe@debian:~$ ln -s entsetzlicher_Verzeichnisname eV
```

eV bezeichnet eine eigene Datei, die 29 Bytes groß ist und in der Ausgabe von `ls -l` ganz links durch ein `l` als Link gekennzeichnet ist. Kommt einem weichen Link seine Zieldatei abhanden, führt sein Aufruf zu einer Fehlermeldung. Die Zugriffsrechte von weichen Links werden nicht ausgewertet, sondern nur die der Zieldatei. Konsequenterweise ändert das Kommando `chmod` die Zugriffsrechte der Zieldatei, nicht des Links, während `chown` und `chgrp` ohne Optionen die Besitzverhältnisse des Links beeinflussen, was jedoch selten vonnöten ist.

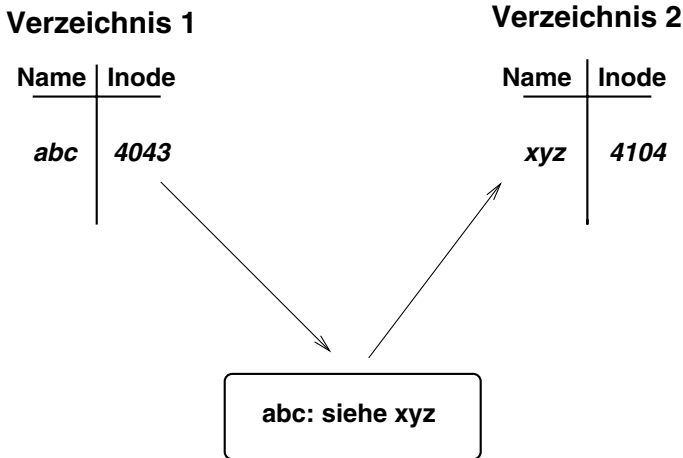


Abbildung 3.2. Weicher oder symbolischer Link (Symlink)

3.1.6 Zeitstempel (ls, touch)

Zu jeder Linux/UNIX-Datei gehören drei Zeitangaben, die Zeitstempel genannt und vom System verwaltet werden:

- die Zeit des jüngsten lesenden Zugriffs (access time, atime),
- die Zeit des jüngsten schreibenden Zugriffs (modification time, mtime),
- die Zeit der jüngsten Änderung des Dateistatus (status/inode change time, ctime).

Der Dateistatus umfasst den Datei-Inhalt, die Zugriffsrechte und den Linkzähler, also auch die in der Inode abgelegten Metadaten einer Datei. Ein Schreibzugriff ändert

zwei Zeitstempel, nämlich die `mtime` und die `stime`. Bei Verzeichnissen gilt Durchsuchen nicht als lesender Zugriff, Löschen oder Hinzufügen gilt als schreibender Zugriff.

Das Kommando `ls -l` zeigt das Datum des jüngsten schreibenden Zugriffs an, mit `ls -lu` erfährt man das Datum des jüngsten lesenden Zugriffs, mit `ls -lc` das Datum der jüngsten Statusänderung. Der Zeitpunkt der Erschaffung einer Datei wird nicht festgehalten und ist auch aus technischer Sicht uninteressant. Eine Datei hat keinen Geburtstag. Änderungen an den Daten hingegen sind für Werkzeuge wie `make` wichtig, um entscheiden zu können, ob eine Datei aktuell ist.

Mit dem Kommando `touch` aktualisiert man die `atime` und/oder die `mtime` einer Datei. Die `ctime` lässt sich nicht direkt beeinflussen. Das Kommando wird auch gern genommen, um eine leere Datei zu erzeugen:

```
joe@debian:~$ touch xyz
```

vorausgesetzt, der Name existiert noch nicht im Arbeitsverzeichnis. In diesem Fall würden die Zeitstempel der existierenden Datei aktualisiert.

3.1.7 Datei-Manager (lfm, mc, xfe, nautilus, konqueror)

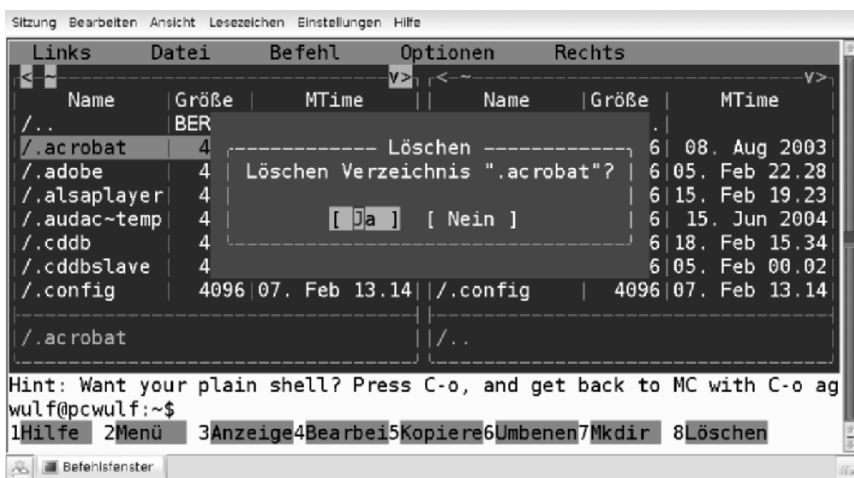


Abbildung 3.3. Screenshot des Datei-Managers Midnight Commander, auf dem Bildschirm bunt

Wer die vorstehend genannten Aufgaben nicht mit einer Kommandozeile sondern durch Klicken bewältigen will, greift zu einem Datei-Manager. Zwei von X11 unabhängige, curses-basierte Datei-Manager sind der Midnight Commander aus dem Debian-Paket `mc` und der Last File Manager aus dem Debian-Paket `lfm`, beide in der Abteilung *Utilities* bzw. *Hilfsprogramme*. DOS-Benutzer werden sich an den

Norton Commander erinnert fühlen. Beide Manager werden curses-üblich mit der Tastatur (Pfeiltasten) gesteuert. In Abbildung 3.3 ist der Midnight Commander dargestellt, der in Wirklichkeit farbig und dadurch angenehmer zu betrachten ist als das Schwarzweißbild.

Auf X11, aber unabhängig von einem Desktop Environment, setzen der X Northern Captain aus dem Debian-Paket `xnc`, der X File Explorer aus dem Debian-Paket `xfe`, der Desktop File Manager aus dem Debian-Paket `dfm` und der `gentoo` auf. Der `xnc` macht einen aufgeräumten Eindruck, der `xfe` kommt schlicht, aber mit allem Notwendigen daher, der `dfm` zeigt in einem Bilder-Verzeichnis kleine Bildchen (thumbnails) an, der `gentoo`³ (<http://www.obsession.se/gentoo/>) ist weitgehend konfigurierbar und ziemlich mächtig. Es gibt noch mehr von der Sorte.

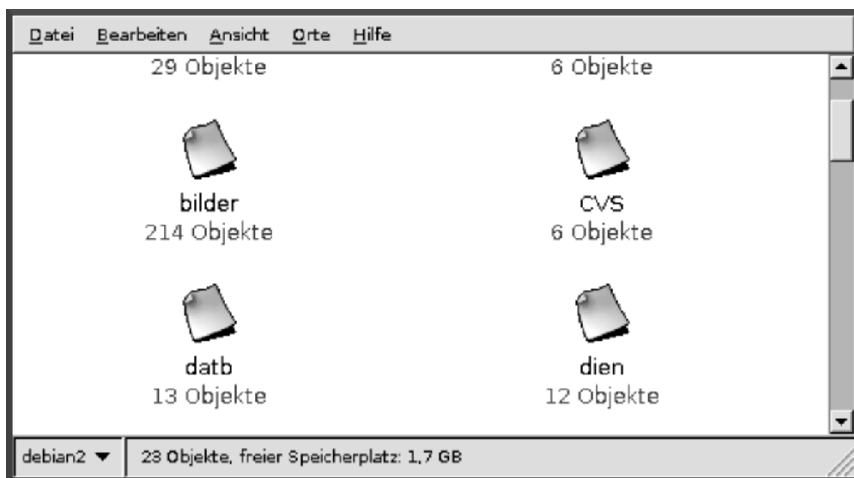


Abbildung 3.4. Screenshot des Datei-Managers Nautilus aus dem GNOME-Projekt

Aus dem GNOME-Projekt stammt der Datei-Manager Nautilus, der X11 voraussetzt. Er stellt sich schmucklos dar und beherrscht alle nötigen Funktionen. Abbildung 3.4 zeigt ihn bei der Wiedergabe eines benutzereigenen Verzeichnisses.

Der Datei-Manager Konqueror stammt aus dem KDE-Projekt und ist zugleich ein Web-Browser, nicht ganz so mächtig wie Mozilla. Die Ähnlichkeit der Darstellung legt die Verbindung der beiden Tätigkeitsfelder nahe, aber der Benutzer darf nicht vergessen, dass das lokale Dateisystem und das weltweite Web verschiedene Dinge sind. Ich kann zwar in beiden suchen und meine Funde auf dem Bildschirm darstellen, aber das eine ist ein hoffentlich gut abgesicherter privater Bereich, das andere öffentlich. Abbildung 3.5 zeigt den Konqueror bei der Darstellung von Systemverzeichnissen, auf dem Bildschirm etwas bunter als der Nautilus.

³Der Dateimanager hat nichts mit Gentoo Linux (<http://www.gentoo.org/>) zu tun.



Abbildung 3.5. Screenshot des Datei-Managers Konqueror aus dem KDE-Projekt

Obwohl manche Manager (F: gestionnaire) die Arbeit erleichtern, sollte ein Benutzer das Denken und die Befehlsgewalt nicht ganz an sie abtreten: Display-Manager, Window-Manager, Datei-Manager, Passwort-Manager, Adressen-Manager, Kalender-Manager, Download-Manager, Collection Manager, Netz-Manager, Listen-Manager, Font Manager, ToDo-Manager, Boot Manager, Site Manager, Database Manager, Archiv-Manager, Account Manager, Financial Manager, Package Manager, Drucker-Manager, Session Manager, Volume Manager, Contact Manager, GPS Manager, Device Manager, Schlüssel-Manager, Zertifikat-Manager, Wallet Manager, Wireless LAN Manager, Musik-Manager, Panel Manager, Preferences Manager, Patch Manager, Bookmark Manager, Human Resource Manager, Project Manager, Web Content Manager, Personal Information Manager, Power Manager, Report Manager, MP3 Manager, Proxy Manager, Knowledge Manager, Playlist Manager. Dazu noch Assistenten und Frontends. Und wer macht die Arbeit?⁴

3.1.8 Packen (compress, gzip, bzip2, zip)

Dateien, die nicht bereits auf Grund ihrer Entstehung gepackt (komprimiert, verdichtet) sind, lassen sich mit einigen Werkzeugen packen, das heißt, auf eine geringere Größe bringen, ohne dass wesentliche Inhalte verloren gehen. Man kann sich vorstellen, dass zehn Leerzeichen durch ein Leerzeichen und die Anzahl zehn ersetzt werden. Tatsächlich sind die Algorithmen zum Komprimieren raffinierter. Es werden zwei Arten von Algorithmen oder Packern unterschieden:

- verlustlose Packer,
- verlustbehaftete Packer.

⁴Real Programmers have no use for managers.

Nach einem verlustlosen Packen lässt sich die ursprüngliche Datei bis auf das letzte Bit genau wiederherstellen. Das Spielchen kann man beliebig oft wiederholen. Diese Art von Packen braucht man für Texte oder binäre Programme (kompilierte Programme). Beim verlustbehafteten Packen wird auf unwesentliche Einzelheiten des Inhalts verzichtet. Das Original lässt sich nicht wiederherstellen, da etwas fehlt. Diese Art von Packen wird für Grafiken oder Sound verwendet. Es bringt nichts, eine Datei mehrmals zu verdichten. Ebenso ist das Packen von sehr kleinen Dateien nutzlos bis schädlich. Das Packen oder Verdichten hat nichts mit Archivieren oder mit Verschlüsseln zu tun, das sind andere Baustellen.

Tabelle 3.1. Größe einer mittels `compress`, `gzip` oder `bzip2` gepackten PostScript-Datei, die auf einer `LaTeX`-Quelle `abklex.tex` beruht

Name	Bytes	%
<code>abklex.tex</code>	383831	56
<code>abklex.ps</code>	685895	100
<code>abklex.ps.Z</code>	271651	40
<code>abklex.ps.gz</code>	223052	33
<code>abklex.ps.bz2</code>	163375	24

Ein Versuch mit einer PostScript-Datei (`abklex.ps`), die auf einer `LaTeX`-Quelle (`abklex.tex`) beruhte, zeigte ein Ergebnis gemäß Tabelle 3.1 nach dem Packen. Das Packen mit dem ehrwürdigen UNIX-Werkzeug `compress` (Dateikennung `Z`) verkleinerte die ursprüngliche PostScript-Datei auf 40 %. Das weit verbreitete GNU-Werkzeug `gzip` (Dateikennung `gz`) schaffte 33 % und das neuere Werkzeug `bzip2` (Dateikennung `bz2`) sogar 24 %. Hierbei handelt es sich durchweg um verlustlose Packer. Wiederholt man den Versuch mit der bereits bei der Erzeugung gepackten Datei im pdf-Format, sieht das Ergebnis nicht so überzeugend aus, wie Tabelle 3.2 zeigt. Das lohnt sich kaum. Man sollte daher wissen, ob eine vorliegende Datei bereits gepackt ist oder nicht.

Tabelle 3.2. Größe einer mittels `gzip` oder `bzip2` gepackten pdf-Datei

Name	Bytes	%
<code>abklex.pdf</code>	392448	100
<code>abklex.pdf.gz</code>	322164	82
<code>abklex.pdf.bz2</code>	320372	82

Den Packer `compress` wird man heute nicht mehr verwenden. Man trifft nur noch gelegentlich auf damit gepackte Dateien (Kennung `Z`), die man auch mittels `gunzip` auf Trinkstärke verdünnen kann. Das Paar `gzip` und `gunzip` (Kennung `gz`) ist in der Linux/UNIX-Welt weit verbreitet und das Werkzeug der Wahl, wenn man mit anderen Benutzern Daten austauscht. Für den Eigengebrauch oder nach Rücksprache mit dem Empfänger ist das Paar `bzip2` und `bunzip2` zu empfehlen.

Die Zahl deutet auf die Version 2 hin, die mit der älteren Version 1 nicht kompatibel ist. Zu `bzip2` findet sich außer der Manualseite ein Mini-HOWTO, das die Zusammenarbeit mit anderen Werkzeugen (`tar`, `less`, `grep`, `emacs`...) erläutert.

Schließlich gibt es aus Rücksicht auf die MS-Windows-Welt noch ein Werkzeugpaar `zip` und `unzip`, das sich mit PKZIP (PHIL KATZ's ZIP) verträgt. Bei obiger PostScript-Datei lieferte es eine Kompression wie `gzip`. Es ist gleichzeitig ein Programm zum Archivieren wie `tar`. Unter Debian braucht man es hauptsächlich zum Aufdröseln von gepackten Archiven fremder Herkunft.

3.1.9 Archivieren (`ar`, `tar`), Backup

Ein Archiv ist eine Zusammenfassung mehrerer Dateien oder Verzeichnisse zu einer einzigen Datei, die einfacher handzuhaben ist als ihre Komponenten einzeln, beispielsweise bei einer Übertragung im Netz.

Der Archivierer `ar` fasst mehrere Dateien samt Verwaltungsinformationen zu einer Datei zusammen, die Archiv oder Bibliothek (E: library, F: bibliothèque) genannt wird. Die Hauptanwendung ist das Erzeugen von Funktions- oder Klassenbibliotheken. Auf die Teile kann zugegriffen werden, ohne das ganze Archiv aufzulösen. Auch die Debian-Pakete sind `ar`-Archive. Der nicht programmierende Benutzer hat mit dem Werkzeug wenig zu tun.

Dagegen ist das Werkzeug `tar` (tape archiver) immer dann interessant, wenn mehrere (viele) Dateien oder Verzeichnisse zu einer Datei zusammengeschnürt werden sollen, um sie zu verschicken oder unter einem einzigen Namen zu speichern (Backup). Meist werden `tar`-Archive auch noch gepackt, siehe oben. Ursprünglich schrieb man mit `tar` vor allem Verzeichnisse auf Bänder, daher der Name und einige historisch bedingte Eigenheiten. Der Aufruf (`c` = create):

```
joe@debian:~$ tar -cvf mein_archiv.tar datei1 datei2
*.txt verz1
```

stopft ausgehend vom Arbeitsverzeichnis die Dateien `datei1`, `datei2`, alle Dateien mit der Kennung `.txt` und das Unterverzeichnis `verz1` in ein Archiv namens `mein_archiv.tar` im Arbeitsverzeichnis. Man achte darauf, dass der Archivname nicht in der Liste der zu archivierenden Dateien auftaucht (Jokerzeichen!). Das Archiv kann man anschließend irgendwohin verfrachten und wieder auflösen (`x` = extract):

```
joe@debian:~$ tar xvf mein_archiv.tar
```

Die Dateien und Unterverzeichnisse stehen an der neuen Stelle so, wie sie oben als Argument genannt worden sind. Zur Verdeutlichung:

```
tar -cvf linux.tar *.tex          # in /buch/linux
tar -cvf linux.tar ./*.tex        # in /buch/linux
tar -cvf linux.tar linux/*.tex    # in /buch
tar -cvf linux.tar /buch/linux/*.tex # an beliebiger Stelle
```

archivieren zwar dieselben Dateien, aber unter verschiedenen Namen, was beim Auflösen zu verschiedenen Ergebnissen führt. Die ersten beiden Formen lassen sich in einem beliebigen Verzeichnis auflösen, ebenso die dritte Form, die aber vor Ort ein Unterverzeichnis namens `linux` anlegt, das dann alle Dateien enthält. Die vierte Form ist unflexibel und führt zu demselben absoluten Pfad wie beim Packen, was selten gewünscht ist und unter Umständen Superuser-Rechte erfordert. Will man sich nur das Inhaltsverzeichnis des Archivs ansehen, ruft man (`t` = table of contents):

```
joe@debian:~$ tar -tvf mein_archiv.tar
```

auf. Die von Debian verwendete GNU-Version von `tar` filtert mittels der Option `z` das Archiv gleich durch `gzip` bzw. `gunzip`, sodass man sich Tipparbeit spart:

```
joe@debian:~$ tar -tvzf mein_archiv.tar.gz
```

Die Option `j` bewirkt dasselbe für Kompression mittels `bzip2`.

Ein Backup einer Datei oder eines Verzeichnisses ist eine Kopie zur Sicherung vor Datenverlust bei Hardwaredefekten (Plattencrash) oder bei Beschädigung infolge Fehlern im Gebrauch von Software (ungewolltes Löschen oder Überschreiben). Während das Backup eines ganzen Systems Aufgabe des Verwalters ist, kann ein Benutzer durchaus daran interessiert sein, einzelne wertvolle Dateien – Manuskripte beispielsweise – zusätzlich in kurzen Zeitabständen zu sichern. Dazu erzeugt er von den Dateien oder Verzeichnissen ein `tar`-Archiv, zweckmäßig gleich komprimiert, und verschiebt oder kopiert dieses auf eine andere Festplatte. Ein Archiv auf derselben Festplatte hilft nichts bei einem Plattendefekt. Es muss wenigstens eine zweite Platte auf demselben Rechner sein, besser eine Platte auf einem anderen Rechner. Falls das zweite Dateisystem über NFS gemountet ist, reichen dazu die üblichen Befehle `cp` oder `mv`, andernfalls muss mittels FTP oder `scp` fernkopiert werden. Eine alte Weisheit besagt, dass eine Datei, die nur einmal existiert, gar nicht existiert. Legen Sie zusammen mit Ihrem Verwalter ein abgestuftes Vorgehen zum Sichern fest.

3.1.10 Mitschreiben (`tee`, `script`)

Manchmal möchte man die Bildschirmausgabe in einer Datei mitschreiben oder die Ausgabe eines Programms gleichzeitig in zwei Dateien lenken, um bei Problemen genau nachvollziehen zu können, was geschehen ist. Das Filter `tee` liest von `stdin` und schreibt nach `stdout` und gleichzeitig in eine als Argument mitgegebene Datei – auch in mehrere – so wie ein T-Stück in einem Schlauch einen Strom in zwei Teilströme trennt, nur dass hier Strom und Teilströme identisch sind. Der Aufruf:

```
joe@debian:~$ ls -lR debian | tee ls1R_debian
```

listet das Verzeichnis `debian` samt Unterverzeichnissen (rekursiv) in langer Form auf und schreibt das Ergebnis gleichzeitig auf den Bildschirm und in die Datei `ls1R_debian`, die man in aller Ruhe bearbeiten kann.

Das Werkzeug `script` startet eine neue Shell und schreibt jedes Zeichen, das auf dem Bildschirm erscheint, in einer Datei mit, die per Default den Namen

typescript trägt, aber auch anders benannt werden kann, Nach dem Aufruf arbeitet man normal wie mit jeder Shell weiter und beendet sie mittels `exit` oder `<ctrl>-<d>`, wie gewohnt. Das Ergebnis einer Mitschrift sieht so aus:

```
Script started on Thu Jan  6 22:55:21 2005
wulf@pcwulf:~/server/TeX/d2/dien$ id
uid=1000(wulf) gid=1900(www) groups=1000(wulf),6(disk),24(cdrom)
wulf@pcwulf:~/server/TeX/d2/dien$ pwd
/home/wulf/server/TeX/d2/dien
wulf@pcwulf:~/server/TeX/d2/dien$ date
Thu Jan  6 22:55:33 CET 2005
wulf@pcwulf:~/server/TeX/d2/dien$ who
wulf      :0                Jan  6 09:33
wulf@pcwulf:~/server/TeX/d2/dien$ ll
total 84
-rw-r--r--  1 wulf      wulf           325 Nov 28 13:56 dien.tex
-rw-r--r--  1 wulf      wulf       31923 Jan  6 22:48 dien1.tex
-rw-r--r--  1 wulf      wulf       3125 Nov 16 12:40 dien2.tex
-rw-r--r--  1 wulf      wulf       5061 Jan  4 21:28 dien3.tex
-rw-r--r--  1 wulf      www          0 Jan  6 22:55 typescript
wulf@pcwulf:~/server/TeX/d2/dien$ exit
Script done on Thu Jan  6 22:56:00 2005
```

Wie man sieht, schreibt `script` alles mit, auch den Prompt. Das Werkzeug kann nicht ein grafisches Fenster mitprotokollieren, in dem mit Mausklicks gearbeitet wird.

3.1.11 Dateisysteme im Netz (NFS, Samba, Netatalk)

Übersicht

Der Wunsch, in einem lokalen Netz von allen Arbeitsplätzen aus auf einen gemeinsamen Massenspeicher und eventuell weitere Ressourcen (Drucker, Internet-Anschluss) zugreifen zu können, liegt nahe. In Verbindung mit zusätzlicher Software kann man dahin kommen, dass in einem lokalen Netz jeder Benutzer auf jedem Arbeitsplatz arbeiten kann und überall seine gewohnte Umgebung vorfindet. Das ist der Schritt vom Personal zum Unpersonal Computer. Benutzerbezogen bleiben nur die Daten.

Für den gemeinsamen Zugriff über das Netz auf einen Massenspeicher stehen drei Protokolle zur Verfügung:

- das Network File System (NFS), entwickelt von der Firma Sun und in das Internet übergegangen, fast immer mit dem Network Information System (NIS) gekoppelt,
- das Server Message Block Protocol (SMB), auch mit Common Internet File System (CIFS), NetBIOS- oder LanManager-Protokoll bezeichnet, in der Linux-Welt verwirklicht durch das Software-Paket Samba,
- die AppleTalk-Protokolle in der Apple-Welt, die aber zunehmend von NFS oder SMB abgelöst werden.

Zwei weitere, bei Debian erhältliche verteilte Dateisysteme sind Coda und Open-AFS. Das Andrew File System (AFS) wurde von IBM für große Netze entwickelt und als Open Source freigegeben, Näheres bei <http://www.openafs.org/>. Coda wurde von der Carnegie Mellon University entwickelt und fußt auf dem AFS, Näheres bei <http://www.coda.cs.cmu.edu/>. Die Open Software Foundation hat das Konzept eines Distributed Computing Environments (DCE) entwickelt, das jedoch ähnlich wie das ISO-OSI-Schichtenmodell oder der X.500-Verzeichnisdienst praktisch keinen Erfolg hatte und für uns zwei Nummern zu mächtig ist. Wenn Sie bei den in lokalen Netzen etablierten Dateisystemen etwas vermissen, soll sich Ihr Verwalter schlaue machen.

Ein konzeptueller Unterschied besteht zwischen Peer-to-Peer-Netzen und Servernetzen. In einem Peer-to-Peer-Netz kann jeder Benutzer auf seinem Arbeitsplatz Dateien oder Verzeichnisse anderen Teilnehmern im Netz zum Lesen und/oder Schreiben zugänglich machen. Das funktioniert nur in kleinen Netzen und erfordert Disziplin. In einem Servernetz liegen die Daten im wesentlichen auf Servern, die unter der Fuchtel eines Verwalters stehen, der für Ordnung und Sicherheit sorgt. In Linux/UNIX-Netzen ist eher das Serverkonzept anzutreffen.

Network File System (NFS)

Will man von einem Arbeitsplatz aus ein Dateisystem nutzen, das auf einem anderen Rechner (Dateiserver) im Netz liegt, braucht man auf dem Arbeitsplatz ein Client-Programm und auf dem entfernten Rechner ein Server-Programm, die sich mittels eines Protokolles verständigen. Ein vernetzter Rechner kann Dateien oder häufiger Verzeichnisse über das Netz beziehen, sodass ein Benutzer fast keinen Unterschied zwischen lokalen und entfernten Daten bemerkt. Die Netzverbindung ist transparent. In einem homogenen Linux/UNIX-Netz ist hierfür das Network File System (NFS) ein gebräuchlicher Weg, in einem heterogenen Netz mit MS-Windows-Rechnern bietet Samba eine ähnliche Funktionalität.

Auf dem Server erlaubt der Verwalter mittels der Datei `/etc/exports` den Export bestimmter Verzeichnisse an bestimmte Clients. Clientseitig ist in der Datei `/etc/fstab` anzugeben, von welchen Servern welche Verzeichnisse in welche lokalen Verzeichnisse (mounting points) eingehängt (eingebunden, gemountet) werden sollen. Dabei lassen sich einige Einstellungen vornehmen. Die beim Booten sehr wichtige Datei `/etc/fstab` könnte folgende Zeile enthalten:

```
server1:/home /server1 nfs rw,soft,sync,rsize=8192,wsize=8192
```

Von `server1` soll das Verzeichnis `/home` auf das lokale Verzeichnis `/server1` gemountet werden. Dabei handelt es sich um ein Network File System (nfs). Das Verzeichnis soll lesbar (readable) und beschreibbar (writable) gemountet werden. Falls eine NFS-Dateioperation wegen eines Fehlers abgebrochen wird, soll sie nicht wiederholt werden, sondern eine Fehlermeldung loslassen (soft). Ein- und Ausgabe sollen synchron (ungepuffert) erfolgen (sync). Schließlich soll die Blockgröße beim Lesen und Schreiben auf 8192 Bytes gesetzt werden, was die Übertragung beschleunigt. Man kann sich aber auch mit der Angabe `defaults` begnügen. Einzelheiten

auf den Manualseiten zum Kommando `mount` und zu `nfs`. Mit obiger Zeile wird auf einem Arbeitsplatz beim Systemstart das Server-Verzeichnis sofort in die lokale Dateihierarchie eingebaut und steht wie ein lokales Verzeichnis zur Verfügung. Wegen der Besitzverhältnisse und Zugriffsrechte ist es praktisch, wenn Server und Arbeitsplätze eine gemeinsame Benutzerverwaltung haben, was sich mit dem Network Information Service (NIS) erreichen lässt. Auch eine einheitliche Zeit im lokalen Netz ist angenehm (Network Time Protocol, NTP). Darum soll sich der Verwalter kümmern.

Der Datenverkehr wird nicht verschlüsselt. Es findet auch keine nennenswerte Authentifizierung statt. NFS ist daher nur in überschaubaren lokalen Netzen zu gebrauchen, nicht im Internet. Superuser-Rechte werden per Default nicht über NFS durchgereicht. Ansonsten arbeitet es sich mit den fernen Daten wie mit lokalen.

Samba

Samba ist ein Software-Paket, das wie NFS den Zugriff auf im Netz verteilte Massenspeicher ermöglicht und darüber hinaus auch die gemeinsame Benutzung von Druckern. Ferner kann es zwei in MS-Windows-Netzen benötigte Dienste übernehmen, die Namensauflösung (WINS) und das Browsen. Es verwendet ein Protokoll (Server Message Block Protocol), das außerhalb der Linux/UNIX-Welt beheimatet ist, und richtet sich an SMB/CIFS-Clients unter MacOS, IBM OS/2 oder MS Windows. Natürlich kann ein Linux-Rechner auch als Samba-Client auftreten, beispielsweise um Ressourcen eines MS-Windows-Servers zu nutzen. Das Samba-Projekt ist unter <http://de.samba.org/> zu finden.

Falls Samba eingerichtet und ordnungsgemäß konfiguriert ist, kann ein Client mit Werkzeugen aus dem Debian-Paket *smbclient* auf die Ressourcen des Servers zugreifen: freigegebene Verzeichnisse mounten, Drucker ansprechen und sich Informationen über das SMB-Netz anzeigen lassen. Ein grafisches Frontend zu den Werkzeugen ist das Programm *xsmbrowser*, siehe Abbildung 3.6. Alternativen sind *smbc* (curses-basiert) oder *tk smb*.

Ihr Verwalter kann mit einem clientseitigen Eintrag wie:

```
//server/homes /samba smbfs rw,noauto,user=sam,password=geheim 0 0
```

in die Datei `/etc/fstab` das Mounten von per Samba bereit gestellten Verzeichnissen vorbereiten. Das erste Glied der Zeile ist nicht ein Verzeichnis wie bei NFS, sondern ein Share im Sprachgebrauch von MS Windows. Wenn alle übrigen Einstellungen stimmen, kann ein Linux-Benutzer auf die gemounteten Verzeichnisse in gewohnter Weise zugreifen.

Netatalk

Apple-Macintosh-Rechner verwendeten bis zur Version MacOS 9 einschließlich ein eigenes Netzprotokoll, vergleichbar SMB für MS Windows und IBM OS/2. Das Protokoll ermöglicht ebenfalls die Freigabe von Dateien oder Druckern. Mit dem

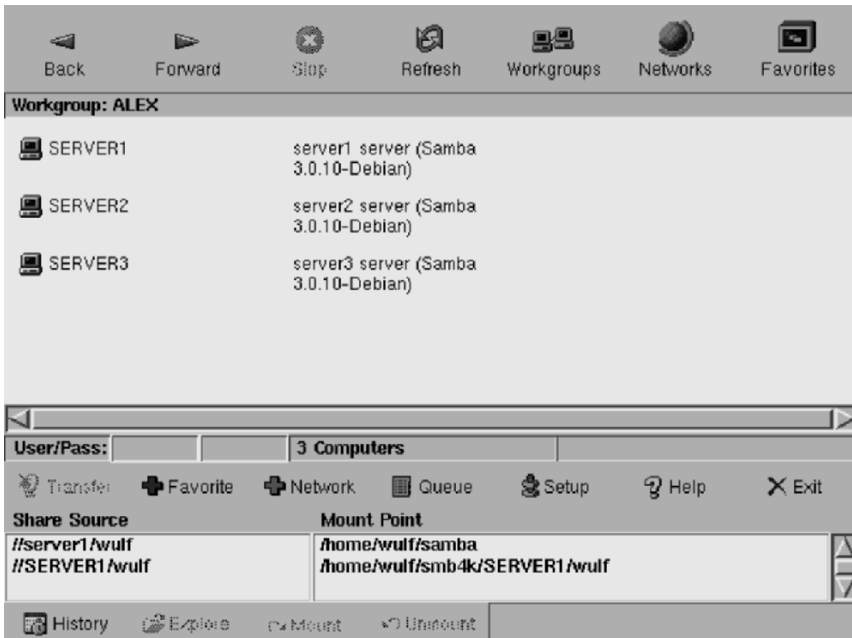


Abbildung 3.6. Screenshot von xsmbbrowser, einem grafischen Frontend zu den Werkzeugen aus dem Debian-Paket *smbclient*

Debian-Paket *netatalk* kann ein Rechner als Server für AppleTalk-Dienste eingerichtet werden, eine Aufgabe für den Verwalter. Näheres unter <http://netatalk.sourceforge.net/> und im Netatalk-HOWTO. Für den Benutzer an einem Macintosh ergeben sich keine Besonderheiten. Wenn keine älteren Macs im lokalen Netz mitreden, sollte man darauf achten, dass der Netatalk-Dienst nicht gestartet wird. Er verzögert den Bootvorgang. Außerdem ist jeder aktivierte und nicht benötigte Dienst ein unnötiges Sicherheitsrisiko.

3.1.12 Arbeiten mit externen Datenträgern (tar, mount)

Externe, entfernbare Datenträger (E: removable media, F:) sind Disketten verschiedener Durchmesser, Bänder, CD/DVDs, Memory Sticks, Speicherkarten usw. Mit Lochkarten und -streifen sowie mit Magnetbändern begann die Entwicklung, die immer noch voranschreitet. Eine Folge dieser raschen Entwicklung ist, dass man zehn Jahre alte Datenträger oft nicht mehr lesen kann, weil die Lesegeräte vom Markt verschwunden sind. Einen Brief meines Urgroßvaters dagegen – mit Eisengallus-Tinte und Feder auf Papier geschrieben – kann ich noch heute lesen. Ein zweites Problem ist die Haltbarkeit der Daten. Bei magnetischen Datenträgern muss man nach fünf Jahren mit Ausfällen rechnen.

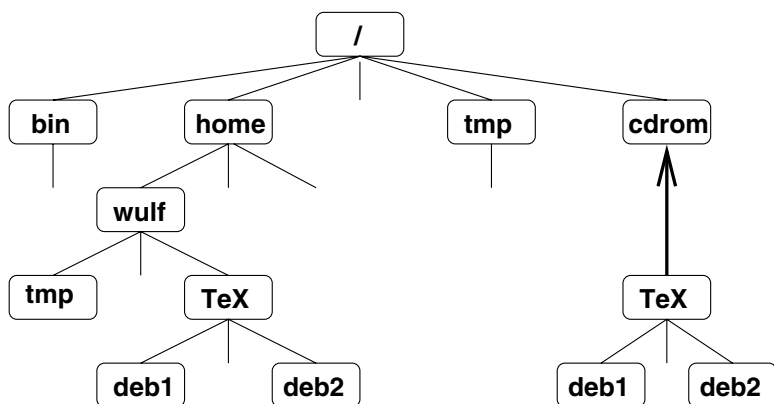


Abbildung 3.7. Mounten externer (entfernbarer) Datenträger. Erklärung im Text

Die Daten auf externen Datenträger werden auf zweierlei Weise angesprochen:

- als Dateisystem,
- als Archiv.

Archivierer wie `tar` können ein Archiv auf eine Gerätedatei (character device) wie `/dev/tape` schreiben. In älteren Ausgaben des Kommandos war das die Default-Einstellung; heute schreibt zumindest GNU-`tar` als Default nach `stdout`, wie viele Linux/UNIX-Werkzeuge. Der externe Datenträger muss formatiert sein – Floppies mittels `fdformat`, `gfloppy` oder `superformat`, heute meist schon vom Hersteller erledigt – braucht aber kein Dateisystem zu enthalten. Infolgedessen gibt es hier auch nichts zu mounten.

Ist dagegen auf dem externen Datenträger ein Dateisystem angelegt – mittels `mkfs` – dann muss es gemountet werden, bevor man auf den Datenträger zugreifen kann. Beim Lesen – nicht beim Brennen – von CD/DVDs ist das der Fall. Oft hat der Verwalter in der Datei `/etc/fstab` das Mounten durch einen Benutzer vorbereitet oder automatisiert. Das Verzeichnis, in das die Wurzel des externen Dateisystems eingehängt werden soll, wird als Mounting Point bezeichnet und soll leer sein. Ist es nicht leer, sind vorübergehend die darin befindlichen Unterverzeichnisse und Dateien nicht zugänglich. Sie sind aber nicht verloren und erscheinen wieder, sobald der Mounting Point frei gegeben ist. Ist das Mounten in `fstab` vorbereitet, ruft ein Benutzer das `mount`-Kommando mit dem Namen des als Mounting Point zu verwendenden Verzeichnisses oder mit dem Namen der Gerätedatei auf:

```
joe@debian:~$ mount /stick
```

Das Kommando entnimmt alle Einzelheiten der Datei `fstab`. Ohne Argument aufgerufen zeigt es die augenblicklich gemounteten Verzeichnisse an. Das externe, gemountete Dateisystem ist vorübergehend ein Zweig des lokalen Dateisystems und genauso zu benutzen wie dieses. Auf fertige CD/DVDs kann man unabhängig von den `mount`-Optionen natürlich nur lesend zugreifen.

Vor dem Entfernen des Datenträgers ist sein Dateisystem mittels `umount` wieder auszuhängen (zu `unmounten`, `unzumounten`). Ein Laufwerk rückt meist einen gemounteten Datenträger nicht heraus. Das gewaltsame Entfernen eines noch gemounteten Datenträgers kann Datenverluste verursachen. Erscheint auf das `umount`-Kommando hin die Fehlermeldung *device is busy*, wird das gemountete Dateisystem von irgendjemand benutzt. Eine Datei kann noch von einem Programm geöffnet sein, oder ein Benutzer verwendet es gerade als Arbeitsverzeichnis. Um den Benutzer eines auf `/stick` gemounteten Memory Sticks herauszufinden, gibt man:

```
joe@debian:~$ fuser -vm /stick
```

ein und bekommt die erforderlichen Auskünfte, um sich mit dem Benutzer zu verständigen.

Benutzern, die aus der DOS/Windows-Welt kommen, erscheint das Mounten zunächst umständlich. Als Gegenleistung brauche ich mich aber nicht um Laufwerke wie `D:` zu kümmern; alle Verzeichnisse und Dateien im Verzeichnisbaum stehen mir in gleicher Weise zur Verfügung, und wenn sie physikalisch in Down Under liegen, von wannen sie über das Netz gemountet sind.

3.1.13 DOS-Dateisysteme (mtools)

Die Mtools sind eine Sammlung von Werkzeugen zum Umgang mit DOS-Dateien auf Diskette oder Festplatte. Näheres bei <http://mtools.linux.lu/>. Als Trennzeichen im Pfadnamen akzeptieren sie beide Schrägstriche, Optionen werden nach Linux/UNIX-Brauch mit einem Minuszeichen eingeleitet. Sie verstehen die Jokerzeichen der Shell in Dateinamen (Globbing). Die wichtigsten der insgesamt 26 Werkzeuge sind:

- `mattrib` DOS-Datei-Attribute ändern,
- `mcd` DOS-Verzeichnis wechseln,
- `mcopy` DOS-Dateien von/nach Linux/UNIX kopieren,
- `mdel` DOS-Dateien löschen,
- `mdir` DOS-Verzeichnis auflisten,
- `mformat` DOS-Dateisystem auf Diskette schreiben,
- `mren` DOS-Datei umbenennen,
- `mtype` Inhalt einer DOS-Datei anzeigen.

Unter Linux lassen sich bestimmte DOS- oder Windows-Dateisysteme auch mounten und dann mit leichten Einschränkungen wie ein Linux-Dateisystem ansprechen. Näheres mittels `man fs`. Entsprechende Werkzeuge für CP/M-Dateien bietet das Debian-Paket *cpmtools*.

3.1.14 Brennen von CD/DVDs (`cdrecord`, `bashburn`, `gcombust`, `xcdroast`, `k3b`, `cdrdao`)

Das Brennen von CDs auf dem eigenen Rechner ist eine mittlerweile ausgereifte Technik, bei der es kaum zu Pannen kommt. Ältere Rohlinge fassen 650 MB an

Daten, neuere 700 MB. Die speicherbare Netto-Datenmenge ist etwas geringer, da für das Dateisystem einige Bytes verbraucht werden. Die Lebensdauer von einmal beschreibbaren CDs (CD-R) wird bei sorgfältiger Aufbewahrung auf zehn Jahre angesetzt und liegt somit höher als die von magnetischen Datenträgern (Floppies, Bänder). Das *Linux CDROM HOWTO* (2001), das *CD Writing HOWTO* (2000) und das *Linux DVD HOWTO* (2000) befassen sich mit Hard- und Software, wobei das letztere nicht auf dem aktuellen Stand und daher wenig hilfreich ist. Eine umfangreiche *CD-Recording FAQ* liegt auf <http://www.brennmeister.com/cdrfaq/>, aktuell vom Jahr 2000.

Das Herstellen (Schreiben, Brennen) einer Daten-CD ist ein zweistufiger Vorgang:

- zuerst werden die zu speichernden Verzeichnisse und Dateien in ein ISO-9660-Dateisystem gebracht, das als Image in einer vorübergehenden Datei abgelegt wird,
- dann wird die Image-Datei auf den CD-Rohling gebrannt.

Das Werkzeug für den ersten Schritt ist `mkisofs`, das ziemlich intelligent ist und zahlreiche Optionen kennt, siehe Manual. Das Brennen besorgt das vergleichsweise einfache Werkzeug `cdrecord`.

Es vereinfacht die Arbeit, wenn die zu brennenden Dateien in wenigen Verzeichnissen zusammengefasst sind. Der Aufruf zum Erzeugen der Image-Datei lautet beispielsweise:

```
joe@debian:~$ mkisofs -l -r -v -V VerzName_Datum -o
/tmp/cd.iso verz1 dat2
```

Die Optionen und Argumente bedeuten:

- `-l` (long) Dateinamen bis 31 Zeichen Länge zulassen, in der Linux/UNIX-Welt immer sinnvoll,
- `-r` Einträge nach dem Rock Ridge Protocol hinzufügen und Zugriffsrechte sowie Besitz auf vernünftige, allgemeine Werte setzen, sodass die CD auch auf anderen Rechnern uneingeschränkt lesbar ist,
- `-v` (verbose) während der Arbeit ein bisschen was erzählen,
- `-V` (Volume name) nachfolgend ein beliebiger Name für die CD, hier bestehend aus Verzeichnisname und Datum,
- `-o` (output) nachfolgend der Name der Image-Datei, hier `/tmp/cd.iso`; darauf achten, dass im Dateisystem `/tmp` ausreichend Platz vorhanden ist,
- abschließend die absoluten oder relativen Pfade der Dateien und Verzeichnisse, die in die Image-Datei aufgenommen werden sollen.

Obige Optionen sind eine gebräuchliche Auswahl, um eine CD als Backup eines Verzeichnisses mit wertvollen Dateien (Manuskripten) vorzubereiten.

Während des Brennens soll der Rechner möglichst wenig mit anderen Aufgaben beschäftigt werden, weil die Daten gleichmäßig mit hoher Geschwindigkeit zum Brenner (E: burner, F: graveur) fließen müssen. Die Eingaben für das Brennkommando erfahren Sie größtenteils von Ihrem Verwalter:

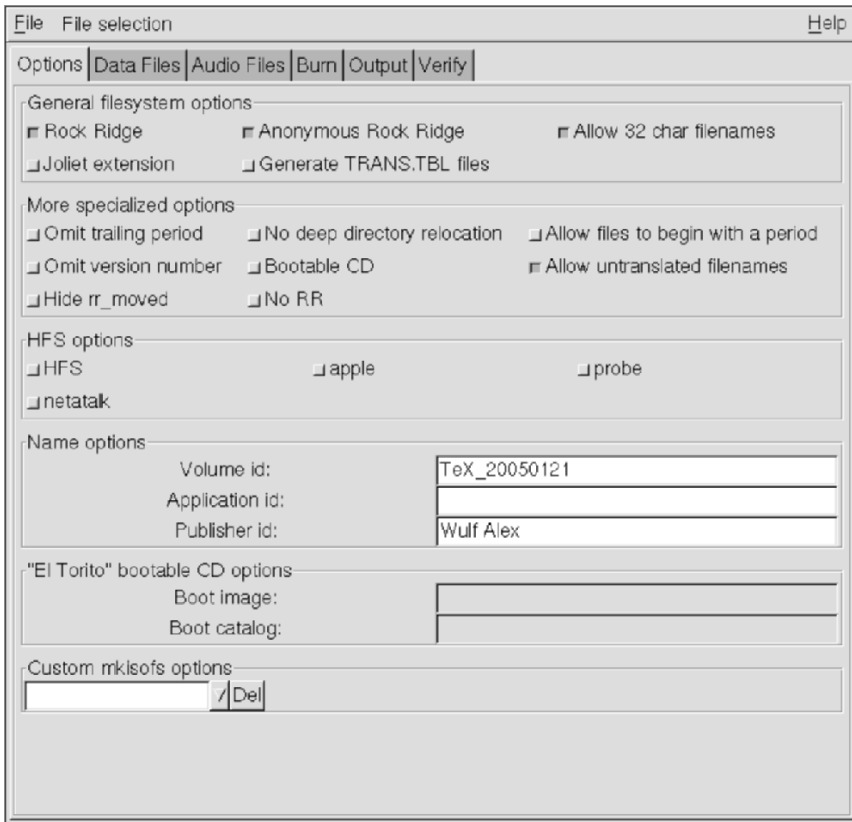


Abbildung 3.8. Screenshot des Frontends gcombust zum Brennen von CDs

```
joe@debian:~$ cdrecord -v speed=12 dev=0,3,0
/tmp/cd.iso
```

Die Optionen und Argumente bedeuten:

- `-v` (verbose) wie vorstehend,
- `speed` maximale Brenngeschwindigkeit, hängt vom Brenner ab,
- `dev` Adresse des Brenners, hängt von der Konfiguration des Rechners ab (hier SCSI-Bus Nr. 0, Target Nr. 3, Logical Unit Number 0),
- abschließend der Name der Image-Datei.

Stellt man mehrere Exemplare der CD her, braucht man die Image-Datei nur einmal zu erzeugen. Wenn alles fertig ist, die Image-Datei löschen:

```
joe@debian:~$ rm /tmp/cd.iso
```

Zufriedene Mienen danken es Ihnen.

```
#!/bin/bash
# Shellskript zum Brennen von CDs

if [ $# -lt 1 ]
then
echo -e "Zu brennendes Verzeichnis eingeben: \c"; read v
else
v=$1
fi

if [ -z $v ]
then
echo "Kein Verzeichnis angegeben, Ende."; exit
fi

if [ -d $v ]
then
echo "Vorbereitung zum Brennen von $v"
echo "Weitermachen? Enter/Return druecken."; read x
else
echo "Verzeichnis $v nicht gefunden, Ende."; exit
fi

# Optionen siehe man mkisofs, anpassen:
mkisofs -l -J -r -v -o /tmp/cd.iso $v
echo "Temporaeres ISO-9660-Dateisystem erzeugt"

while true
do
echo "Rohling einlegen, falls noch nicht geschehen."
echo "Klar zum Brennen? Enter/Return druecken."; read x
# Optionen siehe man cdrecord, anpassen:
cdrecord -v speed=8 dev=0,3,0 /tmp/cd.iso
echo "Fertig mit Brennen. CD entnehmen."
echo -e "Noch eine Kopie brennen? j/n: \a\c"; read x
case $x in
j|J|y|Y) ;;
*) echo "Ende."; rm -f /tmp/cd.iso; exit;;
esac
done
```

Quelle 3.1 . Minimales Shellskript brennen zum Brennen von CDs

Die obige Abfolge von Kommandos samt deren Optionen und Argumenten legt es nahe, daraus ein kleines Shellskript zu machen. In Quelle 3.1 finden Sie ein minimales Skript, das geschrieben wurde, um ein Verzeichnis mit einem Manuskript und Begleitmaterial für einen Kurs mehrmals auf CD zu brennen. Das Skript kann als Grundlage für eigene Anpassungen und Erweiterungen dienen. Unter <http://bashburn.sourceforge.net> liegt ein deutlich leistungsfähigeres Shellskript, das einen guten Eindruck hinterließ, aber schon Probleme mit den Farben der Schrift hatte. Weiße Schrift auf hellgelbem Grund ist schwierig zu erkennen.

Das Werkzeug `gcombust` setzt auf X11 und GTK+ auf; es ist ein grafisches Frontend für `mkisofs`, `cdrecord` und einige weitere Programme. Nachdem es konfiguriert ist, kann der Benutzer die Einzelheiten des Brennens vergessen, er muss natürlich immer noch wissen, was er will. Führt der Maus-Cursor über einen Button, öffnet sich ein kleines Hilfefenster, was das Nachschlagen im Manual erspart. Einige Fehlermöglichkeiten werden von `gcombust` abgefangen. Ähnliches leistet auch das Werkzeug `xcdroast` mit einem etwas anderen grafischen Stil, aber letzten Endes denselben Funktionen. Das KDE-Werkzeug `k3b` ist erst ab *sarge* verfügbar.

Das Werkzeug `cdrdao` zum Brennen von Daten- oder Audio-CDs ermöglicht dem Benutzer, den Brennvorgang weitgehend zu beeinflussen. Zu diesem Zweck werden Anweisungen in eine TOC-Datei (TOC = Table Of Contents, Inhaltsverzeichnis) geschrieben, an deren Stelle auch eine CUE-Datei aus der MS-Windows-Welt (`cue` = Stichwort) treten kann. Die Manualseite ist ausführlich; noch mehr erfährt man bei <http://computer.freepage.de/cdrdao/>.

Bei DVDs gibt es außer einer Vielfalt von Formaten auch noch Probleme mit der Qualität der Rohlinge, insbesondere bei höheren Kapazitäten. Bezüglich der Haltbarkeit liegen wenig Erfahrungen vor, teilweise schlechte. Die Werkzeuge und Frontends sind im wesentlichen dieselben wie für CDs, zusätzlich kommen die Debian-Pakete `dvdplusrw` aus *woody* bzw. `dvd+rw-tools` aus *sarge* in Betracht.

Falls der Brenner vorhanden, angeschlossen sowie eingerichtet ist und vom System beim Start erkannt wird, Sie aber trotzdem nicht brennen dürfen (`permission denied`), liegt der Verdacht nahe, dass Ihnen Zugriffsrechte fehlen. Wenn dann der Verwalter mit seiner Allmacht brennen kann, mangelt es so gut wie sicher an den Rechten. Entweder sind die Rechte bei dem Gerät (`/dev/...`) zu erweitern oder Sie in eine berechtigte Benutzergruppe (`audio`, `cdrom...`) aufzunehmen. Als letzter Ausweg bliebe, das Brennprogramm `setuid-root` zu setzen. Das sind Aufgaben des Verwalters,

3.2 Daten in Bewegung (Prozesse)

3.2.1 Anzeigen von Prozessen (`ps`, `top`, `gnome-system-monitor`)

Programme werden auf Linux/UNIX-Rechnern in Form von Prozessen ausgeführt. Programme ruhen als Dateien im Dateisystem, die aus ihnen hervorgegangenen Prozesse liegen im Arbeitsspeicher und verlangen Rechenzeit von der CPU. In der alltäglichen Sprechweise wird aber häufig nicht zwischen Programm und Prozess unterschieden. Jeder Prozess trägt eine eindeutige Nummer, die Prozess-ID, und gehört dem Benutzer, der ihn gestartet hat. Mit dem Kommando `ps` (Prozess-Status) lassen sich die Prozesse eines Systems anzeigen. Ohne Optionen werden nur die eigenen Prozesse aufgelistet, und zwar ihre Prozess-ID, ihr zugehöriges Terminal (Kontrollterminal), die von ihnen verbrauchte CPU-Zeit und das Kommando, das den Prozess gestartet hat. Oft wird das Kommando mit einigen Optionen aufgerufen und das Ergebnis durch `less` geschickt, um es in Bildschirmseiten zu unterteilen:

```
joe@debian:~$ ps -ef | less
```

Hiermit erhält man ausführlichere Informationen über alle Prozesse auf dem System. Weitere Optionen siehe auf der Manualseite zu `ps`.

Die Belastung eines Systems, sortiert nach der CPU-Beanspruchung durch die einzelnen Prozesse, erfährt man mittels des Kommandos `top`, das keiner Grafik bedarf. Es kann von jedem Benutzer aufgerufen werden, belastet das System minimal und zeigt auch die Speicherauslastung einschließlich Swap an. Per Default wird die Ausgabe alle 3 Sekunden aufgefrischt. Beendet wird es durch Eingabe von `<q>`. Das Manual gibt ausführliche Hinweise und Erklärungen.

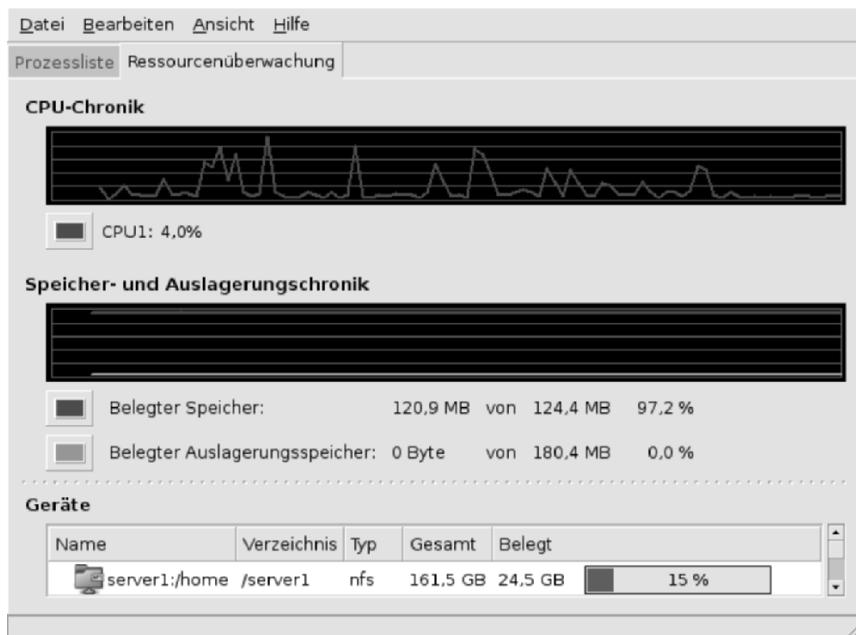


Abbildung 3.9. Screenshot des GNOME System Monitors. Auswahl Ressourcenüberwachung

Das Programm `gnome-system-monitor` zeigt die Prozesse und die Inanspruchnahme verschiedener Systemressourcen unter einer grafischen Oberfläche an (X11 erforderlich), siehe Abbildung 3.9. Es wird unter seinem Namen aus der Kommandozeile oder durch Anklicken des Menüpunktes *Systemüberwachung* gestartet und ist ansprechender als `ps` oder `top`, lässt sich aber nicht mit anderen Kommandos verbinden. Andererseits lassen sich aus dem Monitor heraus unmittelbar Signale an Prozesse senden, beispielsweise um sie zu beenden. Hat man den Eindruck, dass ein Rechner zäh läuft, kann man mit obigen Kommandos untersuchen, ob sich ein Prozess unangemessen breit macht. Manchmal braucht man auch die ID eines Prozesses, siehe Abschnitt 3.2.4 *Signale* auf Seite 100. Es gibt eine Vielzahl weiterer Werkzeuge zur Überwachung einzelner Funktionen des Systems und des Netzes, aber die gehören in die Werkstatt des Verwalters.

3.2.2 Priorität, Load Balancing (nice, batch, queue)

Alle aktiven Prozesse verlangen vom Zentralprozessor (CPU) Rechenzeit. Dazu reihen sie sich in eine Warteschlange ein. Ein Prozess hoher Priorität darf sich in der Warteschlange vordrängeln. Die Priorität setzt sich aus zwei Faktoren zusammen:

- einem vom System vergebenen Faktor, der den Dialogbetrieb mit vielen kurz laufenden Prozessen bevorteilt,
- einem vom Benutzer zu vergebenden nice-Faktor (nice = nett).

Der vom Benutzer zu vergebende nice-Faktor reicht von -20 (höchste Priorität) bis +19 (niedrigste Priorität), Default ist 0. Es gibt auch Systeme mit einem Bereich von 0 bis 39, was nichts an dem Mechanismus ändert. Der Name des Vorkommandos rührt daher, dass ein Benutzer sich nett gegenüber seinen Mitbenutzern erweist, wenn er seinen cpu-intensiven Prozessen eine schlechtere Priorität mitgibt.

Ohne eine Zahlenangabe verschlechtert (erhöht) das Vorkommando `nice` den nice-Faktor eines Programmes um den Wert 10, was in der Regel ausreicht:

```
joe@debian:~$ nice lange_rechnung
```

Der Verwalter kann in verzweifelten Fällen durch einen negativen nice-Faktor seinen Kommandos eine überdurchschnittliche Priorität verleihen. Das Vorkommando ist in Verbindung mit `nohup` gebräuchlich, wenn lange laufende Programme gestartet werden:

```
joe@debian:~$ nohup nice lange_rechnung &
```

Das `et`-Zeichen koppelt den Prozess von der Sitzungshell ab, sodass sie sofort wieder bereit ist. Das Vorkommando `nohup` (no hangup) lässt den Prozess weiterlaufen, auch wenn die Sitzung beendet wird. Angezeigt wird der nice-Faktor und die sich ergebende Priorität der Prozesse mittels:

```
joe@debian:~$ ps -l
```

Mit dem Kommando `renice` lässt sich die Priorität eines bereits laufenden Prozesses ändern.

Das Vorkommando `batch` startet Prozesse erst dann, wenn die CPU-Last unter eine bestimmte Schranke sinkt. Eine netzfähige Weiterentwicklung sind verteilte Batchsysteme und Load Balancer, bei Debian vertreten durch GNU Queue (Aufruf `queue`) aus der Abteilung Administration Utilities, die Aufträge oder Jobs im Netz auf mehrere Maschinen je nach deren Auslastung verteilen. Alle Rechner, auf denen ein `queue`-Server oder -Dämon (`queued`) läuft, bilden einen Pool oder Cluster, der von `queue`-Clients Aufträge entgegen nimmt. Ein Client vergibt einen Auftrag an den Server, der sich seiner Meinung nach momentan am meisten langweilt. Das ist ein Schritt hin zu verteiltem Rechnen. Es versteht sich, dass auf jedem Server das auszuführende Programm vorhanden sein muss; `queue` verteilt nur Aufträge, keine Software. Das Einrichten eines solchen Clusters fällt unter die Aufgaben des Verwalters; er findet Informationen unter <http://www.gnuqueue.org/> und <http://www.gnu.org/software/queue/queue.html>. Ein Benutzer erfährt mittels:

```
joe@debian:~$ queue -- hostname
```

(Leerzeichen um das Paar von Bindestrichen), welcher Server den nächsten Auftrag erhalten würde. Eine einfache Lösung wie `queue` lohnt sich schon in kleinen Netzen, wenn immer wieder einmal Rechenleistung verlangt wird und sich voneinander unabhängige Aufgaben verteilen lassen. Speziell zum Übersetzen (Compilieren) von C/C++-Programmquellen auf mehreren Maschinen gleichzeitig ist der verteilte Compiler `distcc` aus dem Samba-Projekt gedacht (<http://distcc.samba.org/>). Wer ein vertieftes Interesse an diesen Fragen und Lösungen hat, suche im Netz nach den Stichworten *PVM* (Parallel Virtual Machine), *Grid-Computing*, *Mosix*, *Linux Networx* oder *Beowulf*.

3.2.3 Jobkontrolle (fg, bg, jobs, nohup)

Das Betriebssystem führt Kommandos oder Programme in Form von Prozessen aus, die man sich mittels des Kommandos `ps` auflisten lassen kann. Die Shell, mit der unsere Sitzung am Terminal beginnt, kennt daneben Jobs. Ein Job ist ein Prozess – auch eine Prozessgruppe – der durch ein Kommando aus der Kommandozeile gestartet wurde. Die Shell nummeriert ihre laufenden Jobs durch, mit 1 beginnend, unabhängig von der Prozess-ID. Haben wir mehrere Desktops mit jeweils mehreren Terminals (Konsole, `xterm` usw.) gestartet, so steckt hinter jedem Terminal jeweils eine Shell mit eigener Jobverwaltung. Man kann sich daher nicht aus einem Desktop oder einem Terminalfenster die Jobs eines anderen Terminals oder Desktops ansehen. Dagegen zeigt das Kommando `ps` mit der entsprechenden Option stets alle Prozesse auf dem System an, sogar die anderer Benutzer.

Ein Kommando, das man der Shell übergibt, führt diese üblicherweise im Vordergrund (E: foreground, F: avant-plan) oder synchron aus. Sie meldet sich erst dann zurück, wenn das Kommando abgearbeitet ist. Das ist für viele Umstände ein sinnvolles Verhalten. Es kommt aber auch vor, dass man ein Programm – beispielsweise einen Web-Browser – startet, das während der ganzen Sitzung laufen soll, und sofort wieder die Shell haben möchte, um mit ihr weiter zu arbeiten. Dann muss man das Programm – den Web-Browser – im Hintergrund (E: background, F: arrière-plan) oder asynchron starten. Dies geschieht, indem man dem Programmaufruf ein `&`-Zeichen (E: ampersand, F: perluète) mitgibt:

```
joe@debian:~$ netscape &
```

Es kann immer nur ein Job im Vordergrund laufen, das heißt, mit dem Terminal verbunden sein und über dieses seine Eingabe und Ausgabe abwickeln. Im Hintergrund können viele Jobs weiterlaufen oder auch warten. Ihnen steht vorübergehend kein Terminal zur Verfügung. Das Kommando zur Anzeige der Jobs einer Shell lautet `jobs`. Wir können es ausprobieren, indem wir einige Kommandos aufrufen, die nicht in Windeseile fertig sind, und sie mit der Tastenkombination `<ctrl>-<z>` in den Hintergrund schicken und stoppen (suspendieren):

```
joe@debian:~$ man vi
```


Die Manualseite erscheint und wartet. Wir schicken sie mit `<ctrl>-<z>` in den Hintergrund (suspendieren den Job) und bekommen den Prompt der Shell wieder, eventuell erst nach Eingabe von `<cr>`. Das Spielchen wiederholen wir mit verschiedenen Manualseiten. Schließlich lassen wir uns alle Jobs anzeigen:

```
joe@debian:~$ jobs
```

Die Ausgabe sieht wie diese aus:

```
[1]      Stopped      man vi
[2]      Stopped      man gnumeric
[3]+    Stopped      man bash
```

Das Pluszeichen markiert den zuletzt suspendierten Job. Wir holen ihn wieder in den Vordergrund, indem wir seine Nummer mit einem vorangestellten Prozentzeichen aufrufen oder das Shell-Kommando `fg` ohne Argument aufrufen. Wollen wir den Job Nr. 1 in den Vordergrund holen, rufen wir das Kommando mit einem Argument auf:

```
joe@debian:~$ fg %1
```

Der Unterschied zwischen einem Job, der bei seinem Start mittels angehängtem `et`-Zeichen sofort in den Hintergrund geschickt wurde, und einem Job, der mittels `<ctrl>-<z>` suspendiert wurde, besteht darin, dass ersterer im Hintergrund läuft (Running), während ein suspendierter Job wartet (Stopped). Einen suspendierten Job bringe ich mit dem Shell-Kommando `bg` dazu, im Hintergrund weiter zu laufen.

Alle während einer Sitzung gestarteten Prozesse sind Abkömmlinge der Sitzung und werden mit dieser beendet. Für lange laufende Programme – beispielsweise Rechnungen, die sich über eine Nacht erstrecken – ist dieses Verhalten unzumutbar. Mittels des Vorkommandos `nohup` (no hangup) werden sie von der Sitzung abgekoppelt und laufen bei deren Beendigung weiter. Sie verfügen dann natürlich über kein Kontrollterminal mehr und müssen Ein- und Ausgabe über Dateien abwickeln. Meist werden solche Programme zugleich asynchron aufgerufen, damit der Benutzer sofort wieder seine Shell zur Verfügung hat:

```
joe@debian:~$ nohup lange_laufendes_programm &
```

Das sind noch nicht alle Möglichkeiten, den Rechner zu beschäftigen. Vermutlich ist an dieser Stelle eine kleine Übersicht willkommen:

- Der Benutzer kann aus der Kommandozeile heraus einen Prozess oder Job im Vordergrund starten. Für die Laufzeit des Jobs gehören Bildschirm und Tastatur dem Vordergrund-Job. Das ist der häufigste Fall.
- Der Benutzer kann einen Vordergrund-Job durch Eingabe von `<ctrl>-<z>` suspendieren. Dann geht der Job in den Hintergrund und wartet.
- Der Benutzer kann aus der Kommandozeile heraus einen Prozess oder Job durch Anhängen des `et`-Zeichens sofort in den Hintergrund schicken. Der Job läuft dann im Hintergrund.

- Bei einer grafischen Benutzeroberfläche (X11) kann ein Benutzer mehrere Terminalfenster gleichzeitig offen haben, hinter jedem eine Shell mit eigener Jobverwaltung. Nur ein Fenster kann jeweils aktiv sein (den Fokus haben), das heißt, Eingaben von der Tastatur annehmen.
- Bei einer grafischen Benutzeroberfläche (X11) können mehrere (meist vier) Desktops mit jeweils eigenen Fenstern eingerichtet sein. Zwischen den Desktops wird mittels `<ctrl>-<Fn>` (Funktionstaste) oder der Maus umgeschaltet.
- Unter Linux stellt der Kern sechs virtuelle Konsolen bereit, zwischen denen mittels `<alt>-<Fn>` (ohne X11) oder `<ctrl>-<alt>-<Fn>` (mit X11) umgeschaltet wird. Die siebte Konsole ist üblicherweise die, auf der X11 läuft, die anderen sind zunächst Textkonsolen. Auf jeder Konsole kann eine Sitzung laufen, auch von unterschiedlichen Benutzern. Im Prinzip könnten an einer seriellen Schnittstelle über einen Multiplexer auch mehrere reale Konsolen angeschlossen sein; das ist heutzutage aber die Ausnahme.
- Ein Benutzer kann mittels des Vorkommandos `nohup` einen Job starten, der von der Sitzung abgekoppelt ist und bei deren Beendigung weiterläuft, natürlich ohne Terminal oder Shell.
- Schließlich kann sich ein Benutzer über das Netz per `telnet` oder besser `slogin` einloggen und auf dem Rechner arbeiten. Er kann auch auf dem Rechner einen X-Client laufen haben, der den X-Server auf einem anderen Rechner für die Ein- und Ausgabe nutzt.

Heutige CPUs sind nur selten überfordert, eher wird der Arbeitsspeicher knapp. Multimedia-Anwendungen sind der häufigste Fall, in dem auch eine schnelle CPU verlangt ist.

3.2.4 Signale (kill)

Einem Prozess kann man ein Signal schicken und ihn damit zu einem bestimmten Verhalten auffordern. Dazu braucht man den Namen oder die Nummer des Signals und die Prozess-ID. Das Kommando lautet `kill`, weil eines der Verhalten ist, dass sich der Prozess beendet. Der Aufruf:

```
joe@debian:~$ kill -9 1234
```

oder:

```
joe@debian:~$ kill -s KILL 1234
```

schickt das Signal 9/KILL an den Prozess mit der ID 1234 und fordert ihn zum sofortigen Selbstmord ohne vorherige Aufräumarbeiten etc. auf. Das Default-Verhalten ist, dass sich der Prozess geordnet beendet (Signal 15/TERM), aber das kann der Urheber des dem Prozess zu Grunde liegenden Programms geändert haben. Nur die Signale 9/KILL und 24/STOP lassen sich nicht umfunktionieren. Ohne Angabe eines Signals schickt das Kommando `kill` das Signal 15/TERM an den Prozess und fordert ihn zum geordneten Rückzug auf:

```
joe@debian:~$ kill 1234
```

Will man einen Prozess beenden, versucht man es zuerst im Guten mit dem obigen Aufruf. Fruchtet dieser nicht, greift man zur Axt in Form des Signals 9/KILL.

Ein oft gebrauchtes Signal ist 1/HUP (Hangup). Es veranlasst viele Dämonen zu einem Reset, also einem erneuten Einlesen ihrer Konfiguration. Die Tastenkombination `<ctrl>-<c>` sendet das Signal 2/INT (Interrupt), was viele Programme beendet. Es liegt aber in der Hand der Programme, in anderer Weise auf die Signale zu reagieren. Auf den jeweiligen Manualseiten wird die Reaktion beschrieben, beispielsweise beim `inetd`.

Eine Liste der unter Debian GNU/Linux verfügbaren Signale findet sich im Anhang auf Seite 467. Man erhält die Liste auch durch:

```
joe@debian:~$ kill -l
```

und auf der Manualseite zu `signal 7`.

3.2.5 Pipes

Eine Pipe – markiert durch den senkrechten Strich – verbindet `stdout` eines Kommandos mit `stdin` des nächsten Kommandos:

```
joe@debian:~$ ps -ef | grep exim
```

Die Ausgabe von `ps -ef` gelangt nicht auf den Bildschirm, sondern wird gleich als Eingabe an `grep exim` weitergereicht, das die Zeilen herausfiltert, die die Zeichenkette `exim` enthalten, und auf den Bildschirm schreibt. Unter einer derartigen Pipe hat man sich einen Pufferspeicher im Kern vorzustellen, in den das erste Kommando hineinschreibt und von dessen anderem Ende das zweite Kommando liest. Pipes dürfen hintereinander geschaltet werden. Das kleine Shellskript Quelle 3.2 zeigt eine sechsgliedrige Pipe.

```
# Shellskript frequenz, Frequenzwoerterliste
cat $* |
tr "[A-Z]" "[a-z]" |
tr -c "[a-z]" "[\012*]" |
sort |
uniq -c |
sort -nr
```

Quelle 3.2 . Shellskript `frequenz` zur Erzeugung einer nach Häufigkeit des Vorkommens sortierten Wörterliste (Frequenz-Wörterliste)

In mehrzeiligen Pipes muss der Pipe-Operator am Ende einer Zeile stehen. In der ersten Zeile wird eine Textdatei mittels `cat` gelesen und nach `stdout` geschrieben. In der zweiten Zeile werden mittels `tr` alle Großbuchstaben durch Kleinbuchstaben ersetzt. In der dritten werden alle Zeichen, die keine Kleinbuchstaben sind, durch einen Zeilenwechsel (Line Feed, ASCII-Zeichen Nr. 12 oktal) ersetzt, sodass jedes Wort für sich in einer Zeile steht. Dann werden die Zeilen alphabetisch sortiert und durch das Kommando `unique` alle mehrfachen Zeilen gelöscht. Die Option `-c` bewirkt, dass `unique` dabei die Anzahl gleicher Zeilen zählt und der verbliebenen

Zeile voranstellt. Schließlich werden die Zeilen numerisch in absteigender Folge sortiert. Solche Frequenz-Wörterlisten braucht man bei Stilanalysen und zum Vokabellernen.

Neben dieser durch die Shell bereit gestellten, namenlosen Pipe gibt es vom Benutzer eigens erzeugte, benannte Pipes, die seltener gebraucht werden. Die Eingabe (für das Beispiel zweckmäßig im Home-Verzeichnis):

```
joe@debian:~$ mkfifo -m 660 pfeife
```

erzeugt eine Pipe (FIFO = first in, first out) namens `pfeife`, die wie eine Datei im Dateisystem auftaucht, und zwar mit den Zugriffsrechten 660. Schreibt man nun eine Textdatei in die benannte Pipe:

```
joe@debian:~$ cat textdatei > pfeife
```

so kann man in einem anderen Fenster oder virtuellen Terminal die Pipe auslesen:

```
joe@debian:~$ cat < pfeife
```

Genau wie bei einer Datei können je nach Zugriffsrechten auch andere Benutzer auf die benannte Pipe zugreifen. Gelöscht wird sie mit `rm`.

3.3 Drucken

3.3.1 Drucksysteme (lp, lpr, CUPS)

Im Prinzip könnte man unter Linux/UNIX drucken, indem man die zu druckende Datei auf die Gerätedatei des Druckers kopiert. Das hätte zwei Nachteile:

- Auf einem Mehrbenutzersystem könnten zwei Benutzer gleichzeitig auf den Gedanken verfallen, drucken zu wollen. Ein Drucker druckt gnadenlos das, was bei ihm eintrifft, in diesem Fall Textsalat. Deshalb müssen sich die Druckaufträge (E: request, F: requête) in Warteschlangen einreihen (gespoolt werden).
- Text- oder Grafikdateien gibt es in vielerlei Formaten, Drucker haben unterschiedliche Fähigkeiten und sind unterschiedlich konfiguriert. Die meisten Dateien müssen durch ein Programm, ein Filter bearbeitet werden, ehe sie zum Drucker gehen.

Aus diesen und weiteren Gründen nehmen Druckaufträge unter Linux/UNIX den Umweg durch ein Drucksystem. Das Einrichten und Pflegen des Drucksystems ist Aufgabe des Verwalters. Das *Linux Printing HOWTO*, aktuell von 2003, wendet sich an Benutzer und Verwalter, das *Printing Usage HOWTO* an Benutzer, ist aber angeht.

Bevor wir in die Tiefen des Drucksystems steigen, wollen wir uns zwei Dinge verdeutlichen, deren Verständnis die Einrichtung, die Benutzung und die Fehleranalyse erleichtert. Drucker sind angeschlossen:

- lokal an den Arbeitsplatz-Rechner über eine parallele oder serielle Schnittstelle, den Universal Serial Bus (USB) oder drahtlos über Bluetooth oder

- an einen anderen Rechner im Netz über eine der vorstehend genannten Schnittstellen oder
- direkt an das Netz über eine eigene Ethernet-Schnittstelle.

In den ersten beiden Fällen hat man auf dem Rechner alle Möglichkeiten, die Druckaufträge zu beeinflussen. Der Drucker selbst braucht nur wenig eigene Intelligenz. Im dritten Fall ist der Drucker ein Knoten im Netz mit eigener Adresse wie ein Rechner. Er muss so viel Intelligenz mitbringen, dass er die Protokolle beherrscht und sich im Netz gesittet verhält. Andererseits reicht die Intelligenz handelsüblicher Netzdrucker nicht an die eines Rechners heran, sodass der Drucker nur einen Teil der Aufgaben des Drucksystems übernehmen kann. In kleinen Netzen findet man häufiger die erste oder zweite Lösung, in mittleren die dritte.

Tabelle 3.3. Druckkommandos aus BSD und System V

Aufgabe	BSD	System V
drucken	lpr	lp
auf ljet9 drucken	lpr -P ljet9	lp -d ljet9
x Exemplare drucken	lpr -x	lp -n x
Dämon	lpd	lp sched
Auftrag löschen	lprm	cancel
Aufträge ansehen	lpq	lpstat

Der zweite Punkt betrifft die Frage, wo das Ergebnis eines Druckauftrags beeinflusst werden kann. Das sind leider mehrere Stellen:

- Der Autor kann mit einem Editor in seine Texte Steuerzeichen (Escape-Sequenzen) einfügen, beispielsweise um die Breite des linken Randes vorzugeben. Hierhin gehört auch die Darstellung des Zeilenwechsels (LF, CR) in den unterschiedlichen Rechnerwelten (Linux/UNIX, Macintosh, DOS/MS-Windows).
- Das den Druckauftrag erteilende Programm – ein Browser, ein Textprozessor, der Adobe Reader – meint auch, dem Drucker Anweisungen geben zu müssen.
- Filter im Drucksystem machen alles mit dem Text, was sie für richtig halten.
- Der Drucker hat Voreinstellungen, die ihm teils per Hardware, teils per Software mitgeteilt worden sind.

Das Gesagte bezieht sich auf Text; bei Grafik liegen die Verhältnisse teilweise anders, sind aber nicht einfacher. Mit Farbe kommt noch eine Fehlerquelle hinzu. Wenn ein Druckergebnis nicht den Erwartungen entspricht, geht die Suche nach der Ursache los. Niemand will es gewesen sein. Hier für Ordnung zu sorgen, ist Aufgabe des Verwalters.

Unter Linux/UNIX sind zwei klassische und einige neuere Drucksysteme verbreitet. Die klassischen sind:

- BSD-Drucken (Berkeley-Drucken), Druckkommando `lpr`,
- System-V-Drucken (AT&T-Drucken), Druckkommando `lp`.

Die beiden Systeme unterscheiden sich wenig in ihren Fähigkeiten, wohl aber in ihren Kommandos und deren Optionen. Unter Linux ist das BSD-Drucken stärker verbreitet. In Tabelle 3.3 sind die wichtigsten Kommandos beider Systeme aufgelistet. Fragen Sie Ihren Verwalter, welche Kommandos Sie verwenden sollen.



Abbildung 3.10. Screenshot des Common UNIX Printing Systems (CUPS), Druckerstatus

Von den neueren Systemen wollen wir uns das Common UNIX Printing System (CUPS) ansehen, das in vernetzten Umgebungen die sauberste und leistungsfähigste Lösung darstellt. Auf einem Server (oder mehreren) im Netz läuft ein CUPS-Dämon namens *cupsd* aus dem Debian-Paket *cupsys*. Er kennt die in seinem Netz verfügbaren Drucker und wird mittels eines Web-Browsers auf Port 631 angesprochen. Auf den Arbeitsplätzen laufen CUPS-Clients aus dem Debian-Paket *cupsys-client*, eventuell ergänzt durch das Debian-Paket *cupsys-bsd* und weitere. Die Konfiguration der Clients findet sich in `/etc/cups/client.conf`, vor allem der Name des Servers. Von Haus aus versteht CUPS die System-V-Kommandos, mit dem Zusatzpaket auch die BSD-Kommandos. In Abbildung 3.10 sehen wir die Webseite eines CUPS zum Druckerstatus. Es ist zu erkennen, dass es sich hier um einen direkt im Netz befindlichen Laserdrucker handelt, der mit dem CUPS-Server über das Internet Printing Protocol (IPP) nach den RFCs 2910, 2911 und 3380 verkehrt. Das Protokoll muss im Netzinterface des Druckers eingeschaltet werden, was ebenfalls über eine Webseite möglich ist. Wie bei den anderen Drucksystemen lassen sich die Warteschlangen unabhängig von den Druckern ein- und ausschalten. Es kann also ein Drucker vorübergehend außer Betrieb sein, während seine Warteschlange weiterhin Aufträge entgegen nimmt. Das ist bei kurzen Betriebsunterbrechungen (Tonerwechsel) zweckmäßig. Durch Hinzufügen von Benutzern zur Gruppe *lpadmin* auf dem

Server können diesen Verwaltungsrechte am CUPS eingeräumt werden, sodass nicht bei jeder Störung der Verwalter belästert zu werden braucht.

Zwei Tipps: Bei vielen Laserdruckern lässt sich die Tonerdicke per Hard- oder Software einstellen. Bei der Wiedergabe von Fotos führt oft eine geringere Tonerdicke zu besseren Ergebnissen, und man spart noch dabei. Wenn der Toner am Ende ist, kann man noch ein paar Seiten herausholen, wenn man die Patrone einige Male in waagrecht Lage schüttelt.

3.3.2 PostScript (psutils, a2ps)

Drucker verfügen über eine eingebaute Kommando-Sprache. Weit verbreitet ist die Hewlett-Packard Printer Control Language (PCL). Ein Drucker berechnet aus den bei ihm eintreffenden Daten (Nutzdaten samt Kommandos) ein Rasterbild (Bitmap), das auf das Papier gedruckt wird. Für diese Arbeit benötigt ein Drucker etwas Intelligenz und Speicher.

Die von der Firma Adobe entwickelte Seitenbeschreibungssprache PostScript wird mittlerweile ebenfalls von vielen Druckern verstanden. Der Rechenaufwand im Drucker ist höher als bei PCL. Die Eigenschaften eines PostScript-Druckers sind in einer PostScript Printer Description (PPD) niedergelegt. Die Drucker erkennen selbst, ob ein Dokument als PostScript- oder PCL-Datei ankommt. Obwohl es grundsätzlich möglich ist, PCL- oder PostScript-Anweisungen von Hand mit einem Editor in einen Text einzubauen, wird das praktisch kaum gemacht. Dagegen erzeugen viele Anwendungen wie Web-Browser zum Drucken unmittelbar PostScript-Ausgaben. Dieser Sprache kommt daher beim Drucken eine besondere Bedeutung zu.

Das Debian-Paket *psutils* enthält eine Sammlung von Werkzeugen zur Bearbeitung von PostScript-Dateien. Näheres bei <http://www.knackered.org/angus/psutils/>. Die wichtigsten Werkzeuge aus der Sammlung sind:

- *psbook* strukturiert eine PostScript-Datei so, dass sich beim Drucken eine Art von Buch ergibt,
- *pselect* wählt Seiten aus einer PostScript-Datei aus,
- *pstops* strukturiert eine PostScript-Datei um und ermöglicht dabei einige Änderungen,
- *psnup* führt zur Ausgabe mehrerer Textseiten auf einer Seite Papier,
- *psresize* passt die Datei an ein Papierformat an,
- *psmerge* fasst mehrere PostScript-Dateien zusammen,
- *fixfmps* passt Framemaker-Dateien an die PSUtils an,
- *fixmacps* passt Macintosh-Dateien an die PSUtils an,
- *fixwfwps* passt MS-Word-Dateien an die PSUtils an.

Die Sammlung gibt es auch für andere Betriebssysteme.

Das Kommando *a2ps* wandelt mehrere Textformate – nicht nur ASCII – nach PostScript um und erlaubt, die Druckausgabe zu beeinflussen. Die zugehörige Info-Seite ist ausführlicher als das Manual. Werkzeuge zum Lesen von PostScript-Dateien auf dem Bildschirm werden in Abschnitt 5.2 *Werkzeuge zum Lesen* auf Seite 168 vorgestellt.

Ghostscript ist ein Softwarepaket, das PostScript- oder PDF-Dateien in andere Formate umwandelt oder auf einem Drucker oder Bildschirm darstellt. Hat man einen einfachen Drucker, der kein PostScript versteht, schaltet man Ghostscript als Filter davor und kann PostScript- oder PDF-Dateien drucken. Die Software ist unter verschiedenen Lizenzen erhältlich; das Debian-Paket *gs-gpl* bringt die nach der GPL lizenzierte Fassung mit. Ghostview und GV sind Frontends zu Ghostscript, letzteres enthalten im Debian-Paket *gv*. Der Heimathafen für die Pakete ist <http://www.cs.wisc.edu/~ghost/>. Wir kommen im Abschnitt 5.2.3 *Lesen und Umwandeln von bzw. nach PostScript* auf Seite 170 noch einmal darauf zurück.

3.4 Benutzer, Maschine (who, id, rwho, hostname)

Manchmal möchte man wissen, wer gerade auf dem System angemeldet ist. Diesen Wunsch erfüllt das Kommando `who`, zweckmäßig mit Option `who -H` (headers). Hat man seinen eigenen Benutzernamen vergessen, hilft `whoami` weiter. Wenn man im Netz unter verschiedenen Namen auf verschiedenen Rechnern arbeitet, kommt so etwas vor. Meist möchte man sich dann auch vergewissern, auf welchem Rechner man sich gerade aufhält; das Kommando `hostname` teilt das mit. Das Kommando `id` (identity) sagt etwas ausführlicher, wer man ist:

```
uid=1000(wulf) gid=1900(www) groups=1000(wulf),4(adm),6(disk),
25(floppy),29(audio),50(staff),1112(svn),1111(alex),1900(www)
```

Das bedeutet, ich bin unter der Benutzer-ID 1000, Benutzername `wulf`, und der Gruppe mit der ID 1900, Gruppenname `www`, auf dem System angemeldet. Diese Gruppe ist nicht meine primäre Gruppe; ich habe also nach dem Einloggen meine Gruppenzugehörigkeit mit dem Kommando `newgrp` gewechselt. Die weiteren Gruppen, in denen ich Mitglied bin, sind anschließend aufgeführt. Mit einem fremden Benutzernamen als Argument aufgerufen, ermittelt es Benutzer-ID, Gruppen und Gruppen-IDs des fremden Benutzers.

Falls auf den Maschinen im lokalen Netz das Kommando `rwho` samt seinem Gesprächspartner `rwhod` eingerichtet ist, lassen sich `who`-ähnliche Informationen aus dem lokalen Netz abfragen. Der Dienst braucht Zeit zum Aktualisieren.

Die Informationen über das eigene Ich samt dem Namen des Arbeitsverzeichnisses lassen sich im Prompt unterbringen. Dazu ist – wenn es nicht schon der Verwalter systemweit eingerichtet hat – der Datei `$HOME/.bashrc` folgende Zeile hinzuzufügen:

```
PS1=\u@\h:\w\$
```

Die Buchstaben bedeuten user, host und working directory. Das Dollarzeichen ist der unter Linux/UNIX übliche Prompt für gewöhnliche Benutzer. Auf ihn sollte ein Leerzeichen folgen.

Das Kommando `who -r` zeigt den aktuellen Runlevel des Systems an. Das hat nichts mit Benutzern zu tun. Ein Runlevel ist ein Zustand des Systems, in dem bestimmte Dämonen aktiv sind, das heißt bestimmte Dienste laufen. Die Runlevel werden in der Datei `/etc/inittab` deklariert:

- Runlevel 0 bedeutet halt,
- 1 ist der Single-User-Modus, den der Verwalter für größere Arbeiten am System braucht (Minimum an Systemaktivität),
- 2 bis 5 sind Multi-User-Modi, deren Ausgestaltung in der Hand des Verwalters liegt, In der Regel ist 2 der Normalmodus (default runlevel), der beim Booten gestartet wird.
- 6 bedeutet neu booten (Warmstart, reboot).

Definiert werden die Runlevel in den Verzeichnissen `/etc/rc*.d`. Dort kann man nachsehen, welche Dienste im jeweiligen Runlevel gestartet oder gestoppt werden. Mit den Runleveln lassen sich unterschiedliche Systemeigenschaften für Tag- und Nachtbetrieb oder für Werktags- und Wochenendbetrieb verwirklichen. Auf Nicht-Debian-Systemen können die Runlevel geringfügig anders ausgestaltet sein.

3.5 Kalender, Uhr (date, leave, cal, cron, at, calendar)

Auf einem Rechner braucht man für viele Zwecke Datum und Uhrzeit. Dateien führen drei Zeitstempel, Mails werden mit Zeitangaben versehen, Datenbanken fügen bei Bedarf automatisch den Zeitpunkt des Eintrags eines Datensatzes ein. Es ist lästig, wenn die Systemzeit nennenswert von der bürgerlichen Zeit abweicht. Für die korrekte Systemzeit zu sorgen, ist Aufgabe des Verwalters. Er hat zwei Möglichkeiten, die genaue Zeit zu beziehen, um sie in das lokale Netz einzuspeisen:

- ein Rechner im LAN verfügt über eine Funkuhr (Hardware),
- das LAN ist an das Internet angeschlossen und synchronisiert sich mit öffentlichen Zeitservern.

Die erste Lösung setzt einen Standort mit gutem Empfang der Signale des Senders DCF77 voraus, was bei Servern im Keller unwahrscheinlich ist. Im zweiten Fall synchronisiert sich der lokale Zeitserver mit Kollegen im Internet, wofür es ein Internet-Protokoll gibt (<http://www.ntp.org/>). Die amtliche deutsche Zeit wird von der Physikalisch-Technischen Bundesanstalt (PTB) in Braunschweig gemacht und über DCF77 und Zeitserver wie `ntp1.ptb.de` verteilt. Große Provider betreiben ebenfalls Zeitserver.

Die Arbeitsplätze richten sich in beiden Fällen nach dem lokalen Zeitserver. Die Kommandos zum Synchronisieren eines Rechners mit einem Zeitserver lauten `ntpdate` oder `rtime`. Nur wenig aufwendiger ist es, den lokalen Zeitserver als Broadcasting NTP Server und die Arbeitsplätze als Broadcast Clients einzurichten, aber das ist Sache des Verwalters.

Ein Benutzer ruft mit dem Kommando `date` Datum und Uhrzeit als Text ab:

```
joe@debian:~$ date
```

Die Ausgabe lässt sich durch Angabe eines Formates vielfältig gestalten. Unter X11 stehen verschiedene analoge oder digitale Uhren (`xclock`) zur Verfügung, ferner eine Art von Wecker in Form des Kommandos `leave`. Das Kommando erinnert den in seine Arbeit versunkenen Benutzer 5 und 1 min vor der als Argument übergebenen Uhrzeit daran, Schluss zu machen. Man wird den Wecker los, indem man sich ausloggt oder ihm ein Kill-Signal schickt. Fünf Minuten über der Zeit verabschiedet sich der Wecker von selbst. `leave` macht sich nur in dem Fenster bemerkbar, in dem es aufgerufen wurde; zweckmäßiger wäre ein Wecker, der in allen Fenstern auf allen Desktops einer Sitzung bimmelt. Der KDE-Desktop bietet unter dem Menüpunkt *Kleinigkeiten* -> *Weltzeituhr* eine hübsche Weltkarte mit Angabe der lokalen Zeiten an.

Das Kommando `cal` zeigt ohne weitere Argumente aufgerufen einen einfachen Kalender des laufenden Monats an. Man kann aber auch einen bestimmten Monat eines bestimmten Jahres auswählen:

```
joe@debian:~$ cal 9 1752
```

zeigt den September des Jahres 1752 an. Und dann erweist es sich, wer im Geschichtsunterricht aufgepasst hat. Einen bunten Kalender richtet das Debian-Paket `ccal` ein. Das Paket bringt eine einfache Terminverwaltung mit.

Der `cron` ist ein Dämon, der auf vielen Linux/UNIX-Systemen eingerichtet ist und die meiste Zeit verschläft. Jede Minute wacht er kurz auf und schaut in einer Liste nach, ob etwas für ihn zu erledigen ist. Verwalter übertragen ihm regelmäßig wiederkehrende Aufgaben wie das Anlegen von Sicherungskopien (Backup) oder das Aufräumen temporärer Verzeichnisse. Aber auch der gewöhnliche Benutzer kann ihm Aufträge erteilen. Dazu legt der Benutzer in seinem Home-Verzeichnis eine Datei `crontab` – der Name ist beliebig – mit Zeilen folgender Art an:

```
05 5 * * * /usr/bin/fetchmail
```

Die erste Spalte enthält Minuten (0 - 59), die zweite Stunden (0 - 23), die dritte Tage (1 - 31), die vierte Monate (1 - 12) und die fünfte Wochentage (0 - 7), wobei der Sonntag mit 0 oder 7 bezeichnet ist. Der Stern bedeutet *jede/jeder*. Der obige Auftrag lautet demgemäß: An jedem Tag um 5.05 Uhr rufe das Kommando `/usr/bin/fetchmail` auf. Da keine Umgebung eingerichtet ist, sind die Kommandos mit ihrem absoluten Pfad einzutragen. Anschließend übergibt man die Datei dem `cron`:

```
joe@debian:~$ crontab crondat
```

und überprüft mittels:

```
joe@debian:~$ crontab -l
```

ob alles stimmt. In lokalen Netzen mit vielen Cronjobs sollte der Verwalter Zeitfenster für bestimmte Aufgaben festlegen. Für nicht zeitgebundene Aufgaben nimmt man gern die frühen Morgenstunden, weil dann am wenigsten Verkehr herrscht.

Für in der Zukunft liegende einmalige Aufgaben verwendet man das Vorkommando `at`. Der Aufruf:

```
joe@debian:~$ at 2230 /home/wulf/projektX/berechnung
<ctrl>-<d>
```

bewirkt, dass heute um 22.30 Uhr das genannte Programm gestartet wird, vorausgesetzt die Maschine läuft. Für die Zeitangabe sind viele Formate erlaubt.

Bleiben aus irgendeinem Grund Aufträge liegen, werden sie nicht nachgearbeitet, wenn die Maschine wieder betriebsbereit ist. Will man erreichen, dass auch die liegegebliebenen Aufträge bei nächster Gelegenheit erledigt werden, ist das Debian-Paket *anacron* einzurichten. Das empfiehlt sich auf Rechnern, die nicht ständig laufen.

Mit Hilfe des Kommandos `calendar` lässt sich ein einfacher Terminkalender (Reminder Service) einrichten. Man braucht dazu eine Datei `calendar` in seinem Home-Verzeichnis, welche die Termine enthält, und einen Eintrag folgender Art in der systemweiten oder persönlichen Crontab:

```
05 30 * * * /usr/bin/calendar -a -l 2 -w 3
```

An jedem Morgen um 5 Uhr 30 wird das Kommando `calendar` aufgerufen, und zwar für alle Benutzer (es handelt sich hier also um die Crontab von root), wobei die Termine von heute und den nächsten zwei Tagen per Email an die jeweiligen Benutzer versandt werden. Freitags, vor einem Wochenende, soll sich die Zukunft über drei Tage erstrecken. Es ist zweckmäßig, die Termindatei aufzuspalten in eine Datei mit jährlich wiederkehrenden Terminen (Geburtstagen) und eine Datei mit einmaligen Terminen. Die Geburtstagsdatei bekommt obigen Namen und enthält Zeilen wie diese:

```
#include <calendar2>
08/28 Goethe
11/10 Schiller
```

Die erste Zeile zieht die zweite Termindatei ein in einer Schreibweise, die aus der Programmiersprache C stammt. Dann folgen auf jeweils einer Zeile die Termine, wobei für das Datum mehrere Formate zulässig sind. Wichtig ist ein Tabulatorzeichen vor dem Text. Einzelheiten (systemweite Termine, bewegliche Feiertage, kyrillischer Kalender) im Manual. Bei Debian finden sich weitere, leistungsfähigere Terminverwaltungen, siehe Abschnitt 11.9.3 *Termine* aus Seite 388.

Außer unserem abendländischen (gregorianischen) Kalender gibt es auf der Erde weitere Kalender, und selbst im Abendland wurden bis vor hundert Jahren in katholischen, protestantischen und orthodoxen Ländern unterschiedliche Kalender verwendet. Einen schnellen Einstieg in das Thema *Kalender* vermittelt die Wikipedia, umfangreicher geraten sind die *FAQ about Calendars* unter <http://www.tondering.dk/claus/calendar.html> von CLAUS TØNDERING, die Webseite *Chronologie und Kalender* von NIKOLAUS A. BÄR unter <http://www.nabkal.de/> und die Erklärungen von JENS SCHNEIDER unter <http://www.kalendersysteme.de/deutsch/>. Einen Aufsatz von JÜRGEN MALITZ über den Werdegang des julianischen Kalenders, eingeführt durch CAIUS JULIUS CAESAR – den wir mit der Korrektur durch Papst GREGOR XIII.

noch heute gebrauchen – findet der geneigte Leser unter <http://www.gnomon.ku-eichstaett.de/LAG/kalender.html>. Die Suche nach einem allgemeinen Werkzeug für Linux zur Umrechnung verschiedener Kalender verlief ohne Ergebnis.

3.6 Secure Shell (ssh)

Über ein Netz lässt sich auf einem entfernten Rechner arbeiten. Der eigene Rechner vor Ort wird dabei zu einem Terminal des entfernten Rechners. Wir haben so etwas schon in Abschnitt 1.3.2 *X Window System (X11)* auf Seite 11 kennen gelernt. Wenn wir uns auf ein Textterminal beschränken, gibt es dafür zwei ältere, einfache Lösungen.

Im RFC 854 aus dem Jahr 1983 wird das Telnet-Protokoll spezifiziert, das eine ziemlich allgemeine, bidirektionale Kommunikation zwischen zwei Netzknoten beschreibt, insbesondere zwischen einem Prozess und einem Terminal. Unter Linux/UNIX heißt ein verbreitetes Programmpaar, welches das Protokoll realisiert, `telnet` und `telnetd`. Auf dem entfernten Rechner läuft der Telnet-Dämon, meist als Knecht des `inetd`, lokal der Telnet-Client. Wegen seiner Allgemeinheit braucht der Verwalter das Kommando `telnet` gelegentlich beim Aufspüren von Fehlern im Netz. Aus heutiger Sicht hat es den Nachteil, dass der Datenverkehr unverschlüsselt über das Netz geht, was höchstens in lokalen, überschaubaren Netzen tragbar ist. Man kann `telnet` auf seinen Rechnern zwar einrichten, sollte den Dämon aber nicht aktivieren.

Aus Berkeley stammen die *r*-Kommandos (Berkeley Services) für Linux/UNIX-Rechner, die so heißen, weil ihre Namen mit dem Buchstaben *r* wie *remote* beginnen:

- `rlogin` Anmelden auf einem entfernten Rechner, wie mit `telnet`,
- `remsh`, `rsh` oder `rexec` Ausführen von Kommandos auf einem entfernten Rechner, nicht interaktiv,
- `rcp` Kopieren zu oder von einem entfernten Rechner,
- `rdist` Verteilen von Dateien im Netz,
- `ruptime` Auflisten von Hosts im lokalen Netz,
- `rwho` Auflisten von aktiven Benutzern im Netz,
- `rwall` Verschicken von Mitteilungen an alle Teilnehmer im Netz.

Die *r*-Kommandos hatten ursprünglich nichts mit dem Internet zu tun, sie gehören in die UNIX-Welt. Nachträglich sind sie im RFC 1282 aus dem Jahr 1991 spezifiziert und auf einige andere Systeme portiert worden, aber da war ihre Zeit beinahe schon vorüber. Auch sie übertragen Daten unverschlüsselt und sind daher heute praktisch nicht mehr zu gebrauchen. Ruft man unter Debian GNU/Linux die Manualseite zu `rlogin` auf, so kommt die Seite zu einer besseren Alternative, nämlich SSH.

Die Secure Shell (SSH) von TATU YLONEN (<http://www.ssh.fi/>) ist als zeitgemäßer, sicherer Ersatz für Telnet, FTP und die Berkeley Services geschaffen worden. Sie ist kein Kommando-Interpreter wie die Bash. SSH bezeichnet sowohl das Protokoll als auch dessen Implementierung, wie wir es bei Telnet, Finger oder

FTP sehen. Meist wird unter Linux die mit der älteren Protokollversion 1 unverträgliche Protokollversion 2 der OpenSSH eingesetzt (<http://www.openssh.com/> oder <http://www.openssh.org/de/>). Die Software wird primär für OpenBSD entwickelt und anschließend auf weitere Systeme wie Linux, AIX, HP-UX oder MacOS X portiert. In der Versionsbezeichnung OpenSSH_3.9p1 weist der Buchstabe p auf die Portabilität hin. Debian hat OpenSSH in seine Distributionen integriert.

Eine Secure Shell erfüllt zwei voneinander unabhängige Aufgaben, die der Sicherheit dienen:

- Sie authentifiziert einen Benutzer oder einen Host,
- sie verschlüsselt den Datenaustausch zwischen Client und Server.

Die Authentifizierung erfolgt hostbasiert (`/etc/shosts.equiv`, `$HOME/.shosts`), über ein Schlüsselpaar oder – wenn nichts anderes Erfolg hat – über Benutzername und Passwort, siehe man `ssh`. Die Sitzung wird mit einem zwischen Client und Server ausgehandelten Schlüssel verschlüsselt. Wie gewohnt sind auf dem entfernten Rechner ein Dämon namens `sshd` und lokal ein Client namens `ssh` einzurichten, beide in dem Debian-Paket `ssh` enthalten. Unter *sarge* sind weitere Hilfsprogramme zur SSH verfügbar. Die Konfiguration liegt in den Verzeichnissen `/etc/ssh/` (systemweit) und `$HOME/.ssh/` (benutzerbezogen).

Ein Benutzer muss vor der ersten Benutzung ein persönliches Schlüsselpaar erzeugen:

```
joe@debian:~$ ssh-keygen
```

Die Schlüssel werden im persönlichen Konfigurationsverzeichnis in je einer Datei abgelegt, wovon die nicht-öffentliche die Zugriffsrechte 600 und die öffentliche die Rechte 644 haben sollen. Eine im übrigen richtige Konfiguration auf Client- und Serverseite vorausgesetzt kann ein Benutzer dann versuchen, sich auf einem entfernten Rechner einzuloggen:

```
joe@debian:~$ slogin rechnername
```

`slogin` ist ein anderer Name für das Kommando `ssh` ohne ein auszuführendes Kommando als Argument, siehe Manual. Bei der ersten Verbindung wird gefragt, ob der öffentliche Schlüssel des entfernten Rechners gespeichert werden soll, was zu bejahen ist, wenn keine Zweifel an der Verbindung bestehen. Der Schlüssel wird in der Datei `known_hosts` im persönlichen Konfigurationsverzeichnis gespeichert. Der Verwalter kann den Schlüssel bereits im systemweiten Konfigurationsverzeichnis hinterlegt haben, dann entfällt die Abfrage. Ändert der entfernte Rechner seinen Schlüssel, was beispielsweise bei Neuinstallationen vorkommt, verursacht der nunmehr veraltete Schlüssel eine Warnung. Am einfachsten löscht man mit einem Editor die ungültig gewordene Zeile in `known_hosts` und beginnt ein neues Verhältnis. Beendet wird die `slogin`-Verbindung mit `exit`, nicht anders als ein lokales Textterminal (`xterm`).

Auch das Fernkopieren mit der Secure Shell geht denkbar einfach. Mittels des Kommandos:

```
joe@debian:~$ scp lokale_datei
user@ferner_rechner:ferner_pfad
```

kopiert man eine lokale Datei unter dem angegebenen Benutzernamen auf den entfernten Rechner in den genannten Pfad. In umgekehrter Richtung sind die beiden Argumente zu vertauschen. Es lassen sich sogar von einem Rechner aus Fernkopien zwischen einem zweiten und einem dritten Rechner austauschen. Das Kommando `sftp` verhält sich ähnlich wie `ftp` und setzt auf der Gegenseite einen SFTP-Server voraus, kommuniziert also nicht mit unsicheren FTP-Servern herkömmlicher Art.

3.7 Secure Sockets Layer, Transport Layer Security (openssl, gnutls)

Im Gegensatz zur Secure Shell, die aus einem Protokoll und einigen selbständigen, das Protokoll implementierenden Programmen besteht, bringt die Secure Sockets Layer (SSL) nebst zwei Protokollen nur eine Bibliothek zur Verwendung in anderen Programmen sowie ein Werkzeugköffchen für die Kommandozeile zur Unterstützung der Bibliothek mit. Die neuere Bezeichnung des Protokolls lautet Transport Layer Security Protocol (TLS, nach RFC 2246). TLS Version 1.0 basiert auf SSL Version 3.0, ist aber nicht damit identisch. Bei SSL oder TLS findet man nicht eine Alternative zu `slogin`, aber Programmierer können mit der Bibliothek einen `ssl`-fähigen Telnet-Dämon und -Client schreiben, was auch geschehen ist. Dieses Paar ist dann eine Alternative zu `ssh` und `sshd`. Das Debian-Paket *openssl* umfasst 350 Dateien, vor allem Dokumentation, und ist Teil der OpenSSL-Implementation (<http://www.openssl.org/>). Beim DFN gibt es ein deutschsprachiges *OpenSSL Handbuch* unter <http://www.dfn-pca.de/certify/ssl/handbuch/>. Die wichtigsten mittels SSL oder TLS gesicherten Anwendungen sind HTTPS (Web), SMTP (Email), NNTP (Netnews) und SIP (Internet-Telefonie).

Ursprünglich 1996 von der Firma Netscape für die sichere Übertragung von HTML-Daten im Web entwickelt können mittlerweile auch andere Netzprotokolle, die über TCP laufen, wie Telnet, FTP oder POP damit gesichert werden. Dazu sind außer OpenSSL `ssl`-fähige Clients und Server der jeweiligen Dienste einzurichten, beispielsweise die Debian-Pakete *ftp-ssl* und *ftpd-ssl*. Auch vom Apache-Webserver findet sich eine SSL-Version bei Debian. TLS bzw. SSL erfüllt drei voneinander unabhängige Funktionen:

- Authentifizierung von Server und/oder Client,
- Datenverschlüsselung,
- Prüfung der Datenintegrität.

Die Authentifizierung stellt sicher, dass der Sender und/oder Empfänger der Daten wirklich der ist, der er zu sein behauptet. Es wäre fatal, wenn ich im Verkehr mit meiner Bank einer betrügerischen Website auf den Leim ginge. Deshalb weist sich der Webserver der Bank gegenüber meinem Browser aus (`https`). Umgekehrt weist

sich mein Browser gegenüber der Bank durch Passwörter (PIN und TAN) aus, weil man das SSL-Verfahren nicht jedem Benutzer zumuten kann. Ein Fall von Client-Authentifizierung gegenüber einem Server liegt vor, wenn ein POP- oder IMAP-Server Mails nur an Clients ausliefert, die sich durch vorher auf einem sicheren Weg übertragene Zertifikate ausweisen. Die Datenverschlüsselung verhindert, dass ein Unbefugter die Daten abhört und erfolgreich auswertet. Unter Datenintegrität versteht man, dass die beim Empfänger eingetroffenen Daten nicht beschädigt oder verfälscht worden sind.

SSL oder TLS besteht selbst aus zwei Protokollen. Das TLS Record Protocol sorgt dafür, dass eine Verbindung vertraulich (privat) ist – mittels symmetrischer Verschlüsselung – und dafür, dass sie zuverlässig ist – mittels einer Art Prüfsumme (Message Authentication Check, MAC). Das TLS Handshake Protocol ermöglicht Client und Server, einander zu authentifizieren und eine symmetrische Verschlüsselung für die Sitzung auszuhandeln. Die Echtheit der von den Beteiligten übersandten öffentlichen Schlüssel verbürgt ein von einer dritten Stelle (Certification Authority, CA) signiertes Zertifikat, das der Empfänger prüfen kann. Da auch ein betrügerischer Rechner ein Zertifikat unterschreiben kann, sind die CAs ihrerseits wieder durch übergeordnete CAs zertifiziert, bis die Kette bei einer Root-CA endet. Sieht man sich mit:

```
joe@debian:~$ openssl verify apache.pem
```

das Zertifikat unseres intrafamiliären Webservers an, so erfährt man:

```
/C=DE/ST=Baden/L=Weingarten/O=Familie Alex/OU=./CN=www
/emailAddress=webmaster@www
error 18 at 0 depth lookup:self signed certificate
OK
```

und wüsste damit zuverlässig, wem der Webserver gehört, wenn das Zertifikat von einer CA signiert (unterschrieben) wäre. Dass sich ein Unbefugter das Zertifikat kopiert und unter falschem Namen verteilt, wird durch eine unsymmetrische Verschlüsselung unterbunden. Zertifikate sind nur zeitlich begrenzt gültig.

Eine Alternative zu OpenSSL ist die GNU Transport Layer Security Library (<http://www.gnu.org/software/gnutls/>, mit einer ausführlichen Dokumentation). Das Projekt ist noch nicht so weit gediehen wie OpenSSL. Unter *sarge* findet sich außer Paketen für Programmierer ein Paket für den Mail Transfer Agent *Postfix* sowie ein Paket für den Verzeichnisdienst *Ultrapossum*.

3.8 Wireless Local Area Network (WLAN)

Ein drahtloses lokales Netz (Funknetz, Wireless Local Area Network, WLAN) ist bequem und in manchen Situationen (Sportplatz, Hörsaal) die einzige vernünftige Lösung, um mehrere Rechner zu vernetzen. Grundsätzlich ist aber ein mit Kupferdraht oder Lichtwellenleiter verkabeltes Netz schneller, zuverlässiger und, wenn man vom Verlegen der Kabel absieht, billiger. Die heutige WLAN-Technik überbrückt je nach

den Räumlichkeiten zehn bis hundert Meter, mit erhöhtem Aufwand bei den Antennen auch wenige Kilometer. Die Bluetooth-Technik dagegen ist nicht zum Aufbau eines Netzes gedacht, sondern zum Anschluss von Peripheriegeräten an den Rechner auf einer Fläche von wenigen Metern Durchmesser.

Die Einrichtung eines WLANs ist Aufgabe des Verwalters. Unter Linux ist unbedingt die Treiberfrage zu beachten; es gibt nicht für alle WLAN-Karten bzw. -Chips Treiber bei Debian oder in seiner Nachbarschaft. Es kommt sogar vor, dass unter derselben Typenbezeichnung WLAN-Karten mit unterschiedlichen Chips gehandelt werden, die unterschiedliche Treiber erfordern. Im Folgenden geht es um Werkzeuge, mit denen ein Benutzer seine Verbindung prüfen kann. Aus dem Debian-Paket *wireless-tools* stammen einige Kommandos:

- `iwconfig` (analog zu `ifconfig` für drahtgebundene Netze) zeigt ohne Argument aufgerufen die Konfiguration der Netzkarte und den Zustand der Verbindung an,
- `iwlist scan` listet die erreichbaren Access Points und Ad-hoc-Zellen samt weitergehenden Informationen auf,
- `iwlist rate` listet die unterstützten Bitraten auf,
- `iwlist freq` zeigt die unterstützten Frequenzen und Kanalnummern an.

Die Kommandos liegen im Verzeichnis `/sbin` und damit nicht im Kommandopfad gewöhnlicher Benutzer, können aber von diesen mit ihrem absoluten Pfad aufgerufen werden. Bei Sourceforge wird ein grafisches KDE-Frontend `kwlan` für die Kommandos bereit gehalten, das sich leider wegen angeblich fehlender, in Wirklichkeit jedoch im üblichen Pfad vorhandener Bibliotheken nicht übersetzen ließ.

Praktisch ist auch das Kommando `wavemon`. Es zeigt auf mehreren Textbildschirmen (`curses`) Informationen zur Verbindung an, die im Sekundenrhythmus aufgefrischt werden (konfigurierbar in `/etc/wavemonrc` bzw. `$HOME/.wavemonrc`). Das Werkzeug eignet sich zum Herausfinden des besten Standortes für eine Antenne. Ähnliches leistet das KDE-Werkzeug `kwavecontrol`.

Suchen und Finden

Dieses Kapitel stellt die Werkzeuge und Strategien vor, mit denen lokal auf dem Rechner oder im Internet nach vielerlei Informationen erfolgreich gesucht werden kann.

4.1 Grundbegriffe

4.1.1 Zeichenketten (Strings)

Der Mensch verbringt eine beträchtliche Zeit seines Erdenwandels mit Suchen. Er sucht nach dem verlegten Haustürschlüssel, nach einer Telefonnummer, nach dem Leck im Geldbeutel, nach einer Straße im Gewirr einer Großstadt, nach einer Parklücke, nach einem Laserdrucker zu einem günstigen Preis, nach Informationen über THOMAS MANN, nach der verlorenen Zeit oder nach dem richtigen Partner fürs Leben. Bei einem bescheidenen Teil dieser Suchen hilft ein Rechner, möglichst unter Debian GNU/Linux.

Ein einfacher Fall ist die Suche nach einer Zeichenkette (E: string, F: chaîne de caractères). Eine Zeichenkette ist eine endliche Folge von Zeichen aus einem Alphabet, bei der die Reihenfolge wesentlich ist. Als Alphabet kommt vor allem der US-ASCII-Zeichensatz in Betracht. Meist schränkt man die Zeichenmenge auf sicht- oder druckbare Zeichen einschließlich der Whitespace-Zeichen ein. Whitespace-Zeichen sind Zwischenraum (space), Tabulator und Zeilenwechsel. Sie bewirken eine Bewegung des Cursors oder Druckkopfs, ohne ein sichtbares (schwarzes) Zeichen zu erzeugen. Große und kleine Buchstaben sind für Linux/UNIX unterschiedliche Zeichen, wie im richtigen Leben. Gewöhnlicher Text besteht aus Zeichenketten.

In den Programmiersprachen werden Zeichenketten unterschiedlich dargestellt. Da Linux/UNIX in C geschrieben ist, werden Zeichenketten – beispielsweise Dateinamen – unter diesem Betriebssystem mit dem nicht sichtbaren ASCII-Zeichen Nr. 0 (Nullbyte) abgeschlossen. Das ist auch der Grund dafür, dass ein Nullbyte nicht in Namen vorkommen darf; der Name wäre mit dem Nullbyte beendet. Der Benutzer braucht sich jedoch nur in seltenen Fällen um den richtigen Abschluss zu kümmern.

Ebenfalls aus der Programmiersprache C stammt eine Schreibweise für einige Zeichen, die anders nur schwierig einzugeben wären und die von einigen Kommandos akzeptiert werden (`man bash`):

- `\a` (alarm, bell) Glocke,
- `\b` (backspace) Rückwärtsschritt,
- `\e` (escape) Escape, Fluchtsymbol,
- `\f` (form feed) Seitenvorschub.
- `\n` (newline) Zeilenwechsel¹,
- `\r` (return) Return, Enter, Amen, Wagenrücklauf,
- `\t` (horizontal tab) Tabulator,
- `\v` (vertical tab) vertikaler Tabulator,
- `\\` (backslash) Gegen- oder Rückschrägstrich,
- `\nnn` ASCII-Zeichen mit der oktalen Nummer `nnn`,
- `\xnnn` ASCII-Zeichen mit der hexadezimalen Nummer `nnn`.

Die Schreibweise gilt nicht universell, im Zweifelsfall muss man im Manual nachlesen oder experimentieren. Einige Programme – darunter `make` und `calendar` – verlangen an bestimmten Stellen ein Tabulatorzeichen, das nicht durch Zwischenräume ersetzt werden darf. In manchen Zusammenhängen ist zwischen der leeren Zeichenkette – die außer dem Nullbyte kein Zeichen enthält – und *nichts* zu unterscheiden, beispielsweise bei der Definition von Umgebungsvariablen. Ferner muss man aufpassen, ob eine Ziffernfolge als Zeichenkette oder als Zahl gemeint ist. Beim Sortieren von Zeichenketten kommt 9 nach 10, beim Sortieren von Zahlen folgt 10 auf 9. Bei der Eingabe von Suchbegriffen in Suchfelder ist darauf zu achten, ob:

- Groß- und Kleinschreibung unterschieden wird,
- eine Zeichenkette als Wort oder als beliebige Teilzeichenkette (E: substring) verstanden wird,
- zwei nebeneinanderstehende Zeichenketten per Default verundet oder verodert werden.

Geben Sie als Suchbegriff *Else* ein, wollen Sie unter Umständen Begriffe wie *Gelsenkirchen* oder *Handelsembargo* nicht sehen. Oft sind verschiedene Beugungsformen eines Wortes mit Hilfe von Jokerzeichen oder regulären Ausdrücken zu berücksichtigen. Synonyma wie *Fernglas*, *Fernrohr* und *Feldstecher* oder Schreibweisen wie *Äthanol* und *Ethanol* sind bei Suchen ebenfalls zu bedenken.

4.1.2 Reguläre Ausdrücke

Reguläre Ausdrücke (RE: E: regular expression, F: expression régulière) sind ein mächtiges Werkzeug zur Bearbeitung von Textdateien. Es handelt sich dabei – ver-

¹ Genau genommen meint Zeilenwechsel das ASCII-Zeichen Nr. 10 (Line Feed, LF). Unter Newline wird in der Linux/UNIX-Welt das Umschalten an den Anfang einer neuen Zeile verstanden, unabhängig davon, durch welche ASCII-Zeichen es im Text dargestellt wird. Siehe Abschnitt 5.1.3 *Textdateien aus anderen Welten (DOS, Mac)* auf Seite 164.

einfacht gesagt – um eine Erweiterung herkömmlicher Suchfunktionen von Textwerkzeugen. Wichtige Programme, die mit regulären Ausdrücken umgehen, sind Texteditoren wie `vi` oder `Emacs`, die Programminterpretierer `perl` oder `awk`, der Streameditor `sed` oder das Suchprogramm `grep`. Die Bezeichnung stammt aus der Theorie formaler Sprachen (Programmiersprachen). Für uns reicht es zu wissen, dass reguläre Ausdrücke Zeichenfolgen sind, die nach einem System von Regeln ausgewertet werden. Gegebene Zeichenketten lassen sich auf Übereinstimmung mit einem regulären Ausdruck prüfen. Die Manualseite erreichen Sie mit:

```
joe@debian:~$ man 7 regex
```

Durch den POSIX-Standard haben sich leichte Änderungen gegenüber den alten regulären Ausdrücken ergeben, wie sie der UNIX-Zeileneditor `ed` verwendete.

Achtung: Reguläre Ausdrücke sind von Dateinamen mit Metazeichen zu unterscheiden, die von der Bash und einigen anderen Programmen verstanden werden. Beide haben zwar einen ähnlichen Zweck, aber eine verschiedene Syntax.

Die einfachste Form eines regulären Ausdrucks ist eine gewöhnliche Zeichenkette ohne tiefere Bedeutung. Um die Systemprotokolldatei `/var/log/messages` nach Zeilen zu durchsuchen, welche die Zeichenkette `eth0` enthalten, wird das Programm `grep` wie folgt benutzt:

```
debian:~# grep eth0 /var/log/messages
```

Es werden alle Zeilen ausgegeben, die irgendwo den regulären Ausdruck `eth0` enthalten. Wenn Sie alle Zeilen sehen möchten, in denen `eth0` oder `eth1` vorkommt, gebrauchen sie einen regulären Ausdruck mit Metazeichen (Sonderzeichen, Platzhaltern). Beispielsweise wird in eckigen Klammern angegeben, dass eins aus mehreren Zeichen als zutreffend gelten soll. Um die Datei `/var/log/messages` nach `eth0` und `eth1` zu durchsuchen, wird folgender Befehl benutzt:

```
joe@debian:~$ grep "eth[01]" /var/log/messages
```

Die Anführungszeichen sind notwendig, damit die eckigen Klammern nicht von der Shell interpretiert werden. Die wichtigsten Sonderzeichen in regulären Ausdrücken sind:

- Ein Punkt steht für genau ein beliebiges Zeichen. Der reguläre Ausdruck `a . a` trifft sowohl auf die Zeichenkette `aba` als auch auf `aca` zu. Er trifft nicht auf die Zeichenkette `aa` zu, weil in dieser Zeichenkette das dritte Zeichen fehlt.
- * Ein Stern nach einem regulärem Ausdruck bedeutet ein beliebig häufiges Vorkommen – auch nullmalig – des voranstehenden regulären Ausdrucks. Der reguläre Ausdruck `stop*` trifft auf die Zeichenketten `sto`, `stop` und `stopp`, nicht aber auf die Zeichenkette `stau` zu, weil hier das dritte Zeichen kein `o` ist.

Durch die Kombination von Punkt und Stern lassen sich beliebig lange Zeichenketten, die aus jedem Zeichen bestehen dürfen, spezifizieren. Der reguläre Ausdruck `. *` (Punkt Stern) trifft auf alle Zeichenketten zu (ein beliebiges Zeichen darf beliebig oft vorkommen).

- + Das Plus-Zeichen bedeutet, dass der voranstehende reguläre Ausdruck mindestens einmal vorkommen muss. Der Ausdruck `Schif+` trifft deswegen nicht auf die Zeichenkette `Schi` zu (weil das Zeichen `f` nicht vorkommt), wohl aber auf `Schiff` oder `Schiffahrt`. Das Pluszeichen wird nicht von allen Programmen, die mit regulären Ausdrücken umgehen, als Sonderzeichen interpretiert.
- [] Durch eckige Klammern eingeschlossene Zeichen bedeuten ein Zeichen aus der eingeklammerten Zeichenmenge. Beispielsweise trifft der Ausdruck `st[a o] . *` sowohl auf `stop` als auch auf `stau` zu, nicht aber auf `stumm`, weil das dritte Zeichen entweder ein `a` oder ein `o` sein muss.
Die eingeklammerte Zeichenmenge kann als Bereich angegeben werden. Der Ausdruck `[0-9]` umfasst alle Ziffern und der Ausdruck `[a-z]` alle Kleinbuchstaben (ohne Umlaute). Auch hier ist die Kombination mit anderen Sonderzeichen möglich. So trifft der Ausdruck `[a-z] *` auf solche Zeichenketten zu, in denen beliebig viele Kleinbuchstaben oder gar keine Zeichen vorkommen.
Wird den Zeichen in eckigen Klammern ein Zirkumflex (Dach, caret) vorangestellt, dann ist gemeint, dass der entsprechende Ausdruck nur auf solche Zeichen zutrifft, die nicht in den eckigen Klammern enthalten sind (Komplement). Der Ausdruck `[^0-9]` trifft auf alle Zeichen zu, die *keine* Ziffern sind.
- ^ Um den Anfang einer Zeile anzugeben, ist ebenfalls das Zeichen `^` zu verwenden. Der Ausdruck `^Linux` trifft für alle Zeilen zu, in denen das Wort `Linux` am Anfang der Zeile steht.
- \$ Ebenso kann das Ende einer Zeile spezifiziert werden. Hierzu ist das Dollarzeichen zu benutzen. Der Ausdruck `2005$` trifft auf solche Zeilen zu, in denen die Zahl `2005` am Ende der Zeile steht. Man sagt, dass Zirkumflex oder Dollarzeichen das Muster (E: pattern, F: forme) am Zeilenanfang oder -ende verankern.
Durch die Kombination von Zirkumflex und Dollarzeichen lässt sich der Inhalt ganzer Zeilen angeben. Beispiel: `^Heute ist Sonntag$` trifft ausschließlich auf Zeilen zu, die nur die Zeichenkette `Heute ist Sonntag` beinhalten, und der Ausdruck `^Heute ist . *tag$` trifft auf die Zeilen zu, die mit der Zeichenkette `Heute ist` beginnen und mit den Zeichen `tag` enden, also auch auf `Heute ist Dienstag` oder `Heute ist ein besonders warmer Donnerstag`, nicht aber auf `Heute ist ein schöner Tag`, weil die letzte Zeile nicht mit den Zeichen `tag`, sondern mit `Tag` endet.
- \ Gelegentlich müssen Zeichen mit besonderer Bedeutung wie normale Zeichen (buchstäblich, als literale Zeichen) eingesetzt werden. Wir kennen das unter dem Begriff *Quoten* von den Metazeichen der Shell. Ein vorangestellter Gegenschrägstrich nimmt den Sonderzeichen ihre besondere Bedeutung. Der Punkt in `Danke\ .` hat keine besondere Bedeutung mehr, der Ausdruck steht für die Zeichenkette `Danke .` (mit Punkt). Soll das Zeichen `\` als nor-

males Zeichen genutzt werden, ist diesem ebenfalls ein Gegenschrägstrich voranzustellen (\\).

Wenn Sie in der Kommandozeile mit regulären Ausdrücken arbeiten, sollten Sie wegen der Ähnlichkeit mit den Metazeichen der Shell die regulären Ausdrücke mit einfachen Anführungszeichen (single quotes) vor jeder Interpretation durch die Shell schützen.

Nehmen Sie irgendeine Textdatei und lassen Sie `grep` mit verschiedenen regulären Ausdrücken darauf los:

```
joe@debian:~$ grep 'aber' text

joe@debian:~$ grep 'ab.a' text

joe@debian:~$ grep 'bb.[aeiou]' text

joe@debian:~$ grep '^ [0-9]+$' text

joe@debian:~$ grep 'M[ae][iy]e?r' text
```

`grep` gibt die Zeilen aus, in denen sich wenigstens eine Zeichenkette befindet, auf die der reguläre Ausdruck passt. Im ersten Beispiel sind das alle Zeilen, die die Zeichenkette `aber` enthalten wie `aber`, `labern`, `Schabernack`, `aberkennen`, im zweiten trifft unter anderem `abwarten` zu, im dritten Abbruch. Die vierte Form ermittelt alle Zeilen, die nur Ziffern enthalten: Am Anfang der Zeile (Zirkumflex) ein Zeichen aus der Menge 0 bis 9, dann eine beliebige Wiederholung von Ziffern bis zum Ende (Dollar) der Zeile. In der fünften Zeile wird nach dem Namen `Meier` mit all seinen Varianten geforscht.

Wie sieht ein regulärer Ausdruck aus, der auf Namen wie `Linux`, `Unix`, `UNIX`, `HP-UX`, `Sinix`, `AIX`, `Ultrix`, `Xenix` oder `Minix` zutrifft? In Verbindung mit `grep` lautet der Ausdruck:

```
joe@debian:~$ grep '[A-Z][A-Za-z]*[Xx]' text
```

Ein Großbuchstabe, eine beliebige Anzahl, auch null, von Buchstaben, gefolgt von einem großen oder kleinen `X`. Bei `HP-UX` fällt der zweite Teil unter den regulären Ausdruck, nicht der ganze Name. `Solaris` oder `FreeBSD` bleiben außen vor. Um auch diese Namen zu erfassen, müsste man drei reguläre Ausdrücke mit *oder* verknüpfen. Die `BSD`-Familie erwischt man mit `[A-Z][A-Za-z]*BSD`, `Solaris` folgt keinem Muster und ist als konstante Zeichenkette einzugeben.

Wir wollen nun einen regulären Ausdruck zusammenstellen, der auf alle gültigen Internet-Email-Anschriften zutrifft. Dazu schauen wir uns einige erfundene oder obsoleete Anschriften an:

```
wulf.alex@mvm.uni-karlsruhe.de
wuaalex1@mvmc64.ciw.uni-karlsruhe.de
ig21@rz.uni-karlsruhe.de
012345678-0001@t-online.de
Dr_Ralf.Muus@FOOBAR.de
```

Links steht immer ein Benutzername, dessen Form vom jeweiligen Betriebssystem (Eintrag in `/etc/passwd` oder `/etc/aliases`) bestimmt wird, dann folgen das @-Zeichen (Klammeraffe) und ein Maschinen- oder Domänenname, dessen Teile durch Punkte voneinander getrennt sind. Im einzelnen:

- Anfangs ein Zeichen aus der Menge der Ziffern oder kleinen oder großen Buchstaben,
- dann eine beliebige Anzahl einschließlich null von Zeichen aus der Menge der Ziffern, der kleinen und großen Buchstaben und der Zeichen Bindestrich, Unterstrich und Punkt,
- genau ein Klammeraffe als Trennzeichen,
- im Maschinen- oder Domänennamen mindestens eine Ziffer oder ein Buchstabe,
- dann eine beliebige Anzahl von Ziffern, Buchstaben oder Strichen,
- mindestens ein Punkt zur Trennung von Domäne und Top-Level-Domäne,
- nochmals mindestens zwei Buchstaben zur Kennzeichnung der Top-Level-Domäne.

Daraus ergibt sich folgender regulärer Ausdruck (einzeilig):

```
^[0-9a-zA-Z][0-9a-zA-Z_-.]*@[0-9a-zA-Z][0-9a-zA-Z_-.]*
\.[a-zA-Z][a-zA-Z]+
```

Das sieht kompliziert aus, ist aber der einfachste Weg zur Beschreibung solcher Gebilde. Obige Form einer Email-Anschrift ist nicht gegen die Internet-RFCs abgeprüft und daher vermutlich zu eng. Eine Anwendung für den regulären Ausdruck könnte ein Programm sein, das Email-Anschriften verarbeitet und sicherstellen will, dass die ihm übergebenen Zeichenketten wenigstens ihrer Form nach gültig sind. Robuste Programme überprüfen Eingaben, ehe sie sich weiter damit abgeben.

Auf der Manualseite zu `grep` widmet sich ein Abschnitt den regulären Ausdrücken. Vermutlich ist die Aussage betreffend das Pluszeichen falsch, für `grep` scheint es kein Metazeichen zu sein. Nähere Informationen finden sich in der Info-Dokumentation zum Programm `awk` (GNU Version, Debian-Paket *gawk* aus der Abteilung *Interpreter*), in der Manualseite zum regulären Ausdruck nach dem POSIX-Standard (Manualseite `regex`) sowie in der Manualseite zu dem Editor `ed`.

4.1.3 Logische Verknüpfungen

Eine Aussage ist entweder wahr (richtig, E: true) oder falsch (E: false); eine dritte Möglichkeit wollen wir ausschließen, obwohl viele Aussagen im wahren Leben weder ganz richtig noch ganz falsch sind. Eine Negation (Umkehrung, Verneinung) macht aus einer wahren Aussage eine falsche und umgekehrt. In der Linux/UNIX-Welt wird – der Schreibweise der Programmiersprache C folgend – als Operator der Negation oft ein Ausrufezeichen verwendet. In anderen Zusammenhängen kommt das Minuszeichen oder das Wort `not` vor. Die Negation ist die einzige Operation, die auf eine einzelne Aussage angewendet wird (unäre Operation).

Zwei Aussagen, die jede für sich wahr oder falsch sein können, lassen sich auf mehrere Weisen zu einer neuen Aussage kombinieren:

- zwei mit `or` (oder) verknüpfte Aussagen liefern dann das Ergebnis wahr, wenn mindestens eine der beiden Aussagen wahr ist,
- zwei mit `and` (und) verknüpfte Aussagen liefern nur dann das Ergebnis wahr, wenn beide Aussagen wahr sind,
- zwei mit `xor` (entweder - oder) verknüpfte Aussagen liefern nur dann das Ergebnis wahr, wenn genau eine (aber nicht beide zugleich) der Aussagen wahr ist.

Ferner lassen sich logische Ausdrücke genau wie Zahlen beim Rechnen klammern. Kennt man die Vorrangregeln der logischen Operatoren nicht, ist es sicherer zu klammern; überflüssige Klammerpaare stören nicht.

Von den vorgenannten ausdrucksweisen Verknüpfungen sind die bitweisen Verknüpfungen zu unterscheiden, die beispielsweise beim Maskieren einer IP-Adresse mit einer Netzmaske eine Rolle spielen. Hierbei wird jedes Bit der Adresse mit dem entsprechenden Bit der Netzmaske durch *und* verknüpft. Das Ergebnis ist wieder eine IP-Adresse, nicht wie oben *wahr* oder *falsch*.

In vielen Zusammenhängen sind Jokerzeichen in logischen Ausdrücken erlaubt, ähnlich wie bei Dateinamen, die einer Shell übergeben werden, aber da muss man sich genau informieren. Der Stern ist nicht überall ein universeller Joker, und manche Suchwerkzeuge vertragen keinen Joker am Anfang einer Zeichenkette. Manchmal sucht man nach Begriffen, die aus mehreren Teilen (Wörtern) bestehen. Die Teilbegriffe sollen entweder unmittelbar beieinander stehen, dann sind sie meist in Apostrophe oder Anführungszeichen einzuschließen, oder sie sollen nahe beieinander vorkommen. Dafür kennen einige Suchprogramme so genannte Nachbarschafts-Operatoren (E: proximity, F: proximité). Schließlich verstehen sich einige Datenbanken auch auf die Suche nach Begriffen, die nach den phonetischen Regeln einer menschlichen Sprache (meist Englisch) ähnlich klingen (Soundex- oder Metaphone-Funktion). Damit erspart man sich das Jonglieren mit regulären Ausdrücken. Eine derartige Funktion wird auch von Rechtschreibungsprüfern wie `aspell` beim Suchen nach Verbesserungsvorschlägen und von Suchmaschinen im Netz angewendet.

Schließlich gibt es dankbare und undankbare Suchbegriffe. Unangenehm sind Begriffe mit vielen Bedeutungen, Schreibweisen oder Synonyma. Wer nach THOMAS MANN sucht, braucht mit *Mann* gar nicht erst anzufangen. Auch die Suche nach `LATEX` bringt viele unerwartete Ergebnisse. Um gezielt voranzukommen, sind derartige Begriffe wenigstens mit einem Gattungsbegriff zu verunden. Die Negation und die Oder-Verknüpfung werden bei Suchen seltener gebraucht, das exklusive Oder so gut wie nie.

4.1.4 Indizierung

Wenn vorhersehbar ist, dass in einem Datenbestand nach bestimmten Merkmalen gesucht wird, kann man einen Teil der Sucharbeit vorverlegen und so die eigentliche Suche abkürzen. Dazu legt man einen sortierten Index der Suchmerkmale an, der zu jedem Wert die Fundstellen enthält. Der Vorgang wird Indizierung oder Indexierung genannt. Die Register am Ende des Buches sind ein Beispiel. Wenn Sie nach dem

Schlagwort *Flugsimulator* suchen, schlagen Sie im alphabetisch sortierten Sachregister nach und erfahren sofort die zugehörigen Seitenzahlen. Das geht schneller, als wenn Sie vorn im Buch zu lesen anfangen und vielleicht nach einigen Stunden auf das Wort stoßen. Da das Sachregister sortiert ist und Sie wissen, wo ungefähr der Buchstabe F liegt, lesen Sie auch das Register nicht von vorn der Reihe nach (sequentiell) durch, sondern starten die Suche nach dem Begriff einige Seiten später. Auch das Inhaltsverzeichnis ist ein Index, nämlich ein Index der Überschriften, sortiert nach Seitenzahlen. Das Merkmal, nach dem sortiert wird, heißt Sortierschlüssel. In manchen Zusammenhängen muss ein Schlüssel eindeutig oder sogar umkehrbar eindeutig sein (Datei - Inode-Nummer - Dateiname). Das Werkzeug `locate` findet deshalb so schnell einen Dateinamen, weil vorher mittels `updatedb` etwas Ähnliches wie ein Index angelegt worden ist, während `find` jedesmal erneut im Dateisystem sucht. Ein weiteres Beispiel ist Glimpse, siehe Abschnitt 4.2.2 *Volltextsuche* auf Seite 123.

Zu einem Datenbestand können mehrere Indizes gehören. Zum einen lässt sich ein Index nach verschiedenen Kriterien sortieren, eine Telefonliste beispielsweise nach Namen oder Nummern. Für eine Büchersammlung kann ein Index der Autoren und ein zweiter der Titel angelegt werden. Ein Vorteil rechnerunterstützter Indizes ist, dass sie im Nullkommanichts durchsucht oder umsortiert werden können, im Gegensatz zu den Karteikästen früherer Jahrhunderte. Das Anlegen eines Index erfordert jedoch meist noch die Mitarbeit höherer Lebensformen. Das Erzeugen eines Sachregisters für ein Buch beginnt damit, dass der Autor Schlag- oder Stichwörter zur Indizierung vormerkt. Die Auswahl kann ihm keine Software abnehmen (wenn das Sachregister etwas taugen soll).

Suchen und Sortieren sind zwei interessante Aufgaben für einen Rechner. Jedermann versteht sie, es gibt mehrere Lösungswege (Algorithmen), aber keinen besten für alle Fälle, und einige Lösungen sind ziemlich ausgefuchst. Den Klassiker zu dem Thema hat DONALD E. KNUTH geschrieben.

4.2 Suchen nach lokalen Informationen

4.2.1 Suchen nach Zeichenketten in Dateien (Editoren, `grep`)

Sucht man in einer Textdatei nach einer Zeichenkette, so kann man die Suchfunktion eines Editors verwenden, siehe Abschnitt 5.3 *Der Editor vi* auf Seite 174 oder Abschnitt 5.4 *Universalgenie: Emacs* auf Seite 188. Die Editoren suchen nach regulären Ausdrücken.

Erstreckt sich die Suche über mehrere Dateien eines Verzeichnisses, ist das Werkzeug `grep` geeignet. Der Aufruf ist einfach:

```
joe@debian:~$ grep zeichenkette dateinamen
```

Der Dateiname darf Shell-Jokerzeichen enthalten. Die Shell interpretiert die Jokerzeichen und übergibt `grep` eine Liste mit zutreffenden Dateinamen. Die Zeichenkette darf – mit leichten Einschränkungen – ein regulärer Ausdruck sein, siehe Abschnitt 4.1.2 *Reguläre Ausdrücke* auf Seite 116. Beispielsweise sucht:


```
joe@debian:~$ grep 'M[ae] [iy]e?r' *.tex
```

im Arbeitsverzeichnis in allen Dateien mit der Kennung `.tex` (L^AT_EX-Dateien) nach Zeilen, in denen der Name *Maier* in all seinen Varianten vorkommt, also Maier, Meier, Mayer, Maier, Meir, Mayr etc. Das Suchmuster steht in Hochkommas (single quotes), um die Shell an einer Interpretation zu hindern. Hingegen soll der Dateiname von der Shell expandiert werden. Die wichtigsten Optionen von `grep` sind:

- `-i` (ignore case) unterscheide nicht zwischen Groß- und Kleinschreibung,
- `-v` (invert match) suche nach Zeilen, die das Muster *nicht* enthalten,
- `-r` (recursive) suche rekursiv auch in Unterverzeichnissen.

Die letzte Option beherrschen erst neuere Versionen von `grep`; früher musste man sich aus `find` und `grep` ein Kommando zusammenbasteln. GNU `grep` macht keinen Unterschied zwischen gewöhnlichen und erweiterten regulären Ausdrücken. `grep` kann auch in binären Dateien suchen, man läuft aber Gefahr, dass die Ausgabe den Bildschirm oder Drucker verkonfiguriert. Leider werten die Programme, die mit regulären Ausdrücken arbeiten (`grep`, `sed`, `vi`, `perl` ...), diese nicht nach einheitlichen Regeln aus. Bei fortgeschrittenen Aufgaben muss man genau nachlesen.

Das Kommando `grep` hat einige Verwandte, die zum Teil nur Shellskripte um `grep` herum sind:

- `agrep` sucht nach nur näherungsweise angegebenen Mustern (approximate patterns).
- `egrep` ist identisch mit `grep -E` und versteht erweiterte reguläre Ausdrücke, was für GNU-`grep` keine Rolle spielt,
- `fgrep` ist identisch mit `grep -F` und sucht nach festen Zeichenketten, auch wenn sie durch einen Zeilenwechsel getrennt sind. Außerdem war es früher schneller als das allgemeine Kommando.
- `sgrep` sucht in strukturiertem Text (HTML, SGML, T_EX, Programmquellen).
- `zgrep` sucht in mittels `compress` oder `gzip` verdichteten Textdateien. Man spart sich das Verdünnen. Ebenso `bzgrep`.

Sucht man in beliebigen (binären) Dateien – meist compilierten Programmen – nach druckbaren (lesbaren) Zeichenketten, hilft das nicht übermäßig intelligente Werkzeug `strings`.

4.2.2 Volltextsuche (lookup, glimpse)

Das Werkzeug `lookup` dient der schnellen, interaktiven Suche in Textdateien, auch in japanischen. Die Manualseite ist nicht umfassend, eine Information im Netz war nicht aufzutreiben. Schade, denn das Universalwerkzeug `grep` ist sicher nicht für alle Zwecke optimal.

Glimpse (Global Implicit Search) ist ein größerer Hammer und dient dem Indizieren und schnellen Durchsuchen umfangreicher Dateisysteme. Das oben genannte Werkzeug `agrep` stammt aus dem Glimpse-Projekt und wird hier mitbenutzt. Eine

Variante zum Durchsuchen und Indizieren von Websites ist WebGlimpse. Die Startseite für beide ist <http://webglimpse.net/>. Die Manualseite ist ausführlich und hilfreich.

Im ersten Schritt ist der in Frage stehende Verzeichniszweig zu indizieren. Das Werkzeug `glimpseindex` kennt drei Stufen des Indizierens:

- Ohne Option erzeugt es einen winzigen (tiny) Index, dessen Größe 2 bis 3 % des Umfangs der Textdateien beträgt,
- mit der Option `-o` wird ein kleiner (small) Index angelegt, dessen Umfang sich auf knapp 10 % beläuft,
- mit der Option `-b` (medium) wächst der Index auf ungefähr 25 %.

Ein größerer Index beschleunigt die Suche. Der Aufruf:

```
joe@debian:~$ glimpseindex zweig
```

durchkämmt rekursiv das Verzeichnis `zweig` und legt einen aus mehreren Dateien mit dem Präfix `.glimpse` bestehenden Index an. Dabei versucht es, Nicht-Text-Dateien zu übergehen; man kann aber auch Dateien oder Verzeichnisse ausdrücklich von der Indizierung ausschließen. Die Dateien werden mit ihrem absoluten Pfad erfasst, was im Netz (NFS oder Samba) Probleme bereiten kann. Ein bereits vorhandener Glimpse-Index lässt sich aktualisieren. Das Indizieren dauert einige Minuten, je nach Hardware und Datenmenge. Arbeitet man viel mit Glimpse, empfiehlt es sich, `glimpseindex` jeden Morgen als Cronjob zu starten.

Ist der Index erst einmal fertig, geht die Suche blitzschnell:

```
joe@debian:~$ glimpse Karlsruhe
```

lieferte in wenigen Sekunden 836 Treffer aus dem Verzeichnis der Dateien zu diesem Buch und anderen. Ist der Suchbegriff zu wenig spezifisch und liefert viele Treffer, warnt Glimpse vorher, so in diesem Fall.

4.2.3 Suchen in lokalen Wörterbüchern (ding, dict)

Das Werkzeug `ding` (Dictionary Nice Grep) setzt auf `agrep` auf und sucht in Wörterbüchern. Per Default verwendet es das deutsch-englische Wörterbuch aus dem Debian-Paket *trans-de-en*, aber das Werkzeug ist nicht darauf beschränkt. Das Format des Wörterbuchs ist offen, sodass es keine Schwierigkeiten bereitet, eigene Wörterbücher zu erstellen oder andere Wortlisten anzupassen. Das Werkzeug wird entweder in der Kommandozeile mit dem gesuchten Wort als Argument aufgerufen:

```
joe@debian:~$ ding fuzzy
```

oder ohne Argument. Dann erscheint ein Fenster mit Eingabefeld. Die Konfiguration steht im Home-Verzeichnis unter `.dingrc`. Außer im Manual findet sich Information unter <http://www-user.tu-chemnitz.de/~fri/ding/>.

Das Paar `dict` (Client) und `dictd` (Server) ermöglicht den schnellen Zugriff auf ein- oder zweisprachige Wörterbücher lokal oder über ein Netz gemäß

RFC 2229. Unter diesen befindet sich auch das bekannte *Hacker's Dictionary*² (jargon file) von ERIC STEVEN RAYMOND (ESR). Die Wörterbücher sind von unterschiedlicher Qualität und ersetzen bei einer gründlicheren Beschäftigung mit einem Wort oder Begriff nicht die Suche in weiteren Werken, siehe Abschnitt 4.3.5 *Wörterbücher, Glossare und Lexika* auf Seite 137. Näheres ist bei <http://www.dict.org/> zu erfahren.

Die Konfiguration von `dict` steht systemweit in `/etc/dictd/dict.conf` und benutzerbezogen im Home-Verzeichnis in `.dictrc`. In der systemweiten Datei finden sich nach der Einrichtung die Server `localhost` und `dict.org`. Hier ist an erster Stelle der lokale Server einzutragen, also die Maschine im LAN, auf der `dictd` läuft, und zwar mit einem Namen, der etwas spezifischer ist als nur `server`. Der Default-Port 2628 braucht nicht genannt zu werden.

Auf dem Server sind außer dem Dämon die gewünschten Wörterbücher aus der Abteilung *Textverarbeitung* einzurichten, beispielsweise:

- `dict-jargon` The New Hacker's Dictionary,
- `dict-vera` Verzeichnis EDV-Relevanter Akronyme,
- `dict-foldoc` Free Online Dictionary Of Computing,
- `dict-freedit`-* zahlreiche zweisprachige Wörterbücher.

Die Wörterbücher liegen in `/usr/share/dictd/` und sind indiziert. Ihr Format wird auf der Manualseite zu `dictd` erläutert, sodass man eigene Wörterbücher abfassen kann. Ferner ist auf dem Server in der Datei `/etc/dictd/dictd.conf` der Zugang (access) für die Rechner im lokalen Netz zu gestatten; per Default ist nur der Zugriff von `localhost` vorgesehen. Dann kann ein Benutzer auf einem Client einen Suchbegriff eingeben:

```
joe@debian:~$ dict ascii
```

und erhält je nach verfügbaren Wörterbüchern eine umfassende Antwort, natürlich nicht nur in Computerfragen:

```
joe@debian:~$ dict "abatement of tax"
```

bedeutet *Steuernachlass*. Die Gegenrichtung funktionierte nicht. Grammatische Auskünfte kommen nur spärlich mit, und bei manchen Fragen ist das Wörterbuch im Regal überlegen:

```
joe@debian:~$ dict ratatouille
```

liefert vorläufig *No definitions found*. Aber das Konzept überzeugt.

4.2.4 Suchen nach binären Mustern in Dateien (gnuifit, imgseek)

Die Überschrift zielt auf die Suche nach bestimmten Bildmustern in Grafik-Dateien oder Klangmustern in Sound-Dateien ab. Sie suchen in einem Verzeichnis mit digitalisierten Fotos (jpg-Dateien) nach Bildern, auf denen der Eiffelturm zu sehen

²Zur Ergänzung suche man in der deutschen Wikipedia den Begriff *Hacker* und folge dem Link zur Kategorie *Hacker* am unteren Seitenrand.

ist oder das Matterhorn. Sie suchen in einem Verzeichnis mit Sound-Dateien (mp3- oder wav-Dateien) die Dateien, die die Europa-Hymne oder den Klang einer Ducati enthalten, und zwar ohne Zuhilfenahme von Menschen erstellter textueller Beschreibungen. Was für einen Menschen eine verständliche und lösbare Aufgabe darstellt, ist für Rechner äußerst schwierig und Gegenstand der Forschung. Der Benutzer darf keine so einfachen und schnellen Lösungen erwarten wie bei der Suche nach Zeichenketten.

Das GNU Image Finding Tool `gnuity-exgift-exviper` – indiziert und sucht Inhalte von Bilddateien. Es ist ein *Content Based Image Retrieval System* (CBIRS) und liegt samt dem Begleitpaket *gnuity-doc* in der Abteilung *Grafik*. Die Startseite des Projekts hat den URL <http://www.gnu.org/software/gift/gift.html>, eine Demo findet sich unter <http://viper.unige.ch/demo/>. Wie bei Glimpse ist zunächst ein automatisch erzeugter Index der Bildersammlung anzulegen, dann kann mit Beispielsbildern nach ähnlichen Bildern gesucht werden. Das Werkzeug beinhaltet keine Funktionen zum Verwalten eines Archivs.

Das Werkzeug `imgseek` hilft bei der Verwaltung umfangreicher Sammlungen von Bildern und ihren Metadaten. Es kann an Hand von Skizzen oder Beispielsbildern nach ähnlichen Bildern suchen und bietet die Treffer in einer Vorschau an. Der Aufruf:

```
joe@debian:~$ imgSeek
```

(mit großem S) startet ein Fenster, in dem man sich zu einem Bildverzeichnis durchklickt. Dort lassen sich HTML-Alben oder Diavorführungen zusammenstellen, Metadaten editieren und weitere Aufgaben zur Verwaltung eines Bildarchivs durchführen. Eine Suche nach ähnlichen Bildern zeitigte keinen Erfolg, obwohl im Verzeichnis dreimal das Schachbrett von Abbildung 13.1 enthalten war, nur zu verschiedenen Zeitpunkten im Spiel gespeichert. Vielleicht ist das Programm nicht so intuitiv zu benutzen, wie es scheint. Jedenfalls unterscheidet es sich in seiner Zielsetzung deutlich von `gnuity`.

4.2.5 Suchen nach Dateien und Verzeichnissen (`find`, `whereis`, `which`, `locate`)

Zum Suchen von Dateien oder Verzeichnissen mit bestimmten Eigenschaften dient das Werkzeug `find`. Es ist vielseitig und versteht reguläre Ausdrücke; entsprechend lang ist die Manualseite. Die `finds` verschiedener UNIXe unterscheiden sich in Einzelheiten, Vorsicht. `find` ist an die Zugriffsrechte des aufrufenden Benutzers gebunden, nur `root` darf den gesamten Dateibaum durchsuchen, im Fall von über NFS oder Samba gemounteten Zweigen auch das nur nach besonderer Konfiguration. Sehen wir uns einige Beispiele an:

```
joe@debian:~$ find -name 'vorwort.*' -print
```

Das Kommando sucht im Arbeitsverzeichnis und darunter (rekursiv) nach Dateien mit dem Namen `vorwort.*` und gibt die Treffer nach `stdout` (Bildschirm) aus. Der gesuchte Name `vorwort.*` steht in Hochkommas (single quotes), damit

er nicht von der Sitzungsshell, sondern von `find` ausgewertet wird. Der Stern bedeutet hier eine beliebige Anzahl beliebiger Zeichen. Die Aktion `-print` könnte weggelassen werden, da sie Default ist. Die Antwort besteht in dem vom Arbeitsverzeichnis ausgehenden Pfad von Dateien wie `vorwort.tex`, `vorwort.aux` oder `vorwort.pdf`.

```
joe@debian:~$ find $HOME -size +1000
```

Das Kommando sucht im Home-Verzeichnis und darunter alle Dateien, die größer als 1000 Blöcke (meist zu 512 Bytes) sind und schreibt deren relativen Pfad nach `stdout`. Die Angabe `1000k` würde 1000 kByte bedeuten.

```
joe@debian:~$ find /home -atime +50
```

Im dritten Beispiel wird im `/home`-Zweig rekursiv nach Dateien gesucht, deren Zugriffszeit (access time) 50 Tage überschreitet, die also mindestens 50 Tage lang nicht geöffnet worden sind. Die Suche bezieht sich auf Dateien, nicht auf Verzeichnisse.

Das allgemeine Muster des Aufrufs von `find` sieht folgendermaßen aus:

```
find pfade -optionen test aktionen
```

Der Pfadteil besteht aus einem oder mehreren Pfaden, absolut oder relativ zum Arbeitsverzeichnis. Das erste Argument, das mit einem Minuszeichen oder bestimmten anderen Zeichen beginnt, wird als Beginn der Optionen oder nachfolgender Ausdrücke gewertet. Die Option `-follow` veranlasst `find` beispielsweise, weichen (symbolischen) Links zu folgen. Der Test kann komplex sein, am Ende kommt immer `true` oder `false` heraus. In Abhängigkeit vom Testergebnis werden dann Aktionen ausgeführt oder nicht. Zwei Beispiele:

```
joe@debian:~$ find tmp temp misc -atime +32 -ok rm
'{}' \;
```

Das erste Beispiel ist wie folgt zu verstehen: Suche von Arbeitsverzeichnis ausgehend in den Verzeichnissen `tmp`, `temp` und `misc` die Dateien, auf die seit mehr als 32 Tagen nicht zugegriffen wurde, und lösche sie, hole dir aber vorher das OK vom Benutzer. Die Hochkommas und der Gegenschrägstrich schützen die Klammern und das Semikolon vor dem Zugriff der Shell.

```
joe@debian:~$ find -name '*.tex' | xargs grep -i hallo
```

Im zweiten Beispiel ermittelt `find` alle Dateien, deren Namen die Kennung `tex` (L^AT_EX-Dateien) trägt und übergibt ihre Namen per Pipe an das Kommando `xargs`. Dieses fügt sie als Argument hinter `hallo` an das Kommando `grep` an und ruft es auf. Man erfährt, in welchen Dateien in welcher Zeile die Zeichenkette `hallo` ohne Unterscheidung von Groß- und Kleinschreibung vorkommt. `xargs` ist ein Weg unter mehreren, die Argumentliste eines Kommandos aufzubauen. Der Unterschied zwischen den beiden Beispielen ist, dass `rm` jedesmal mit einem Dateinamen als Argument aufgerufen wird, während `grep` nur einmal, aber mit einer unter Umständen langen Argumentliste gestartet wird.

Nun noch ein `find` mit einem aus zwei durch *oder* verknüpften Bedingungen gebildeten Test:

```
joe@debian:~$ find /home \( -name core -o -name a.out
\ ) -exec /usr/bin/rm '{}' \;
```

Das Kommando löscht ohne Rückfrage alle Dateien namens `core` oder `a.out` in den Home-Verzeichnissen, vorausgesetzt, es wird vom Verwalter aufgerufen. Komplexe `find`-Aufrufe kommen vor allem in Shellskripten vor.

Ein auf dem System eingerichtetes Kommando – im Beispiel der Editor Emacs – lässt sich mit folgenden drei Werkzeugen finden:

```
joe@debian:~$ which emacs
```

Mittels `which` erfährt der Benutzer den absoluten Pfad eines Kommandos, das über die Shellvariable `PATH` gefunden werden kann. Eingebaute Shell-Kommandos gehören nicht dazu.

```
joe@debian:~$ whereis emacs
```

Etwas gesprächiger gibt sich das Kommando `whereis`, das in Standardverzeichnissen nach ausführbaren Dateien, zugehörigen Manualseiten und weiterem Zubehör sucht.

```
joe@debian:~$ locate emacs
```

Das Kommando `locate` setzt voraus, dass der Verwalter vorher das Kommando `updatedb` aufgerufen hat, das eine kleine Datenbank anlegt, in der `locate` schnell sucht und findet. Die Datenbank sollte von Zeit zu Zeit vom Verwalter aktualisiert werden. Sie umfasst nicht nur Kommandos, sondern Dateien aller Art. Die Antwort auf obige Zeile besteht in allen absoluten Pfaden, in denen die Zeichenkette `emacs` vorkommt. Das waren auf der Maschine, auf der diese Zeilen geschrieben wurden, 2114 Treffer. `locate` versteht die Metazeichen der Shell in Dateinamen (keine regulären Ausdrücke). Im Vergleich zu `find` ist `locate` erheblich schneller, kann aber nur Dateien ermitteln, deren Namen in der Datenbank erscheinen.

4.2.6 Suchen nach Debian-Paketen (`ara`, `dpkg-iasearch`, `packagesearch`, `apt-file`, `dlocate`)

Sucht man gezielt nach Debian-Paketen, gibt es besser geeignete Werkzeuge als `grep`, `find` oder `locate`. Das Kommando `ara` fragt die Debian-Paket-Datenbank `/var/lib/dpkg/available` ab und beherrscht logische Verknüpfungen sowie reguläre Ausdrücke. Einen Eindruck von der Datenbank gewinnt man mit:

```
joe@debian:~$ head -20 available
```

im entsprechenden Verzeichnis. Die Manualseite zu `ara` ist nicht ein Muster von Verständlichkeit, aber mit:

```
joe@debian:~$ ara -examples
```

bekommt man einige Beispiele auf den Schirm. Ein einfaches wäre:

```
joe@debian:~$ ara -list 'Section=graphics'
```

wobei *graphics* die englische Bezeichnung einer Abteilung ist. Als Antwort erscheint eine Liste der in der Abteilung enthaltenen Pakete, ob sie eingerichtet sind oder nicht. Für eine so simple Aufgabe ist *ara* überdimensioniert; erst bei komplexen Fragen zeigt sich seine Stärke.

Das Paket *dpkg-iasearch*, zur Zeit nur in *oldstable* enthalten, stellt einige Werkzeuge zum Suchen nach Paketen bereit. Zuerst ist mittels *dpkg-make-index* ein Index der Pakete anzulegen. Der Aufruf:

```
joe@debian:~$ dpkg-iaquery
```

erzeugt dann ein kleines Fenster, in dem man die Zeile *query* selektiert, worauf ein weiteres Fenster zum Eingeben eines Suchbegriffs in englischer Sprache erscheint. Dort tippen wir beispielsweise *chess* ein und erfahren, dass es zwei diesbezügliche Debian-Pakete gibt, nämlich *xboard* und *gnuchess*. Im Paket *dpkg* aus *sarge* ist ein ähnliches Werkzeug namens *dpkg-query* enthalten.

Das grafische Frontend *packagesearch* ist einfach zu benutzen. Gefüttert mit dem Suchwort *chess* findet es 25 Pakete und liefert deren Beschreibung mit. Beschränkt man die Suche auf eingerichtete Pakete, bleiben 3, was stimmen könnte.

Zu APT gehört das Kommando *apt-file*. Mittels:

```
debian:~# apt-file update
```

wird die APT-Datenbank aktualisiert, falls das noch nicht erledigt ist. Der Aufruf:

```
joe@debian:~$ apt-file search chess
```

liefert die Namen aller Pakete samt den absoluten Pfaden der darin enthaltenen Dateien, in denen die Zeichenkette *chess* vorkommt.

```
joe@debian:~$ apt-file list gnuchess
```

listet alle Dateien im Paket *gnuchess* auf. Ähnliche Informationen stellt auch das Kommando *apt-cache* bereit:

```
joe@debian:~$ apt-cache search gnuchess
```

zeigt die einschlägigen Pakete mit einer kurzen Beschreibung an,

```
joe@debian:~$ apt-cache showpkg gnuchess
```

liefert den zum Paket *gnuchess* gehörenden Eintrag aus der Datenbank zurück. Schließlich kann man noch mit *dlocate* nach Paketen und den darin enthaltenen Dateien suchen. Alle Suchwerkzeuge greifen auf dieselben Datenbasen zu und unterscheiden sich etwas im Gebrauch und in der Auswahl der Daten. Ein Benutzer sucht sich das für seine Fragen am besten passende Werkzeug aus und bleibt dann dabei.

Im Netz gibt es auch eine Suchmaschinen speziell für RPM- und Debian-Pakete: <http://rpmseek.com/>. Sie wird nicht von den jeweiligen Projekten betrieben und ernährt sich von Reklame, aber vorbeizuschauen lohnt sich.

4.2.7 Suchen nach Informationen über Benutzer (finger)

Das Kommando `finger` bringt einige Informationen über lokale oder entfernte Benutzer auf den Bildschirm:

```
joe@debian:~$ finger wulf
```

```
joe@debian:~$ finger wulf@ganz_toller_rechner
```

vorausgesetzt, auf dem jeweiligen Rechner läuft ein Fingerdämon (`fingerd`) – meist als Knecht des `inetd`, siehe dessen Konfiguration in `/etc/inetd.conf` – und der Port (79/tcp) ist in `/etc/services` bekannt. Die Information kann so aussehen:

```
Login: wulf                Name: W. Alex
Directory: /home/wulf      Shell: /bin/bash
Office: Dachstueble, 07244-000000 Home Phone: 0160-0000000
On since Wed Mar  9 08:56 (CET) on :0 (messages off)
Mail last read Wed Mar  9 21:01 2005 (CET)
Project: Winter 2004/05: Debian Band II (fuer Anwender)
Plan: Schreiben, dass die Rechner qualmen.
```

An Fingerdämonen besteht eine gewisse Auswahl, siehe die Paketlisten. Der Fingerdämon bezieht seine Information aus den Dateien `/etc/passwd`, `/var/log/wtmp` und anderen. In `/etc/passwd` wird das Kommentarfeld (GECOS-Feld) ausgewertet, das durch Komma getrennt den bürgerlichen Namen des Benutzers, seine Raumnummer, seine dienstliche und seine private Telefonnummer enthält. Heute könnten es auch Festnetz- und Mobilnetznummer sein. Mittels des Kommandos `chfn` kann jeder Benutzer seine eigenen Informationen ändern.

In der Datei `$HOME/.project` darf genau eine Textzeile stehen, die die Projekte benennt, an denen der Benutzer mitarbeitet, während in der Datei `$HOME/.plan` ein beliebig langer Text stehen darf. Möglicherweise wird auch ein öffentlicher PGP-Schlüssel aus der Datei `$HOME/.pgpkey` angezeigt. In früheren Zeiten hat LINUS TORVALDS den Stand des Linux-Projektes in seiner `.plan`-Datei auf <http://kruuna.helsinki.fi/> veröffentlicht. Heute noch kann man mit:

```
joe@debian:~$ finger @www.kernel.org
```

sich nach dem Stand des Linux-Kerns erkundigen.

Innerhalb eines überschaubaren Netzes bestehen keine Bedenken gegen die Fingerei. Ins böse Internet sollte man personenbezogene Informationen – auch wenn sie harmlos sind – nicht gelangen lassen. In der Regel blockt eine Firewall von außen kommende Anfragen auf Port 79 ab.

4.2.8 Lokale Webserver

Web-Browser sind zu einem allgegenwärtigen Werkzeug zum Austauschen von Informationen geworden, sodass es sich anbietet, sie auch für Zwecke einzusetzen, die

mit dem Web nichts zu tun haben. Beispielsweise bringen viele Netzkomponenten (Drucker, Router, Access Points, Switches, IP-Telefone ...) Web-Interfaces mit, kleine, spezialisierte HTTP-Server. Ganze Rechner lassen sich mit Hilfe des Webmin-Systems konfigurieren und überwachen.

Will man ein lokales Verzeichnis mit Informationen wie `/usr/share/doc/` im lokalen Netz für Web-Browser zugänglich machen, so genügt dafür ein einfacher HTTP-Dämon, der Apache-Webserver wäre ein zu gewaltiger Hammer. Eine minimale, aber für diesen Zweck ausreichende Lösung und darüber hinaus ein nettes Spielzeug ist der `micro-httpd` mit der Manualseite `micro_httpd` (Unterstrich, im Paketnamen Bindestrich).

Der `micro_httpd` läuft als Knecht des `inetd` und ist im Verzeichnis `/sbin/` zu Hause. Seine Konfiguration besteht aus zwei Zeilen. In die Datei `/etc/inetd.conf` ist am Ende einzutragen (einzeilig, vergleiche die bereits vorhandenen Einträge):

```
micro_httpd stream tcp nowait nobody /usr/sbin/micro_httpd
micro_httpd /usr/share/doc
```

Darin ist das letzte Argument der absolute Pfad des Verzeichnisses, das wir in das lokale Netz stellen wollen, der Webspace. Dessen Zugriffsrechte müssen jedermann das Durchsuchen und Lesen gestatten (755). In die Datei `/etc/services` ist ebenfalls am Ende einzutragen:

```
micro_http 8888/tcp # Micro HTTP Server
```

Damit wird dem neuen Dienst der Port 8888 zugewiesen; es kann auch ein anderer, noch unbenutzter Port sein. Anschließend ist der `inetd` zu restarten. Rufen wir nun in einem Web-Browser den neuen Server auf:

```
joe@debian:~$ http://servername:8888/
```

meldet sich die Neuanschaffung. Sie beherrscht die Ausgabe von Verzeichnislisten und findet als Default eine etwaige Datei `index.html`. Damit erschöpfen sich die besonderen Fähigkeiten des Winzlings.

Bei der Darstellung des Verzeichnisses `/usr/share/doc/` blieben zwei Wünsche offen. Die per HTTP übertragenen Verzeichnislisten bestanden aus inaktiven Einträgen, nicht aus Hyperlinks. Dem ließ sich durch ein kleines, rekursiv arbeitendes Shellsript abhelfen, das in jedem Verzeichnis eine Datei `index.html` anlegt, die den Verzeichnisinhalt (die Ausgabe von `ls`) in Hyperlinks verpackt enthält. Damit waren die Zeilen in den Verzeichnislisten anklickbar.

Der zweite Wunsch betraf die Verarbeitung der übertragenen Dateien durch den Browser. Nur ein Teil der Dateien in `/usr/share/doc/` liegt als html- oder Textdatei vor, die meisten sind gezippte Textdateien, die der Browser zunächst einmal zum Herunterladen anbietet. Wir wollen aber nur lesen, und das Entzippen sollte der Browser übernehmen, nicht der Webserver. Da fehlt beim Browser noch etwas.

Ist das lokale Netz größer oder der Webspace stark gegliedert – vielleicht mit unterschiedlichen Rechten für einzelne Verzeichnisse – kommt als Webserver der

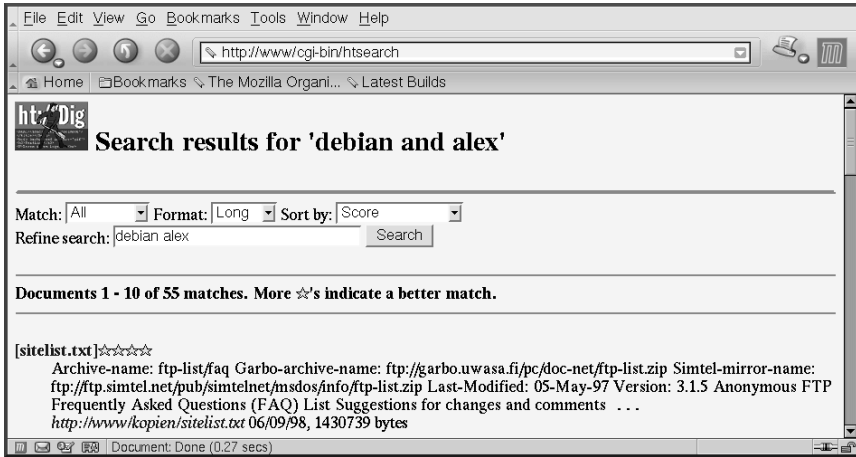


Abbildung 4.1. Screenshot der Suchmaschine htDig, die auf lokalen Servern läuft

Apache (<http://www.apache.org/>) in Frage, bei Debian in voller Pracht erhältlich. Dazu gehören eine lokale Suchmaschine wie htDig – siehe Abbildung 4.1 – oder WebGlimpse und ein Statistikprogramm wie Analog, alle bei Debian abzurufen (Glimpse im non-free-Bereich). Die Einrichtung ist nicht schwierig, aber Sache des Verwalters. Ein lokaler Webserver eignet sich auch gut als Testbett für Webseiten, die nach bestandener Prüfung in das große Web gestellt werden sollen.

4.3 Informationen im Internet

4.3.1 Übersicht (CUSI, Suchfibel, Suchlexikon)

Im Internet liegt eine Unmenge von öffentlich zugänglichen und größtenteils kostenfreien Informationen herum. Zum Zurechtfinden im Netz braucht man ebenfalls einige Informationen, eine Art von Netz-GPS. Die zahlreichen Verweise im vorliegenden Buch und auch in den Büchern und Artikeln anderer Autoren zu verwandten Themen zeigen deutlich, wie wichtig das Internet als Informationsquelle geworden ist. Eine Bemerkung vorweg: Das Web ist nicht das Internet, sondern ein Teil davon, und Google ist nicht das Web, sondern *eine* Suchmaschine im Web. Dass nicht jede Information aus dem Netz das Evangelium ist, dürfte der geneigte Leser bald merken. Die erste Frage ist stets, wie man eine bestimmte Information sucht und findet. Dazu muss man wissen, wo und wie im Internet zu suchen ist. Hat man eine Information gefunden, lauten die nächsten Fragen: Ist sie zuverlässig? Woher stammt sie? Wie alt ist sie? Welche Absichten könnten dahinter stecken?

Die *Configurable Unified Search Engine* (CUSI) der UNIX-Arbeitsgemeinschaft in der Universität Siegen bietet eine einheitliche Oberfläche zum Eingeben von Suchbegriffen und eine kommentierte Auswahl von Suchdiensten und anderen Quellen, die von Google bis zum Usenet reichen. Auf der Maschine kann man sich einen

guten Überblick über die Suchmöglichkeiten im Internet verschaffen. Da einige Einträge ihrerseits wieder Suchmöglichkeiten auflisten, sollte man einige Stunden oder besser ein verregnetes Wochenende für den ersten Besuch bei CUSI einplanen, aber es lohnt sich.

Unter dem URL <http://www.klug-suchen.de/> meldet sich eine Webseite, die 1600 Suchmaschinen auflistet, in 15 Gruppen eingeteilt. Ein Teil des Textes ist ziemlich klein gesetzt, und dann noch in dunkelblauer Schrift auf mittelblauem Grund. Sie bietet eine Funktion zum Suchen nach Suchmaschinen, die bei den Begriffen *Spreewald*, *Ratatouille* oder *Chianti* keine zuständige Suchmaschine fand und für *SSL* (Secure Socket Layer) zwei Suchmaschinen anbot, die sich mit Russland befassen. Nicht überzeugend.

Bei <http://karzauninkat.de/> findet sich unter dem Punkt *Suchmaschinen* ein Text mit Hyperlinks zu einem Suchlexikon mit 2600 Suchmaschinen, jede ausführlich kommentiert, und zu einer Suchfibel, die direkt unter <http://www.suchfibel.de/> zu erreichen, gut verständlich und auch als Buch zu haben ist. Eine kürzere Anleitung desselben Autors zum gezielten Suchen liegt unter <http://www.heise.de/ct/99/23/172/>. *Stefan Karzauninkat* hat auf der erstgenannten Website im Verzeichnis Goldhtml / *Die Goldenen Regeln für schlechtes HTML* zusammengestellt, 70 an der Zahl. Sie scheinen sich noch nicht unter allen Web-Designern herumgesprochen zu haben.

Schließlich bietet die Bibliothek der Universität Konstanz eine Liste deutschsprachiger Suchmaschinen unter dem URL <http://www.ub.uni-konstanz.de/dt-suchm.htm> an, die aktuell ist und geschätzt 100 Einträge umfasst, natürlich als Hyperlinks, ohne Kommentar. Ebenfalls in der Universität Konstanz – unter <http://www.inf-wiss.uni-konstanz.de/suche/> – findet der Benutzer weitere Hilfen zum Suchen im Internet. Wer sich für die Technik und den Hintergrund der Suchmaschinen interessiert, wird bei WOLFGANG SANDER-BEUERMANN im Suchmaschinenlabor der Universität Hannover fündig (<http://metager.de/suma.html>).

4.3.2 Suchen nach Netzknoten (ping, arping, host, mtr)

Hat man die IP-Adresse oder den Namen eines Netzknotens (Rechner, Drucker, Router ...), so testet man mit dem Kommando `ping`, ob man den Knoten erreicht:

```
joe@debian:~$ ping 141.76.2.4
```

```
joe@debian:~$ ping ftp.de.debian.org
```

Das Kommando schickt im Sekundentakt ICMP-Pakete (Internet Control Message Protocol) an den Knoten, die dieser echot. Beendet wird der Versand mittels der Tastenkombination `<ctrl>-<c>`. Der Benutzer erfährt etwas über die Zuverlässigkeit der Verbindung und die Laufzeit der Pakete (Hin- und Rückweg zusammen). Mit der Option `-c anzahl` beendet sich `ping` nach dem Versand von `anzahl` Paketen selbständig. Fremde Rechner soll man nicht unnötig lange anpingen. Ein Rechner kann so konfiguriert werden, dass er auf Pings nicht antwortet.

Um einen neuen Rechner zu testen, untersucht man als erstes den Rechner selbst und dann immer weiter entfernte Knoten:

```
ping 127.0.0.1
ping localhost
ping 192.168.7.1
ping nachbarknoten
ping 192.168.7.254
ping router
ping 141.76.2.4
ping ftp.de.debian.org
```

Sind die Knoten nicht nur über ihre IP-Adresse, sondern auch über ihre Namen erreichbar, funktioniert die Namensauflösung mittels `/etc/hosts` oder DNS.

Dem Kommando `arping` kann man als Argument Namen, IP-Adressen oder MAC-Adressen (Ethernet-Adressen der Form `00:50:04:43:b5:d1`) übergeben. Es hilft bei Problemen im Netz gelegentlich weiter als `ping`, aber das ist dann schon ein Fall für den Netzverwalter.

Zum Ermitteln des Namens zu einer IP-Adresse oder umgekehrt dient das Kommando `host` (anstelle des veralteten Kommandos `nslookup`):

```
joe@debian:~$ host 141.76.2.4

joe@debian:~$ host ftp.de.debian.org
```

Es kommt vor, dass in der Antwort die Reihenfolge der Ziffergruppen umgekehrt ist. Deshalb sollte man in wichtigen Fällen die Gegenprobe machen. Will man eine Antwort von einem bestimmten DNS-Server haben, ist dessen IP-Adresse als zweites Argument zu übergeben:

```
joe@debian:~$ host ftp.de.debian.org 129.143.2.1

joe@debian:~$ host www.mvm.uni-karlsruhe.de
129.13.96.2
```

Die erste Zeile spricht gezielt einen DNS-Server des Belvue-Netzes an, die zweite einen DNS-Server der Universität Karlsruhe. In letzterem Fall fällt die Antwort zweizeilig aus:

```
www.mvm.uni-karlsruhe.de      CNAME mvmc100.ciw.uni-karlsruhe.de
mvmc100.ciw.uni-karlsruhe.de A      129.13.251.54
```

Die erste Zeile besagt, dass der Grundname (canonical name, CNAME) des Web-Servers `mvmc100` lautet. Die zweite Zeile nennt zu dem Grundnamen die IP-Adresse. Und auch das ist noch nicht die ganze Wahrheit. Die IP-Adresse `129.13.252.54` gehört nämlich einem NAT-Server (Network Address Translation, Masquerading), der die wirkliche Adresse des Rechners, die aus einem der privaten IP-Adressbereiche stammt, vor der Außenwelt verbirgt. Mit der privaten Adresse könnte ein Außenstehender auch nichts anfangen, da sie nicht weitergeleitet (geroutet) wird. Käme die Anfrage aus dem inneren, privaten Netz, lieferte die Antwort sinnvollerweise die private Adresse. Zu einer IP-Adresse können mehrere Namen

gehören wie bei obigem Webserver, nämlich ein Grundname und beliebig viele Aliasnamen, das ist nichts Ungewöhnliches. Umgekehrt können hinter einem Namen auch mehrere IP-Adressen stehen, beispielsweise hinter `www.ibm.com`. Wie die untereinander zurecht kommen, ist nicht unser Problem.

Mit dem Kommando `traceroute` oder besser `mtr` (Matt's traceroute) lässt sich der Weg verfolgen, den Datenpakete vom eigenen Rechner zu einem Zielrechner nehmen, hier zu `non-us.debian.org`:

```
Matt's traceroute [v0.48]
pcwulf Sun Nov 28 14:23:02 2004
Hostname:
router.localdomain
217.5.98.37
217.237.153.18
ams-e4.AMS.NL.net.DTAG.DE
ams-ix.tc2.xs4all.net
0.so-7-0-0.xr1.d12.xs4all.net
0.ge-1-3-0.cr1.d12.xs4all.net
klecker.debian.org
```

Das Kommando `mtr` vereint die Funktionen von `traceroute` und `ping` und gibt auch noch Statistiken aus; es wird mit der Tasteneingabe `<q>` gestoppt. Die erste Station ist der DSL-Router in unserem Keller. Von dort geht es über zwei namenlose (nicht im DNS eingetragene) Knoten, die vermutlich bei T-Online beheimatet sind. Hinter `ams-e4` verbergen sich dreizehn IP-Adressen. DTAG ist die *Deutsche Telekom AG*, der Knoten `ams-e4` sieht nach einem Gateway von der Deutschen Telekom in die Niederlande aus. Die folgenden drei Knoten gehören zum niederländischen Netz XS4ALL, und `klecker` ist schließlich identisch mit dem Ziel `non-us.debian.org`. Die Reise nach `www.debian.org` führt auch zu `klecker`, die Reise nach `ftp.de.debian.org` dagegen nicht.

Schließlich kann man beim Verzeichnisdienst `whois` (siehe Abschnitt 7.7 *Verzeichnisdienste* auf Seite 286) sein Glück versuchen:

```
joe@debian:~$ whois 129.13.64.5
```

Ob man aus der Antwort, die etwas Zeit benötigt, schlau wird, ist eine andere Frage.

4.3.3 Internet-Domänen

Jeder im Internet (nicht innerhalb abgeschlossener lokaler Netze, Intranets) verwendete Rechnername (Hostname) ist irgendwo im Domain Name System (DNS) registriert. Eine Maschine, die nicht registriert ist, kann nur über ihre IP-Adresse wie 123.456.78.9 erreicht werden. Das DNS ist ein auf viele DNS-Server weltweit verteiltes Datenbanksystem. Einen Überblick über alle Domännennamen aus der seit 1986 existierenden und heute fast zehn Millionen Einträge umfassenden Top-Level-Domäne `de` hat das Deutsche Network Information Center, das DENIC, <http://www.nic.de/>. Es bietet eine Domänenabfrage an, unter der man auch eine Auskunft über den Domäneninhaber bekommt. Fragen Sie einmal nach

den Domänen `alex.de` und `alex-weingarten.de`. Das DENIC gibt keine Auskunft über Hostnamen (`www.alex-weingarten.de`) oder andere Top-Level-Domänen (`.net` oder `.org`). Hostnamen erfährt man – wie bereits erläutert – mittels des Kommandos `host` von Nameservern, wie sie in der Datei `/etc/resolv.conf` eingetragen sind.

Domännennamen aus der Top-Level-Domäne `fr` findet man bei dem Server `http://www.afnic.fr/`, der zum Glück auch auf `http://www.nic.fr/` horcht, sonst wäre das Raten der Webadresse schwierig. Es geht nicht so einfach weiter. Ein Besuch bei `http://www.nic.org/` zeigt, dass die Webadresse zu einem *National Investment Center* in Annapolis gehört. Bei `http://www.nic.net/` haben wir mehr Erfolg, es erweist sich auskunftsbereit für die Top-Level-Domänen `com`, `biz`, `net`, `org`, `info` und `us`. Einen Server `http://www.nic.us/` gibt es ebenfalls. Die Seite `http://www.nic.nu/` von den Niue-Inseln spricht sogar sechs europäische Sprachen. Die Wikipedia sagt Ihnen, wo die Inseln zu suchen sind. Informationen aus erster Hand liefert `http://www.niue.nu/`, die *Internet Users Society of Niue*. Weiteres zu Informationen über Domännennamen siehe Abschnitt 7.7 *Verzeichnisdienste* auf Seite 286.

4.3.4 Email-Anschriften (MESA)

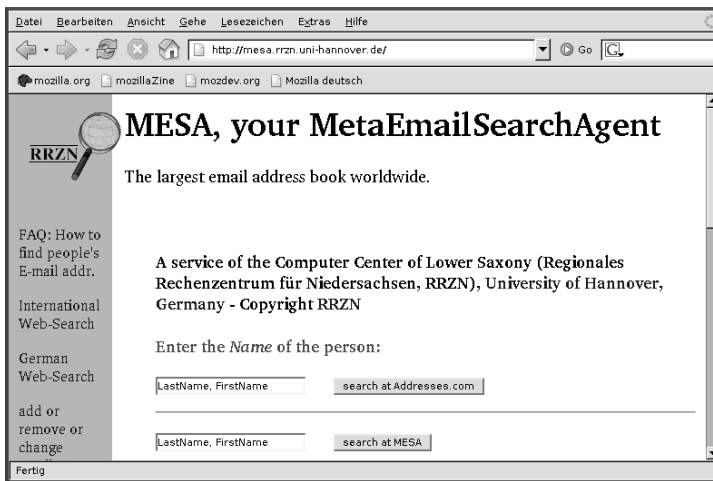


Abbildung 4.2. Screenshot der Email-Anschriften-Suchmaschine MESA

Es gibt keine Verzeichnisse von Email-Anschriften vergleichbar mit Telefonbüchern. Sucht man die Email-Anschrift einer Person, wendet man sich an den *Meta Email Search Agent* (MESA), eine Meta-Suchmaschine der Universität Hannover, zu finden unter `http://mesa.rrzn.uni-hannover.de/`. Sie beauftragt eigentliche Suchmaschinen mit der Suche und liefert die Ergebnisse in vereinheitlichter Form an den Benutzer zurück.

In der Universität Konstanz wird unter dem URL http://www.inf-wiss.uni-konstanz.de/suche/email_druck.htm eine Übersicht von Email-Verzeichnissen geführt, im Frühjahr 2005 mit 12 Einträgen, und der Möglichkeit, Anfragen abzuschicken. Ein Test mit mehreren Maschinen verlief negativ, obwohl die Email-Anschriften des Verfassers seit 10 Jahren veröffentlicht und in mehreren Internet-Diensten verwendet werden. Jeder Spammer kennt sie. Eine allgemeine Suche nach dem Personennamen im Web verspricht mehr Erfolg. Die Netzteilnehmer behandeln zunehmend ihre Email-Anschriften vertraulich, um die Flut von Spam (unerwünschte Email) einzudämmen. Schade, und nicht der optimale Weg, mit Spam fertig zu werden.

4.3.5 Wörterbücher, Glossare und Lexika (LEO, Wikipedia)

Das Nachschlagen in Wörterbüchern im LAN wurde bereits in Abschnitt 4.2.3 *Suchen in lokalen Wörterbüchern* auf Seite 124 behandelt. Im Web findet sich heute eine unüberschaubare Anzahl von kleinen und großen Wörterbüchern, Glossaren und Lexika, von denen hier nur einige als Appetitanreger vorgestellt werden.

Die Wikipedien haben wir schon in Abschnitt 1.5 *Hilfen* auf Seite 36 als Informationsquelle bei Computerfragen kennen gelernt. Sie decken aber das gesamte Wissen ab und sind den bekannten Enzyklopädien wie Brockhaus, Meyers, Larousse oder Encyclopedia Britannica vergleichbar, wenn auch nicht mit deren Tradition und Erfahrung, dafür oft aktueller und manchmal ausführlicher. Da die Verlage mit ihren Werken Geld verdienen müssen, stehen die traditionellen Enzyklopädien nicht oder nur mit Einschränkungen (Magerversion, Campuslizenz, kostenpflichtiges Abonnement) im Netz:

- <http://www.brockhaus.de/>
- <http://www.meyerslexikonverlag.de/>
- <http://www.larousse.fr/>
- <http://www.britannica.com/>

Die Papiaerausgabe wird meist durch eine vergleichsweise preisgünstige Ausgabe auf CD oder DVD ergänzt, sodass man nach deren Erwerb mit dem Rechner offline suchen oder multimediale Einträge wiedergeben kann. Noch einige umfassendere Webseiten:

- Eine Übersicht über Lexika und Wörterbücher stellt die Universität Erlangen unter <http://www.phil.uni-erlangen.de/~p2gerlw/ressourc/lex.html> ins Web.
- Die Informatik der Technischen Universität München betreibt den Online-Service LEO unter <http://www.leo.org/> mit einer Linksammlung zu Wörterbüchern und Glossaren sowie eigenen zweisprachigen Wörterbüchern.
- Einen deutschen Wortschatz von 9 Millionen Wörtern findet der Benutzer in der Universität Leipzig unter <http://wortschatz.informatik.uni-leipzig.de/>.
- Das *Deutsche Wörterbuch* der Brüder GRIMM stellt die Universität Trier unter <http://www.dwb.uni-trier.de/> zur Verfügung.

- Ein Meta-Wörterbuch für 40 Sprachen bietet die Webseite <http://www.foreignword.com/Tools/dictsrch.htm>.
- JOHN RYAN hat unter <http://web.syr.edu/~jryan/infopro/> *Information Resources for Information Professionals* vorwiegend aus dem angloamerikanischen Raum zusammengestellt.

Das sollte für den Einstieg reichen. Die Aufzählung ließe sich noch erheblich verlängern.

Bei der Deutung der in der Computerei beliebten Abkürzungen und Akronyme helfen mehrere Webseiten, darunter:

- Babel aus den USA, http://www.geocities.com/ikind_babel/babel/babel.html, von IRVING & RICHARD KIND,
- V.E.R.A. <http://cgi.snafu.de/ohei/user-cgi-bin/veramain.cgi>, auch in der Newsgruppe de.etc.lists, von OLIVER HEIDELBACH,
- Abklex <http://www.abklex.de/abklex/>, auf Tschechisch <http://www.sochorek.cz/archiv/slovniky/abklex.htm>.

Auch diese Aufzählung könnte verlängert werden, vor allem um kleinere, spezialisierte Zusammenstellungen.

4.3.6 Suchen nach Büchern

Sucht man ein Buch – ganz gleich, ob man es neu oder gebraucht erwerben oder aus einer Bibliothek entleihen möchte – setzt man sich nicht mehr ans Telefon, sondern an den Rechner. Es gibt keine einheitliche Strategie, die auf jede Fragestellung passt.

Geht es um beliebige Bücher zu einem Thema, beginnt man mit einer allgemeinen Suchmaschine wie Metager und hofft, dass unter den Antworten einige sind, die zu Büchern führen, möglicherweise auf Umwegen. Der Suchbegriff *Varanger* – eine nette Gegend in Norwegen – liefert bei Metager als ersten Treffer den zugehörigen Eintrag in der deutschen Wikipedia. Von dort gelangt man auf vier norwegische Websites. Das Norwegische Museumsnetz führt uns zum *Varanger Samiske Museum* (<http://www.varjjat.org/>), dessen Linksammlung zu *De Samiske Samlinger* in Karasjok (<http://www.desamiskesamlinger.no>), dessen Linksammlung zum *Vadsø Museum* (<http://museumsnett.no/vadsomuseet/>). Das wäre auch schneller gegangen, aber da wir virtuell reisen, ist der Zeitverlust minimal. Das Museum in Vadsø bietet eine Webseite mit Büchern an. Dort finden wir JUHA HÖYKINPURO, *Varanger – nabo til Nordishavet*, gedruckt in Finnland. Damit kennen wir einen Buchtitel samt Verfasser und wissen, dass wir das Buch in Norwegen oder Finnland zu suchen haben, falls es nicht in Deutschland aufzutreiben ist. Unterwegs haben wir noch *Berlevåg Mannsangforening* in Bild und Ton kennen gelernt – heftig og begeistret.

Nun brauchen wir eine Bezugsquelle und versuchen unser Glück mit dem Suchwort *varanger* bei:

- <http://www.findmybook.de/>
- <http://www.zvab.de/>
- <http://www.amazon.de/>

jedoch ohne obiges Buch zu finden. <http://www.amazon.no/> gibt es nicht, <http://www.amazon.fi/> handelt mit Haustieren, Papageien und so. Aber wir wissen, dass es in Finnland zwei große Buchhandlungen gibt (andernfalls sucht man nach Helsinki und bokhandel):

- die Akademische Buchhandlung in Helsinki, deren deutschsprachige Abteilung größer ist als manche deutsche Buchhandlung, zu erreichen unter <http://www.akateeminen.com/>,
- die Finnische Buchhandlung, eine über ganz Finnland verbreitete Kette, <http://www.suomalainen.com/sk/>.

Erstere bietet ihre Seiten auf Finnisch und Schwedisch an, kennt das Buch jedoch nicht. Die zweite spricht nur Finnisch, bietet aber das Buch samt einem Webformular zum Bestellen an. Langsam wird es Zeit für ein Wörterbuch. Was *Postipaketti Eurooppaan* bedeutet, kann man ja noch erraten, aber der Rest ist nicht immer intuitiv-selbsterklärend. Vor fünfzehn Jahren wären weder Suche noch Bestellung so einfach gewesen.

Jetzt bleibt noch zu klären, ob das Buch irgendwo in Europa in einer Bibliothek steht. Wir suchen bei mehreren Bibliotheksverbunden:

- Karlsruher Gesamtkatalog, ein Metakatalog über die Bibliotheken im Raum Karlsruhe, <http://www.ubka.uni-karlsruhe.de/>,
- Karlsruher Virtueller Katalog, ein Metakatalog über viele Bibliotheken, <http://www.ubka.uni-karlsruhe.de/kvk.html>,
- Katalog des Südwestdeutschen Bibliotheksverbundes, <http://www.swbv.uni-konstanz.de/>,

und werden mit Hilfe des Karlsruher Virtuellen Kataloges in Norwegen bei der Nationalbibliothek in Oslo und bei der Universitätsbibliothek in Tromsø fündig, beide im Netz. Die Suche war insofern einfach, als das Suchwort *varanger* eindeutig und eng begrenzt ist. Ein Suchwort wie *multimedia* ist praktisch unbrauchbar, weil es viele Themen berührt, über die viel veröffentlicht ist. In solchen Fällen muss man zuerst bessere Suchwörter suchen.

Seit Mitte der siebziger Jahre werden Bücher und verwandte Erzeugnisse durch eine weltweit eindeutige *International Standard Book Number* (ISBN) und Periodika (Zeitschriften) durch eine *International Standard Serial Number* (ISSN) gekennzeichnet. Als Suchmerkmal ist die ISBN manchmal schon zu genau, da verschiedene Ausgaben (gebunden, Taschenbuch) oder Auflagen eines Buches verschiedene Nummern erhalten. Die zehnstellige, künftig dreizehnstellige ISBN wird aus vier Zifferngruppen für Sprache oder Gruppe, Verlag, Buch und Prüfziffer gebildet, beispielsweise 3-540-43267-1. Die Zahl 3 bezeichnet den deutschen Sprachraum, 540 darin den Springer-Verlag, 43267 das Debian-Buch von GANTEN und ALEX, 2. Auflage, ohne DVDs, und die 1 ist eine nach einem einfachen Algorithmus berechnete Prüfziffer. Große Verlage mit vielen Büchern haben eine niedrige Verlagsnummer, damit die Gesamtstellenzahl der ISBN bewahrt bleibt. Daher lässt eine ISBN wie 3-9805220-0-8 auf einen sehr kleinen Verlag schließen. Es handelt sich um ein Hochschulinstitut, das dann und wann ein Buch selbst publiziert. Die auf vielen Handelsartikeln zu

sehende EAN-Nummer samt ihrem Strichcode wird bei Büchern aus der ISBN abgeleitet. Weiteres in der Wikipedia oder bei <http://www.german-isbn.org/>.

4.3.7 Anonymous FTP, BitTorrent, Jigdo

Vor 1995 war das Web noch nicht in aller Munde. Man holte sich Texte oder Software per File Transfer Protocol (FTP) nach RFC 959 von Servern, die das anonyme Fernkopieren zuließen und oft in Hochschulen angesiedelt waren. Auf diese Weise konnte ein PC-Benutzer sein DOS anreichern. Es gab auch einen Suchdienst namens Archie; der deutsche Archie stand in Darmstadt, *requiescat in pace* zusammen mit anderen vom Web abgelösten Diensten wie Gopher, Veronica, Jughead und WAIS.

FTP spielt auch heute noch eine gewisse Rolle. Beispielsweise können Debian-Pakete per FTP oder HTTP von Debian-Servern heruntergeladen werden, ebenso die RFCs von <ftp://ftp.denic.de/>. Auf vielen Servern sind inzwischen die öffentlich zugänglichen FTP-Verzeichnisse (/pub/) und der Webspaces identisch. Web-Browser beherrschen beide Protokolle. Umgekehrt werden Manuskripte vom Autor zum Verlag per FTP hochgeladen, natürlich nicht anonym. Die Suche nach Informationen auf FTP-Servern übernehmen die Suchmaschinen des Webs.

Eine Verbindung zu einem öffentlich zugänglichen FTP-Server – hier ein Debian-Server in der Technischen Universität Dresden – wird durch:

```
joe@debian:~$ ftp ftp.de.debian.org
```

aufgebaut. Der Server fragt nach Benutzername (ftp) und Passwort, wobei meist die Email-Anschrift des Benutzers als Passwort erwartet wird. Manchen Servern ist das jedoch gleich. Kommt die Verbindung zustande, navigiert man sich mit den FTP-Kommandos `help`, `pwd`, `cd` und `ls` zur gesuchten Datei durch, lädt sie mit `get` herunter und verabschiedet sich mit `quit`. Dies sind *keine* Linux/UNIX-Kommandos, obwohl sie gleich lauten. Ihre Syntax und ihre Fähigkeiten sind anders, es gibt kein `ls -l`. Einzelheiten auf der Manualseite zu ftp.

Beim Herunterladen großer Brocken wie der ISO-Abbilder (E: image, F: image) von Debian oder der Wikipedia wächst die Wahrscheinlichkeit, dass die Datenübertragung durch einen Fehler unterbrochen wird. Einfache FTP-Clients beenden sich und müssen neu von vorn beginnen, was nicht viel Erfolg verspricht. Für solche Aufgaben braucht man intelligentere Clients, auch Download-Manager genannt. beispielsweise `wget`, zu Hause auf <http://www.gnu.org/software/wget/>. Wer es grafisch liebt, holt sich das GNOME-Frontend `gwget` dazu.

Das Werkzeug `wget` beherrscht die Protokolle FTP, HTTP und HTTPS, arbeitet nach dem Start ohne Dialog im Hintergrund weiter und kann rekursiv ganze Verzeichniszweige oder zusammengesetzte Webseiten holen. Unterbrochene Downloads setzt es an der Bruchstelle fort. Es lässt sich in Shellskripten oder Cronjobs verwenden und versteht sich auf Zeitstempel. Als Argument braucht es den URL des gewünschten Dokuments:

```
joe@debian:~$ wget
http://www.abklex.de/abklex/abklex.txt
```

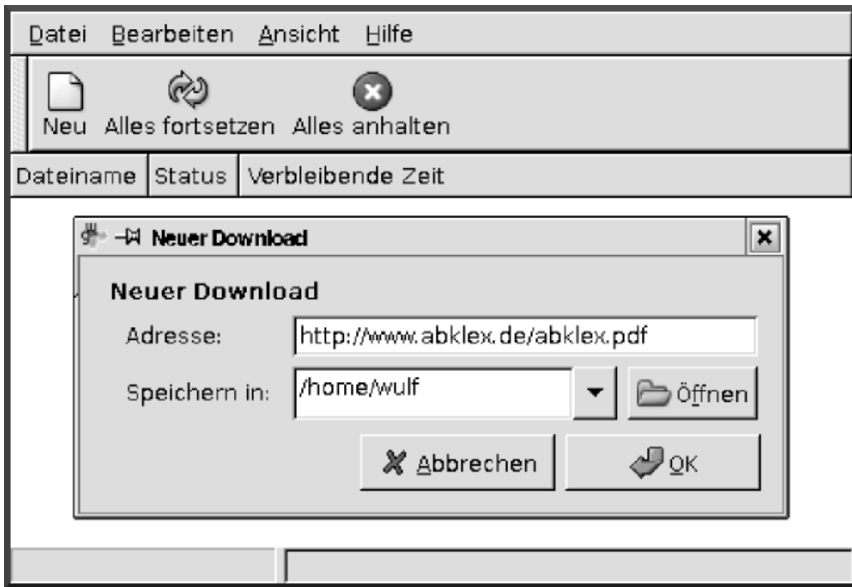


Abbildung 4.3. Screenshot des Download-Managers wget mit grafischer Oberfläche gwget

Abbildung 4.3 zeigt das Werkzeug mit seiner grafischen Oberfläche. Es hat gerade ein zweites Fenster zur Eingabe des URLs geöffnet.

Vorsicht ist im Verkehr mit FTP-Servern geboten, die nicht unter einem unix-ähnlichen Betriebssystem laufen. Alles, was nicht in Zeilen unterteilter Text ist, muss im binären Modus heruntergeladen werden, der nicht in jedem Fall voreingestellt ist. Das FTP-Kommando zum Umschalten in den binären Modus lautet `bi`. Den gegenteiligen Modus erreicht man mit `ascii`. Hat man versehentlich eine Textdatei im binären Modus heruntergeladen, ist sie mit einfachen Werkzeugen zu retten. Hat man eine Binärdatei im ASCII-Modus heruntergeladen, ist sie Schrott. Zwischen Clients und Servern aus gleichen Welten spielt der Unterschied keine Rolle. Hinter einer Firewall ist eine passive Verbindung aufzubauen, bei der die Datenübertragung vom Client angestoßen und damit von der Firewall durchgelassen wird.

Eine Alternative zum Herunterladen insbesondere bei extrem großen Dateien ist das Filesharing nach dem BitTorrent-Protokoll (Wikipedia und <http://www.bittorrent.cd/>). Bei Debian finden sich zwei zugehörige Pakete. Während das Herunterladen von Dateien den FTP- oder HTTP-Server stark belastet, verteilt das BitTorrent-Protokoll die Last auf Server und auf Clients, die die gefragte Datei bereits heruntergeladen haben und nun ihrerseits als Server mithelfen. Die Daten werden nicht mehr sternförmig von einem einzigen oder wenigen Servern verteilt, sondern auch zwischen den Benutzern wie in einem Peer-to-Peer-Netz. Das setzt voraus, dass die Clients oder Peers über eine brauchbare Upload-Bandbreite verfügen – was bei dem verbreiteten asymmetrischen DSL (ADSL) nicht gegeben ist – und durch die Uploads kostenmäßig nicht belastet werden, da sie für ihre Netzanbindung pau-

schal bezahlen (Flatrate). Ein Anwendungsbeispiel war die deutsche Wikipedia, die ihr DVD-Image im Umfang von 2,6 GB vorübergehend zum Herunterladen per BitTorrent angeboten hat, mittlerweile jedoch ausschließlich auf eDonkey2000 (ed2k) übergegangen ist.

Als weitere BitTorrent-Clients finden sich bei Debian Azureus (<http://azureus.sourceforge.net/>), BitTornado (<http://bittornado.com/>), QTorrent (<http://thegraveyard.org/qtorrent.php>) sowie in der *unstable*-Distribution im Paket *mldonkey-server* das Werkzeug *MLDonkey* (<http://www.mldonkey.org/> und <http://www.nongnu.org/mldonkey/>). Eine kurze Übersicht über das Thema vermittelt das *Datentausch-Dienste-Mini-FAQ*, aktualisiert Jahresmitte 2004.

Jigsaw Download, kurz *Jigdo*, von RICHARD ATTERER ist ein weiteres Konzept, um große Dateien unters Volk zu bringen. Sein Heimathafen ist <http://atterer.net/jigdo/>. Das Debian-Projekt beabsichtigt, es für seine CD/DVD-Images einzusetzen. So ein Image ist eine einzige riesige Datei, aber es besteht ursprünglich aus vielen vergleichsweise kleinen Debian-Paketen. Die beim Benutzer eingerichtete Jigdo-Software aus dem Debian-Paket *jigdo-file* lädt die entsprechend vorbereiteten Pakete – unter Umständen von verschiedenen Servern – herunter und setzt sie beim Benutzer zum Image zusammen. Die Server sparen sich dadurch das Speichern der Images für CDs und/oder DVDs zusätzlich zu den Paketen. Jigdo spart darüber hinaus auch Bandbreite im Netz, da es beim Updaten von Images nur die Änderungen herunterlädt, ähnlich einem Versionskontrollsystem wie CVS. Jigdo benutzt kein eigenes Protokoll zur Übertragung, sondern das Werkzeug *wget* und dieses per Default HTTP.

4.3.8 Web-Browser

Das World Wide Web, kurz WWW oder Web genannt, hat sich in kurzer Zeit zu einem Internetdienst entwickelt, den fast jeder kennt und benutzt. Manche verwechseln sogar das Web mit dem Internet. Das Web erblickte 1991 am CERN (Kernforschungszentrum) in Genf das Licht der Welt und ist damit so alt oder jung wie Linux. Die technische Entwicklung wird heute vom World Wide Web Consortium (W3C) koordiniert, das unter <http://www.w3.org/> zu erreichen ist.

Im Web stellen Server einfache bis multimediale Inhalte öffentlich oder eingeschränkt, kostenlos oder entgeltlich zur Verfügung. Die Clients werden als Web-Browser oder kurz als Browser (F: navigateur) bezeichnet. Das wichtigste Übertragungsprotokoll ist das Hypertext Transfer Protocol (HTTP). Im Unterschied zu Rundfunk oder Fernsehen strahlen die Server ihre Mitteilungen nicht ungezielt ins Netz; ein Client muss bei einem bestimmten Server anklopfen und um ein bestimmtes Dokument bitten. Dazu braucht der Client, der Browser, die Adresse des Dokumentes im Web. Diese wird Uniform Resource Locator (URL) genannt und besteht im Wesentlichen aus Protokoll, Servername und Dokumentname:

<http://www.w3.org/Addressing/>

Ein URL muss die standardgerechte Form einhalten und kann weitere Elemente aufweisen; das Konzept ist allgemeiner als geschildert. Ein Server kann statt über seinen Namen auch über seine IP-Adresse angesprochen werden, was aber möglichst zu vermeiden ist. Bei den Dokumentnamen spielt Groß- und Kleinschreibung eine Rolle, wie stets auf Linux/UNIX-Systemen. Uniform Resource Identifiers (URI) sind ein Oberbegriff zu URLs. Die Webdokumente werden in der Hypertext Markup Language (HTML) oder ihren Weiterentwicklungen abgefasst. Wir lernen sie in Abschnitt 5.12 *Schreiben für das Web* auf Seite 223 kurz kennen.



Abbildung 4.4. Screenshot des Web-Browsers Firefox

An Browsern besteht eine reiche Auswahl, vom einfachen Textbrowser (für blinde Benutzer wichtig) bis zum Browser, der außer dem Web auch alle anderen Internetdienste beherrscht oder sich wenigstens darum bemüht:

- lynx, ein curses-basierter Text-Browser, <http://lynx.browser.org/>,
- elinks, ähnlich wie lynx, <http://elinks.or.cz/>,
- links2, ähnlich wie lynx, zusätzlich mit X11 grafikfähig, <http://atrey.karlin.mff.cuni.cz/~clock/twibright/links/>,
- Amaya, der Referenz-Browser des W3C, <http://www.w3.org/Amaya/>,
- Netscape Navigator, Mozilla, Firefox, eine Generationenfolge weit verbreiteter grafikfähiger Browser mit vielen Plugins zur Ausweitung der Fähigkeiten, <http://www.mozilla.org/products/firefox/>, siehe Abbildung 4.4,
- Galeon, ein GNOME Browser, setzt auf der Rendering-Maschine Gecko von Mozilla auf, <http://galeon.sourceforge.net/>,
- Epiphany, ähnlich wie Galeon, <http://www.gnome.org/projects/epiphany/>,

- Konqueror, der Browser und zugleich Dateimanager von KDE, <http://www.konqueror.org/>,
- Opera, <http://www.opera.no/>, kostenlos bis erschwinglich, aber nicht bei Debian erhältlich.

Eine umfassendere Übersicht findet sich auf der Webseite http://www.itp.uni-hannover.de/~kreutzm/en/lin_browser.html. In Abbildung 4.4 sehen wir unterhalb der Titelleiste eine Menüleiste, darunter eine Leiste mit fünf Symbolen für häufig gebrauchte Aktionen – die man auch aus der Menüleiste aufrufen könnte – und einem Fenster zum Eingeben und Ändern von URLs. Links vom Hauptfenster öffnet sich auf Wunsch ein schmales Fenster mit dem Verlauf der Sitzung oder mit Lesezeichen (E: bookmark, F: signet), rechts vom Hauptfenster ein Rollbalken. Unten befindet sich eine Statusleiste. Hat man nur einen kleinen Bildschirm, sollte man alle überflüssigen Fenster weglassen.

Nach der Einrichtung eines Browsers ist er zu konfigurieren. Der Weg führt über die Menüpunkte *Bearbeiten* -> *Einstellungen* bzw. *Edit* -> *Preferences*. Der erste Schritt ist, die Webseite festzulegen, mit welcher der Browser nach dem Aufruf startet. In unserem DAN ist das natürlich die Startseite (Homepage) unseres häuslichen Webserver, aber es könnte auch die Startseite einer Hochschule oder eines Brötchengebers sein. Der zweite Schritt ist die Eintragung eines Proxy-Servers (Proxie, E: proxy server, F: serveur mandaté), sofern der Internet Service Provider (ISP) einen solchen anbietet, was meist der Fall ist. Der Rest betrifft weniger dringende Punkte. Die sicherheitsrelevanten Einstellungen sollte man zwar bald vornehmen, aber erst, nachdem man verstanden hat, worum es geht. Und so lange keine vertraulichen Daten über das Web schicken.

Ob man auch andere Internetdienste (FTP, News, Email) über einen Web-Browser nutzt, ist eine Frage der Gewohnheit. Viele Benutzer bevorzugen die jeweiligen Spezialwerkzeuge, weil sie schneller und leistungsfähiger sind. Dagegen gewinnen die Web-Browser als Oberfläche zur Verwaltung eines Netzes an Boden, Stichwort *Webmin*.

4.3.9 Suchmaschinen im Web

Allgemeine Suchmaschinen (Google, Altavista, Fireball, Ask Jeeves)

Bei dem rasanten Wachstum des Internets und insbesondere des Webs³ zeigte sich bald die Notwendigkeit, die bisher nie gekannte Fülle an öffentlich zugänglichen Informationen automatisch zu erfassen. Deshalb wurden von mehreren Firmen oder Organisationen leistungsfähige Rechner eingerichtet, die regelmäßig das Web absuchen – inzwischen auch Anonymous-FTP-Server und News-Archive – und ihre

³Das Web wurde Anfang der 90er Jahre von TIMOTHY BERNERS-LEE und ROBERT CAILLIAU am CERN in Genf entwickelt. Der erste Webserver in den USA ging 1991 am Stanford Linear Accelerator Center in Betrieb, der erste deutsche Webserver 1992 am DESY in Hamburg.

Funde in Datenbanken eintragen. Die zugehörigen Suchmaschinen (E: search engine, F: moteur de recherche) beantworten Fragen der Benutzer aus den Datenbanken. Man schätzt, dass heute etwa ein Drittel aller Webseiten von Suchmaschinen erfasst wird, Tendenz abnehmend. Ein Betriebsgeheimnis der Suchmaschinen ist die Rangfolge, das Ranking, in der sie die Treffer bringen. Neben Selbstverständlichkeiten spielen raffinierte Algorithmen und gelegentlich Geld eine Rolle. Es gibt unlautere Wege, der eigenen Webseite zu einem vorderen Rang zu verhelfen – bis die Betreiber der Suchmaschinen dahinter kommen. Erfahrungsgemäß beachten Leser nur die vordersten Suchergebnisse.



Abbildung 4.5. Screenshot der Web-Suchmaschine Google Deutschland

Zur gegenwärtig populärsten Suchmaschine hat sich Google entwickelt, siehe Abbildung 4.5, deren deutscher Ableger unter <http://www.google.de/> zu erreichen ist, ausländische Kollegen mit den entsprechenden Länderkürzeln (fr, it, no, ru ...) anstelle von de. Warum Google Sie glücklich machen kann, lesen Sie am besten bei Google selbst nach. Der im Süden des deutschen Sprachraums bekannte *Googlehupf* (F: Kougelhopf, Kouglof) hat nichts mit der Suchmaschine zu tun, sondern ist älteren Ursprungs, siehe Wikipedia; Rezepte unter anderem bei:

- <http://ochsenkopf.physik.uni-konstanz.de/~andi/rezepte/node39.html>.
- <http://www.ruhr-uni.bochum.de/~schaelcz/kochfreunde/bsuserkochbuch/kochbuch.htm>
- <http://pl.physik.tu-berlin.de/gtroups/pg242/rezepte.html>
- <http://www.kernchemie.uni-mainz.de/kueche/schinken.htmlx>
- <http://www.bufa-weinheim.de/129.html>

- <http://www.netzkoch.de/elsass2.htm>
- http://www.sos-gateaux.net/sos_gateaux_recette_kouglof.php
- <http://www.lionsclub-eupen.org/back/rezepte/gugelhupf.htm>
- <http://www.mamas-rezepte.de/>
- http://www.schweizer-kochrezepte.ch/rezepte/schweiz/gugelhupf_geruehrt.html
- <http://www.recipesource.com/ethnic/europe/austrian/>

und natürlich auf dem berühmtem Rezepteserver der UNIX-Arbeitsgemeinschaft an der Universität Kaiserslautern <http://kochbuch.unix-ag.uni-kl.de/>. Die Suchmaschine Fireball lieferte 2400 deutschsprachige Treffer für das Begriffs-paar *Gugelhupf Rezept*. Das reicht für sechseinhalb Jahre, wenn man täglich einen backt.



Abbildung 4.6. Screenshot der Web-Suchmaschine Altavista

Altavista (<http://www.altavista.com/>) sucht seit 1995 im Web und hat den ersten Webindex zusammengestellt. Der deutsche Vertreter hat den URL <http://de.altavista.com/>, einen Screenshot zeigt Abbildung 4.6. Ursprünglich als Demonstrationsobjekt des heute nicht mehr existierenden Computerherstellers Digital Equipment (DEC) aufgebaut, gehört Altavista mittlerweile einem anderen Netzdienstleister.

Fireball (<http://www.fireball.de/>) ist der Suchindex von Lycos Europe für deutschsprachige Webseiten. Die Maschine bietet eine gut verständliche Hilfe sowie eine Textversion des Suchfensters an und macht einen aufgeräumten Eindruck, siehe Abbildung 4.7. Als Ergänzung zum automatisch erstellten Index bringt Fireball einen von Allesklar gepflegten Webkatalog mit, siehe nächster Abschnitt.



Abbildung 4.7. Screenshot der Web-Suchmaschine Fireball

Die Website Ask Jeeves mit dem URL <http://ask.com/> und einem Ableger unter <http://ask.co.uk/> bietet mehr als eine Suchmaschine, aber worin das Mehr besteht, ist schwierig zu ergründen. Benutzt man sie einfach zum Suchen und füttert sie mit den Wörtern *debian* und *alex*, liefert sie 80.700 Treffer. Nimmt man noch *ganten* hinzu, verringert sich die Anzahl auf 177. Das Suchwort *ratatouille* erbringt 64.200 Links; google.de erreicht zum Vergleich 327.000. Das sind Zahlen jenseits von Sinn und Unsinn. Die Maschine erlaubt die Eingabe von Kategorien, zum Beispiel *Restaurants* oder *Shopping*, und Orten, kennt aber offenbar nur Nordamerika. Sucht man nach Restaurants mit dem Namen *Brasserie* in Paris, werden 13 Lokale in Kalifornien aufgeführt, zum Teil mit Stadtplan und Bewertungen (review) durch Kunden. Auch auf Fragen in natürlichen Sprachen erhält man Antwort, auf deutsche Fragen in Deutsch, auf schwedische Fragen in Schwedisch, auf lateinische Fragen in Latein usw. (ohne eine Sprache anzuklicken, versteht sich).

Ask Jeeves wurde 1996 in Kalifornien gegründet und *provides consumers and advertisers with world-class information retrieval products across a diverse portfolio of Web sites, portals and desktop search applications*, zu übersetzen mit Gemischtwarenladen oder Internet-Butler, siehe den Screenshot 4.8. Die Maschine gilt als die zweitgrößte nach Google. Man kann sich mit Email-Anschrift und Passwort anmelden; dann übernimmt die Site auch noch das Speichern der Suchergebnisse zwecks wiederholtem Zugriff. Die Firma hat *Excite Europa* übernommen und ist selbst von der Firma *Interactive Corporation* aufgekauft worden, Business as usual. Der Butler REGINALD JEEVES ist eine Figur aus den Novellen von PELHAM GRENVILLE WODEHOUSE (PGW).

Die Kindersuchmaschine von T-Online ist keine Maschine, um seine Kinder zu suchen (wünscht man sich manchmal), sondern eine Suchmaschine zum Gebrauch durch Kinder, um einem dringenden Bedarf abzuhelpen. Weitere Suchma-



Abbildung 4.8. Screenshot der Suchmaschine Ask Jeeves, United Kingdom

schinen findet man am besten online in der Suchfibel (<http://http://www.suchfibel.de/>).

Kataloge (vlib, Yahoo, Allesklar, dmoz)

Ein Katalog unterscheidet sich von den vorgenannten, automatisch arbeitenden Suchmaschinen dadurch, dass der Datenbestand redaktionell bearbeitet wird. Das kostet Zeit. Eine Suche in einem Katalog liefert daher weniger und weniger aktuelle Treffer, aber oft eine bessere Qualität.

Unter <http://vlib.org/> findet sich *The WWW Virtual Library*, eine nach Sachgebieten gegliederte, weltweit verteilte englischsprachige Sammlung von Informationsquellen, vorwiegend aus dem Bereich von Forschung und Lehre. Beispielsweise wird die Seite für *Engineering, Mathematics, and Computing* von EEVL (ursprünglich *Edinburgh Engineering Virtual Library*) im UK betreut, die Seite für Klassische Musik von der *Georgetown Preparatory School*, einer Jesuitenschule, in den USA. Deutsche Datenquellen aller Sachgebiete werden in der Universität Karlsruhe unter <http://www.uni-karlsruhe.de/Outerspace/VirtualLibrary/> zusammengefasst. Von dort gelangt man unter dem Sachgebiet *Geschichte* zur Universität Erlangen und so fort.

Ein bekannter Webkatalog ist Yahoo (<http://www.yahoo.com/> bzw. <http://de.yahoo.com/>). Die Sachgebiete und die Auswahl sind weniger akademisch als in der Virtual Library. Die Suche nach BEETHOVENS Komposition *Für Elise* bei [de.yahoo](http://de.yahoo.com/) erbrachte 170.000 Treffer (weder nachgezählt noch überprüft). Einen optischen Eindruck vermittelt der Screenshot Abbildung 4.9.

Als größten deutschen Webkatalog bezeichnet sich Allesklar mit dem URL <http://www.allesklar.de/> und Filialen in der Schweiz und Österreich, siehe Abbildung 4.10. Eine unspezifische Suche nach BEETHOVENS *Elise* zeitigte



Abbildung 4.9. Screenshot des Web-Kataloges Yahoo Deutschland

20 Treffer, größtenteils brauchbar. Eine Besonderheit: man kann bei Allesklar einen Suchbegriff mit einem Ort verbinden, beispielsweise nach *Debian* in *Bremen* suchen. Es ist zu vermuten, dass Allesklar bei Informationsquellen aus dem deutschsprachigen Raum stark ist.

Das Open Directory Project (ODP) ist der Webkatalog der Open Source Gemeinschaft und unter <http://dmoz.org/> oder <http://dmoz.de/> zu erreichen. Es hat Verbindungen zu Netscape und erinnert an die Wikipedia und an Debian. Eine Suche nach *Beethoven* förderte immerhin 200 Treffer zu Tage, *Debian* brachte es auf 36 in 5 Kategorien. Der deutschsprachige Katalog <http://www.metafinder.de/> basiert auf dem Open Directory.

Ein Problem solcher Sammlungen ist die Linkfäule (E: link rot): ein zum Zeitpunkt des Eintrags gültiger Link kann ein Jahr später ins Leere führen. Dokumente wechseln ihren Ort im Webspace oder verschwinden ganz aus dem Web. Die Pflege einer Sammlung lässt sich nicht ganz automatisieren. Im Grunde bräuchte man statt URLs, die einen Ort im Web bezeichnen, URIs, die ein Dokument unabhängig vom Ort kennzeichnen und finden lassen. So weit ist das Netz noch nicht.

Die Stärke der Kataloge liegt bei oft nachgefragten Begriffen, weil hier die Redaktion eine Vorauswahl der Ergebnisse trifft. Bei seltenen Suchbegriffen macht sich die umfangreichere Datengrundlage der Automaten bemerkbar. Aus diesem Grund arbeiten oft – unbemerkt vom Benutzer – ein Katalog und ein Automat zusammen. Man braucht sich also nicht den Kopf darüber zu zerbrechen, ob man bei einer konkreten Frage besser einen Katalog oder einen automatischen Index anspricht.

Fachspezifische Suchmaschinen

Neben den allgemeinen Suchmaschinen, die alle Sachgebiete abdecken, finden sich fachspezifische Suchmaschinen. Die erste Hürde ist das Entdecken dieser meist kleinen Maschinen. Das bereits erwähnte Suchlexikon ist ein Einstieg. Auf der Web-



Abbildung 4.10. Screenshot des Web-Kataloges Allesklar

seite <http://www.fh-augsburg.de/informatik/projekte/mebib/www/spezielsuchdienste.html> werden Spezialsuchdienste aus aller Welt nach Themenbereichen geordnet aufgeführt. Hier einige selbst recherchierte Beispiele, ohne Wertung:

- Medizin: <http://www.dr-antonius.de/>
- Recht: <http://www.jura-lotse.de/>
- Geschichte: <http://www.weltchronik.de/>
- Musik: <http://www.allmusic.com/>
- Mathematik: <http://www.math-net.de/>
- Physik: <http://findemaschine.pro-physik.de/>
- Finanzen: <http://www.finanz-links.de/>
- Latein: <http://www.uky.edu/ArtsSciences/Classics/>

Die Grenzen zwischen Linksammlungen interessierter Amateure oder berufsmäßiger Fachleute und ausgewachsenen Suchmaschinen von Instituten oder Verbänden sind fließend.

Ausländische Suchmaschinen

Geht es um eine das Ausland betreffende Frage, bringt eine ausländische Suchmaschine oft Ergebnisse, die die internationalen Suchmaschinen übersehen. Kenntnisse der ausländischen Sprache sind hilfreich bis notwendig. Wie kommt man an eine solche Maschine? Zum einen kann man bei den deutschen großen Maschinen wie Yahoo oder Google das *de* im URL durch den entsprechenden fremden Landeskennner ersetzen, beispielsweise durch *fr* oder *it*. Zum anderen finden sich unter <http://www.ku-eichstaett.de/WWF/Suche/sdi.htm> Informatio-

nen über viele Länder, darunter nationale Suchmaschinen. Wieder einige Beispiele, ohne Wertung:

- Polen: <http://www.wp.pl/>
- Schweden: <http://katalogen.sunet.se/search.html>
- Schweiz: <http://www.search.ch/>
- Türkei: <http://web.bilkent.edu.tr/inet-turkey/>
- Venezuela: <http://buscador.infoguia.net/>
- Mittel- und Südamerika: <http://www.elsitio.com/>

Suche nach bestimmten Arten von Informationen

Auf die Suche nach bestimmten Arten von Informationen wie Bücher, Fotos, Filme oder Sound-Dateien haben sich ebenfalls einige Maschinen spezialisiert. Dabei geht es nicht nur um Unterhaltung, sondern auch um beruflich veranlasste Suchen beispielsweise von Bibliotheken oder Presseagenturen. Einige Tipps für den Einstieg:

- <http://www.bsz-bw.de/>, das Bibliotheksservice-Zentrum Baden-Württemberg,
- <http://www.findmybook.de/>, eine übergeordnete Suchmaschine für neue oder gebrauchte Bücher,
- <http://www.zvab.com/>, das Zentrale Verzeichnis Antiquarischer Bücher,
- <http://www.digimania.de/>, die Online-Community der digitalen Fotografie,
- <http://www.photosearch.de/> vom Bundesverband der Pressebild-Agenturen und Bildarchive,
- <http://www.fotofinder.net/>, eine Fotosuchmaschine in Berlin,
- <http://www.ifilm.com/>, eine nordamerikanische Filmsuchmaschine, in Seattle,
- <http://www.musicline.de/de/melodiesuche>.

Letztere Maschine ist insofern interessant, als sie versucht, eine ihr vorgepiffene Melodie zu erkennen. Dieselbe Möglichkeit bietet auch die Musipedia (<http://www.musipedia.org/>).

Meta-Suchmaschinen (MetaGer, Metaspinner, MetaXplorer, Metacrawler)

Bei schwierigen oder umfangreichen Suchen liegt es nahe, mehrere Suchmaschinen zu beauftragen. Nun verlangt jede Suchmaschine eine etwas andere Syntax, außerdem hat man unter den Treffern mit Dubletten zu rechnen. Diesen Unannehmlichkeiten helfen Meta-Suchmaschinen ab. Eine Meta-Suchmaschine nimmt eine Anfrage entgegen, verteilt sie auf ein Dutzend oder mehr echte Suchmaschinen und fasst die Ergebnisse zusammen; sie hält also selbst keine Daten. Vier solcher Meta-Suchmaschinen sind:

- MetaGer in der Universität Hannover <http://metager.de/>,
- Metaspinner <http://www.metaspinner.de/>,

- MetaXplorer von der Universitätsbibliothek Karlsruhe <http://www.ubka.uni-karlsruhe.de/suchmaschinen/metaxa/> mit ein paar naheliegenden Spezialitäten,
- Metacrawler, entwickelt 1994 in der University of Washington <http://metacrawler.com/>.

Nachteilig ist, dass die Syntax einer Meta-Suchmaschine nur aus dem gemeinsamen Nenner der zugrunde liegenden Suchmaschinen bestehen kann. Ausgeklügelte Abfragen reicht sie nicht weiter. Ferner sind die Meta-Suchmaschinen teilweise den echten Suchmaschinen ein Dorn im Auge, weil sie die einträgliche Werbung aussondern. Deshalb verweigern manche großen Suchmaschinen die Zusammenarbeit. Je nach Arbeitsgebiet des Fragestellers bleiben aber genügend brauchbare Treffer übrig. Meist will man ja nicht möglichst viele, sondern die besten Treffer finden. Es gibt auch fachbezogene Meta-Suchmaschinen wie <http://www.meta-jur.de/> oder <http://www.medivista.de/>.

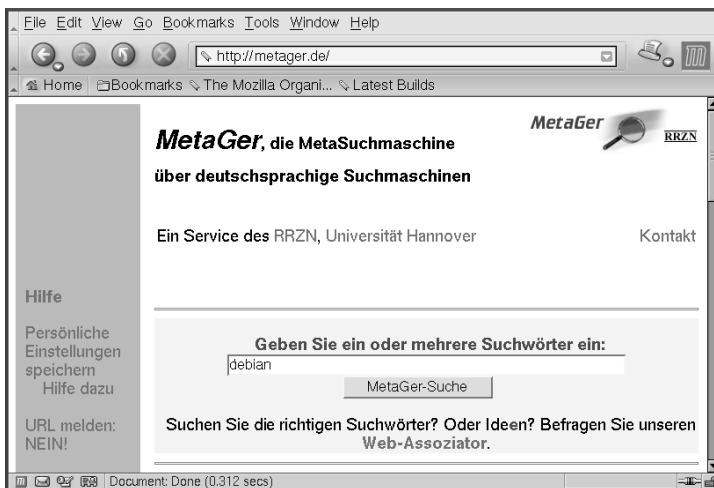


Abbildung 4.11. Screenshot der deutschen Meta-Suchmaschine MetaGer

Der MetaGer-Bildschirm macht von grafischen Elementen Gebrauch, siehe Abbildung 4.11. Für die Screenreader oder Braillezeilen blinder Benutzer steht daher eine angepasste Ausführung unter dem Namen *Easy MetaGer* und dem URL <http://www.c-lab.de/~wegge/mgb.html> zur Verfügung. MetaGer bietet ferner eine Überprüfung der Links, zahlreiche Optionen sowie zu jeder Ergebnisliste einen Spruch an, über den man nachdenken kann, während die Treffer von den einzelnen Suchmaschinen eintrudeln und aufbereitet werden.

Die Entwicklung geht wohl dahin, dass die meisten Benutzer das Internet nur noch über eine Suchmaschine betreten. Die Suchmaschinen werden lernen, sich auf die Bedürfnisse des Benutzers oder das, was sie dafür halten, einzustellen. Das ist einfache Statistik. Dazu ein Web-Browser, der auch Email und News erledigt und

dezent auf günstige Einkaufsmöglichkeiten hinweist, dann ist das Rundum-Sorglos-Paket fertig. Der mündige Benutzer muss aufpassen, dass er nicht unter dem Vorwand der Benutzerfreundlichkeit und des Komforts schleichend entmündigt wird.

Textverarbeitung

Hier wird (fast) alles erklärt, was man braucht, um einen ordentlichen Text zu erstellen. Anspruchsvolle Textverarbeitung ist eine Stärke von Linux/UNIX und Mitarbeitern.

5.1 Grundbegriffe

5.1.1 Zeichensätze – von ASCII bis Unicode

Wenn es um Texte geht, muss man sich zuerst mit dem Problem der Zeichensätze herumschlagen. Das hat nichts mit Linux/UNIX zu tun, sondern gilt für alle Betriebssysteme.

Der Rechner kennt nur Bits und Bytes (Oktetts, 8 Bits). Die Bedeutung erhalten die Bits durch die Programme. Ob eine Bitfolge in der Menschenwelt eine Zahl, ein Wort oder einen Schnörkel darstellt, entscheidet die Software. Um mit Texten zu arbeiten, muss ein Zeichensatz (E: character set, F: jeu de caractères) vereinbart werden. Dieser besteht aus einer zunächst ungeordneten Menge von Zeichen, auch Répertoire oder Zeichenvorrat genannt, die nur besagt, welche Zeichen bekannt sind. Zu dem Zeichenvorrat der europäischen Sprachen gehören:

- Kleine und große Buchstaben, auch mit Akzenten (Diakritika) usw.,
- Ziffern,
- Satzzeichen,
- Symbole: aus der Mathematik, Euro-Symbol, Klammeraffe (E: commercial at, F: arobase), et-Zeichen,
- Leerzeichen (Zwischenraum, Space), Tabulator (so genannte Whitespaces),
- Steuerzeichen wie Zeilenwechsel (Line Feed), Seitenwechsel (Form Feed), Backspace.

In einem zweiten Schritt wird die Menge geordnet, jedem Zeichen wird eine Position (E: code position, code point, Unicode scalar value) zugewiesen. Nahe liegend ist eine mit null beginnende Nummerierung. Die geordnete Menge ist der Zeichensatz.

In der Regel bekommt jedes Zeichen auch noch einen Namen, hier einige Zeichen aus dem Zeichensatz Latin-1:

Nr. 008	bs	Backspace (Rückschritt)
Nr. 055	7	Digit seven (Ziffer sieben)
Nr. 104	h	Latin small letter h (lateinischer Kleinbuchstabe h)
Nr. 123	{	Left curly bracket (linke geschweifte Klammer)
Nr. 220	Ü	Latin capital letter u with diaeresis (lateinischer Großbuchstabe u-Umlaut)

Bekannt ist das Internationale Alphabet Nr. 5 (IA5), auf dem ASCII aufsetzt. Wir wissen aber noch nicht, wie die Zeichen im Rechner und auf dem Bildschirm oder auf Papier dargestellt werden. Auch die Aussprache der Zeichen ist noch zu vereinbaren und nicht einheitlich.

Die Zeichencodierung (E: character encoding) legt fest, wie eine Folge von Zeichen in eine Folge von Bits oder Bytes umzuwandeln ist, und zurück. Der Zeichensatz ist von seiner Codierung zu unterscheiden. Im einfachsten Fall wird die vorzeichenlose, ganzzahlige Positionsnummer als Code für ein Zeichen genommen. Dann stellt sich die Frage, wie solche Zahlen im Rechner dargestellt werden. Der häufigste Fall ist eine Darstellung als duale Zahl mit sieben oder acht Bits. Wir kommen so zur Codetafel. In der linken Spalte stehen die Zeichen, in der rechten die Bits oder umgekehrt. Mit diesem Schritt ist die rechnerinterne Darstellung der Zeichen festgelegt, aber noch nicht deren Aussehen auf Schirm oder Papier, die Glyphen.

Zu Zeiten, als Bits noch knapp und teuer waren, haben die Nordamerikaner eine Codetafel aufgestellt, in der die ihnen bekannten Buchstaben, Ziffern und Satzzeichen zuzüglich einiger Steueranweisungen wie Zeilen- und Seitenvorschub mit sieben Bits dargestellt werden. Das war Sparsamkeit am falschen Platz, aber schon ein Fortschritt gegenüber dem Fernschreib-Alphabet, das nur fünf Bits (oder Löcher auf dem Lochstreifen) vorsah. Mit sieben Bits unterscheide ich $2^7 = 128$ Zeichen, nummeriert von 0 bis 127. Diese Codetafel ist unter dem Namen *American Standard Code for Information Interchange* (ASCII) weit verbreitet und seit 1963 in ISO 646 genormt, gleichzeitig auch als CCITT-Empfehlung V.3. Um Missverständnisse auszuschließen, heißt sie ausführlich 7-bit-US-ASCII. Jeder Rechner kennt sie.

Die ersten 32 Zeichen der ASCII-Tafel dienen der Steuerung der Ausgabegeräte, es sind unsichtbare Zeichen. Ein Beispiel für Steuerzeichen ist das ASCII-Zeichen Nr. 12, Form Feed, das einen Drucker zum Einziehen eines Blattes Papier veranlasst. Auf der Tastatur werden sie entweder in Form ihrer Nummer oder mit gleichzeitig gedrückter <ctrl>-Taste erzeugt. Die Ziffern 0 bis 9 tragen die Nummern 48 bis 57, die Großbuchstaben die Nummern 65 bis 90. Die Kleinbuchstaben haben um 32 höhere Nummern als die zugehörigen Großbuchstaben, was das Umrechnen erleichtert. Der Rest sind Satzzeichen. Im Anhang ist die ASCII-Tafel wiedergegeben. Das Kommando:

```
joe@debian:~$ man ascii
```

bringt sie auf den Schirm.

Textausgabegeräte wie Bildschirme oder Drucker erhalten vom Rechner die ASCII-Nummer eines Zeichens und setzen diese mit Hilfe einer eingebauten Soft-

ware in das entsprechende Zeichen um. So wird beispielsweise die ASCII-Nr. 100 in den Buchstaben d umgesetzt. Die Ausgabe der Zahl 100 als Zeichenkette erfordert das Abschicken der ASCII-Nr. 49, 48, 48.

Die US-ASCII-Tafel enthält nicht die deutschen Umlaute und andere europäische Absonderlichkeiten. Es gibt einen Ausweg aus dieser Klemme, leider sogar mehrere. Bleibt man bei sieben Bits, muss man einige nicht unbedingt benötigte US-ASCII-Zeichen durch nationale Sonderzeichen ersetzen. Für deutsche Zeichen ist eine Ersetzung gemäß DIN 66003, siehe Anhang B.2 *German ASCII* auf Seite 441, üblich. German ASCII ist als Variante Nr. 21 in ISO 646 registriert. Für Frankreich oder Schweden lautet die Ersetzung anders, siehe die deutsche Wikipedia unter dem Begriff *ISO 646*. Im Ausgabegerät, das die Umsetzung der ASCII-Nummern in Zeichen vornimmt, liegt eine solche Ersatztafel. Deshalb kann ein entsprechend ausgestatteter Bildschirm oder Drucker dieselbe Textdatei einmal mit amerikanischen ASCII-Zeichen ausgeben, ein andermal mit deutschen ASCII-Zeichen. Im ersten Fall erscheinen geschweifte und eckige Klammern, im zweiten stattdessen Umlaute. Werden bei Ein- und Ausgabe unterschiedliche Codetafeln verwendet, gibt es Zeichensalat. Andersherum gesagt: Wenn ich einen Text ausgabe, muss ich die Codetafel der Eingabe kennen und verwenden.

Spendiert man ein Bit mehr, so lassen sich $2^8 = 256$ Zeichen darstellen. Das ist der bessere Weg. Hewlett-Packard hat die nationalen Sonderzeichen den Nummern 128 bis 255 zugeordnet und so den Zeichensatz ROMAN8 geschaffen, dessen untere Hälfte mit dem US-ASCII-Zeichensatz identisch ist. Das hat den Vorzug, dass reine US-ASCII-Texte genau so verarbeitet werden wie ROMAN8-Texte. Diese Codetafel hat sich nicht allgemein durchgesetzt.

Die Firma IBM hat schon früh bei größeren Anlagen (mainframes) den *Extended Binary Coded Decimal Interchange Code* (EBCDIC) mit acht Bits verwendet, der nirgends mit ASCII übereinstimmt. Hätte sich diese Codetafel statt ASCII durchgesetzt, wäre uns Europäern einige Mühe erspart geblieben. Eine globale Lösung wäre es aber auch nicht gewesen.

Bei ihren PCs schließlich wollte IBM außer nationalen Sonderzeichen auch einige Halbgrafikzeichen wie Mondgesichter, Herzchen, Noten und Linien unterbringen und schuf einen weiteren Zeichensatz namens IBM-PC, der in seinem Kern mit ASCII übereinstimmt, ansonsten aber weder mit EBCDIC noch mit Latin-*

Die internationale Normen-Organisation ISO hat mehrere 8-bit-Zeichensätze festgelegt, von denen einer unter dem Namen *Latin-1* nach ISO-8859-1 Verbreitung gewonnen hat, vor allem in weltweiten Netzdiensten. Seine untere Hälfte ist mit US-ASCII identisch, die obere enthält die Sonderzeichen west- und mitteleuropäischer Sprachen. Der Aufruf:

```
joe@debian:~$ man iso_8859_1
```

zeigt die Tafel. Polnische und tschechische Sonderzeichen sind in *Latin-2* nach ISO 8859-2 enthalten, siehe Anhang *Latin-1* und *Latin-2* ab Seite 443. Die Zeichensätze Latin-1 bis 4 sind ebenfalls im Standard ECMA-94 beschrieben (ECMA = European Computer Manufacturers Association). Kyrillische Zeichen sind in ISO 8859-5, griechische in ISO 8859-7 festgelegt (logischerweise nicht als Latin-* be-

zeichnet; Latin-5 ist Türkisch nach ISO 8859-9). ISO 8859-15 oder Latin-9 entspricht Latin-1 zuzüglich des Eurosymbols und weiterer Kleinigkeiten.

Die Latin-Zeichensätze enthalten außer dem gewohnten Zwischenraum- oder Leerzeichen (space) ein in Textverarbeitungen oft benötigtes Zeichen für einen Zwischenraum, bei dem kein Zeilenumbruch erfolgen darf (Latin-1 Nr. 160, no-break space). Im Satzsystem \LaTeX wird hierfür die Tilde verwendet, in HTML die Umschreibung (entity) ` `. Dieses Zeichen kommt beispielsweise zwischen Zahl und Maßeinheit oder zwischen den Initialen eines Namens vor.

Auch wenn die Ein- und Ausgabegeräte 8-bit-Zeichensätze kennen, ist noch nicht sicher, dass man die Sonderzeichen benutzen darf. Die Programme müssen ebenfalls mitspielen. Der hergebrachte vi-Editor, die curses-Bibliothek für Bildschirmfunktionen und einige Email-Programme verarbeiten nur 7-bit-Zeichen. Erst jüngere Versionen von Linux/UNIX mit Native Language Support unterstützen 8-bit-Zeichensätze voll. Textverarbeitende Software, die 8-bit-Zeichensätze verträgt, wird als *8-bit-clean* bezeichnet. Bei Textübertragungen zwischen Rechnern (insbesondere Email) ist Misstrauen angebracht. Die Konsequenz heißt in kritischen Fällen Beschränkung auf 7-bit-US-ASCII, das funktioniert immer und überall. Das Problem verliert an Bedeutung.

Wenn wir das Abendland verlassen, stellen wir fest, dass es drei Gruppen von Schriften gibt, von Exoten wie den Quipus der Inkas abgesehen:

- alphabetische Schriften (E: alphabetic scripts): Lateinisch, Griechisch, Kyrlisch, Arabisch, Hebräisch, mit jeweils wenigen hundert Zeichen,
- Silbenschriften (E: syllabic scripts): Devanagari, Hiragana, Hangul, Thai, mit jeweils wenigen tausend Zeichen,
- Begriffsschriften (E: ideographic scripts): Chinesisch, Japanisch, Koreanisch, Vietnamesisch (CJKV), mit etwa 50000 Zeichen im Chinesischen.

Mehrere Einrichtungen bemühen sich, diese Vielfalt unter einen Hut zu bringen:

- die International Standardization Organization (ISO), die Internationale Normen-Organisation mit Sitz in Genf, an der auch das Deutsche Institut für Normung (DIN) beteiligt ist, <http://www.iso.org/> bzw. <http://www2.din.de/>,
- die International Electrotechnical Commission (IEC), an der die Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im DIN + VDE beteiligt ist, <http://www.iec.ch/> bzw. <http://www.dke.de/>,
- das Unicode Consortium, ein Zusammenschluss führender Hersteller und Institutionen sowie von Einzelpersonen, <http://www.unicode.org/>,
- das World Wide Web Consortium (W3C), <http://www.w3.org/International/>,
- das Internet in Form einiger Requests For Comments.

Eine große, gemeinsame Richtung ist erkennbar, wenn auch im Einzelnen die unterschiedlichen Bezeichnungen und Zuständigkeiten verwirren.

ISO und IEC haben mit der Norm ISO/IEC 10646 einen Zeichensatz (Zeichen, Namen und Positionsnummer) festgelegt, der die Zeichen aller historischen, gegen-

Tabelle 5.1. Darstellung des Zeichens A in verschiedenen Standards

Standard	dual	dez
ASCII (7 bit)	1000001	65
ISO 8859-1 (8 bit), UTF-8	01000001	65
UTF-16, UCS-2 (16 bit)	00000000 01000001	65
UTF-32, UCS-4 (32 bit)	00000000 00000000 00000000 01000001	65

wärtigen und zukünftigen Schriftsprachen der Erde berücksichtigt. Dieser Zeichensatz heißt *Universal Character Set* (UCS), ausführlich *Universal Multiple Octet Coded Character Set*. Unabhängig von der ISO arbeitet das Unicode Consortium seit 1991 an dem gleichen Ziel und hat einen Zeichensatz namens *Unicode* geschaffen. Glücklicherweise haben beide Einrichtungen zusammengefunden, Im Jahr 2003 erschien Unicode in Version 4, die der Norm ISO/IEC 10646:2003 folgt. Der einheitliche Zeichensatz reicht für mehr als eine Million Zeichen. Die Standards von ISO/IEC und Unicode unterscheiden sich in ergänzenden Abschnitten, die teilweise verschiedene Themen aufgreifen. Geht es nur um den Zeichensatz, bedeuten Unicode, UCS und ISO/IEC 10646 dasselbe. In den untersten 256 Zeichen ist Unicode mit Latin-1 identisch, in den untersten 128 Zeichen folglich mit US-ASCII.

Die nächste Frage betrifft die Umsetzung der Zeichen oder ihrer Nummern in Bits und Bytes. Nahe liegend ist eine Darstellung durch 2 oder 4 Bytes, wobei das höchstwertige Byte zuerst übertragen wird (bigendian), wenn nicht anders bestimmt (UTF = Unicode Transformation Format):

- UCS-4 (UTF-32) verwendet eine feste Anzahl von 32 Bits (4 Bytes) und codiert damit bis zu $2^{32} = 4294967296$ Zeichen (Positionen). Der Bedarf an Speicherplatz ist maximal.
- UCS-2 (UTF-16) ist ein Kompromiss, der für die häufigeren Zeichen 2 Bytes (65 536 Positionen) und für die selteneren ein Paar von UCS-2-Codes (4 Bytes) verwendet. Von Java und MS Windows benutzt.

Die Linux/UNIX-Familie bekommt damit ein Problem. In UCS-2- oder UCS-4-codierten Zeichenketten treten häufig Nullbytes (00000000) auf, siehe Tabelle 5.1. Diese haben in der Programmiersprache C und damit auch in Linux/UNIX eine besondere Bedeutung: Sie beenden eine Zeichenkette. Außerdem kennen die herkömmlichen Linux/UNIX-Werkzeuge nur Zeichen mit der Länge von einem Byte. Wir brauchen eine andere Codierung des Zeichensatzes.

Die Lösung des Linux/UNIX-Problems – ausgedacht von KENNETH THOMPSON mit Unterstützung durch ROB PIKE, beide UNIX-Urgestein – besteht in UTF-8, das 1 bis 6 Bytes verwendet und alle UCS-Zeichen codiert. Der Bedarf an Speicherplatz ist minimal, dafür muss mehr gerechnet werden. UTF-8 codiert nach folgender Vorschrift:

- Die US-ASCII-Zeichen Nr. 0 bis 127 (hex 00 bis 7F) bleiben, wie sie sind: ein Byte, höchstwertiges Bit null.

- Alle Zeichen >127 werden als Folgen mehrerer Bytes codiert, wobei in jedem Byte das höchstwertige Bit gesetzt ist. Somit tritt kein ASCII-Byte als Teil eines Zeichens >127 auf.
- Das erste Byte einer Bytefolge, die ein Zeichen >127 darstellt, beginnt mit so vielen Einsen, wie die Folge Bytes umfasst, gefolgt von einer Null, und liegt damit im Bereich von 192 bis 253 (hex C0 bis FD). Die restlichen Bits tragen zur Codierung der Positionsnummer bei.
- Die weiteren Bytes der Folge beginnen stets mit 10, die restlichen Bits tragen zur Codierung der Positionsnummer bei. Die Bytes liegen damit im Bereich von 128 bis 191 (hex 80 bis BF).
- Die Bytes 254 und 255 (hex FE und FF) werden nicht verwendet.
- Jedes Zeichen wird durch seine kürzest mögliche Bytefolge codiert, die Codierung ist umkehrbar eindeutig.

Die Folgen können bis zu sechs Bytes lang sein; die 65000 häufiger vorkommenden Zeichen erfordern jedoch nur bis zu drei Bytes. UTF-8 ist im Web verbreitet und gewinnt in der Linux/UNIX-Welt an Boden. Ein Beispiel: Unser Eszett (ß) hat im Latin-1-Zeichensatz die Nr. 223 und damit auch in UCS oder Unicode. Der Aufruf:

```
joe@debian:~$ unicode -d 223
```

liefert als Ergebnis:

```
U+00DF LATIN SMALL LETTER SHARP S
UTF-8: c3 9f   UTF-16BE: 00df   Decimal: &#223;
SS (SS)
Uppercase: U+00DF
Category: Ll (Letter, Lowercase)
Bidi: L (Left-to-Right)
```

was Folgendes bedeutet:

- Das Zeichen mit der hexadezimal ausgedrückten Position U+00DF (dezimal 223) in Unicode trägt den Namen *LATIN SMALL LETTER SHARP S*, wird in UTF-8 durch hexadezimal *c3 9f*, in bigendian UTF-16 durch hexadezimal *00df* dargestellt.
- Der zugehörige Großbuchstabe (den es nicht gibt) ist identisch mit dem Zeichen selbst.
- Das Zeichen fällt in die Kategorie der Kleinbuchstaben.
- Das Zeichen gehört zu den Schriften, die von links nach rechts geschrieben werden.

Probieren Sie obige Eingabe mit den Nr. 167, 188, 216, 555, 1648, 4711 und 12345 aus. Die UTF-8-Darstellung des Eszettts liest sich dual:

```
c3 9f = 11000011 10011111 = 110 00011 10 011111
```

Von links gelesen ergibt die Anzahl der Einsen bis zur ersten Null die Anzahl der zum Zeichen gehörenden Bytes, hier zwei. Vom ersten Byte sind dann die ab der

dritten Eins stehenden Stellen zu nehmen, hier 11. Vom zweiten Byte sind die beiden führenden Bits zu streichen, der Rest ist an die 11 vom ersten Byte anzuhängen, sodass sich $11011111 = 223$ dezimal ergibt. Bei solchen Rechnungen ist der Zahlenbasiskonverter `gbase` hilfreich.

Wir haben nun einen globalen Zeichensatz (UCS, Unicode) samt einer Codierung (UTF-8), die ins Konzept von Linux/UNIX passt. Jetzt muss noch die Software mitspielen. Werkzeuge wie `cat` lesen Bytes ein und geben Bytes aus, ohne auf ihre Bedeutung zu achten. Die brauchen nichts von UTF-8 zu verstehen. Aber überall dort, wo die Länge von Zeichenketten (Strings) gefragt ist, müssen die Funktionen und Programme jetzt aufpassen. Die alte Gleichheit `Byte = Zeichen = Schritt` auf dem Bildschirm gilt nicht mehr. Ein Zeichen kann aus mehreren Bytes bestehen und – bei den CJKV-Sprachen – doppelt breit sein. Bestimmte Zeichen wie selbständige Akzente veranlassen keinen Schritt auf dem Bildschirm, haben also die Breite null. Betroffen sind Programme wie `ls`, `wc` und die Texteditoren. Da der Benutzer keinen Einfluss auf seine Werkzeuge hat (und sich das vorliegende Buch nicht an Programmierer wendet), bleibt nur die Hoffnung, dass die Textwerkzeuge unter Debian GNU/Linux zunehmend UTF-8 beherrschen. Auf der Webseite zum Debian-Paket `vim` aus *sarge* ist jedenfalls von Unicode die Rede; der aktuelle Emacs murmelt etwas von *some Unicode support*. Es gibt auch Editoren, die speziell für UTF-8 und bidirektionale Schreibweise entwickelt worden sind, siehe Abschnitt 5.8 *Editoren für fremde Schriften* auf Seite 202.

UTF-7 nach RFC 1642 ist ein Zugeständnis an die Email-Standards, die verlangen, dass Email nur mit 7 Bits codierte Zeichen enthalten darf. Die US-ASCII-Zeichen bleiben unverändert, Zeichen auf höheren Positionen werden durch ein Plus- und ein Minuszeichen eingerahmt nach UTF-16 und Base64 codiert. Stirbt aus.

Zusammenfassend noch einmal die drei in der Linux/UNIX-Welt gebräuchlichsten Zeichensätze und Codierungen:

- Alt, bewährt, verbreitet, aber beschränkt in seinen Möglichkeiten: 7-bit-ASCII nach ISO 646,
- verbreitet, aber nur für das Abendland ausreichend: ISO/IEC 8859 mit 8 Bits, daraus für West- und Mitteleuropa ISO-8859-1, auch Latin-1 oder IBM Codepage 819 genannt.
- UCS bzw. Unicode mit der Codierung UTF-8, nach ISO/IEC 10646 und Unicode Version 4, ein bis sechs Bytes, zukunftssicher und zunehmend unterstützt.

Wer sich eingehender mit Unicode und UTF-8 beschäftigen möchte, findet einen guten Einstieg bei:

- <http://www.unicode.org/Public/> sowie <http://www.unicode.org/faq/>,
- RFC 3629 *UTF-8, a transformation format of ISO 10646*.
- BRUNO HAIBLE, *The Unicode HOWTO*, <http://www.tldp.org/HOWTO/Unicode-HOWTO.html>,
- MARKUS KUHN, *UTF-8 and Unicode FAQ for Unix/Linux*, <http://www.cl.ac.uk/~mgk25/unicode.html>,

- ROMAN CZYBORRA, *Unicode Transformation Formats: UTF-8 & Co.*, <http://czyborra.com/utf/>,
- ALAN WOOD, *Unicode and Multilingual Support in HTML, Fonts, Web Browsers and Other Applications*, <http://www.alanwood.net/unicode/>,

Die Standards gibt es bei den jeweiligen Organisationen, in der Regel nur gegen Entgelt, da sie sich zum Teil aus dem Verkauf finanzieren. In den Bibliotheken technischer Hochschulen kann man die Standards einsehen.

5.1.2 Fonts

Der Zeichensatz sagt, welche Zeichen bekannt sind, nicht wie sie aussehen. Ein Font (E: font, F: police) legt das Aussehen der Zeichen, die Glyphen, fest. Ursprünglich war ein Font der Inhalt eines Setzkastens. Verwirrend ist, dass einige einfache Font-Formate die Gestaltinformationen verschiedenen Zahlen (Positionen) zuordnen und so der Font darüber befindet, welches Zeichen für welche Zahl ausgegeben wird. Die einzige saubere Lösung ist die Trennung von Zeichensatz, Codierung und Font. Die Vielzahl der Fonts hat technische und künstlerische Gründe.

Bei einem anspruchsvollen Font werden nicht nur die einzelnen Zeichen dargestellt, sondern auch bestimmte Zeichenpaare oder -tripel (Ligaturen). JOHANNES GUTENBERG verwendete 190 Typen, und da war noch kein Klammeraffe und kein Eurosymbol dabei. Der Buchstabe *f* ist besonders kritisch. Erkennen Sie den Unterschied zwischen *hoffen* und *hofffähig* oder *Kaufläche* und *Kaufleute*? Die jeweils ersten Glieder der Paare werden mit Ligatur geschrieben, die zweiten ohne. \LaTeX -Fonts kennen f-Ligaturen, wie man sieht.

Unter einer Schrift, Schriftfamilie oder Schriftart (E: typeface) wie Times Roman, New Century Schoolbook, Garamond, Bodoni, Helvetica, Futura, Univers, Schwabacher, Courier, OCR (optical character recognition) oder Schreibschriften versteht man einen stilistisch einheitlichen Satz von Fonts in verschiedenen Ausführungen. Die Schriftart soll zum Charakter und Zweck des Schriftstücks und zur Wiedergabetechnik passen. Die Times Roman ist beispielsweise ziemlich kompakt sowie leicht und schnell zu lesen (sie stammt aus dem Zeitungsdruck), während die New Century Schoolbook 10 % mehr Platz benötigt, dafür aber deutlicher lesbar ist, was für Leseanfänger und Sehschwache eine Rolle spielt. Die klassizistische Bodoni wirkt gehoben und ist die Lieblingsschrift von IBM. In einer Tageszeitung wäre sie fehl am Platz. Serifenlose Schriften wie die Helvetica eignen sich für Plakate, Overhead-Folien, Präsentationen, Beschriftungen von Geräten und kurze Texte. Die Computerzeitschrift c't ist mit der Nummer 6/2003 auf einen serifenlosen Font umgestiegen, um sich einen modernen Anstrich zu geben. Die Schriften liegen in verschiedenen Schriftschnitten (E: treatment) vor: mager, fett, breit, schmal, kursiv, dazu in verschiedenen Größen oder Schriftgraden (E: point size). Die Schriftweite, der Zeichenabstand (E: pitch), ist entweder fest wie bei einfachen Schreibmaschinen, beispielsweise 10 oder 12 Zeichen pro Zoll, oder von der Zeichenbreite abhängig wie bei den Proportionalschriften. Diese sind besser lesbar und sparen Platz, erfordern aber in Tabellen Aufwand. Ein vollständiger Satz von Buchstaben, Ziffern, Satz- und

Sonderzeichen einer Schrift, eines Schnittes, eines Grades und gegebenenfalls einer Schriftweite wird heute *Font* genannt.

Im wesentlichen gibt es zwei Kategorien von Font-Formaten: Bitmap-Fonts (Raster-Fonts) und Vektor-Fonts. Bitmap-Fonts speichern die Gestalt eines Zeichens in einer Punktematrix. Der Vorteil besteht in der einfachen und schnellen Verarbeitung. Auf vielen Systemen existieren mehrere Bitmap-Fonts derselben Schrift, optimiert für die am häufigsten benötigten Schriftgrößen. Nachteilig ist die unbefriedigende Skalierbarkeit (Vergrößerung oder Verkleinerung). Das Problem ist das gleiche wie bei Grafiken.

Bessere Systeme verwenden daher Vektor-Fonts, die die Gestalt der Zeichen durch eine mathematische Beschreibung ihrer Umrisse festhalten. Vektor-Fonts lassen sich daher problemlos skalieren. Bei starken Maßstabsänderungen muss jedoch auch die Gestalt etwas verändert werden. Gute Vektor-Fonts speichern deshalb zusätzliche, beim Skalieren zu beachtende Informationen (hints) zu jedem Zeichen.

Die beiden wichtigsten Vektor-Font-Formate sind True Type (TT), hauptsächlich auf Macintoshs und unter MS Windows zu finden, und das von Adobe stammende PostScript-Type-1-Format (PS1), das auch von X11 dargestellt werden kann und daher unter Linux/UNIX verbreitet ist. True Type kam um 1990 heraus und war die Antwort von Apple und Microsoft auf Adobe. Im Jahr 1996 rauchten sich Adobe und Microsoft zusammen und schufen das Open Type Format als eine einheitliche Verpackung von PostScript- und Truetype-Fonts.

X11 hat ursprünglich nichts mit der Druckausgabe zu tun. Die Tatsache, dass ein Font unter X11 verfügbar ist, bedeutet noch nicht, dass er auch gedruckt werden kann. Einen gemeinsamen Nenner von X11 und der Druckerwelt stellt das Type-1-Format dar: Fonts dieses Formates können sowohl von X11 auf dem Bildschirm dargestellt als auch als Softfonts (in Dateien gespeicherte Fonts) in PostScript-Drucker geladen werden. Für den Privatanwender, der sich keinen PostScript-Drucker leistet, bietet sich der Weg an, den freien PostScript-Interpreter *Ghostscript* als Druckerfilter zu verwenden. Er wandelt PostScript-Daten in verschiedene Druckersteuersprachen wie PCL um.

Da die Papierformate länglich sind, spielt die Orientierung (E: orientation) eine Rolle. Das Hochformat wird englisch mit *portrait*, das Querformat mit *landscape* bezeichnet¹. Ferner trägt der Zeilenabstand oder Vorschub (E: line spacing) wesentlich zur Lesbarkeit bei. Weitere Gesichtspunkte zur Schrift und zur Gestaltung von Schriftstücken findet man in der im Anhang angegebenen Literatur und im Netz, zum Beispiel in der FAQ der Newsgruppe `comp.fonts`, auf Papier 260 Seiten, zusammengestellt von NORMAN WALSH (<http://www.nwalsh.com/comp.fonts/FAQ/>). Trinken Sie einen auf sein Wohl und denken Sie darüber nach, wie viel freiwillige und unentgeltliche Arbeit in den FAQs steckt. Es gibt auch ein *Font HOWTO*, genauer betitelt *Optimal Use of Fonts on Linux*, aktuell von 2005.

Jetzt wäre es an der Zeit, über die akustische Darstellung von Texten zu reden, was bei der Sprachsynthese eine Rolle spielt – siehe Abschnitt 9.7 *Sprachsynthese*,

¹Woraus man schließt, dass Engländer ein Flachland bewohnende Langschädler sind, während alpine Querköpfe die Bezeichnungen vermutlich andersherum gewählt hätten.

Screenreader auf Seite 347 – aber das verschieben wir aus naheliegenden Gründen in die Zukunft.

5.1.3 Dateiformate (txt, pdf, ps, rtf)

Nicht genug mit Zeichen, Codes und Fonts, ein Textdokument lässt sich auch noch auf mancherlei Weise in einer Datei abspeichern. Der Benutzer bemerkt das, wenn ihm ein Text in einem ungewohnten Format zuflattert. Gewöhnlicher Text in ASCII- oder Latin-Codierung ohne weitere Verzierungen wird oft in Dateien mit der Kennung `txt` gespeichert, was in der Linux/UNIX-Welt jedoch nicht vorgeschrieben ist.

Textdateien aus anderen Welten (DOS, Macintosh)

In Linux/UNIX-Textdateien wird der Zeilenwechsel (E: line break, F: saut de ligne) durch ein newline-Zeichen markiert, hinter dem das ASCII-Zeichen Nr. 10 (LF, Line feed) steckt, das auch durch die Tastenkombination `<ctrl>-<j>` eingegeben wird. In DOS-Textdateien wird ein Zeilenwechsel durch das Zeichenpaar Carriage return – Line feed (CR LF, ASCII Nr. 13 und 10, `<ctrl>-<m>` und `<ctrl>-<j>`) markiert, das Dateiende durch das ASCII-Zeichen Nr. 26 (`<ctrl>-<z>`). Auf Macs ist die dritte Möglichkeit verwirklicht, das Zeichen Carriage return (CR, ASCII Nr. 13, `<cr>`) allein veranlasst den Sprung an den Anfang der nächsten Zeile.

Auf einer Linux/UNIX-Maschine lassen sich die störenden Carriage returns (oktal 15) der DOS-Texte leicht durch folgenden Aufruf entfernen:

```
joe@debian:~$ tr -d '\015' < text.dos > text.unix
```

Die Texteditoren `vi`, `emacs` oder `sed` können das auch, ebenso ein einfaches C-Programm.

Wenn Ihr Text auf einem Bildschirm oder Drucker treppenförmig dargestellt wird – nach rechts fallend – erwartet das Gerät einen Text nach Art von DOS mit CR und LF, der Text enthält jedoch nach Art von Linux/UNIX nur LF als Zeilenende. In einigen Fällen lässt sich das Gerät entsprechend konfigurieren, auf jeden Fall kann man den Text entsprechend ergänzen. Wenn umgekehrt auf dem Bildschirm kein Text zu sehen ist, erwartet das Ausgabeprogramm einen Linux/UNIX-Text ohne CR, die Textdatei stammt jedoch aus der DOS-Welt mit CR und LF. Jede Zeile wird geschrieben und gleich wieder durch den Rücksprung an den Zeilenanfang gelöscht. Viele Linux/UNIX-Pager (`more`, `less`) berücksichtigen das und geben das CR nicht weiter. Auf Druckern kann sich dieses Missverständnis durch Verdoppelung des Zeilenabstands äußern. Kein Problem, nur lästig.

Neben diesen, dem Benutzer auffallenden Unterschieden gibt es auch viele Möglichkeiten der rechnerinternen Darstellung der Zeichen. Damit in einem Netz verschiedene Rechner Texte austauschen können, hat man sich im Internet – im RFC 854 Telnet – darauf geeinigt, im Netz die Zeichen gemäß Network Virtual Terminal ASCII (NVT ASCII) darzustellen, wenn nichts anderes vereinbart ist, und die

Umsetzung auf die lokale Darstellung dem Sender bzw. dem Empfänger zu überlassen. Dabei werden die 128 Zeichen des US-ASCII-Zeichensatzes durch ein Byte dargestellt, dessen oberstes Bit (most significant bit) null ist.

PostScript und Portable Document Format (PDF)

Außer dem eigentlichen Text enthält ein besseres Textdokument Formatierungsangaben und Grafik. Um ein derartig zusammengesetztes Dokument für die Ausgabe auf einem Drucker zu beschreiben, hat die Firma Adobe 1985 die Programmiersprache *PostScript* geschaffen, die auch als Seitenbeschreibungssprache bezeichnet wird. Die Entwicklung von Laserdruckern und PostScript hat sich gegenseitig beflügelt. Eine PostScript-Seite besteht aus:

- Linien,
- Text unterschiedlichen Aussehens,
- Bitmap-Grafiken.

Ihre Beschreibung in PostScript ist geräteunabhängig. Im Ausgabegerät (oder kurz davor) werden aus den PostScript-Daten gerätespezifische Daten errechnet. Deshalb kann dieselbe PostScript-Datei einmal einen Laserdrucker steuern, ein andermal einen Schneideplotter zum Ausschneiden eines Schriftzuges oder Firmenlogos aus einer Folie. Außer den Nutzdaten enthalten die Dateien nur Befehle, die aus dem US-ASCII-Zeichensatz zusammengesetzt sind. Es sind lesbare Textdateien ohne Zeilenstruktur. Der Anfang einer PostScript-Datei (Vorspann) sieht so aus:

```
%!PS-Adobe-2.0
%%Creator: dvips(k) 5.86e Copyright 2001 Radical Eye Software
%%Title: folien.dvi
%%Pages: 19
%%PageOrder: Ascend
%%BoundingBox: 0 0 596 842
%%EndComments
%DVIPSWebPage: (www.radicaleye.com)
%DVIPSCmdLine: /usr/bin/dvips -o folien.ps folien
%DVIPSPParameters: dpi=600, compressed
%DVIPSSource: TeX output 2004.07.24:1326
%%BeginProcSet: texc.pro
```

und die erste Seite so (eine Vortragsfolie):

```
%%Page: 0 1
0 0 bop 380 386 a Fu(P)-11 b(a)g(rtik)g(elme\031technik)703
834 y(\(Kr)1264 845 y(\177)1262 834 y(umelkunde\))356
1521 y Ft(W.)82 b(Alex,)h(Universit)2252 1530 y(\177)2254
1521 y(at)e(Ka)-7 b(rlsruhe)1266 2085 y(24.)83 b(Juli)e(2004)-97
5702 y Fs({)69 b(T)-17 b(yp)6 b(eset)67 b(b)-6 b(y)69
b(F)-6 b(oil)g Fr(T)1501 5747 y(E)1615 5702 y(X)68 b
Fs({)i(W.)f(Alex:)f(Kr)3048 5709 y(\177)3046 5702 y(umelkunde)p
eop
```

Wer sich gut mit PostScript auskennt, könnte mit einem Texteditor die Datei ändern. Bruchstücke des Textes erkennt auch ein ungeübter Leser. Dennoch ist das unmittelbare Schreiben oder Lesen von PostScript-Dateien die Ausnahme. Erzeugt werden die Dateien mittels entsprechender Treiberprogramme aus gewöhnlichen Texten oder Grafiken. Zum Lesen braucht man besondere Leseprogramme, siehe Abschnitt 5.2.3 *Lesen von PostScript-Dateien* auf Seite 170..

Encapsulated PostScript (EPS, EPSF) ist eine Untermenge von PostScript zur Beschreibung von Grafiken, die in andere Dokumente eingebunden oder zwischen verschiedenen Anwendungen ausgetauscht werden sollen. Eine EPS-Datei enthält immer genau ein Objekt (Bild, Seite) und am Anfang eine Angabe zur Größe des Objekts (bounding box). Hier der Anfang der EPS-Datei zu Abbildung 1.5, die von \LaTeX eingebunden wird:

```
%!PS-Adobe-3.0 EPSF-3.0
%%Creator: GIMP PostScript file plugin V 1.12 by Peter Kirchgessner
%%Title: /server1/wulf/TeX/debian2/bilder/debian.eps
%%CreationDate: Sat Dec 25 22:34:32 2004
%%DocumentData: Clean7Bit
%%LanguageLevel: 2
%%Pages: 1
%%BoundingBox: 14 14 115 138
```

Ein Grafikprogramm wie GIMP kann eine beliebige Grafik im EPS-Format exportieren, die dann von einer Textanwendung importiert wird.

Das *Portable Document Format* (PDF) von Adobe ist eine Weiterentwicklung von PostScript. PostScript-Dateien lassen sich daher einfach mittels eines Programmes wie dem Adobe Distiller oder Ghostscript nach PDF umwandeln. Aus \LaTeX -Quellen lässt sich mittels `pdflatex` unmittelbar PDF-Text erzeugen, bei schwierigen Texten (Formeln) der bessere Weg. PDF-Werkzeuge sind für viele Systeme verfügbar, PDF-Dokumente sind zwischen diesen Systemen austauschbar.

Im Gegensatz zu \LaTeX - oder HTML-Quellen und in Übereinstimmung mit PostScript-Dateien wird in PDF-Dateien das Aussehen der Dokumente genau festgelegt. Der Autor bestimmt, was der Leser sieht. Die Erweiterung gegenüber PostScript besteht vor allem darin, dass PDF-Dokumente ebenso wie HTML-Seiten aktiv sein können; es gibt Hyperlinks, Eingabefelder und bewegte Grafiken. Vereinfacht gesagt ist PDF eine Mischung aus PostScript und HTML. Zusätzlich wird PDF deutlich komprimiert und ist daher – von einzelnen Zeichenketten zu Anfang abgesehen – nicht lesbar. Beim Drucken auf Papier geht die Aktivität natürlich verloren, je nach Drucker auch die Farbe. Zum Lesen siehe Abschnitt 5.2.4 *Lesen von PDF-Dateien* auf Seite 171.

Weitere Formate

Das von Microsoft geschaffene Rich Text Format (RTF) soll den Austausch von Texten und Grafiken zwischen verschiedenen Anwendungen und Geräten ermöglichen. Die Spezifikation findet man bei <http://msdn.microsoft.com/library/> und bei <http://latex2rtf.sourceforge.net/rtfspec>.

html. Das Format hat innerhalb der Microsoft-Welt eine gewisse Bedeutung erlangt, wird aber nicht als Standardformat verwendet. Außerhalb der Microsoft-Welt hat es sich nicht gegen PostScript, PDF, HTML und \LaTeX durchsetzen können. Bei Debian gibt es die Konverter `latex2rtf`, `rtf2latex` und `unrtf`. Letzteres Werkzeug wandelt RTF in einfachen Text, PostScript, \LaTeX oder HTML um, siehe Abschnitt 5.5.8 *Werkzeuge für das Rich Text Format* auf Seite 198.

5.1.4 Allgemeines über Texteditoren

Ein Editor (E: editor, F: editeur) ist ein Programm, ein Werkzeug zum Erstellen oder Verändern von Texten, im weiteren Sinn auch von Musiknoten, mathematischen oder chemischen Formeln, Grafiken, Sound oder sonstigen binären Dateien. Das ist eine der ältesten und wichtigsten Arbeiten, die mit Hilfe eines Rechners erledigt werden. Notwendigerweise lässt sich ein Text mit einem Editor auch lesen.

Wird ein bereits vorhandener, abgespeicherter Text verändert, so arbeiten die Editoren – mit Ausnahme der Hex-Editoren – auf einer temporären Kopie des Textes in einem Pufferspeicher (E: buffer, F: mémoire tampon). Am Ende der Arbeit hat man die Wahl zwischen Zurückschreiben oder Verwerfen der veränderten Kopie. Manche Editoren sind so vorsichtig oder lassen sich so einstellen, dass sie auf jeden Fall das Original unter einem leicht veränderten Namen retten. Man hat dann eine Chance, die Änderungen auch noch später zu bereuen. Es gibt sogar die Möglichkeit, mit Hilfe von Versionskontrollsystemen alle Änderungen seit Anbeginn zurückzunehmen, aber das muss besonders eingerichtet werden. Bei den Textwerkzeugen werden zwei Richtungen unterschieden:

- gewöhnliche Editoren, die den Text ohne besondere Formatierung auf dem Bildschirm darstellen, die also Inhalt und Form trennen und sich für letztere nicht interessieren,
- WYSIWYG-Textprozessoren (What You See Is What You Get², auf französisch *tel écran – tel écrit*), welche die Formatierung des Textes selbst sofort durchführen und auf dem Schirm den formatierten Text anzeigen.

WYSIWYG-Textprozessoren wie Abiword, Kword, MS Word oder Wordperfect erscheinen auf den ersten Blick als die intelligenteren und bequemer zu handhabenden Werkzeuge. Sie sind tatsächlich in Büros weiter verbreitet, oft im Rahmen von Office-Paketen, siehe Abschnitt 12 *Office-Pakete* auf Seite 405. Sowie es jedoch um anspruchsvolle Texte (Bücher, schwierige Formeln, komplexes Layout, mehrere Schreiber in einem heterogenen Netz) geht, kommt man mit der Trennung von Inhalt und Form besser voran. Außerdem birgt WYSIWYG eine Versuchung in sich. Die reichen Mittel aus dem typografischen Kosmetikkoffer namens *Desktop Publishing* sind sparsam einzusetzen, unser Ziel heißt Lesbarkeit, nicht Barock.

Jeder Editor steht vor dem Problem, dass mit einer einzigen Tastatur sowohl der Text wie Editierkommandos eingegeben werden müssen. Die wenigen Editiertasten

²Real Programmers consider WYSIWYG to be just as bad a concept in text editors as it is in women.

auf den üblichen PC-Tastaturen reichen bei weitem nicht aus, außerdem sind die Tastaturen verschieden. Auch bei der Lösung dieses Problems haben sich zwei Wege herauskristallisiert:

- Editoren vom `vi`-Typ unterscheiden zwei Eingabe-Modi. Im Kommando-Modus wird jede Tastatureingabe als Kommando verstanden, im Eingabe-Modus als Text.
- Editoren vom Emacs-Typ verwenden als Kommandos Tastenkombinationen wie `<ctrl>-<x>` gefolgt von `<ctrl>-<c>`, die als Text nicht vorkommen.

Den `vi` als Linux/UNIX-Standard-Editor sollte jeder Benutzer in seinen Grundzügen kennen, siehe Abschnitt 5.3.1 *Schnellstart* auf Seite 174. Die Textdateien lassen sich zwischen den Editoren beliebig austauschen, die Werkzeuge verwenden kein besonderes Dateiformat. Man kann einen Text mit dem `vi` beginnen, mit dem `joe` verbessern und mit dem Emacs fertig stellen. Bei WYSIWYG-Textprozessoren ist das nicht uneingeschränkt der Fall, eher schon ein Glücksfall, wenn es funktioniert. Für die tägliche Arbeit bleibe man bei seinem Lieblingseditor. Man kann sich an jeden Editor gewöhnen, nur nicht jede Woche an einen neuen.

Die vorstehenden Zeilen waren vielleicht etwas viel zu einem so einfachen Thema wie der Erzeugung und Wiedergabe von Texten, aber es gibt nun einmal auf der Welt mehr als die sechszwanzig Zeichen des lateinischen Alphabets und mehr als ein Textprogramm. Auf längere Sicht kommt man nicht darum herum, sich die Zusammenhänge klar zu machen. Dann versteht man, warum in der Textverarbeitung so viel schief geht, von der künstlerischen Seite ganz abgesehen.

5.2 Werkzeuge zum Lesen und Umwandeln

5.2.1 Einfache Werkzeuge (`cat`, `more`, `pg`, `less`, `lv`, `bidiv`, `view`)

Das alte Linux/UNIX-Werkzeug `cat` ist das Muster eines Filters: es liest beliebige Dateien von `stdin` und schreibt sie nach `stdout`. Werden dem Kommando Dateinamen als Argument mitgegeben, liest es diese und gibt sie der Reihe nach aus, daher der Name (`cat` = concatenate = verketten). Es versteht ein paar einfache Optionen. Als Lesewerkzeug hat es bei längeren Texten den Nachteil, dass die Zeilen über den Bildschirm rauschen und nur die letzten stehen bleiben. Sein Hauptanwendungsgebiet sind daher Shellskripte (Programme), nicht der Dialog.

Zum Lesen längerer Texte am Bildschirm wird in der Linux/UNIX-Welt seit altersher das Kommando `more` eingesetzt. Es kann als Filter in einer Pipe oder selbständig mit dem Namen der darzustellenden Datei als Argument aufgerufen werden:

```
joe@debian:~$ more lange_textdatei
```

Nach jeweils einem Bildschirmvoll hält `more` an, sodass man den Text in Ruhe lesen kann. Die Größe des Bildschirms entnimmt das Kommando der Umgebungsvariablen `TERM`. Mit Hilfe der Eingabetaste schaltet man eine Zeile weiter, mit Hilfe der Leertaste einen Bildschirm. `more` kennt einige einfache Optionen und ebenso

einige Kommandos des Editors `vi`. Als Glied einer Pipe kann `more` nicht rückwärts blättern. Die Eingabe von `<q>` beendet das Programm. Eine ähnliche Funktionalität bietet das Werkzeug `pg` (pager).

Das Werkzeug `less` ist eine Weiterentwicklung von `more` im GNU-Projekt mit einer deutlich umfangreicheren Manualeite. Wichtigste Verbesserung ist die Fähigkeit, auch rückwärts blättern zu können. Außer den Kommandos von `more` versteht es eine Reihe von `vi`-Kommandos. Auf Linux-Maschinen ist es das Werkzeug der Wahl zum seitenweisen Lesen von Texten. Ähnlich sieht auch der Viewer `lv` aus, der mehrere Zeichensätze und Sprachen beherrscht, darunter Chinesisch, Japanisch und Koreanisch. Unter dem Namen `lgrep` aufgerufen, stellt er ein fremdsprachiges `grep` zum Suchen nach Zeichenketten bereit.

Geht es um die Darstellung von bidirektionalen Texten – also um aus Deutsch oder Englisch und Arabisch oder Hebräisch gemischten Texten – kommt ein gewöhnliches Terminal in Schwierigkeiten, da es nur eine Schreibrichtung beherrscht, üblicherweise von links nach rechts. Das Filter `bidiv` arbeitet ähnlich wie `cat` und setzt dabei eine logische Schreibrichtung bei Bedarf in eine visuelle um. Weitere Hinweise zu bidirektionalen Texten finden sich im Netz unter den Suchbegriffen *GNU FriBiDi* und *Unicode Bidirectional Algorithm*.

Das Kommando `view` ist der Editor `vi` im Nur-Lese-Modus (readonly). Alle Fähigkeiten des `vi` stehen zur Verfügung, nur Zurückschreiben kann (will) der Editor nicht. Wer diesen Editor gewohnt ist, greift gern zu `view`, andere Benutzer bleiben besser bei `less`. Auch weitere Editoren kennen einen Nur-Lese-Modus.

5.2.2 Dumper (`hexcat`, `od`)

Leseprogramme wie `more` oder `less` fassen die ihnen als Argument übergebenen Dateien als Textdateien auf. Mitunter will man jedoch beliebige Dateien ohne jede Interpretation durch das Werkzeug anschauen oder kopieren, was als dumpen (E: to dump, F: clicher) bezeichnet wird. Das Werkzeug `hexcat` ist einfach zu gebrauchen, kennt keine Optionen und leistet das Wichtigste. Der Aufruf:

```
joe@debian:~$ hexcat bild.jpg | less
```

schreibt den Inhalt der Datei `bild.jpg` in hexadezimaler Form und am rechten Rand soweit möglich als ASCII-Text auf den Schirm. Etwas vielseitiger ist das Werkzeug `od` (octal dumper). Es versteht einige Optionen in verschiedenen Schreibweisen:

```
joe@debian:~$ od -c datei
```

```
joe@debian:~$ od -t c datei
```

```
joe@debian:~$ od -format=c
```

um die Ausgabe verständlicher zu gestalten, siehe Manual. In vorstehendem Beispiel wird verlangt, die Bytes soweit möglich als ASCII-Zeichen darzustellen wie hier:

```

00000 377 330 377 340 \0 020 J F I F \0 001 001 001 001
00020 001 U \0 \0 377 376 \0 027 C r e a t e d
00040 w i t h T h e G I M P 377 333
00060 C \0 \b 006 006 \a 006 005 \b \a \a \a \t \t \b

```

aus dem Anfang einer Bilddatei (jpg), die – wie man erkennt – unter anderem lesbare Zeichenketten enthält.

5.2.3 Lesen und Umwandeln von bzw. nach PostScript (gv, a2ps)

Zum Lesen von PostScript-Dateien braucht man besondere Leseprogramme, welche die PostScript-Anweisungen in Anweisungen für das jeweilige Ausgabegerät übersetzen. Das Gleiche gilt für pdf-Dateien. Das Portable Document Format (PDF), ebenfalls von Adobe, ist eine Weiterentwicklung von PostScript. Beide Formate wurden in Abschnitt 5.1.3 *PostScript und PDF* auf Seite 165 vorgestellt.

GPL Ghostscript ist ein Übersetzer (Interpreter) für PostScript und PDF. Seine Webseite liegt unter <http://www.cs.wisc.edu/~ghost/>. Ghostview ist eine auf X11 beruhende Benutzeroberfläche zu Ghostscript, ebenso gv. GSview (nicht bei Debian) leistet ähnliche Dienste, ist aber von Ghostview unabhängig. Um eine PostScript- oder PDF-Datei für einen einfachen Drucker (der nicht selbst PostScript interpretieren kann) aufzubereiten, braucht man Ghostscript als Filter. Zum Lesen einer PostScript-Datei am Bildschirm nimmt man Ghostview oder besser gv. Das Debian-Paket *gnome-gv* enthält ein GNOME-Frontend *ggv* für Ghostscript, das Debian-Paket *kghostview* eine Anpassung von Ghostview an den KDE-Desktop.

Das Debian-Paket *psutils* enthält rund ein Dutzend Werkzeuge zur Bearbeitung von PostScript-Dateien, darunter die zunächst überflüssig erscheinenden Konverter *ps2ps* und *eps2eps*. Diese benutzen *gs* bzw. Ghostscript, um die ihnen als Argument übergebenen Dateien zu optimieren und von kleineren syntaktischen Fehlern zu bereinigen. Konverter nach PostScript sind:

- *a2ps* ASCII-Text nach PostScript mit zahlreichen Optionen für die Druckerausgabe,
- *dvi2ps* L^AT_EX-DVI nach PostScript, daneben kommt mit L^AT_EX ein Werkzeug *dvips* mit,
- *e2ps* wie *a2ps*, mit Unterstützung für Japanisch,
- *enscript* ASCII-Text nach PostScript, HTML oder RTF,
- *html2ps* HTML (Webseiten) nach PostScript,
- *u2ps*, Paket *gnome-u2ps*, UTF-8-Text nach PostScript.

In der Gegenrichtung gibt es nicht so viele Werkzeuge:

- *ps2ascii*, Paket *ghostscript*, PostScript nach ASCII-Text,
- *ps2eps* PostScript nach Encapsulated PostScript,
- *pstotext* PostScript oder PDF nach Latin-1-Text, alternativ zu *ps2ascii*.

5.2.4 Lesen von PDF-Dateien (Adobe Reader, xpdf, gpdf, kpdf, pdftk, gv, Evince)

PDF-Dateien sind komprimiert und daher deutlich kleiner als PostScript-Dateien. Dazu kommen weitere Fähigkeiten, an die man bei der Entwicklung von PostScript noch nicht dachte. Da Adobe das Lesewerkzeug *Adobe Reader*, Aufruf `acroread`, kostenlos verteilt (<http://www.adobe.de/>) und es auf vielen Rechnern läuft, hat sich das PDF-Format zum Speichern und Austauschen von Dokumenten schnell durchgesetzt. Für Linux/UNIX steht der Adobe Reader 7.0 Basic Version in Englisch, Deutsch oder Französisch als Tarball oder rpm-Paket im Umfang von 40 MB bei Adobe im Netz. Die Einrichtung durch den Verwalter verlief problemlos wie schon bei früheren Versionen. Zum Schluss blieb noch das Anlegen eines symbolischen Links, damit das Werkzeug im üblichen Pfad gefunden wird:

```
debian:/usr/local/bin# ln -s
/usr/local/Adobe/Acrobat7.0/bin/acroread areader
```

Beim Drucken aus dem Adobe Reader heraus sollte man die PostScript-Einstellung *Große Seiten auf Papiergröße verkleinern* im Normalfall ausschalten, das sie auch dann verkleinert, wenn kein Bedarf besteht (PDF-Ausgabe auf DIN A4, sollte eigentlich passen, wird trotzdem verkleinert). Uns ist noch kein Drucker begegnet, der selbst PDF interpretiert, der Weg führt stets über PostScript.



Abbildung 5.1. Screenshot des Leseprogramms GNOME Evince

Das Debian-Paket *xpdf* ist ein Oberbegriff für die Pakete *xpdf-common*, *xpdf-reader* und *xpdf-utils*, die zusammen eine Werkzeugkiste zur Handhabung von PDF-

Dateien darstellen. Das Kommando `xpdf` ruft einen Betrachter auf, eine Alternative zum Adobe Reader. GNOME `gpdf` setzt auf `xpdf` auf, KDE `kpdf` ist aus `xpdf` abgeleitet. Im Debian-Paket *pdftk* (PDF Toolkit) ist ein vielseitiges Werkzeug zur Bearbeitung von PDF-Dateien enthalten, Heimathafen <http://www.accesspdf.com/pdftk/>, ein deutsches Tutorial von STEFAN LAGOTZKI gibt es bei <http://www.lagotzki.de/pdftk/>. Wie schon erwähnt, kann `gv` außer PostScript auch PDF lesen. Das noch junge Werkzeug Evince aus dem GNOME-Projekt liest PostScript, Encapsulated PostScript, PDF und künftig vielleicht weitere Formate. Es setzt auf Ghostscript und `xpdf` auf. Sein Ziel ist, zahlreiche Einzelwerkzeuge wie `ggv`, `gpdf` und `xpdf` zu erübrigen. Das Werkzeug hinterließ einen angenehmen, aufgeräumten Eindruck, siehe Abbildung 5.1.

5.2.5 Lesen von Word-Dateien und Ähnlichem (`catdoc`, `wordview`, `antiword`, `wv`, `word2x`, `xlhtml`, `unrtf`)

Ein Debian-Benutzer ist manchmal gezwungen, Textdateien zu lesen, die in einem proprietären Format vorliegen. Am häufigsten handelt es sich dabei um Dateien, die mit MS Word geschrieben sind, kenntlich an der Dateikennung `doc`. Einige der WYSIWYG-Editoren – siehe Abschnitt 5.6 *WYSIWYG-Textprozessoren* auf Seite 199 – können mit Word-Dateien umgehen, wobei unter Umständen komplexe Strukturen verloren gehen. Will man solche Dateien nur lesen, gibt es einfachere Wege.

Das Werkzeug `catdoc` liest mit MS Word oder MS Write geschriebene Texte sowie Textdateien im RTF-Format. Es beherrscht mehrere Zeichensätze; intern verwendet es Unicode. Für die Ausgabe kann es \LaTeX -Metazeichen in entsprechende Sequenzen umsetzen, sodass eine Weiterverarbeitung mittels \LaTeX erleichtert wird. Es ist jedoch kein Word-to- \LaTeX -Konverter, dafür sind die Unterschiede zwischen den beiden Textwelten zu grundlegend. `catdoc` versucht nicht, die Strukturen der eingelesenen Texte zu erhalten oder zu übersetzen.

Das Debian-Paket enthält auch das Werkzeug `xls2csv`, mit dem MS-Excel-Tabellen in Tabellen aus Zeilen mit durch Kommas getrennten Werten (comma separated values) umgewandelt werden. Schließlich bringt das Debian-Paket noch `wordview` mit, eine auf Tcl/Tk beruhende grafische Benutzeroberfläche zu `catdoc`. Da Microsoft mehrmals das Format von Word geändert hat, ist es unwahrscheinlich, dass `catdoc` mit *jeder* Word-Datei klarkommt. Mit Word 8 bzw. Word aus Office 97 erzeugte Dateien sollten keine Schwierigkeiten verursachen.

Das Kommando `antiword` wandelt Word-Dateien in einfachen Text oder PostScript um. Seine Startseite liegt unter <http://www.winfield.demon.nl/>.

Das Debian-Paket `wv` – früher als `mswordview` bekannt – enthält eine Bibliothek und einige Anwendungen, um Word- oder RTF-Dateien in HTML oder \LaTeX umzuwandeln. Seine Funktionen werden unter anderem von den WYSIWYG-Textprozessoren Abiword und KWord verwendet. Die Startseite des Projektes findet sich unter <http://wv.sourceforge.net/>.

Nicht bei Debian, aber unter <http://word2x.sourceforge.net/> findet man das Werkzeug `word2x`, das Word-Dokumente in einfachen Text, \LaTeX oder

HTML umwandelt. Unter <http://pair.mbl.ca/doc2xml/> ist das Werkzeug `doc2xml` erhältlich, ein Tarball in frühem Zustand.

Das Werkzeug `xlhtml` – Startseite <http://chicago.sourceforge.net/xlhtml/> – wandelt Excel- oder PowerPoint-Dateien nach HTML, XML oder CSV um, falls man nicht eine Tabellenkalkulation bzw. ein Präsentationswerkzeug dafür einsetzen möchte.

Das Debian-Paket `unrtf` enthält einen Umwandler von RTF nach ASCII-Text, HTML, \LaTeX und PostScript. Weitere Formate sind in Entwicklung. Am weitesten fortgeschritten ist die Umwandlung nach HTML. Näheres auf <http://www.gnu.org/software/unrtf/unrtf.html>.

5.2.6 Optical Character Recognition (`clara`, `gocr`)

Optische Zeichenerkennung (E: optical character recognition, OCR, F: reconnaissance optique de caractères) bezeichnet das maschinelle Erkennen von Zeichen und damit Lesen von Texten, beispielsweise im Zusammenhang mit der automatischen Bearbeitung von Schecks. Hierzu wurden Schriften entwickelt, die sowohl von Menschen wie von Maschinen gelesen werden können und insbesondere bei Ziffern die Wahrscheinlichkeit von Verwechslungen minimieren. Auch die neueren Kfz-Kennzeichen verwenden eine solche Schrift. Mit der steigenden Leistungsfähigkeit der Rechner ist der Bedarf an solchen Schriften jedoch zurückgegangen. Eine wichtige Anwendung von OCR ist die Umwandlung von gescannten oder gefaxten Textvorlagen – die als Rastergrafik vorliegen – zurück in Text. Das Projekt Gutenberg (<http://www.gutenberg.org/>, im deutschen Sprachraum <http://gutenberg.spiegel.de/>), das urheberrechtsfreie Literatur ins Netz stellt, ist ein weltweites, umfangreiches Beispiel für die Zeichenerkennung. Sie finden dort unter anderem Werke von SHAKESPEARE, EINSTEIN, EURIPIDES, FONTANE und KARL MAY.

Das Werkzeug `clara` erwartet Vorlagen im `pbm`- oder `pgm`-Format und setzt X11 voraus. Auf der Webseite <http://www.claraocr.org/> finden sich neben Informationen in Englisch auch einige Textbeispiele zum Ausprobieren, falls man keinen gescannten Text zur Hand hat. Das Programm ist lernfähig. Leider geht seine Entwicklung zur Zeit nur langsam voran.

Das Kommandozeilen-Werkzeug `gocr` verlangt Dateien im `pbm`-, `pnm`-, `pgm`-, `ppm`- oder `pcx`-Format und kommt unter gewissen Voraussetzungen auch mit `png`, `jpg`-, `tiff`- oder `gif`-Dateien zurecht. Der Gebrauch ist denkbar einfach:

```
joe@debian:~$ gocr datei
```

liest die Grafikdatei `datei` und schreibt das Ergebnis nach `stdout`. Bei einfach aufgebauten Seiten liefert es nach etwas Konfiguration brauchbare Ergebnisse. Seine Webseite ist unter <http://jocr.sourceforge.net/> zu finden; der Name `jocr` ist kein Schreibfehler, `gocr` war bei Sourceforge schon besetzt. Bei Debian liegen noch einige ergänzende Pakete zu `gocr`, darunter ein grafisches Frontend und Dokumentation.

5.2.7 Umcodieren von Zeichen und Zeichensätzen (`tr`, `recode`, `uniconv`, `iconv`, `convmv`)

Zur Umsetzung von Zeichen gibt es mehrere Linux/UNIX-Werkzeuge wie `tr` und `sed`. Das erstere ist auch ein Nothelfer, wenn man ein Zeichen in einen Text einbauen möchte, das man nicht auf der Tastatur findet. Will man beispielsweise den Namen *Citroën* richtig schreiben und sieht keine Möglichkeit, das `e` mit dem Trema per Tastatur zu erzeugen, dann schreibt man *CitroXn* und schickt den Text durch:

```
joe@debian:~$ tr '\130' '\315' < text > text.neu
```

Die Zahlen sind die oktalen Positionen, hier im Zeichensatz ROMAN8. Mit dem Streaming Editor `sed` lassen sich komplexe Ersetzungen bewerkstelligen, siehe Abschnitt 5.9 *Streaming Editor* auf Seite 205.

Aus dem GNU-Projekt stammt ein Filter namens `recode`, das etwa hundert Codetafeln oder Zeichensätze ineinander umrechnet:

```
joe@debian:~$ recode --help
```

```
joe@debian:~$ recode -l
```

```
joe@debian:~$ recode ascii-bs:EBCDIC-IBM textfile
```

Man beachte jedoch, dass beispielsweise ein HTML-Text, der mit ASCII-Ersatzdarstellungen für die Umlaute (`ä` für a-Umlaut) geschrieben ist, bei Umwandlung nach ASCII unverändert bleibt. Es werden Zeichensätze umgewandelt, mehr nicht. Auch werden \LaTeX -Formatanweisungen nicht in HTML-Formatanweisungen übersetzt, dafür gibt es andere Werkzeuge wie `latex2html`. Die Ursprungsdatei wird überschrieben, daher sicherheitshalber mit einer Kopie arbeiten. Nicht jede Umsetzung ist umkehrbar.

Unter dem Namen *uniconv* laufen mehrere, teilweise kommerzielle Produkte. Das Debian-Paket *yudit* – siehe Abschnitt 5.8 *Editoren für fremde Schriften* auf Seite 202 – bringt ein Werkzeug `uniconv` mit, das verschiedene Zeichensätze und -codierungen ineinander umrechnet: UTF-8, UTF-16, ISO 8859-1, ISO 8859-2, ISO 8859-5, CP 1251, ISO 2022 und weitere. Ähnliches leistet das Kommando `iconv` aus dem GNU-Projekt (<http://www.gnu.org/software/libiconv/>). Aufgerufen mit der Option `-l` listen beide Kommandos die ihnen bekannten Zeichensätze und -codierungen auf. Das Werkzeug `convmv` übersetzt Dateinamen nach UTF-8 und hilft beim Umstellen eines Systems auf Unicode.

5.3 Der Editor `vi` samt Nachwuchs

5.3.1 Schnellstart

Der Name der zu editierenden Datei wird beim Aufruf als Argument mitgegeben:

```
joe@debian:~$ vi datei
```

Der `vi` kennt zwei Modi: den Kommando-Modus³ (E: command mode) und den Schreib-Modus (E: input mode, editing mode). Tasteneingaben im Kommando-Modus werden als Kommandos aufgefasst, im Schreib-Modus als zu speichern-der Text. Kommandos, die mit einem Doppelpunkt (E: colon) beginnen, sind mit `<cr>` abzuschicken; sie werden auf der untersten Zeile angezeigt. Die übrigen Kommandos werden nicht angezeigt und wirken sofort. Mit einem Dutzend `vi`-Kommandos kommt man weit. Was darüber hinaus geht, lernt man bei Bedarf. Alle `vi`-Kommandos nutzt kein Schreiber aus. Das Dutzend (die aktuelle Zeile ist die, in der sich der Cursor befindet):

- `<esc>` schalte in den Kommando-Modus um,
- `o` (open) öffne eine neue Zeile zum Schreiben unterhalb der aktuellen Zeile,
- `i` (insert) füge in aktueller Zeile Text vor der Cursorposition ein,
- `a` (append) füge in aktueller Zeile Text nach der Cursorposition ein,
- `r`, `R` (replace) ersetze genau ein Zeichen bzw. beliebig langen Text,
- `x` lösche genau ein Zeichen,
- `dd` (delete) lösche die aktuelle Zeile,
- `u` (undo) nimm jüngstes Editierkommando zurück,
- `nG` (go) springe an den Anfang der Zeile `n`,
- `/abcd <cr>` suche nach der Zeichenkette `abcd` vorwärts,
- `?abcd <cr>` suche nach der Zeichenkette `abcd` rückwärts,
- `:wq <cr>` (write, quit) schreibe die Arbeitskopie zurück und beende den Editor.

Falls Ihre Pfeiltasten (Cursortasten) nicht funktionieren, können Sie sich mit folgenden Buchstabentasten im Kommandomodus helfen:

- `h` gehe ein Zeichen nach links,
- `j` gehe eine Zeile nach unten,
- `k` gehe eine Zeile nach oben,
- `l` gehe ein Zeichen nach rechts.

Die Zeichen liegen auf der Tastatur in der Mitte beieinander. Diese Alternative wird auch von einigen anderen Programmen genutzt. Wenn Sie jetzt noch wissen, dass man vor die Lösch-Kommandos `x` und `dd` eine Zahlenangabe setzen darf, die die Kommandos auf entsprechend viele Zeichen bzw. Zeilen ausweitet, beherrschen Sie die Pflicht eines `vi`-Benutzers. Was folgt, ist Kür.

Während Sie editieren, können Sie Hilfe zu vielen Themen anfordern, indem Sie im Kommandomodus `:h thema <cr>` (help) eingeben. Zurück aus der Hilfe in Ihren Text gelangen Sie mit `:q <cr>` (quit).

5.3.2 Übersicht

Der bildschirmorientierte Texteditor `vi` (visual editor) wurde um 1976 von BILL JOY in Berkeley entwickelt. Von ihm stammt auch die Aussprache *vee-eye*, deutsch

³Genau genommen gibt es zwei Kommando-Modi, was jedoch für den praktischen Gebrauch keine Bedeutung hat.

wie-ei. Der `vi` löste nach und nach einen einfacheren, heute schon sehr klassischen Editor namens `ed` ab, der in Zeiten von Druckerterminals und langsamen Datenleitungen seine Berechtigung hatte⁴, und ist heute auf jedem Linux/UNIX-System zu finden. `vi`-ähnliche Editoren sind außer für Linux/UNIX auch für andere Betriebssysteme einschließlich DOS geschrieben worden. Unter Debian GNU/Linux ist meist der `nvi` (new `vi`) von KEITH BOSTIC oder der `vim` (`vi` improved) von BRAM MOOLENAAR eingerichtet und meldet sich unter dem Aufruf `vi`, siehe `/etc/alternatives/`. Unterschiede bemerkt erst ein fortgeschrittener Benutzer. `gvim` ist im Debian-Paket `vim-full` enthalten und ein um eine X11-Oberfläche erweiterter `vim`. Den originalen `vi` gibt es noch bei <http://ex-vi.sourceforge.net/>. Infolge des Alters und der weiten Verbreitung der `vi`-Familie auch auf anderen Systemen findet sich viel Information im Netz und in den Buchregalen. Eine Auswahl:

- Manualseite (`man vi`), nur zum Nachschlagen, nicht zum Lernen,
- Online-Tutorial: aus dem Editor heraus `:help tutor` eingeben, verlassen mit `:q`,
- Vim-HOWTO, zur Zeit (Mitte 2005) in Revision,
- FAQ: <http://www.faqs.org/faqs/editor-faq/vi/part1/> und [part2](http://www.faqs.org/faqs/editor-faq/vi/part2/),
- WILLIAM JOY und MARK HORTON: *An Introduction to Display Editing with Vi*, <http://docs.freebsd.org/44doc/usd/12.vi/paper.html>,
- *Mastering the VI editor*, <http://www.eng.hawaii.edu/Tutor/vi.html>,
- *The Berkeley VI Editor Home Page*, <http://www.bostic.com/vi/> (es geht um `nvi`),
- <http://freshmeat.net/projects/nvi/>,
- <http://www.vim.org/> mit viel Dokumentation,
- *The VI Lovers Home Page*, <http://thome.com/vi/vi.html>,

Der Editor hat Grenzen. Beim klassischen `vi` von 1987 beträgt die Zeilenlänge maximal 1024 Zeichen, beim `vim` über 2 Milliarden Zeichen (das Manuskript zu vorliegendem Buch umfasst rund 2 Millionen Zeichen). Die maximale Anzahl der Zeilen beträgt 250.000, beim `vim` über 2 Milliarden. Die maximale Dateigröße liegt bei 0,5 MB, beim `vim` 2 GByte, wobei es Unfug ist, Textdateien so groß werden zu lassen. Zusätzlich begrenzen Arbeitsspeicher einschließlich Swapbereich die Dateigröße. Hat man solche Monsterdateien zu editieren, können Hex-Editoren (Seite 202) weiterhelfen. In den folgenden Abschnitten wird der `vi` systematisch und ausführlich, aber nicht erschöpfend dargestellt.

5.3.3 Konfiguration

Wer mit dem `vi` beginnt, wird die Default-Einstellungen übernehmen. Im Lauf der Zeit kann man über einige Werte nachdenken, um sich die Arbeit zu erleichtern.

⁴Ich habe ihn verwendet, als meine Verbindung von daheim zur Universität über einen Akustikkoppler lief. Der Verwalter braucht ihn in verzweifelten Situationen.

Die Einstellungen werden als vi-Umgebung bezeichnet. Es gibt drei Wege, sie zu beeinflussen:

- durch Setzen einer systemweiten oder benutzerspezifischen Umgebungsvariablen namens EXINIT,
- durch eine benutzereigene Datei namens `.exrc` bzw. `.vimrc`,
- während des Editierens durch Eingeben des Kommandos `:set`.

Die augenblicklichen Einstellungen bekommt man während des Editierens mittels des Kommandos `:set all` zu Gesicht. Wir beschränken uns auf die Konfiguration mit Hilfe der Datei `.exrc`. Es kann je nach vi-Variante erforderlich sein, den Editor zum automatischen Lesen dieser Datei durch Setzen einer Option mittels der EXINIT-Variablen in einer Datei wie `$HOME/.profile` oder `$HOME/.bashrc` zu bewegen:

```
export EXINIT="set exrc"
```

Äußerst wichtig für den vi ist die Shell-Umgebungsvariable TERM, die ihm sagt, mit welchem Terminaltyp er zusammenarbeitet. Auf Grund der Eintragung unter TERM schaut der Editor entweder in einer Datei `/etc/termcap` oder `/etc/terminfo` nach, um die jeweiligen Steuerkommandos herauszulesen. Es müssen also sowohl die TERM-Variable wie die Terminalbeschreibung in `termcap` oder `terminfo` stimmen. Für die TERM-Variable ist der Benutzer verantwortlich, für die Terminalbeschreibung der Verwalter. Terminaltyp und TERM-Variable ermittelt man mit folgenden Kommandos:

```
joe@debian:~$ tset -q
```

```
joe@debian:~$ echo $TERM
```

Sie sollten übereinstimmen. Faselt der Editor nach dem Aufruf etwas von *Open Mode*, so kennt er den Terminaltyp nicht und nimmt ein dummes Terminal an. Damit kann man arbeiten, aber nicht glücklich werden. Arbeitet man mittels `slogin` oder einem ähnlichen Kommando auf einem entfernten Rechner und zeigt sich der vi widerspenstig, gilt der erste Blick der Variablen TERM, die erforderlichenfalls anzupassen ist. Innerhalb der Shell (bash) geschieht dies mittels einer Eingabe wie `TERM=xterm`, innerhalb des Editors mittels einer Kommandofolge wie:

```
Q set term=xterm <cr> vi <cr>
```

Damit wird zuerst der vi-Modus verlassen und in den ex-Modus geschaltet (Q), dort der Terminaltyp gesetzt (`set term=xterm <cr>`) und schließlich in den vi-Modus zurückgeschaltet (`vi <cr>`). Einfacher ist es, schon vor dem Aufruf des Editors die Shellvariable TERM zu überprüfen, siehe oben.

In seinem Home-Verzeichnis oder einem Unterverzeichnis kann ein Benutzer eine Datei `.exrc` anlegen. Auf diese Weise lassen sich für bestimmte Verzeichnisse – beispielsweise mit Programmquellen – besondere Verhaltensweisen oder Optionen festlegen. In die Datei werden Kommandos geschrieben, wie sie auch während des Editierens eingegeben werden dürfen. Einige sind:

- `:set autoindent` rücke aktuelle Zeile automatisch wie die vorhergehende ein (kann nerven),
- `:set ignorecase` unterscheide nicht zwischen Groß- und Kleinschreibung bei der Suche mit regulären Ausdrücken,
- `:set number` nummeriere die Zeilen auf dem Bildschirm, nicht im Speicher,
- `:set readonly` rufe den Editor per Default nur zum Lesen auf,
- `:set showmatch` springe beim Eintippen einer schließenden Klammer kurz zur zugehörigen öffnenden Klammer,
- `:set tabstop=4` setze den Tabulatorsprung auf vier Zeichen statt auf acht (Default),
- `:set wrapmargin=n` bricht lange Zeilen auf dem Bildschirm automatisch um, und zwar n Zeichen vor dem rechten Rand.

Die Vorsilbe `no` hat die entgegengesetzte Wirkung:

```
:set nonumber
```

bewirkt, dass die Zeilen auf dem Schirm nicht nummeriert werden (Default).

Der `vim` versteht einige Optionen mehr als der klassische `vi`. Interessant ist die Syntax-Hervorhebung. Der Editor kennt mehrere hundert Dateiformate, darunter Quelltexte gängiger Programmiersprachen sowie \LaTeX . Schaltet man die Option ein:

```
:syntax on
```

werden die Schlüsselwörter des jeweiligen Formates farbig hervorgehoben. Die Formatbeschreibungen liegen in `/usr/share/vim/vim63/syntax/`. Man kann sie sich mit `less` ansehen, neue erfinden oder – was eher gefragt sein könnte – aus vorhandenen eigene Varianten ableiten.

5.3.4 Makros

In der Datei `.exrc` lassen sich auch Makros unterbringen. Das sind kurze Kommandofolgen, die einer einzelnen Taste oder einer Tastenkombination (keychord, key sequence) zugeordnet sind. Wir unterscheiden zwischen Makrobefehlen, die im Kommandomodus einzugeben sind, und solchen, die im Schreibmodus eingetippt werden (ohne vorheriges `<esc>`). Erstere werden durch Zeilen folgender Art in der Datei `.exrc` geschaffen:

```
:map makroname kommandos <cr>
```

Als Makroname kommt bevorzugt ein einzelnes Zeichen in Betracht. Die Buchstaben `g` und `v` sind vom `vi` nicht belegt und können bedenkenlos als Makroname herangezogen werden, eventuell auch die Funktionstasten. Definieren wir:

```
:map g ?index<ctrl>v<cr><cr>
```

so führt die Eingabe von `<g>` im Kommando-Modus zum Zurückspringen auf die nächstliegende Zeichenkette *index*. Das erste `<cr>` ist Bestandteil des Kommandos. Um es als Zeichen eingeben zu können, ist ihm die Zeichenkombination `<ctrl>-<v>` voranzustellen. Das zweite `<cr>` schließt das Makro ab.

Im Schreibmodus einzugebende Makros werden folgendermaßen definiert:

```
:map! makroname zeichenkette <cr>
```

Ein Zeichen, dem man auf diese Weise eine besondere Bedeutung verleiht, kann dann nur durch Voranstellen von `<ctrl>-<v>` in seiner ursprünglichen Bedeutung eingegeben werden. Beispiel:

```
:map! C Chemieingenieurwesen <cr>
```

bewirkt, dass die Eingabe von `<C>` im Schreibmodus zum Erzeugen der Zeichenkette *Chemieingenieurwesen* führt, bei der ich mich fast immer vertippe. Künftig tippe ich nur noch `<C>`, und die Zeichenkette erscheint. Brauche ich dann in einem Brief an CAIUS JULIUS CAESAR den Buchstaben `<C>`, muss ich `<ctrl>-<v>` voranstellen. Die Frage ist, was man häufiger benötigt. Mittels `<ctrl>-<v>` lassen sich auch das Zeichen `<esc>` und andere Sonderzeichen buchstäblich eingeben. Die aktuellen Mappings erfährt man mittels `:map`.

Verwandt mit den vorstehenden Makros sind Abkürzungen. Der Unterschied liegt darin, dass eine Abkürzung nur dann erweitert (expandiert) wird, wenn sie als Wort für sich allein steht, nicht jedoch, wenn sie als Teil eines längeren Wortes auftritt. Wollen wir CIW als Abkürzung für Chemieingenieurwesen verwenden, so definieren wir:

```
:abbr CIW Chemieingenieurwesen<cr>
```

und in unserem Text wird die Eingabe CIW immer zu *Chemieingenieurwesen* erweitert. Das entspricht unserem Wunsch vermutlich besser als das Makro. Vertippt man sich bei einem Wort häufig in derselben Weise, so lässt sich die falsche Schreibweise als Abkürzung für die richtige definieren:

```
:abbr Karslsruhe Karlsruhe<cr>
```

und der Editor verbessert unsere Eingabe stillschweigend. Alternativ wäre Üben mit der Tastatur anzuraten, siehe Abschnitt 5.13.1 *Tipptrainer* auf Seite 229.

5.3.5 Aufrufen und Beenden

Der Editor wird aus der Kommandozeile in einer der folgenden Formen aufgerufen:

```
joe@debian:~$ vi
```

Die Arbeit vollzieht sich in einem Pufferspeicher. Beim Zurückschreiben muss man sich für einen bereits vorhandenen oder neuen Namen entscheiden.

```
joe@debian:~$ vi neue_datei
```


Die Arbeit vollzieht sich in einem Pufferspeicher. Beim Zurückschreiben wird die Datei `neue_datei` angelegt.

```
joe@debian:~$ vi alte_datei
```

Der Editor arbeitet auf einer temporären Kopie von `alte_datei`. Beim Zurückschreiben wird `alte_datei` mit der bearbeiteten Kopie überschrieben. Falls mehrere Benutzer gleichzeitig auf Kopien von `alte_datei` arbeiten, gewinnt der, der als letzter zurückschreibt.

```
joe@debian:~$ vi +144 alte_datei
```

Wie oben, der Editor springt gleich in Zeile Nr. 144. Praktisch, wenn ein Fehlersuchprogramm die Nummer der fehlerhaften Zeile mitgeteilt hat.

```
joe@debian:~$ vi +/muster alte_datei
```

Wie oben, der Editor springt gleich zum ersten Vorkommen des Zeichenmusters.

```
joe@debian:~$ vi datei1 datei2 datei3
```

Der Editor arbeitet auf einer Kopie von `datei1`. Nach Zurückschreiben mittels des Kommandos `:w` holt man durch das Kommando `:n <cr>` eine Kopie der nächsten Datei in den Pufferspeicher. Das Kommando `rew` (rewind) geht in der Dateiliste einen Schritt zurück.

```
joe@debian:~$ view alte_datei
```

Der Editor wird nur zum Lesen (readonly) aufgerufen. Sie können alles mit der temporären Kopie der Datei anstellen, nur nicht zurückschreiben. Eine Alternative zu Pagnern wie `more` oder `less`.

Zum Zurückschreiben der temporären Arbeitskopie auf den Massenspeicher dient das Kommando `:w` (write). Man kann und soll es auch während der laufenden Arbeit geben, um Datenverluste bei irgendwelchen Zwischenfällen zu vermeiden. Falls die Datei schreibgeschützt ist oder man den Editor readonly (`view`) aufgerufen hat, muss man `:w!` eintippen, um seinen Willen durchzusetzen. Der Doppelpunkt weist darauf hin, dass es sich hier eigentlich um ein Kommando des Editors `ex` handelt. Tatsächlich sind `vi` und `ex` dasselbe Programm, nur mit unterschiedlichem Verhalten gegenüber dem Benutzer. Das braucht uns jedoch nicht zu interessieren. Wir arbeiten nicht ausdrücklich mit dem `ex`.

Den Editor beendet man mit dem Kommando `:q` (quit). Auch hier kann ein Ausrufezeichen erforderlich werden. Meist kombiniert man Zurückschreiben und Beenden mittels `:wq`.

Wird der Editiervorgang gewaltsam unterbrochen (Stromausfall), so steht die mit dem jüngsten `:w`-Kommando zurückgeschriebene Kopie im Massenspeicher. Aber der `vi` tut von sich aus noch mehr. Er legt im jeweiligen Arbeitsverzeichnis eine Punktdatei mit der Kennung `.swp` an (swap file) – mittels `ls -la` zu sehen – die laufend nachgeführt wird. Der `vim` aktualisiert die Swap-Datei nach der Eingabe von jeweils 200 Zeichen oder nach 4 s Ruhe, konfigurierbar. Die anderen Familienmitglieder des Editors halten sich mit diesbezüglichen Auskünften zurück. Man

kann erreichen, dass die Datei nicht im Arbeitsverzeichnis, sondern in einem anderen Verzeichnis, möglichst auf einer anderen Platte, angelegt wird. Dann ist sogar bei einem Plattencrash der Text gerettet. Die Swap-Datei enthält den Text und binäre Informationen, ist also nur eingeschränkt verständlich. Beim ordnungsgemäßen Beenden des Editors wird sie gelöscht, bei Stromausfall bleibt sie erhalten. Rufen wir später den Editor mit dem Namen der während des Stromausfalls editierten Datei als Argument auf, bemerkt der Editor die Existenz der Punktdatei und bietet an, sie als die jüngste Fassung des Textes weiter zu verarbeiten (recover). So gehen im schlimmsten Fall nur wenige Eingaben infolge der Unterbrechung verloren. Einige Versionen des Editors legen vor jedem Zurückschreiben eine Sicherungskopie des vorhergehenden Textes an. Man kann so nach einer Katastrophe auf drei Fassungen des Textes zurückgreifen:

- die Swap- oder Recovery-Datei, die aktuellste Fassung,
- die Textdatei auf dem Massenspeicher mit Stand des jüngsten Zurückschreibens,
- die Sicherungskopie mit Stand vor dem jüngsten Zurückschreiben.

Wir sind dem `vi` schon dankbar gewesen. Bei einem Plattencrash hilft dieses Vorgehen ohne die oben erwähnte Auslagerung der Swap-Datei nichts, da müsste man mit Spiegeln (RAID) oder eigenen Skripts und `cron`-Jobs arbeiten.

Hat man beim Editieren so viel Unsinn gebaut, dass nichts mehr zu retten ist, holt man mit dem Kommando `:vi!` eine frische Kopie der Datei vom Massenspeicher in den Puffer. Man beginnt so mit dem Stand des jüngsten Zurückschreibens von vorn. Ist nur der Bildschirm (nicht die Datei) durcheinander geraten, schreibt das Kommando `<ctrl>-<l>` den Schirm neu. Das kann vorkommen, wenn man während der Arbeit die Fenstergröße ändert.

5.3.6 Navigieren im Text

Beim Schreiben geht ein beträchtlicher Teil der Zeit auf das Navigieren im Text drauf – der möglicherweise auf mehrere Dateien und Verzeichnisse verteilt ist. Nächst dem Eingeben des Textes ist das Navigieren die wichtigste handwerkliche Fähigkeit des Schreibers. Zunächst einige Kommandos zum Bewegen des Bildschirms (Fensters) im Text (das Zeichen `n` steht hier immer für eine Anzahl):

- `n<ctrl>-d` (down) bewege den Text samt Cursor auf dem Bildschirm (scrolle) `n` Zeilen abwärts,
- `n<ctrl>-u` (up) bewege den Text samt Cursor auf dem Bildschirm (scrolle) `n` Zeilen aufwärts,
- `n<ctrl>-e` bewege den Text auf dem Bildschirm `n` Zeilen abwärts, der Cursor bleibt auf seiner Position im Text,
- `n<ctrl>-y` bewege den Text auf dem Bildschirm `n` Zeilen aufwärts, der Cursor bleibt auf seiner Position im Text,
- `n<ctrl>-f` (forward) bewege den Text samt Cursor um `n` Seiten (Bildschirme) vorwärts,
- `n<ctrl>-b` (backward) bewege den Text samt Cursor um `n` Seiten (Bildschirme) rückwärts,

- `<ctrl>-l` schreibe einen verunstalteten Bildschirm neu,
- `z<cr>` schreibe den Bildschirm neu so, dass die aktuelle Zeile oben erscheint,
- `z.` schreibe den Bildschirm neu so, dass die aktuelle Zeile in der Mitte erscheint,
- `z-` schreibe den Bildschirm neu so, dass die aktuelle Zeile unten erscheint,

Nun einige ausgewählte Kommandos zum Bewegen des Cursors:

- `nh` bewege den Cursor um `n` Zeichen nach links, Default `n = 1`,
- `nj` bewege den Cursor um `n` Zeilen nach unten, Default `n = 1`,
- `nk` bewege den Cursor um `n` Zeilen nach oben, Default `n = 1`,
- `nl` bewege den Cursor um `n` Zeichen nach rechts, Default `n = 1`,
- `ng` bewege den Cursor nach Zeile `n`,
- `G` bewege den Cursor zur letzten Zeile,
- `0` (null) bewege den Cursor zum Anfang der aktuellen Zeile,
- `$` bewege den Cursor an das Ende der aktuellen Zeile,
- `nw` bewege den Cursor vorwärts auf den Anfang des `n`-nächsten Wortes,
- `nb` bewege den Cursor rückwärts auf den Anfang des `n`-nächsten Wortes,
- die beiden vorstehenden Kommandos können auch mit Großbuchstaben gegeben werden, wobei die Definition von *Wort* etwas anders lautet,
- `n}` bewege der Cursor nach rechts oder unten an den Anfang des `n`-nächsten Absatzes, gekennzeichnet durch eine Leerzeile,
- `n{` bewege den Cursor nach links oder oben an den Anfang des `n`-nächsten Absatzes rückwärts,
- `m[a-z]` (`m` gefolgt von einem Kleinbuchstaben) setze eine Marke im Text,
- ``[a-z]` (back quote) bewege den Cursor an die wie vorstehend beschrieben gesetzte Marke,
- `%` falls der Cursor unter einer Klammer (rund, eckig, geschweift) steht, springe zur ergänzenden Klammer.

Sucht man bestimmte Zeichen oder Zeichenmuster im Text, so helfen folgende Kommandos:

- `nf x` bewege den Cursor zum `n`-nächsten Vorkommen des Zeichens `x` in der Zeile,
- `nF x` dasselbe rückwärts,
- `;` (Semikolon) wiederhole eines der obigen Kommandos,
- `,` (Komma) wiederhole eines der obigen Kommandos in umgekehrter Richtung,
- `/muster<cr>` suche vorwärts nach dem Zeichenmuster *muster*,
- `?muster<cr>` suche rückwärts nach dem Zeichenmuster *muster*,
- `*` (Stern) suche vorwärts das nächste Vorkommen des Wortes, auf dem der Cursor steht (nur `vim`),
- `#` (Doppelkreuz) dasselbe rückwärts,
- `n` (nochmal) wiederhole die jüngste Suche,
- `N` wiederhole die jüngste Suche in umgekehrter Richtung.

Das Zeichenmuster darf ein regulärer Ausdruck sein, siehe Abschnitt 4.1.2 *Reguläre Ausdrücke* auf Seite 116. Sucht man beispielsweise in einer Liste nach der Zeichenkette *Alex* am Anfang einer Zeile, so lautet das Kommando:

/^ Alex

Die letzten fünf Suchkommandos werden häufig gebraucht.

5.3.7 Text ändern

Einen Text ändern heißt, Textteile hinzufügen, löschen, ersetzen, verschieben. Diesen Zwecken dienen folgende vi-Kommandos:

- a (append) füge hinzu im Anschluss an die Cursorposition,
- A füge hinzu im Anschluss an die aktuelle Zeile,
- i (insert) füge hinzu vor der Cursorposition,
- I füge hinzu vor der aktuellen Zeile,
- o (open) öffne neue Zeile zum Schreiben unterhalb der aktuellen Zeile,
- O öffne neue Zeile zum Schreiben oberhalb der aktuellen Zeile,
- nx lösche n Zeichen ab Cursorposition, Default n = 1,
- nX lösche n Zeichen links von der Cursorposition, Default n = 1,
- dnw (delete) lösche n Wörter ab der Cursorposition, Default n = 1,
- d\$ lösche aktuelle Zeile von Cursorposition bis zum Ende,
- dG lösche Text von aktueller Zeile bis zum Textende,
- d1G lösche Text von aktueller Zeile bis zum Textanfang,
- ndd lösche n Zeilen ab aktueller Zeile vorwärts, Default n = 1,
- nr (replace) ersetze n Zeichen ab Cursorposition, Default n = 1,
- R ersetze beliebig viele Zeichen ab Cursorposition,
- ncw (change) ersetze n Wörter ab Cursorposition, Default n = 1,
- ncc ersetze n Zeilen ab aktueller Zeile, Default n = 1,
- u (undo) nimm jüngstes textänderndes Kommando zurück,
- U stelle die aktuelle Zeile wieder her,
- :n,m s/regex/ersatz/g <cr> (substitute) ersetze im Text von Zeile n bis Zeile m das Muster (regulärer Ausdruck) *regex* durch die – eventuell leere – Zeichenkette *ersatz*, und zwar global, nicht nur beim ersten Auftreten in der Zeile.

Das letzte Kommando soll mit einem Beispiel verdeutlicht werden:

```
:1,$ s/129\.13\.118\.15/172.22.132.26/g
```

Die Aufgabe ist, in einem Text überall die IP-Adresse 129.13.118.15 durch 172.22.132.26 zu ersetzen. Nach dem Doppelpunkt werden Anfangszeile (1) und Endzeile (\$) genannt. Das zu ersetzende Muster 129.13.118.15 wird stets als regulärer Ausdruck verstanden. In diesen hat der Punkt eine Sonderbedeutung, die ihm durch Voranstellen des Gegenschrägstrichs genommen werden muss. Der Ersatz kann nur eine feste Zeichenkette sein, auch die leere. Wir lernen das Kommando genauer in Abschnitt 5.9 *Streaming Editor* auf Seite 205 kennen.

Wollen Sie die Kontrolle darüber behalten, was ersetzt wird, bietet sich folgendes Vorgehen an:

1. Suchen Sie vom Dateianfang ausgehend mit dem Kommando `/muster` das erste Auftreten des zu ersetzenden Musters,
2. ändern Sie manuell mittels genau eines Editierkommandos wie `cw` das Muster,
3. suchen Sie mittels `n` das nächste Auftreten des Musters,
4. wiederholen Sie mittels `.` (Punkt) das jüngste ändernde Kommando oder übergehen Sie diesen Schritt,
5. wiederholen Sie die beiden vorstehenden Schritte, bis Sie am Ende der Datei angelangt sind.

Wenn Sie *Anna* durch *Otto* ersetzen, denken Sie daran, dass sich auch Personal- und Relativpronomina ändern.

5.3.8 Pufferspeicher und Zwischendateien

Der Editor verwaltet neben dem Pufferspeicher, der die Arbeitskopie des gesamten Textes enthält, mehrere Puffer, in denen Textausschnitte vorübergehend abgelegt werden. Ebenso können auch Textauschnitte in Dateien herausgeschrieben und von diesen wieder an anderer Stelle eingelesen werden.

In einem namenlosen Puffer befindet sich immer der zuletzt gelöschte Text (Wörter, Zeilen). Jedes neue Löschkommando überschreibt den Puffer. Ferner lässt sich der Puffer gezielt füllen, ohne die jeweiligen Zeilen zu löschen. Dazu dient das `yank`-Kommando:

- `yw` kopiert ein Wort in den namenlosen Puffer,
- `nY` kopiert `n` Zeilen in den namenlosen Puffer, Default `n = 1`.

Den Pufferinhalt schreibt man folgendermaßen an beliebiger Stelle zurück:

- `p` (`put`) schreibe den Pufferinhalt rechts (bei Wörtern) bzw. unterhalb (bei Zeilen) der Cursorposition zurück,
- `P` schreibe den Pufferinhalt links (bei Wörtern) bzw. oberhalb (bei Zeilen) der Cursorposition zurück.

Der Puffer kann beliebig oft ausgelesen werden. Nützlich, wenn man Zeilen umsortiert oder Zeilen mit geringen Änderungen mehrmals benötigt.

In neun, von 1 bis 9 durchnummerierten Puffern speichert der Editor die durch die neun jüngsten Zeilen-Löschkommandos gelöschten Zeilen. Die Zeilen werden zurückgewonnen, indem man den obigen `put`-Kommandos Anführungszeichen und die Puffernummer voranstellt:

```
"5p
```

schreibt unterhalb der gerade aktuellen Zeile die beim fünftjüngsten Zeilen-Löschkommando gelöschten Zeilen zurück. Hat man vergessen, in welchem der Puffer sich der gewünschte Text befindet, kann man sich mit nachstehender Kommandofolge durch die Puffer durchhangeln:

```
"1pu.u.u.
```

usw. Der Punkt nach dem undo-Kommando bewirkt hier ein Fortschreiten in den nummerierten Puffern.

Schließlich kennt der Editor noch 26 mit den Buchstaben von a bis z benannte Puffer. Diese füllt der Editor nur auf Anweisung. Dazu sind den entsprechenden Kommandos (yank, delete, change) ein Anführungszeichen und der dem Puffer zugehörige Buchstabe voranzustellen:

- "a2Y kopiere die aktuelle und die folgende Zeile in den Puffer a,
- "b4dd lösche die aktuelle und drei folgende Zeilen und kopiere sie zusätzlich in den Puffer b,
- "c2cw ersetze zwei Wörter und kopiere sie zusätzlich in den Puffer c.

Schreibt man die Namen der Puffer groß, wird der Text an bereits im Puffer befindlichen Text angehängt. Zum Herausschreiben dienen die put-Kommandos mit vorangestellten Anführungszeichen und Puffernamen:

- "ap kopiere den Inhalt des Puffers a rechts bzw. unterhalb des Cursors,
- "bP kopiere den Inhalt des Puffers b links bzw. oberhalb des Cursors.

Die benannten Puffer bewahren ihren Inhalt, wenn man innerhalb des Editors den zu bearbeitenden Text (die Arbeitskopie) mittels des Kommandos :e wechselt. So lassen sich einfach Textausschnitte zwischen Textdateien austauschen.

Alle Puffer werden beim Verlassen des Editors gelöscht. Will man Textausschnitte längerfristig aufbewahren, schreibt man sie in eine Datei. Hierzu markiert man die erste und die letzte Zeile des zu kopierenden Textausschnittes mittels ma und mb und schreibt den Textausschnitt in eine Datei beliebigen Namens:

```
: 'a, 'bw datei
```

a und b sind hier keine Puffer, sondern Marken, deshalb stehen Hochkommas. Eingelesen wird eine beliebige Datei unterhalb der aktuellen Zeile mit:

```
:r datei
```

Die Datei bleibt dabei erhalten; sie ist außerhalb des Editors mittels rm zu löschen.

5.3.9 Sonstige Kommandos

Wir haben noch einige häufiger gebrauchte vi-Kommandos zu erläutern, die sich keiner der vorgenannten Gruppen zuordnen lassen:

- J (join) verbinde die aktuelle Zeile mit der folgenden,
- . (Punkt, dot) wiederhole das jüngste Kommando, das die Arbeitskopie verändert hat (funktioniert nicht mit Kommandos, die mit einem Doppelpunkt beginnen),
- :sh (shell) stelle vorübergehend eine Shell zur Verfügung (aus-shellen),
- :!command führe das Shellkommando command aus, beispielsweise date oder ls -l,
- :ve zeige die Versionsnummer des Editors an,

- `<ctrl>-g` zeige den Dateinamen und die Zeilennummer an,
- `:e datei <cr>` (edit) lies die Datei `datei` ein und bearbeite sie. Das setzt voraus, dass die vorangehende Arbeitskopie mit `:w` zurückgeschrieben worden ist,
- `:help` zeige einen Hilfetext an.

Einige Zeichen wie `@` oder `#` können vom Betriebssystem her mit Bedeutungen versehen sein (`@` = lösche Zeile, line kill), die sich auch im Editor auswirken. Das ist eher verwirrend als hilfreich. Wenn es sich nicht abstellen lässt (man `tset`), müssen die Sonderzeichen durch andere Darstellungen ersetzt werden, siehe Abschnitt C *Metazeichen* auf Seite 454.

Arbeitet man mit dem `vim` auf einer Tastatur ohne Umlaute (US-Tastatur) und braucht gelegentlich einen Umlaut, so gibt man `<ctrl>-<k>` ein, ein Fragezeichen erscheint auf dem Schirm, dann tippt man `<:> <a>` bzw. einen anderen Grundlaut ein und erhält den Umlaut. Das Eszett erzeugt man mit `<s> <s>`.

5.3.10 Zusammenarbeit vim – ispell

Es ist wünschenswert, Tipp- und Rechtschreibfehler schon bei der Eingabe zu erkennen und zu verbessern. Das unterbricht zwar den Schreibfluss, kostet aber weniger Zeit als eine nachträgliche Korrektur. Dazu müsste der Editor etwas von Rechtschreibung verstehen. Nun sind die Editoren ohnehin schon recht dicke Brocken, außerdem ist das Rad in Form von Rechtschreibprüfern wie `ispell` schon erfunden, siehe Abschnitt 5.13.4 *Rechtschreibung prüfen* auf Seite 231. Optimal erscheint die Zusammenarbeit eines Editors mit einem Prüfprogramm. Lösungen für `vim` sind zu finden unter:

- http://vim.sourceforge.net/scripts/script.php?script_id=195, von CHARLES E. CAMPBELL,
- <http://home.tu-clausthal.de/~mwra/vim/spell.vim>, ein Vorschlag von RALF ARENS,
- http://hermitte.free.fr/vim/VS_help.html, von LUC HERMITTE.

Ein Versuch mit dem Skript `Ispell.vim` von RALF ARENS als `.vimrc` verlief insofern erfolgreich, als man aus dem Editor heraus `ispell` aufrufen konnte. Kleinere Anpassungen waren wegen der neuen deutschen Rechtschreibung (`german -> ngerman`), wegen `LATEX` und wegen des auf der US-Tastatur nicht vorhandenen, in den Mappings am Ende des Skripts verwendeten Buchstabens `ä` (`-> g`) erforderlich. Aber das war noch nicht das Gesuchte. Der Prüfer sollte parallel zum Schreiben mitlaufen (check on the fly).

Um das Skript `vimspell.vim` von LUC HERMITTE einzurichten, gibt man in einer beliebigen Editiersitzung das Kommando `:h add-global-plugin`, tut wie dort geheißen und kopiert das Skript nach `$HOME/.vim/plugin/`. An `.vimrc` ist nichts zu ändern. Eine Dokumentation steht am Ende des Skripts. Beim nächsten Editierversuch – der genau diese Zeilen hier betraf – wurden verdächtige Wörter unterstrichen, on the fly, bei Korrekturen nach einer kurzen Denkpause.

\LaTeX -Kommandos blieben unbeanstandet. Sieht hoffnungsvoll aus. Jetzt fehlt nur noch die Integration eines \LaTeX -Prüfers. Wenn man die Syntax-Hervorhebung eingeschaltet hat, sollte man nicht auch noch verdächtige Wörter markieren lassen, das Erscheinungsbild wird sonst sehr unruhig. Entweder – oder.

Das war etwa die Hälfte der verfügbaren vi-Kommandos. Spezialitäten des vim oder nvi sind kaum berücksichtigt. Fangen Sie mit wenigen vi-Kommandos an, beobachten Sie, welche Vorgänge Sie häufig brauchen und suchen Sie nach dem einfachsten Weg, den Editor zu der gewünschten Tätigkeit zu bewegen. Vermeiden Sie, seltene Aufgaben zu optimieren. Sollte Ihnen das Gebotene nicht reichen, greifen Sie auf Monografien oder die Referenzliteratur zurück. Wir haben noch andere Themen vor uns.

5.3.11 VimOutliner – ein Outline Prozessor

Ein Outline Prozessor ist ein Werkzeug, um Entwürfe, Umrisse oder Grundzüge von umfangreichen Textdokumenten zu erstellen und zu ändern. Die Arbeit am Buch begann mit einem Entwurf der Kapitel, allerdings noch mit Bleistift und Papier. Als so viel Stoff vorhanden war, dass \LaTeX ein Inhaltsverzeichnis schreiben konnte, diente dieses als Outline, in das – immer noch mit Bleistift – Einfälle und Änderungen eingetragen wurden. Ein Outline spiegelt mehr die Gedanken wider – unabhängig von einer Kapitelstruktur – während ein Inhaltsverzeichnis die Einteilung des Manuskriptes darstellt, was nicht dasselbe ist, aber im vorliegenden Fall ausreichend genau übereinstimmt. Falls es zu einer zweiten Auflage kommt, wird sie natürlich mit einem Outline Prozessor erstellt. Seine Aufgaben greifen in die Gebiete von Mindmapping (Abschnitt 11.11 *Mindmapping* auf Seite 398) und Jottern oder Notizbüchern (Abschnitt 11.9.2 *Notizbuch, Personal Information Manager* auf Seite 386) über.

Das Debian-Paket *vim-vimoutliner* stellt einige Skripte, Konfigurationsdateien und Werkzeuge bereit, die den Editor vim um Fähigkeiten zum Outlining bereichern. Der VimOutliner (VO) ist auf <http://www.vimoutliner.org/> beheimatet, Dokumentation findet sich nach Einrichtung des Pakets in `/usr/share/doc/vim-vimoutliner/`. Manualseiten gibt es nur zu den einzelnen Werkzeugen:

- `ot12docbook`, ein Perl-Skript, das die Ausgabe von VO in das DocBook-XML-Format umwandelt,
- `ot12html`, ein Python-Skript, das die Ausgabe von VO nach HTML umwandelt,
- `ot12pdb`, ein Perl-Skript, das die Ausgabe von VO in eine AddressDB.pdb-Datei für einen Palm-Rechner umwandelt,
- `vo_maketags`, ein Perl-Skript, das eine vim-Tag-Datei erzeugt.

Ein Tutorial erhält man, indem man aus dem Editor heraus das Kommando `:help vimoutliner` eingibt. Auf der noch frischen und daher stellenweise inhaltsarmen Website stehen weitere Skripte und Plugins zum Herunterladen bereit, beispielsweise um aus einer Outliner-Datei eine Präsentation im Format von OpenOffice.org Impress zu machen.

5.4 Universalgenie: Emacs

5.4.1 Schnellstart

Um den Emacs aufzurufen, gibt man einfach

```
joe@debian:~$ emacs
```

ein. Wer will, kann noch einen Dateinamen dahinter hängen. Nach spätestens ein paar Sekunden sollte sich ein Fenster öffnen - mit dem Schnellstart ist das bei Emacs so eine Sache, denn üblicherweise ruft man ihn am Anfang einer Sitzung auf und schließt ihn erst wieder, wenn man sich ausloggt. Falls Sie mit einer grafischen Benutzeroberfläche arbeiten, öffnet sich ein Fenster, dessen weitgehend selbsterklärende Menüs und Piktogramme einen schnellen Einstieg erlauben. Andernfalls öffnet man mit `<ctrl>-<x> <ctrl>-<f>` (zwei Tastenkombinationen hintereinander) erstmal eine Datei, falls nicht schon beim Aufruf von Emacs geschehen. Dabei vervollständigt `<Tab>` unvollständig eingegebene Verzeichnis- und Dateinamen, ein Entgegenkommen der Bash.

Dann kann man direkt mit dem Schreiben loslegen; emacs kennt keine getrennten Kommando- und Eingabemodi wie vi. Eine neue Zeile beginnt man mit `<cr>` (nach dem Cursor) oder `<ctrl>-<o>` (davor). Kleine Tippfehler korrigieren wir mittels Pfeiltasten und Backspace oder `` bzw. `<Entf>`, ganze Zeilen lassen sich mit `<ctrl>-<k>` löschen und mit `<ctrl>-<y>` wieder einfügen. Bei größeren Missgeschicken hilft `<ctrl>-<x> <u>`, welches das letzte Kommando rückgängig macht. Vermissen Sie in Ihrem Text etwas, können Sie mit `<ctrl>-<s>` eine Suche vorwärts und mit `<ctrl>-<r>` dasselbe rückwärts beginnen. Schließlich drücken wir `<ctrl>-<x> <ctrl>-<s>` zum Speichern des Geschriebenen und `<ctrl>-<x> <ctrl>-<c>`, um Emacs zu beenden. Verlässt man Emacs, ohne vorher zurückzuschreiben, wird man gefragt, ob man nicht doch abspeichern möchte.

Falls Sie sich in irgend einem Unterkommando verfangen (was früher oder später passiert), kommen Sie mit `<ctrl>-<g>` oder im schlimmsten Fall `<Esc> <Esc> <Esc>` wieder heraus. Abschließend seien die recht umfangreichen Hilfsfunktionen von Emacs erwähnt: Drückt man `<ctrl>-<h> <a>`, kann man einen Begriff eingeben, zu dem Emacs dann Hilfe sucht. Dabei werden Sie meistens mehrere Befehle zu sehen bekommen, die Sie eintippen können, wenn Sie vorher ein `<alt>-<x>` eingeben. Nach ein paar Mal wird das zu langweilig, und man fragt Emacs per `<ctrl>-<h> <w>`, auf welcher Taste der jeweilige Befehl zu finden ist. Wollen Sie noch mehr lernen, zeigt Ihnen `<ctrl>-<h> <t>` eine Einführung in den Umgang mit Emacs an, bei einer deutschen Installation auf Deutsch.

5.4.2 Übersicht

Ursprünglich nur aus einer Sammlung von Makros für einen anderen Editor (daher der Name *Editor MACroS*) bestehend, hat sich der Emacs unter der Federführung von

RICHARD STALLMAN zur eierlegenden Wollmilchsau unter den Texteditoren entwickelt. Inzwischen bringt er seinen eigenen Interpreter für eine Programmiersprache mit (LISP) und besitzt spezielle Modi für alle häufiger verwendeten Textformate (insbesondere auch \LaTeX). Sie können sogar Mail und News damit lesen, den vi emulieren oder Quellcode damit debuggen. Dementsprechend hat er nicht gerade den Ruf, besonders schnell zu laden, wovon diverse Verballhornungen des Wortes *Emacs* wie *Emacs Makes A Computer Slow* zeugen. Der große Unterschied zwischen vi und emacs besteht darin, dass letzterer nicht über einen getrennten Kommando- und Eingabemodus verfügt, sondern die Editierkommandos über das gleichzeitige Drücken von `<ctrl>` oder `<alt>` und anderen Tasten aufgerufen werden (key-cords). Das erfordert eine gewisse Fingerakrobatik, weswegen *Emacs* manchmal auch als *Escape-Meta-Alt-Control-Shift* interpretiert wird.

Der Emacs ist in mehreren Varianten erhältlich, von denen die aus dem GNU-Projekt (<http://www.gnu.org/software/emacs/>, Debian-Paket *emacs21*) die ursprünglichste ist. Daneben hat sich inzwischen XEmacs (<http://www.xemacs.org/>, Debian-Paket *xemacs21*), ehemals Lucid Emacs, etabliert. Die Entwicklung von XEmacs wurde vor allem von JAMIE ZAWINSKI und BEN WING vorangetrieben; er verfügt nur über eine grafische Oberfläche, lässt sich also nicht mehr auf einem alphanumerischen Terminal benutzen. XEmacs hat sich bei Version 19 von GNU Emacs aufgrund von Meinungsverschiedenheiten unter den Entwicklern abgespalten und bietet nach eigener Aussage mehr Funktionalität. Seither hat aber auch GNU Emacs wieder nachgelegt, in der aktuellen Version 21.4.1 (Stand Mitte 2005) muss man schon lange suchen, um größere Unterschiede festzustellen. Weiterhin gibt es Portierungen auf andere Systeme einschließlich DOS sowie die Variante Microemacs (<http://uemacs.tripod.com/> und <http://www.jasspa.com/>, nicht bei Debian).

Wer vor dem gewaltigen Emacs erschrickt oder knapp an Speicher und Prozessorleistung ist, kann sein Glück mit *Jonathan's Own Version of Emacs* versuchen, aus den Debian-Paketen *jove* und *xjove* (X11-Frontend zu *jove*). Der kleine Bruder verfügt über die wesentlichen Fähigkeiten des großen. Umsteigen kann man immer.

Bei Fragen zu Emacs empfiehlt sich zu allererst das Emacs-FAQ, zu erreichen unter <http://www.gnu.org/software/emacs/emacs-faq.text>, oder das XEmacs-FAQ unter <http://www.xemacs.org/FAQ/xemacs-faq.html>. Wesentlich ausführlicher gestaltet sich das Emacs-Handbuch (<http://www.gnu.org/software/emacs/manual/emacs.html>). Weitere Verweise hält <http://www.gnu.org/software/emacs/#Help> bereit.

5.4.3 Einrichtung und Konfiguration

Unter Debian kann man wie gewohnt mit `apt-get install emacs21` oder `apt-get install xemacs21` das entsprechende Paket installieren. Falls der Emacs nicht fertig eingerichtet vorliegt, muss man sich selbst darum bemühen. Man holt ihn sich als Tarball per Anonymous FTP oder Web-Browser von <http://ftp.informatik.rwth-aachen.de/pub/gnu/>, von

informatik.tu-muenchen.de/pub/comp/os/unix/gnu/ oder von anderen Servern und richtet das Ganze wie in Abschnitt 1.8.3 *tar-Archive* auf Seite 50 beschrieben ein. Aktuell ist Mitte 2005 die Version 21.4.1, sowohl bei GNU wie bei Debian. Unter Debian findet man in `/etc/alternatives/emacs` einen Hinweis, wo der Editor im Dateisystem ruht.

Die systemweite Konfiguration findet sich im Verzeichnis `/etc/emacs/` oder `/etc/emacs21/`. XEmacs besitzt sein eigenes Verzeichnis `/etc/xemacs21`, lädt aber normalerweise auch die Einstellungen von GNU Emacs. In diesen Verzeichnissen sollte man mit Änderungen sparsam umgehen, hauptsächlich gehören hier Einstellungen hin, die die Installation von Emacs selbst betreffen (Verzeichnisnamen, Terminaleinstellungen). Richtig unfreundlich verhalten sich Systemverwalter, die die Standard-Tastenbelegungen ohne Rücksicht auf die Benutzer ändern.

Die benutzerspezifischen Einstellungen lagern in der Datei `$HOME/.emacs`. Spätestens, wenn Sie diese anschauen, werden Sie zum ersten Mal mit dem LISP-Quellcode konfrontiert, mit dem sich Emacs nach Belieben konfigurieren und erweitern lässt. Da sich darüber viele Benutzer beschwert haben, erreicht man inzwischen mit dem Befehl `<alt>-<x> customize <cr>` einen Konfigurationsmodus, der die gewünschten Änderungen in LISP formuliert und wieder speichert. Noch einfacher geht das unter grafischen Oberflächen: Das *Options*-Menü kennt eine ganze Menge verschiedener Einstellungen und bietet mit *Save Options* das Zurückschreiben an.

Trotzdem seien hier ein paar Beispiele angeführt, wie `.emacs` aussehen könnte. Wir verzichten darauf, die Syntax von LISP näher zu beschreiben, da das den Rahmen des Buches sprengen würde. Zuerst müssen wir allerdings das Konzept von Haupt- und Nebenmodi (*major* bzw. *minor modes*) kurz erklären: Jedes Fenster, in dem man eine Datei bearbeitet, befindet sich in genau einem Hauptmodus. Diesen kann man mit `<alt>-<x> neuer-modus <cr>` ändern. Zur Verfügung stehen unter anderem `latex-mode` (für \LaTeX -Quellen, Teil von *AUCTex*), `viper-mode` (zur vi-Emulation) oder `c-mode` (für C-Programme). Daneben kann man zu einem Hauptmodus einen oder mehrere Nebenmodi aktivieren, z. B. `auto-fill-mode` zum automatischen Zeilenumbruch oder `font-lock-mode` zur Syntax-Hervorhebung, vorausgesetzt, Emacs kennt das jeweilige Dateiformat.

Wenn Sie nun beispielsweise bei jedem Aufruf von Emacs die Zeile und Spalte, in der sich der Cursor befindet, in der Statuszeile angezeigt bekommen wollen, schreiben sie einfach

```
(setq line-number-mode t)

(setq column-number-mode t)
```

in Ihre `.emacs`-Datei. Andere internen Variable von Emacs lassen sich auf gleiche Art und Weise setzen. Falls Sie nur im `tex-mode` und nicht permanent die Syntax-Hervorhebung einschalten möchten, erledigt das

```
(add-hook 'tex-mode-hook (lambda () (font-lock-mode)))
```

für Sie. Auf ähnliche Weise lässt sich der automatische Zeilenumbruch ausschließlich für den Standard-Modus einschalten:

```
(add-hook 'fundamental-mode (lambda () (auto-fill-mode)))
```

Zu guter Letzt lässt sich mit

```
(global-set-key [(meta g)] (quote goto-line))
```

der Befehl `goto-line`, mit dem man in eine bestimmte Zeile springt, auf die Tastenkombination `<alt>-<g>` legen. Mit anderen Tasten funktioniert das genauso, für `<ctrl>-<t>` gibt man (`control t`) anstatt (`meta g`) ein. Wenn Sie jetzt Appetit auf mehr von dieser Sprache mit den unzähligen Klammern bekommen, bietet sich die Einführung *Programming in Emacs Lisp* (<http://www.gnu.org/software/emacs/emacs-lisp-intro/emacs-lisp-intro.html>) an.

5.4.4 Aufrufen und Beenden

Meistens ruft man Emacs einfach mit

```
joe@debian:~$ emacs
```

auf und öffnet mit `<ctrl>-<x>` `<ctrl>-<f>` eine Datei. Wahlweise schreibt man einen oder mehrere Dateinamen gleich hinter den Befehl:

```
joe@debian:~$ emacs dateiname1 dateiname2 ...
```

Die Dateien brauchen noch nicht zu existieren und werden dann beim ersten Sichern angelegt. Man arbeitet sowieso immer mit einer Kopie im Arbeitsspeicher, nur beim Speichern werden Daten auf dem Massenspeicher verändert. Viele Optionen beim Aufruf kennt Emacs nicht, am interessantesten ist die Möglichkeit, mittels

```
joe@debian:~$ emacs -f lisp-funktion
```

einen Emacs-Befehl sofort auszuführen, z. B. `auto-fill-mode`, um den automatischen Zeilenumbruch sofort einzuschalten.

Ein paar Worte zur Terminologie von Emacs: Mit Puffer wird die Kopie einer Datei im Arbeitsspeicher bezeichnet. Nur diese lässt sich bearbeiten. Was sie zu sehen bekommen, wenn Sie Emacs aufrufen, ist ein Fenster, das einen Puffer anzeigt. Man kann mit `<ctrl>-<x>` 2 die Anzeige auch in zwei Fenster aufspalten (oder mehr, aber das wird unübersichtlich). Dann wechselt man mit `<ctrl>-<x>` o dazwischen hin und her. Hat man genug, bringt `<ctrl>-<x>` 1 das Fenster, in dem sich der Cursor gerade befindet, wieder auf volle Größe. Haben Sie mehrere Dateien geöffnet, das heißt auch mehrere Puffer, kommt man mit `<ctrl>-<x>` b von einem zum anderen. Zu allem Überflus bietet Emacs noch die Möglichkeit, innerhalb eines Aufrufs bzw. Prozesses mehrere Fenster der grafischen Benutzeroberfläche zu erzeugen, die dann Frames genannt werden. Das wird aber eher selten gebraucht.

Das Zurruckschreiben eines Puffers auf die Festplatte geschieht mittels `<ctrl>-<x>` `<ctrl>-<s>`, bei mehreren Dateien erledigt das `<ctrl>-<x>`

s auf einmal. Abgesehen davon legt Emacs selbständig Sicherheitskopien der bearbeiteten Dateien an. Diese bekommen an den ursprünglichen Dateinamen einfach eine Tilde angehängt. Außerdem wird jeder Puffer von Zeit zu Zeit automatisch gesichert, was dann Dateinamen der Form #foobar# ergibt. Wünschen Sie sich beim Speichern einen anderen Namen als den aktuellen des Puffers, können Sie mit `<ctrl>-<x> <ctrl>-<w>` einen neuen wählen. Sollten Sie es sich einmal ganz anders überlegen, können Sie mit `<ctrl>-<x> k` einen Puffer löschen, auch ohne ihn zurückzuschreiben.

Am Ende des Tages lässt dann `<ctrl>-<x> <ctrl>-<c>` auch Emacs seine Ruhe finden. Wollen Sie jedoch nur einmal kurz was anderes erledigen, schickt ihn `<ctrl>-<z>` in den Hintergrund. Das ist besonders nützlich beim Arbeiten auf einem alphanumerischen Terminal, da sich dort nicht so einfach wie unter einer grafischen Oberfläche zwischen verschiedenen Prozessen hin- und herschalten lässt. Aber auch dann erspart diese Methode lange Ladezeiten beim Starten von Emacs.

5.4.5 Navigieren im Text

Wie in jedem Texteditor spielt das Bewegen des Cursors eine zentrale Rolle. Natürlich erreicht man mit den Cursortasten jede beliebige Stelle in einem Text, aber sobald der Text länger als eine Seite wird, geht das langsam auf die Nerven oder führt zu einer Sehnenscheidenentzündung (oder beides). Stattdessen bietet Emacs folgende vier Tastenkombinationen an:

`<ctrl>-<f>` Bewege den Cursor ein Zeichen nach rechts (forward).
`<ctrl>-` Ein Zeichen nach links (back).
`<ctrl>-<n>` Eine Zeile nach unten (next).
`<ctrl>-<p>` Eine Zeile nach oben (previous).

Leider liegen diese Kommandos auf der Tastatur nicht ganz so dicht beieinander wie beim `vi`; das ist aber immer noch ergonomischer als jedes Mal die Hand aus der üblichen Zehn-Finger-Schreibhaltung zu den Cursortasten hinüber zu bewegen. Weitere häufig benutzte Befehle sind:

`<ctrl>-<a>` Bewege den Cursor an den Anfang der Zeile,
`<ctrl>-<e>` ans Ende der Zeile,
`<alt>-<f>` ein Wort weiter,
`<alt>-` ein Wort zurück,
`<alt>-<a>` an den Anfang des Satzes unter dem Cursor,
`<alt>-<e>` an das Ende des Satzes unter dem Cursor,
`<alt>-<}>` einen Absatz weiter (getrennt durch eine Leerzeile),
`<alt>-<{>` einen Absatz zurück,
`<alt>-<=>` an den Anfang des Puffers,
`<alt>-<=>` an das Ende des Puffers,
`<alt>-<x> goto-line <cr> n <cr>` springe zu Zeile n (bei XEmacs auch unter `<alt>-<g>`).

Will man weniger den Cursor bewegen als den angezeigten Textausschnitt, sind folgende Kommandos nützlich:

<ctrl>-<v> Bewege den Textausschnitt um eine Seite weiter,
 <alt>-<v> dasselbe eine Seite zurück,
 <ctrl>-<l> zentriere die Zeile auf dem Schirm, in der sich der Cursor befindet.

Auch die Suchfunktionen kommen bei Emacs nicht zu kurz:

<ctrl>-<s> Inkrementelle Suche vorwärts (search), das heißt Emacs sucht schon beim Tippen des Suchwortes,
 <ctrl>-<r> inkrementelle Suche rückwärts (reverse search),
 <ctrl>-<alt>-<s> Vorwärtssuche nach einem regulären Ausdruck, siehe Abschnitt 4.1.2 *Reguläre Ausdrücke* auf Seite 116,
 <ctrl>-<alt>-<r> Rückwärtssuche nach einem regulären Ausdruck,
 <ctrl>-<s> wiederhole eine begonnene Suche in Vorwärtsrichtung,
 <ctrl>-<r> wiederhole eine begonnene Suche in Rückwärtsrichtung,
 <cr> höre mit der Suche auf und lasse den Cursor dort, wo er sich momentan befindet,
 <ctrl>-<g> brich die Suche ab und setze den Cursor an seine ursprüngliche Position.

An dieser Stelle dürfte auffallen, dass viele Kommandos, die sich nur durch den Gebrauch von <alt> anstelle von <ctrl> unterscheiden, auch einen logischen Zusammenhang besitzen: <ctrl>-<f> bewegt den Cursor ein Zeichen weiter, <alt>-<f> ein ganzes Wort. Solche Analogien begegnen uns oft, wenn wir noch mehr Tastenkombinationen von Emacs lernen. Ähnlich besitzen manche Tastenkombinationen ein kontextabhängiges, aber immer logisches Verhalten: <ctrl>-<s> beginnt eine neue Suche, außer man befindet sich gerade schon in einer. Dann will man normalerweise nicht die eben begonnene abbrechen.

5.4.6 Text ändern

Da man bei Emacs im Gegensatz zu vi nicht erst einen Befehl eingeben muss, um Text schreiben zu können, verfügt er naturgemäß über weniger Kommandos zum Einfügen von Text. Diejenigen zum Löschen sind aber nicht minder zahlreich:

<ctrl>-<o> Füge eine neue Zeile vor dem Cursor ein (open),
 <ctrl>-<t> vertausche die beiden Zeichen vor dem Cursor,
 <alt>-<t> vertausche die letzten beiden Wörter,
 <Backspace> lösche das Zeichen vor dem Cursor,
 <ctrl>-<d> lösche das Zeichen nach dem Cursor (delete),
 <alt>-<Backspace> lösche das Wort vor dem Cursor,
 <alt>-<d> lösche das Wort nach dem Cursor,
 <ctrl>-<k> lösche alle Zeichen vom Cursor bis zum Zeilenende (kill). Falls Zeile leer, lösche die ganze Zeile,
 <ctrl>-<x> <u> nimm letzte Textänderung zurück (undo).

Weiterhin beherrscht Emacs auch das Finden und Ersetzen von Zeichenketten. Dazu drückt man entweder `<alt>-<%>`, oder man gibt sogar `<alt>-<x> query-replace-regexp <cr>` ein, um reguläre Ausdrücke benutzen zu können. Man wird dann nach dem Suchmuster und dem einzusetzenden Text gefragt. Bei einem Treffer antwortet man entweder `<y>` (yes), um diese Stelle zu ersetzen, `<n>` (no), um fortzufahren, `<!>`, um alle restlichen Treffer ohne Nachfrage zu ersetzen, oder `<Esc>`, um die Suche abubrechen. Außerdem erweisen sich bei größeren Mengen an zu löschendem Text die im nächsten Abschnitt besprochenen Befehle als nützlich.

5.4.7 Cut & Paste

Mit Cut & Paste bezeichnet man Funktionen zum Ausschneiden und Wiedereinfügen von Textpassagen. Das Kommando `<ctrl>-<k>` haben Sie schon kennen gelernt, es gibt aber etliche mehr:

`<ctrl>-<space>` Markierung setzen, dann den Cursor bewegen...
`<ctrl>-<w>` ...und den markierten Text löschen,
`<alt>-<w>` markierten Text in einen Zwischenspeicher kopieren, aber nicht löschen,
`<ctrl>-<y>` zuletzt gelöschten Text aus dem Zwischenspeicher vor dem Cursor einfügen (yank),
`<alt>-<y>` zuletzt eingefügten Text durch den davor gelöschten Text ersetzen,
`<ctrl>-<x>` `<ctrl>-<x>` Cursor und Markierung vertauschen,
`<ctrl>-<x>` h den ganzen Puffer markieren.

Die Arbeit mit diesen Befehlen ist recht einfach. Falls Sie diese Operationen schon einmal in einem anderen Texteditor ausgeführt haben, sollten sie Ihnen vertraut vorkommen. Andernfalls probieren Sie sie mit einer simplen Textdatei aus.

5.4.8 Sonstiges

Die Befehle, die über obige Grundkenntnisse von Emacs hinausgehen, aber von einem der Autoren dieses Buchs häufiger benutzt werden, beinhalten:

`<ctrl>-<u>` zahl Führt den nächsten Befehl zahl-mal aus. Nützlich insbesondere beim Löschen von Text (etwa `<ctrl>-<u> 2 <ctrl>-<k>`),
`<ctrl>-<x>` i fragt nach einer Datei, die dann vor dem Cursor eingefügt wird,
`<ctrl>-<x>` `<ctrl>-` listet alle offenen Puffer auf,
`<alt>-<q>` bricht die Zeilen des Textabschnitts unter dem Cursor automatisch um,
`<alt>-<!>` führt ein nachfolgend einzugebendes Shellkommando aus,
`<alt>-<x>` `ispell-buffer` lässt `ispell` über den aktuellen Puffer laufen (Rechtschreibung prüfen).

Außerdem seien drei Befehle erwähnt, um Makros aufzunehmen und abzuspielen:

<ctrl>-<x> <(> Makro-Aufnahme beginnen.
 <ctrl>-<x> <)> Makro-Aufnahme beenden.
 <ctrl>-<x> <e> zuletzt aufgenommenes Makro ausführen (execute).

Jetzt kennen Sie erst einen Bruchteil der in Emacs verfügbaren Kommandos. Nächste Anlaufstellen bei der Suche nach Hilfe bieten Emacs selbst (unter <ctrl>-<h> <ctrl>-<h>) und das Internet (<http://www.gnu.org/software/emacs/> sowie <http://www.xemacs.org>). Es gibt nichts, was es nicht gibt; angefangen beim *Emacs Psychiatrist* im Help-Menü von GNU Emacs über Erweiterungen für exotische Alphabete bis hin zu Emacspeak, einer Sprachausgabe für sehbehinderte Benutzer, siehe Abschnitt 9.7.3 *Screenreader* auf Seite 349. Falls Sie wider Erwarten etwas nicht finden sollten, haben Sie immer noch die Möglichkeit, Ihre eigene Erweiterung zu schreiben. Der eingebaute LISP-Interpreter steht anderen Programmiersprachen in nichts nach.

Ob Ihnen nun emacs oder vi besser gefällt, bleibt Ihnen überlassen. Nachdem die Glaubenskriege zwischen den Anhängern der beiden Editoren abgeklungen sind, mehren sich die Stimmen, beide hätten ihre Vorzüge und Nachteile. Falls wir den Eindruck erweckt haben, man könne mit emacs mehr bewerkstelligen, lassen Sie sich nicht täuschen: Auch der vim lässt sich mittlerweile vollständig programmieren.

5.5 Weitere Texteditoren

5.5.1 Einfachst: nano

pico ist ein kleiner Editor, der ursprünglich zu einem Email-Programm gehörte, aber auch selbständig zu gebrauchen ist. Wegen Lizenz-Einschränkungen steht unter Debian nur der Klon nano zur Verfügung, ebenso eine abgemagerte Version nano-tiny. Wer nur einfache, kurze Texte schreibt, kommt mit ihm aus.

5.5.2 Joe's Own Editor (joe)

Der joe⁵ von JOSEPH H. ALLEN soll als Beispiel für eine Vielzahl von Editoren stehen, die im Netz herumschwimmen und entweder mehr können oder einfacher zu benutzen sind als die Standard-Editoren. Er ist nicht auf X11 angewiesen und bringt eine eigene Verhaltensweise in normaler und beschränkter Fassung mit, kann aber auch WordStar, pico oder den Emacs emulieren (nachahmen), je nach Aufruf und Konfiguration. Diese lässt sich in eine Datei \$HOME/.joerc den eigenen Wünschen anpassen.

Der joe kennt keine Modi. Nach dem Aufruf legt man gleich mit der Texteingabe los. Editorkommandos werden durch control-Sequenzen gekennzeichnet. Beispielsweise erzeugt die Folge <ctrl>-<k> und <h> ein Hilfefenster am oberen

⁵Leider hat sich ein HTML-Editor aus Frankreich denselben Namen zugelegt.



Abbildung 5.2. Screenshot des Editors nano

Bildschirmrand. Nochmalige Eingabe der Sequenz löscht das Fenster. Am Ende verlässt man den Editor mittels `<ctrl>-<c>` ohne Zurückschreiben oder mit der Sequenz `<ctrl>-<k>` und `<x>` unter Speichern des Textes. Weitere Kommandos im Hilfefenster oder im Manual.

5.5.3 Der Nirwana-Editor (nedit)

Der `nedit` setzt auf X11 auf und verwendet eine grafische Oberfläche im Stil von Motif. Er ist in vielen Linux-Distributionen enthalten und für weitere UNIXe sowie VMS zu haben. Insbesondere lässt er sich an die Eigenheiten vieler Programmiersprachen anpassen; seine Makro-Sprache ähnelt C. Wer viel programmiert, sollte sich ihn ansehen. Informationen findet man unter <http://nedit.org/>.

5.5.4 Ein Editor für KDE (kate)

Der KDE Advanced Text Editor `kate` ist für alle Arten von Text einsetzbar, beweist jedoch Stärken beim Schreiben von Quellcode wie die Hervorhebung von Syntaxelementen oder das Kollabieren/Expandieren von Funktionen. Er hat mehrere Dokumente gleichzeitig im Griff und lässt sich durch Plugins erweitern. Mit der Option `--help-all` aufgerufen zeigt er alle Optionen an. Vermutlich wird er den KDE-Editor `kwrite` ablösen. Näheres bei <http://kate.kde.org/>.

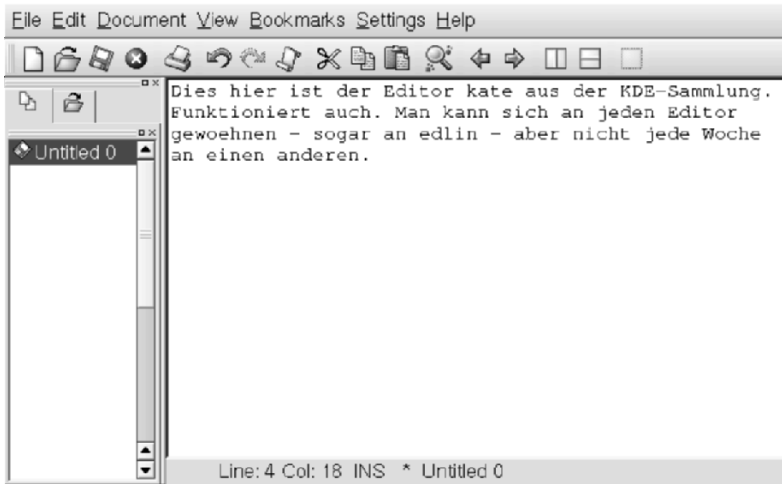


Abbildung 5.3. Screenshot des KDE-Editors kate

5.5.5 Ein Editor für GNOME (gedit)

Der aus dem GNOME-Projekt stammende Editor `gedit` hinterließ einen guten Eindruck. Die Konfiguration kann im laufenden Betrieb geändert werden (beispielsweise Schriftgröße und -art). Alle wesentlichen Funktionen eines Editors sind einfach zu erreichen. Auch unter KDE treten keine Probleme auf. Wenn man ohnehin mit einer grafischen Benutzeroberfläche und X11 arbeitet, ist `gedit` eine gute Wahl.

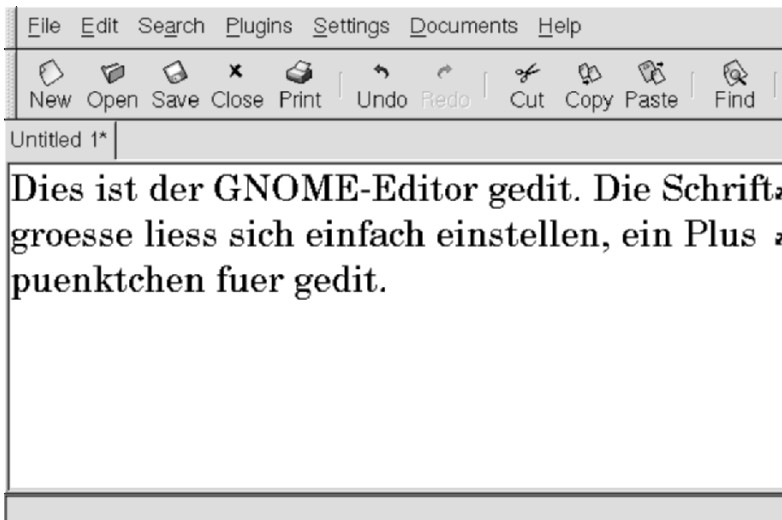


Abbildung 5.4. Screenshot des GNOME-Editors gedit

5.5.6 aXe – ein einfacher Editor unter X11

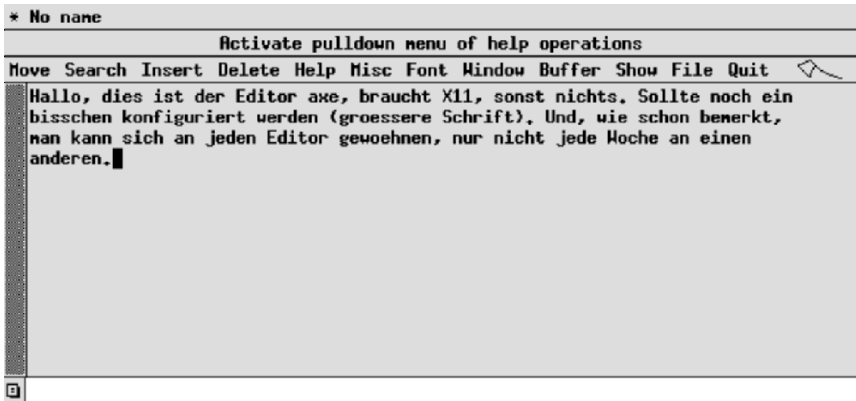


Abbildung 5.5. Screenshot des X11-Editors axe

Der Editor `axe` ist eine Weiterentwicklung des Editors `xedit`. Er setzt X11 voraus und macht vom X Athena Widget Set Gebrauch.

Das waren noch längst nicht alle Editoren aus der Debian-Distribution. Außerhalb von Debian finden sich zahllose weitere Texteditoren. Falls der geneigte Leser eine besondere Perle entdeckt, sind wir für eine diesbezügliche Mail dankbar.

5.5.7 jEdit und jed – für Programmierer

Für Programmierer interessant ist der Editor `jedit`, der 130 Programmiersprachen (im weitesten Sinn) und ihre Gepflogenheiten kennt. Er setzt Java 2 voraus und ist nicht bei Debian, sondern samt einem *User's Guide* bei <http://www.jedit.org/> zu haben. Bei Sourceforge liegt ein Debian-Paket, das von `apt-get` erfasst werden kann. Unter anderem beherrscht der Editor Unicode und UTF-8 und bringt eine Vielzahl von Makros und Plugins mit, die sich durch Eigenerzeugnisse ergänzen lassen. Er ist auch für andere Betriebssysteme verfügbar, was in einer heterogenen Umgebung die Zusammenarbeit erleichtert.

Eine Alternative ist der Editor `jed` bzw. `xjed`, der ebenfalls viele beim Programmieren angenehme Fähigkeiten besitzt. Bei Debian in sechs Pakete verschnürt zu haben.

5.5.8 Editor für das Rich Text Format (ted)

Unter DOS rief `ted` einen *Tiny Editor* von TOM KIHLENS auf, 3 kB groß, was sogar damals wenig war. Er kannte zehn Kommandos, die auf den zehn Funktionstasten lagen. Natürlich konnte das Editorle nicht die *Türme von Hanoi* spielen wie der

Emacs von heute, aber zum Schreiben reichte er. Friede seiner Asche. Das Debian-Paket *ted-common* in Verbindung mit *ted* oder *ted-gtk* richtet einen Editor mit grafischer Oberfläche ein, der speziell das Rich Text Format (RTF) unterstützt. Beide Varianten brauchen X11, die erste, als stabil bezeichnete dazu die LessTif-Bibliothek, die zweite, weniger stabile GTK+. Von *ted* erzeugte RTF-Dokumente sollten von jedem MS-Windows-Programm akzeptiert werden, umgekehrt kann es bei selteneren Formatanweisungen zu Schwierigkeiten kommen. *ted* ignoriert, was er nicht versteht, sodass der Inhalt eines Dokumentes auf jeden Fall zu erkennen ist. Der Editor wandelt RTF nach PostScript oder PDF um und lässt sich als MIME-Handler für RTF verwenden. Näheres auf seiner Webseite <http://www.n11gg.nl/Ted/>, wo auch eine deutsche Fassung als Tarball heruntergeladen werden kann.

5.6 WYSIWYG-Textprozessoren

5.6.1 GNOME AbiWord

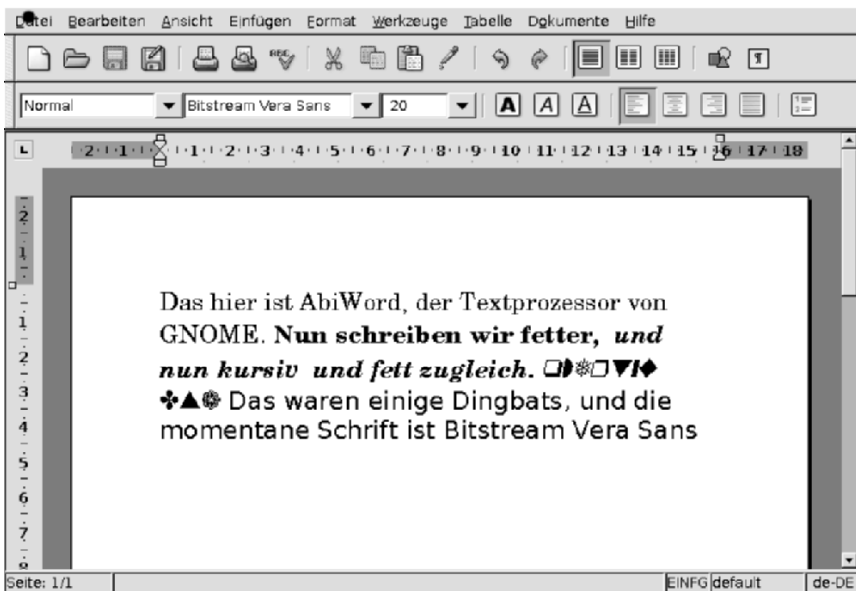


Abbildung 5.6. Screenshot des WYSIWYG-Textprozessors AbiWord aus dem GNOME-Office-Paket. Wenn es denn unbedingt WYSIWYG sein soll, keine schlechte Wahl. Woher der Name stammt, wissen die Götter; das Abitur ist nicht zu seinem Gebrauch erforderlich.

Wir kommen nun zu den WYSIWYG-Editoren oder Textprozessoren, die eine bei der Eingabe vorgenommene Formatierung sogleich auf dem Bildschirm wiedergeben. Das ist beim Schreiben von Briefen praktisch, hat aber Grenzen bei an-

spruchsvollen Texten. Deshalb findet man derartige Editoren vor allem in Office-Paketen, siehe Abschnitt 12 *Office-Pakete* auf Seite 405. Die Prozessoren lassen sich aber auch einzeln verwenden.

Zum GNOME-Projekt gehört AbiWord (<http://www.abiword.org/>), ein Textprozessor, der alle Funktionen mitbringt, die man im Alltag braucht. Die Beschränkung auf das Wesentliche trägt zur Geschwindigkeit bei. Seine Oberfläche erinnert an den Marktführer, siehe Abbildung 5.6. Ein Schönheitsfehler ist, dass die Menüs teils deutsch, teils englisch daherkommen. AbiWord ist eine gute Wahl, wenn man nur schreiben will und kein volles Office-Paket braucht. Die Herkunft des Namens ließ sich nicht klären.

5.6.2 KDE KWord

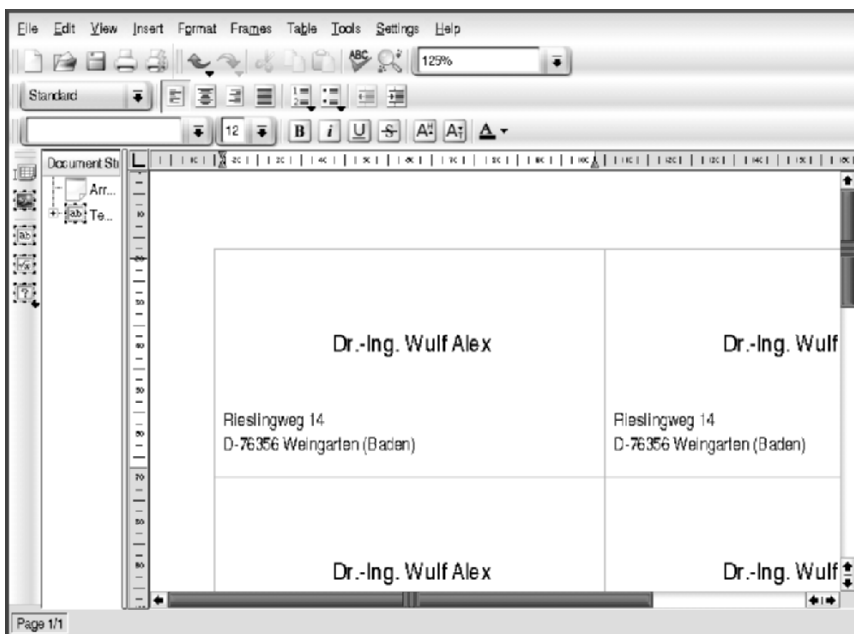


Abbildung 5.7. Screenshot des WYSIWYG-Textprozessors *kword* aus dem KDE-Office-Paket beim Gestalten von Visitenkarten

Der Textprozessor KWord aus dem KDE-Office-Paket wartet mit den meisten Features und Buttons auf, siehe Abbildung 5.7, die ihn beim Erstellen von Visitenkarten zeigt. Er fügt sich reibungslos in den KDE-Desktop ein. Sein Look + Feel weicht stärker vom Marktführer ab, was als Vorteil empfunden werden kann.

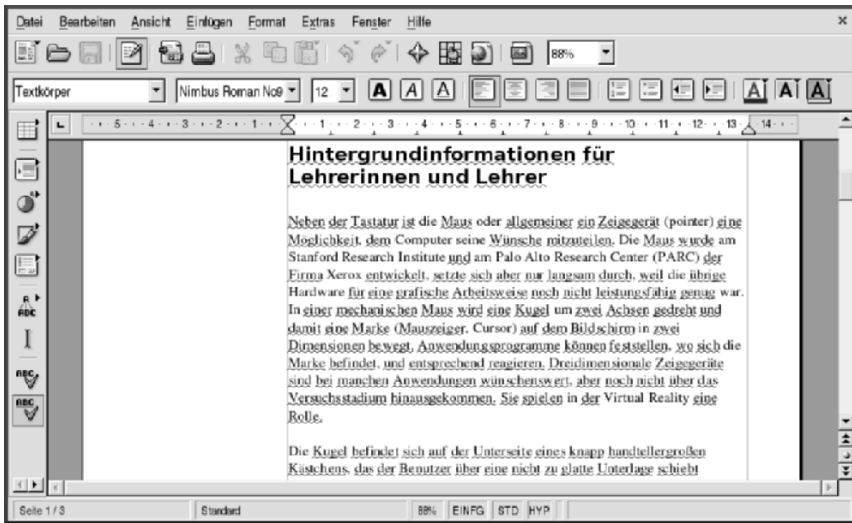


Abbildung 5.8. Screenshot des WYSIWYG-Textprozessors oowriter aus dem OpenOffice.org-Paket beim Abfassen eines Artikels

5.6.3 Open.org writer

Der oowriter ist das Textwerkzeug von OpenOffice.org. Man kann es unter seinem Namen aufrufen oder mit office. Die Namen aller Einzelwerkzeuge von OpenOffice.org sind Symlinks auf office. Öffnet man eine Textdatei oder wählt bei einer Neuanlage den Menüpunkt *Textdokument* aus, meldet sich der Textprozessor. Die Arbeitsweise unterscheidet sich nicht von der anderer Textprozessoren und ist relativ gut zu verstehen. Das originale Format für Textdateien trägt die Kennung *sxw* und ist ein komprimiertes Archiv von XML-Dateien; die Texte lassen sich aber auch als *pdf*-Datei exportieren oder als Word-, RTF-, ASCII- oder HTML-Datei abspeichern. Abbildung 5.8 vermittelt einen Eindruck vom Aussehen des Bildschirms.

5.6.4 LyX

Der Textprozessor LyX ist ein WYSIWYG-Frontend für \LaTeX , so weit das möglich ist. Benutzern, die bisher mit Textprozessoren gearbeitet haben, wird damit der Wechsel zu \LaTeX erleichtert. Die Webseite des Werkzeugs liegt unter <http://www.lyx.org/>. LyX kann bestehende \LaTeX -Dateien importieren, hat aber bei komplexen Dokumentstrukturen Probleme. Am besten entscheidet man sich zu Beginn eines Projektes für oder gegen LyX. Wer sich mit \LaTeX auskennt und einen der Texteditoren beherrscht, gewinnt mit LyX nichts dazu. LyX ist nicht mit dem textorientierten Web-Browser lynx zu verwechseln.

5.7 Hex-Editoren (**bvi**, **hexedit**, **hexcurse**, **vche**)

Text-Editoren fassen ihnen vorgelegte Dateien immer als Text auf. Es gibt aber Aufgaben, bei denen eine Datei Byte für Byte ohne jegliche Interpretation angezeigt und editiert werden soll. Das Werkzeug zu diesem Zweck sind Binär-Editoren. Die meisten zeigen einen Dateiinhalt wahlweise binär, oktal, dezimal, hexadezimal oder soweit möglich als ASCII-Zeichen an, daher auch die Bezeichnung Hex-Editor.

Die beiden Text-Editoren **vim** und **emacs** kennen einen Binär-Modus, Aufruf **vim -b** bzw. **Meta-x** im Emacs. Spezielle Binär-Editoren sind jedoch besser an die Aufgabe angepasst. Neben kommerziellen Produkten wie **vedit** oder **ultraedit** finden sich im Netz auch freie Binär-Editoren in Form des Quellcodes oder als Debian-Paket:

- **bvi**, ein binärer **vi**,
- **hexedit**, äußerlich dem **bvi** ähnlich, aber mit anderen Kommandos,
- **khexit** aus der KDE-Arbeitsumgebung, mit einer netten grafischen Oberfläche, zu finden unter Utilities – Binary Editor,
- **shed**, der *Simple Hex Editor*, siehe <http://shed.sourceforge.net/>,
- **lfhex**, der *Large File Hex Editor*, siehe <http://www.freshports.org/editors/lfhex/>,
- **vche**, der *Virtual Console Hex Editor*, siehe <http://www.grigna.com/diego/linux/vche/>,
- **bed**, der *Menu Driven Binary Editor*, siehe <http://bedlinux.tripod.com/>.

Ihrer Aufgabe gemäß können Binär-Editoren mit extrem großen Dateien umgehen, wo Text-Editoren versagen. Sie arbeiten nicht auf einer Kopie der gesamten Datei, sondern nur auf Ausschnitten.

In Abbildung 5.9 sehen wir einen Screenshot des Hex-Editors **hexedit** bei der Untersuchung des Anfangs einer **jpg**-Datei. Der Schirm des **bvi**-Hex-Editors sieht genauso aus. Links stehen die hexadezimalen Adressen des ersten Bytes der jeweiligen Zeile, in der Mitte die Bytes in hexadezimaler Schreibweise – pro Byte ein Zeichenpaar von 0 bis F – und rechts deren Übersetzung in ASCII-Zeichen, soweit möglich. Einzelne Bytes lassen sich editieren, vorzugsweise durch Ersetzen. Ohne tieferes Verständnis der Datei kann man sie aber auch leicht ruinieren.

Der Hex-Editor **vche** (Virtual Console Hex Editor) schreibt direkt auf eine virtuelle Konsole (**/dev/vcsa1** oder ähnlich) und läuft daher in seiner Grundversion nicht unter X11 oder über ein Netz (remote). Ein Benutzer braucht Zugriffsrechte auf **/dev/vcsa1**, die er nicht von vornherein hat. Der Editor zeigt auch nicht druckbare Zeichen an und kann Gerätedateien bearbeiten.

5.8 Editoren für fremde Schriften (**geresh**, **katoob**, **yudit**)

Die großen Editoren lernen allmählich, mit UTF-8 und bidirektionalen Texten – zum Beispiel Englisch und Hebräisch gemischt – umzugehen. Es gibt aber schon länger

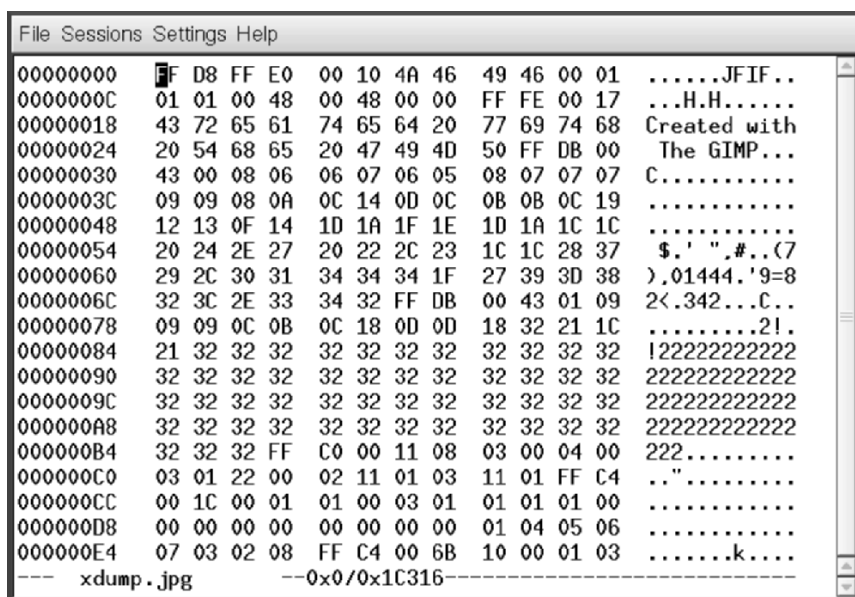


Abbildung 5.9. Screenshot des Hex-Editors hexedit beim Anschauen einer jpg-Datei

spezielle Editoren für diese Aufgabe. Aus Israel stammt *geresh* mit der Startseite <http://www.typo.co.il/~mooffie/geresh/>. Der Editor arbeitet zeichenbasiert ohne X11, kennt Unicode und unterstützt in erster Linie Hebräisch und Arabisch. Der Begleiter *pgeresh* ist ein Pager, ein Leseprogramm mit denselben Sprachkenntnissen. Die Manualseiten beschränken sich auf das Nötigste; Hilfe bieten die Werkzeuge beim Aufruf mit der Option `--help` oder indem man bei der Arbeit mit der Taste <F9> oder <F10> in die Menüleiste springt, dort per Pfeiltaste den gewünschten Punkt auswählt und mit <cr> bestätigt. Verlassen der meisten Punkte oder des Editors mit <alt>-<x>. In `/usr/share/doc/geresh/` findet man ein User Manual, leider nur auf Hebräisch. Abbildung 5.10 zeigt den Schirm mit der Auswahl eines Menüpunktes.

Aus dem arabischen Lager (<http://www.arabeyes.org/>) stammt der Editor *katoob*, der X11 und GTK+ benötigt und ebenfalls mehrere Sprachen und Schriften beherrscht. In die Menüs gelangt man mit der Maus, die Menüsprache ist Deutsch. Als Tastaturen werden US-ASCII, Arabisch und Hebräisch angeboten. Die Manualseite ist kurz, unter dem Menüpunkt *Hilfe* erscheint auch nicht direkt ein Lehrbuch für Arabisch. Der Editor lässt sich als Filter zum Umcodieren von Textdateien gebrauchen. Das User Manual zu *geresh* wird problemlos in hebräischer Schrift mit lateinischen Einlagen angezeigt.

Der Editor *yudit* ist speziell dafür gebaut, unter X11 mit verschiedenen Zeichensätzen zurecht zu kommen, darunter UTF-7 und UTF-8 (<http://www.yudit.org/>). Das Werkzeug eignet sich ebenfalls zum Umcodieren gegebener Texte. Der Editor legt beim ersten Verlassen ein Verzeichnis `.yudit` im Home-

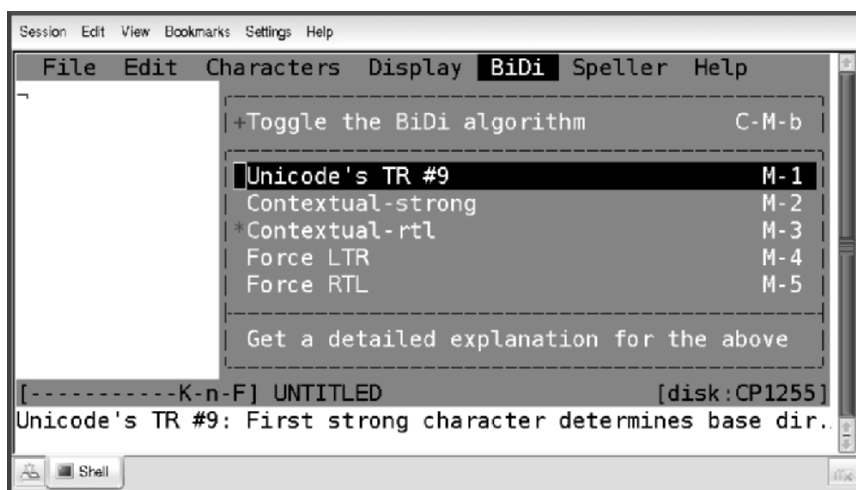


Abbildung 5.10. Screenshot des für Hebräisch und Arabisch geeigneten Editors *geresh*

Verzeichnis des Benutzers an und darin eine Datei *yudit.properties* mit der Konfiguration.

Für slawische Sprachen und kyrillische Schriften findet man ausführliche Informationen unter <http://www.slovo.info/>, einer Website von CHRISTOPH SINGER, die leider nicht mehr gepflegt wird, jüngster Stand 2000.



Abbildung 5.11. Screenshot (Ausschnitt) des für fremde Schriften geeigneten bidirektionalen Editors *yudit*

5.9 Streaming Editor (sed)

Der Streaming Editor `sed` bearbeitet eine Textdatei zeilenweise nach Regeln, die man ihm als Option oder in einer getrennten Datei (`sed`-Skript) mitgibt. Er ist im Gegensatz zu den bisher genannten Editoren nicht interaktiv, er führt keinen Dialog. Die letzte Zeile der zu bearbeitenden Textdatei muss leer sein oder anders gesagt, das letzte Zeichen des Textes muss ein newline-Zeichen (Linefeed) sein.

Die einfachste Aufgabe für den `sed` wäre der Ersatz eines bestimmten Zeichens im Text durch ein anderes (dafür gibt es allerdings ein besseres, weil einfacheres Werkzeug `tr`). Der `sed` bewältigt ziemlich komplexe Aufgaben, daher ist seine Syntax umfangreich. Sie baut auf der Syntax des alten Zeileneditors `ed` auf. Der Aufruf

```
joe@debian:~$ sed 'Kommandos' datei
```

veranlasst den `sed`, die Datei `datei` Zeile für Zeile einzulesen und gemäß den Kommandos bearbeitet nach `stdout` auszugeben. Der Aufruf

```
joe@debian:~$ sed '1d' datei
```

löscht die erste Zeile in der Datei `datei` und schreibt das Ergebnis nach `stdout`. Die Quotes um das `sed`-Kommando verhindern, dass die Shell sich das für den `sed` bestimmte Kommando ansieht und möglicherweise Metazeichen interpretiert. Hier wären sie nicht nötig und stehen einfach aus Gewohnheit. Jokerzeichen in `datei` dagegen werden von der Shell zu Recht interpretiert, sodass der `sed` von der Shell eine Liste gültiger Namen erhält.

Folgender einzeliger Aufruf ersetzt alle Großbuchstaben durch die entsprechenden Kleinbuchstaben:

```
joe@debian:~$ sed 'y/ABCDEFGHIJKLMNOPQRSTUVWXYZ/abcdefghijklmnopqrstuvwxyz/' datei
```

Das `y`-Kommando kennt keine Zeichenbereiche, wie sie bei regulären Ausdrücken oder beim Kommando `tr` erlaubt sind, man muss die beiden notwendigerweise gleichlangen Zeichenmengen auflisten. Übrigens ist obiges Kommando ein Weg zur ROT13-Verschlüsselung, indem man die zweite Zeichenmenge mit `n` beginnen lässt. Geht es um den Ersatz eines festen Zeichenmusters oder eines regulären Ausdrucks durch eine feste Zeichenkette, so nimmt man:

```
joe@debian:~$ sed 's/[a-z][a-z]*{...}/LaTeX-K/g' datei
```

Im Kommando steht `s` für *substitute*. Dann folgt ein regulärer Ausdruck zur Kennzeichnung dessen, was ersetzt werden soll, hier das Muster eines `LaTeX`-Kommandos. An dritter Stelle ist der Ersatz (replacement) aufgeführt, hier die feste Zeichenfolge `LaTeX-K`, und schließlich ein Flag, das besagt, den Ersatz global (überall, nicht nur beim ersten Auftreten des regulären Ausdrucks in der Zeile) auszuführen. Das Trennzeichen, der Schrägstrich, zwischen den vier Teilen kann jedes beliebige Zeichen sein, es darf nur nicht in den Teilen selbst vorkommen.

Der `sed` ist mächtig und nützlich, aber lernbedürftig. Erleichtert wird die Arbeit dadurch, dass man ein `sed`-Skript oder auch die Anweisung in der Kommandozeile Schritt für Schritt aufbauen und testen kann. Im Netz findet sich viel Material dazu, lassen Sie eine Suchmaschine nach `unix AND editor AND sed` suchen.

Merke: Der `vi` oder der Emacs ist ein interaktiver Editor, der Tastatureingaben erfordert und nicht Bestandteil einer Pipe sein oder im Hintergrund laufen kann. Der `sed` ist ein Filter, das keine Tastatureingaben verlangt, Glied einer Pipe oder eines Shellskripts sein und unbeaufsichtigt laufen kann.

5.10 Listenbearbeitung (awk)

Das Werkzeug `awk` ist nach seinen Urhebern ALFRED V. AHO, PETER J. WEINBERGER und BRIAN W. KERNIGHAN benannt und firmiert als programmierbares Filter oder Listengenerator. Es lässt sich auch als eine einfache Programmiersprache für einen bestimmten, engen Zweck auffassen. Der `awk` bearbeitet eine Textdatei zeilenweise, wobei er jede Zeile – auch Satz genannt – in Felder zerlegt. Eine typische Aufgabe ist die Bearbeitung von Listen. Hier ist er angenehmer als der `sed`, allerdings auch langsamer. Für die Verwaltung eines kleinen Vereins ist er recht, für das Telefonbuch von Berlin nicht. Unter Debian steht der originale `awk` im Paket *original-awk* zur Verfügung. In der Regel wird man aber die leistungsfähigere GNU-Version `gawk` vorziehen.

In einfachen Fällen werden dem `awk` beim Aufruf die Befehle zusammen mit den Namen der zu bearbeitenden Dateien mitgegeben, die Befehle in Hochkommas, um sie vor der Shell zu schützen:

```
joe@debian:~$ awk 'befehle' dateien
```

Ein `awk`-Befehl besteht aus den Teilen *Muster* und *Aktion*. Jede Eingabezeile, auf die das Muster zutrifft, wird entsprechend der Aktion behandelt. Die Ausgabe geht auf `stdout`. Ein Beispiel:

```
joe@debian:~$ awk '{if (NR < 8) print $0}' datei
```

Die Datei `datei` wird Zeile für Zeile gelesen. Die vorgegebene `awk`-Variable `NR` ist die Zeilennummer, beginnend mit 1. `$0` ist die ganze jeweilige Zeile. Falls die Zeilennummer kleiner als 8 ist, wird die Zeile nach `stdout` geschrieben. Es werden also die ersten 7 Zeilen der Datei ausgegeben. Nun wollen wir das letzte Feld der letzten Zeile ausgeben:

```
joe@debian:~$ awk 'END {print $NF}' datei
```

Das Muster `END` trifft zu, wenn die letzte Zeile eingelesen ist. Üblicherweise gilt die zugehörige Aktion irgendwelchen Abschlussarbeiten. Die Variable `NF` enthält die Anzahl der Felder der Zeile, die Variable `$NF` ist das letzte Feld. Nun wird es etwas anspruchsvoller:

```
joe@debian:~$ awk '$1 != prev { print; prev = $1 }'
wortliste
```

Die Datei `wortliste` enthalte in alphabetischer Folge Wörter und gegebenenfalls weitere Bemerkungen zu den Wörtern, pro Wort eine Zeile. Der `awk` liest die Datei zeilenweise und spaltet jede Zeile in durch Leerzeichen oder Tabs getrennte Felder auf. Die Variable `$1` enthält das erste Feld, also hier das Wort zu Zeilenbeginn. Falls dieses Wort von dem ersten Wort der vorangegangenen Zeile abweicht (Variable `prev`), wird die ganze Zeile ausgegeben und das augenblickliche Wort in die Variable `prev` gestellt. Zeilen, die im ersten Feld übereinstimmen, werden nur einmal ausgegeben. Dieser `awk`-Aufruf hat eine ähnliche Funktion wie das Linux/UNIX-Kommando `uniq`. Da Variable mit dem Nullzeichenkette initialisiert werden, wird auch die erste Zeile richtig bearbeitet.

Wenn die Anweisungen an den `awk` umfangreicher werden, schreibt man sie in eine eigene Datei (`awk`-Skript). Der Aufruf sieht dann so aus:

```
joe@debian:~$ awk -f awkscript textdateien
```

`awk`-Skripte werden in einer Sprache geschrieben, die teils an Shellskripte, teils an C-Programme erinnert. Sie bestehen – wie ein deutscher Schulaufsatz – aus Einleitung, Hauptteil und Schluss. Sehen wir uns ein Beispiel an, das mehrfache Eintragungen von Stichwörtern in einem Sachregister aussortiert und die zugehörigen Seitenzahlen der ersten Eintragung zuordnet:

```
# awk-Skript fuer Sachregister

BEGIN { ORS = ""
        print "Sachregister"
      }
      {
        if ($1 == altwort)
          print ", " $NF
        else
          {
            print "\n" $0
            altwort = $1
            nor++
          }
      }
END    { print "\n\n"
        print "gelesen: " NR " geschrieben: " nor "\n"
      }
```

Quelle 5.1 . `awk`-Skript für Sachregister

Das Doppelkreuz markiert eine Kommentarzeile. Der Einleitungsblock wird mit `BEGIN` gekennzeichnet, der Hauptteil steht nur in geschweiften Klammern, und der Schluss beginnt mit `END`. Die vorbestimmte, `awk`-eigene Variable `ORS` (Output Record Separator, das heißt Trennzeichen zwischen Sätzen in der Ausgabe), standardmäßig das Newline-Zeichen, wird mit dem Nullstring initialisiert. Dann wird die Überschrift *Sachregister* ausgegeben.

Im Hauptteil wird das aktuelle erste Feld gegen die Variable `altwort` geprüft. Bei Übereinstimmung werden ein Komma, ein Leerzeichen und das letzte Feld der aktuellen Zeile ausgegeben, nämlich die Seitenzahl. Die `awk`-eigene Variable `NF` enthält die Anzahl der Felder des aktuellen Satzes, die Variable `$NF` mithin das letzte Feld.

Bei Nichtübereinstimmung (einem neuen Stichwort `also`) werden ein Newline-Zeichen und dann die ganze Zeile (`$0`) ausgegeben. Anschließend werden das erste Feld in die Variable `altwort` gestellt und die vom Programmierer definierte Variable `nor` um 1 inkrementiert. So wird mit der ganzen Textdatei verfahren.

Am Ende der Textdatei angelangt, werden noch zwei Newline-Zeichen, der Wert der `awk`-eigenen Variablen `NR` (Number of Records) und der Wert der Variablen `nor` ausgegeben. Die Aufgabe wäre auch mit dem `sed` oder einem C-Programm zu lösen, aber ein `awk`-Skript ist der einfachste Weg. Das Werkzeug vermag noch viel mehr.

Eine Besonderheit des `awk` sind Vektoren mit Inhaltindizierung (E: associative array). In Programmiersprachen wie C oder FORTRAN werden die Elemente eines Arrays oder Vektors mit fortlaufenden ganzen Zahlen (Indizes) bezeichnet. Auf ein bestimmtes Element wird mittels des Arraynamens und des Index zugegriffen:

```
arrayname[13]
```

In einem `awk`-Array dürfen die Indizes nicht nur ganze Zahlen, sondern auch beliebige Zeichenketten sein:

```
telefon["Meyer"]
```

ist eine gültige Bezeichnung eines Elementes. Es könnte die Anzahl der Telefonanschlüsse namens Meyer in einem Telefonbuch enthalten.

Eine Alternative zu `awk` ist Perl, eine interpretierte Programmiersprache zur Verarbeitung von Textdateien, die Elemente aus C, `sed`, `awk` und der Shell `bash` enthält. Ihre Möglichkeiten gehen über das Verarbeiten von Texten hinaus in Richtung Shellskripte.

5.11 Formatieren

5.11.1 Inhalt, Struktur und Aufmachung

Ein Schriftstück – sei es Brief oder Buch – hat einen Inhalt, nämlich Text, gegebenenfalls auch Abbildungen, der in einer bestimmten Form dargestellt ist. Bei der Form unterscheiden wir zwischen der logischen Struktur und ihrer Darstellung auf Papier oder Bildschirm, Aufmachung oder Layout genannt. Beim Schreiben des Manuskripts macht sich der Autor Gedanken über Inhalt und logische Struktur, aber kaum über Schrifttypen und Schriftgrößen, den Satzspiegel, den Seitenumbruch, die Nummerierung der Abbildungen. Das ist Aufgabe des Metteurs oder Layouters im Verlag, der seinerseits möglichst wenig an Inhalt und Struktur ändert. *Schreiben* und

Setzen sind unterschiedliche Aufgaben, die unterschiedliche Kenntnisse und Werkzeuge erfordern. Wer wissen will, was außer den Gedanken des Verfassers hinter einem guten Buch steckt, findet bei JAN TSCHICHOLD Rat.

Der Rechner wird als Werkzeug für alle drei Aufgaben (Inhalt, Struktur, Layout) eingesetzt. Mit einem Editor schreibt man einen strukturierten Text, weitergehende Programme prüfen die Rechtschreibung, helfen beim Erstellen eines Sachverzeichnisses, analysieren den Stil. Ein Satz- oder Formatierprogramm erledigt den Zeilen- und Seitenumbruch, sorgt für die Nummerierung der Abschnitte, Seiten, Abbildungen, Tabellen, Fußnoten und Formeln, legt die Schriftgrößen fest, ordnet die Abbildungen in den Text ein, stellt das Inhaltsverzeichnis zusammen usw. Ein Satzprogramm wie `groff` oder \LaTeX erfüllt hohe Ansprüche und besteht daher aus einem ganzen Softwarepaket. Ein kommerzielles Produkt mit gleichem Ziel ist FrameMaker. Der Textprozessor KWord aus der KDE Office Suite hat Ähnlichkeiten mit FrameMaker und ist bei einem Wechsel das Werkzeug der Wahl.

Der Linux/UNIX-Formatierer `nroff` bzw. `groff` und das Satzprogramm \LaTeX (\TeX , \LaTeX , `pdf \TeX` etc.) halten Inhalt, Struktur und Layout auseinander. Man schreibt mit einem beliebigen Editor den Text samt Strukturhinweisen und formatiert anschließend. \LaTeX verfolgt darüber hinaus den Gedanken, dass der Autor seine Objekte logisch beschreiben und von der typografischen Gestaltung möglichst die Finger lassen soll. Der Autor soll sagen: *Jetzt kommt eine Kapitelüberschrift* oder *Jetzt folgt eine Fußnote*. \LaTeX legt dann nach typografischen Regeln die Gestaltung fest. Man kann darüber streiten, ob die Regeln das Nonplusultra der Schwarzen Kunst sind, ihr Ergebnis ist jedenfalls besser als vieles, was Laien einfällt.

Sowohl `nroff/groff` wie \LaTeX zielen auf die Wiedergabe der Dokumente mittels Drucker auf Papier ab. Mit Hilfe von Programmen wie `gv` oder `xdvi` lassen sich die Ergebnisse auf dem Bildschirm vor dem Ausdrucken beurteilen. Die Hypertext Markup Language HTML, deren Anwendung wir im World Wide Web begegnen, hat viel mit \LaTeX gemeinsam, eignet sich jedoch in erster Linie für Dokumente, die auf dem Bildschirm dargestellt werden sollen. Auch sie trennt Inhalt, Struktur und Layout. Leider begreifen das manche Web-Autoren nicht.

Beim Arbeiten mit Formatierern wie \LaTeX – gegebenenfalls in Verbindung mit einem Versionskontrollsystem wie CVS – kommen die Schwierigkeiten am Anfang, wenn man eine Reihe von Konzepten und Kommandos lernen muss, wenige Vorlagen hat und viel Zeit in die Strukturierung des Projektes (Verzeichnisbaum) steckt, die sich nicht sofort auf Papier niederschlägt. Beim Arbeiten mit WYSIWYG-Programmen hat man schnell Erfolgserlebnisse, die Schwierigkeiten kommen später, wenn die Manuskripte umfangreicher und komplexer werden, aber dann wird ein Umstieg teuer.

5.11.2 UNIX-Formatierer (`nroff`, `groff`)

Der Standard-Formatierer in Linux/UNIX für Druckerausgabe ist `nroff`. Das *n* steht für *new*, da der Vorgänger ein `roff` war, und dieser hieß so, weil damit *run off to the printer* gemeint war. Unter Debian ist `nroff` ein Script, das `groff` aus dem GNU-Projekt aufruft. Die Variante `troff` war ursprünglich ein Formatierer zur

Ausgabe auf einem Fotosatzbelichter. Unter Debian ist `troff` der Teil von `groff`, der die eigentliche Arbeit erledigt.

Eine Datei für `nroff` enthält den unformatierten Text und `nroff`-Kommandos. Diese stehen stets in eigenen Zeilen mit einem Punkt am Anfang. Ein `nroff`-Text könnte so beginnen:

```
.po 1c
.ll 60
.fi
.ad c

.cu 1
Ein Textbeispiel

von W. Alex

.ad b
.ti 1c
Dies ist das Beispiel eines Textes, der mit nroff formatiert
werden soll. Er wurde mit dem Editor vi geschrieben.

.ti 1c
Hier beginnt der zweite Absatz.
Die Zeilenlaenge in der Textdatei ist unerheblich.
```

Die `nroff`-Kommandos bedeuten folgendes:

- `po 1c` page offset 1 cm (zusätzlicher linker Seitenrand),
- `ll 60` line length 60 characters,
- `fi` fill output lines (für Blocksatz),
- `ad c` adjust center (zentrieren),
- `cu 1` continuous underline 1 line (auch Leerzeichen unterstreichen),
- `ad b` adjust both margins (Blocksatz),
- `ti 1c` temporary indent 1 cm (einmaliger Einzug).

Die Kommandos können wesentlich komplexer sein als im obigen Beispiel, es sind auch Makros, Abfragen und Rechnungen möglich. S. R. BOURNE führt in seinem im Anhang N *Zum Weiterlesen* auf Seite 508 genannten Buch die Makros auf, mit denen die amerikanische Ausgabe seines Buches formatiert wurde. Es gibt ganze Makrobibliotheken zu `nroff`.

Da sich Formeln und Tabellen nur schlecht mit den Textbefehlen beschreiben lassen, verwendet man für diese beiden Fälle eigene Befehle samt Präprozessoren, die die Spezialbefehle in `nroff`-Befehle umwandeln. Für Tabellen nimmt man `tbl`, für Formeln `neqn`, meist in Form einer Pipe:

```
joe@debian:~$ tbl textfile | neqn | nroff | col | lp
```

wobei `col` ein Filter zur Behandlung von Backspaces und dergleichen ist.

5.11.3 GNU Texinfo

Das Debian-Paket *texinfo* (gesprochen *tekinfo*) enthält das GNU Documentation System Texinfo, dessen Startseite unter <http://texinfo.org/> oder <http://www.gnu.org/doc/tekinfo.html> zu finden ist.

`//www.gnu.org/software/texinfo/` liegt. Das System bezweckt, aus einer Textvorlage (Quelle) sowohl eine Ausgabe auf den Bildschirm (Terminal oder Web-Browser) als auch eine Ausgabe auf einen Drucker zu erzeugen. Eine andere Lösung dieser Aufgabe findet man unter den Stichworten *Standard General Markup Language* (SGML) und *Extensible Markup Language* (XML). Texinfo ist unabhängig von $\text{T}_{\text{E}}\text{X}$ und $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$, arbeitet mit diesem System aber zusammen. Wir finden Texinfo-Dokumente oft als Ersatz für Manualseiten, siehe Abschnitt 1.5 *Hilfen* auf Seite 36. Außerhalb von GNU ist Texinfo wenig verbreitet.

Das Kommando zum Lesen von Texinfo-Dokumenten lautet `info`. Man kann es zunächst einmal anstelle von `man` aufrufen:

```
joe@debian:~$ info ls
```

Dann bekommt man mit allen Möglichkeiten von Texinfo wie Vorwärts- und Rückwärtsblättern die gewohnte Manualseite angezeigt. Mit dem Kommando:

```
joe@debian:~$ info info
```

erscheint eine kurze, strukturierte Einführung auf dem Bildschirm. Bei Schwierigkeiten versuchen Sie es mit der Option `-f` und dem absoluten Pfad der Texinfo-Datei `/usr/share/info/...` Nachdem Sie den ersten Schirm gelesen haben, bewegen Sie den Cursor mit den Pfeiltasten in die Zeile *Getting Started* und tippen `<cr>`. Darauf hin erscheint die entsprechende Seite. In der obersten Zeile (header) lesen Sie immer, wo Sie sich befinden. Mit den Tasten `<n>` oder `<p>` gelangt man zur nächsten (next) bzw. vorherigen (previous) Seite. Füllt eine Seite mehr als einen Schirm, schiebt man – wie von `less` gewohnt – mit der Leertaste den Text nach oben, der Backspace schiebt nach unten, mittels `` (begin) gelangt man zum Seitenanfang, mittels `<e>` (end) zum Seitenende. Weitere Möglichkeiten erklärt die Infoseite. Eine Alternative zu `info` enthält das Debian-Paket *pinfo*. Auch die Hilfesysteme von GNOME und KDE verstehen das Texinfo-Format. Im Konqueror kann man mittels der Eingabe `info:gzip` als URL die Texinfo-Seite zum Kommando `gzip` betrachten, da automatisch das Filter `info2html` aufgerufen wird. Bei fehlendem Texinfo-Dokument fällt der Konqueror nicht auf das Manual zurück.

Ein Texinfo-Quelltext wird mit einem beliebigen Editor geschrieben. Der Emacs kennt einen Texinfo-Modus, der die Arbeit erleichtert und auch statt `info` zum Lesen verwendet werden kann. Metazeichen sind der Klammeraffe und die beiden geschweiften Klammern. Der Klammeraffe dient auch zum Quoten der Metazeichen, `@@` bedeutet den Klammeraffen buchstäblich. Tabs verursachen Probleme bei der Formatierung, ansonsten dürfen alle druckbaren ASCII-Zeichen im Text vorkommen. Die Formatierkommandos lauten anders als bei `nroff`, $\text{T}_{\text{E}}\text{X}$ oder $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$, obwohl sie zum großen Teil dieselben Aufgaben haben. Man darf auch nicht vergessen, dass man für zwei verschiedene Leserkreise schreibt: die Buchleser und die Bildschirmleser. Kurze Passagen im Text lassen sich so markieren, dass sie sich nur an einen der beiden Leserkreise wenden, aber zwei gänzlich unterschiedliche Dokumente abzufassen, wäre ein Rückfall in die Zeit vor Texinfo. Die Referenz entnimmt man am besten dem Netz.

Hier ein minimales Beispiel für einen Texinfo-Quelltext (stark gekürzt nach dem Short Sample aus <http://www.gnu.org/software/texinfo/manual/texinfo.html>), nur um einen Eindruck zu gewinnen:

```
\input texinfo
@setfilename sample.info
@settitle Sample Manual

@titlepage
@title Sample Title
@end titlepage

@ifnottex
@node Top
@top Texinfo-Beispiel
@end ifnottex

@menu
* 1. Abschnitt::
@end menu

@node 1. Abschnitt
@chapter 1. Abschnitt

@bye
```

Wer schon mit `nroff` oder \LaTeX gearbeitet hat, gewöhnt sich schnell ein. Die online lesbare info-Datei wird von dem Programm `makeinfo` erstellt, das zum Texinfo-Paket gehört. Zur Erzeugung der Papierausgabe ist die Texinfo-Quelldatei durch das \TeX -System samt dem Makro `texinfo.tex` zu schicken, die DVI-Datei dann wie unter \TeX oder \LaTeX gewohnt durch `dvips`, um eine druckbare PostScript-Datei zu erhalten. Zweckmäßig fasst man die Kommandos in einem Makefile zusammen:

```
all : sample.ps sample.info

sample.info : sample.texinfo
            makeinfo sample

sample.ps : sample.dvi
            dvips -o sample.ps sample

sample.dvi : sample.texinfo
            tex sample.texinfo

clean :
            rm *.aux *.toc *.log
```

und ruft nach dem Editieren nur noch `make` auf. Alternativ erzeugt auch die Eingabe:

```
joe@debian:~$ texi2dvi sample.texinfo
```

eine dvi-Datei. Im Netz finden sich Konverter von Texinfo nach HTML, `nroff`, IPF und RTF.

5.11.4 DocBook

DocBook ist ein Format für Bücher, Artikel und andere Dokumentationen vorzugsweise technischen Inhalts, das in einer Document Type Definition (DTD) beschrieben ist, siehe Abschnitt 5.12.3 *Standard Generalized Markup Language* auf Seite 227. Die DTD (SGML) ist im Debian-Paket *docbook* zu finden, die DTD (XML) im Debian-Paket *docbook-xml*. Das Format ist in der Open-Source-Welt verbreitet. Beispielsweise werden viele HOWTOS in diesem Format erstellt. Ein DocBook-Quelltext hat Ähnlichkeiten mit einem HTML-Dokument, siehe Abschnitt 5.12.2 *Hypertext Markup Language* auf Seite 224. Aus DocBook-Quellen lassen sich als Endprodukte PDF- oder PostScript-Dateien für den Druck oder HTML für das Web erzeugen. Das Debian-Paket *db2latex-xsl* enthält Style Sheets zur Umwandlung von DocBook-XML-Dateien nach \LaTeX . Weitere Konverter finden sich im Debian-Paket *docbook-utils*, darunter *docbook2html* und *docbook2pdf*. Der Versuch, einige *docbook*-Hilfen aus dem KDE-Projekt zu konvertieren, scheiterte leider an Syntaxfehlern. Näheres bei:

- <http://www.oasis-open.org/docbook/>, siehe auch das Debian-Paket *docbook-simple*,
- <http://docbook.org/> von NORMAN WALSH und anderen, siehe auch Debian-Paket *docbook-defguide*,
- <http://www.sagehill.net/docbookxsl/> von BOB STAYTON,
- <http://www.docbook.de/>,
- <http://www.goshaky.com/docbook-tutorial/> von LARS TRIE-LOFF (in deutscher Sprache).

5.11.5 \LaTeX

\TeX und \LaTeX

\TeX ist eine Formatierungssoftware (E: typesetting system), die von DONALD ERVIN KNUTH (<http://www-cs-staff.stanford.edu/~uno/index.html>) um 1980 entwickelt wurde – dem Mann, der seit Jahrzehnten an dem siebenbändigen Werk *The Art of Computer Programming* (TAOCP) schreibt und hoffentlich noch lange lebt. Die Stärke der Software sind umfangreiche mathematische oder technische Texte. Das System optimiert immer ganze Seiten und innerhalb der Seiten die Absätze, nicht einzelne Zeilen wie einfachere Werkzeuge. In \TeX steckt viel handwerkliches Wissen aus dem typografischen Gewerbe. Seine Grenzen findet es bei der Gestaltung von:

- Zeitungen mit mehreren Spalten, vielen Überschriften, Fotos, Anzeigen und dergleichen. Für diese Aufgabe braucht man Desktop-Publishing-Programme wie Scribus (Heimathafen <http://www.scribus.net/>, als Debian-Paket verfügbar), Adobe InDesign oder Quark XPress (beide kommerziell).
- Bildbänden oder Fotoalben mit vielen großformatigen Abbildungen. Hier kann man sich helfen, indem man Bild- und Textteil voneinander trennt. Das gesamte

Werk lässt sich dann wieder mit \TeX einrichten. Wenn jedes Bild samt Unterschrift genau eine oder eine halbe Seite beansprucht, wird es wieder einfach.

Für rein grafische Erzeugnisse wie Plakate gibt es geeignetere Werkzeuge, siehe Abschnitt 8 *Das grafische Atelier* auf Seite 291.

\TeX ist sehr leistungsfähig, verlangt aber vom Benutzer die Kenntnis vieler Einzelheiten, ähnlich wie Programmieren in Assembler. \LaTeX ist eine Makrosammlung, die auf \TeX aufbaut. Die \LaTeX -Makros von LESLIE LAMPORT erleichtern bei Standardaufgaben und -formaten die Arbeit beträchtlich, indem viele \TeX -Befehle zu einfach anzuwendenden \LaTeX -Befehlen zusammengefasst werden. Kleinere Modifikationen der Standardeinstellungen sind vorgesehen, Sonderwünsche erfordern das Hinabsteigen auf \TeX -Ebene. Die Entwicklung seit LAMPORT diente vor allem der Internationalisierung. Zu \LaTeX sind im Laufe der Jahre zahlreiche Makros hinzugefügt worden, sodass es heute kaum einen Wunsch im Textsatz gibt, der nicht mit \LaTeX erfüllbar wäre. Das Problem besteht eher darin, das am besten geeignete Makro aus rund 1000 Zusatzpaketen herauszufinden. \LaTeX eignet sich für alle Arten von Dokumenten, vom Brief bis zum mehrbändigen Lebenswerk. Auch zur Gestaltung von Folien oder Präsentationen lässt sich \LaTeX heranziehen. Es steht für viele Betriebssysteme kostenfrei zur Verfügung. Alle gängigen Linux-Distributionen bringen \LaTeX mit. Da \TeX samt \LaTeX ein umfangreiches Programmpaket bilden, stehen eigene \LaTeX -Distributionen im Netz; eine bekannte ist teTeX von THOMAS ESSER mit der Startseite <http://www.tug.org/teTeX/>. Das Einrichten einer solchen Distribution ist immer noch Arbeit. Debian-Benutzer laden sich einfach mit `apt-get install` die entsprechenden teTeX -Pakete herunter und brauchen sich um nichts weiter zu kümmern.

Im deutschsprachigen Raum werden \TeX und \LaTeX von der *Deutschsprachigen Anwendervereinigung TeX e. V.* (DANTE) mit Sitz in Heidelberg gepflegt. Auf ihrem Webserver <http://www.dante.de/> findet man alles, was zu der Software gehört, insbesondere einen Wegweiser zum *Comprehensive TeX Archive Network* (CTAN) und eine DE-TeX-FAQ-Sammlung, ausgedruckt knapp 200 Seiten. Auch an Einführungen und Anleitungen in diversen Sprachen mangelt es nicht. Wer Geld loswerden will, wird durch eine umfangreiche Bücherliste unterstützt. Als schnelle Einführung ist die bewährte *\LaTeX -Kurzanleitung* von HUBERT PARTL et. al. hervorzuheben, die auf Deutsch oder Englisch an vielen Stellen im Netz zu finden ist und jeder \LaTeX -Distribution beiliegt.

Computeralgebrasysteme wie Maple oder Mathematica arbeiten mit \LaTeX zusammen; aus den Arbeitsblättern lassen sich \LaTeX -Artikel erzeugen. Diese kann man entweder als selbständige Dokumente behandeln oder nach etwas Editieren (Löschen des Vorspanns) als Kapitel oder Abschnitte in andere \LaTeX -Dokumente einbinden. Man hat damit sozusagen ein \LaTeX , das rechnen kann.

Das Adobe-PDF-Format (Portable Document Format) hat sich sehr verbreitet, siehe Abschnitt 5.2 *Werkzeuge zum Lesen und Umwandeln* auf Seite 168. PostScript-Dateien lassen sich in pdf-Dateien umwandeln; besser jedoch ist es, aus \LaTeX -Manuskripten mittels `pdflatex` unmittelbar pdf-Dateien zu erzeugen. Zwischen `latex` und `pdflatex` bestehen kleine Unterschiede beim Einbinden von Grafi-

ken in den Text; erforderlichenfalls sind die Grafiken in zwei Formaten (jpg und eps) zur Verfügung zu stellen, GIMP hilft dabei.

Das wichtigste Stilelement von \LaTeX ist die Dokumentklasse. Diese bestimmt die Gliederung und wesentliche Teile der Aufmachung. Die Dokumentklasse *book* kennt Bände, Kapitel, Abschnitte, Unterabschnitte usw. Ein Inhaltsverzeichnis wird angelegt, auf Wunsch und mit etwas menschlicher Unterstützung auch ein Sachverzeichnis. Bei der Dokumentklasse *report* beginnt die Gliederung mit dem Kapitel, ansonsten ist sie dem Buch ähnlich. Das ist die richtige Klasse für Dissertationen, Diplomarbeiten, Forschungsberichte, Skripten und dergleichen. Mehrbändige Werke sind in diesem Genre selten. Die Dokumentklasse *article* eignet sich für Aufsätze und kurze Berichte. Die Gliederung beginnt mit dem Abschnitt, die Klasse kennt kein Inhaltsverzeichnis, dafür aber eine Zusammenfassung (Abstract). Nützlich ist auch die Dokumentklasse *foils* zur Formatierung von Folien für Overhead-Projektoren. Hier wird das Dokument in Folien gegliedert, die Schriftgröße beträgt 25 oder 30 Punkte. Zum Schreiben von Briefen gibt es eine Klasse *dinbrief*, mit deren Hilfe sich anspruchsvoll gestaltete, normgerechte Briefe erstellen lassen, auch Privatbriefe.

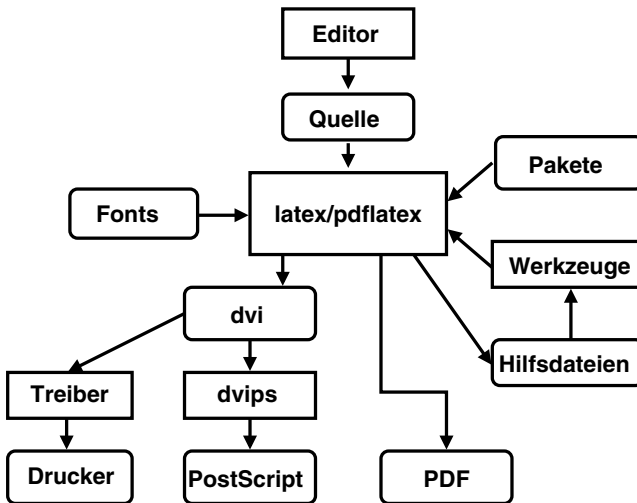


Abbildung 5.12. Arbeitsablauf beim Erzeugen eines \LaTeX -Dokumentes

In Abbildung 5.12 ist der Arbeitsablauf schematisch dargestellt. Man schreibt seinen Text mit einem beliebigen Editor. Dabei wird nur von den druckbaren Zeichen des US-ASCII-Zeichensatzes zuzüglich `<cr>` Gebrauch gemacht. In den Text eingestreut sind \LaTeX -Anweisungen. Der Name der Textdatei (Quelle) muss die Kennung `tex` tragen. Dann schickt man die Textdatei durch den \LaTeX -Übersetzer. Dieser erzeugt eine Binärdatei, deren Namen die Kennung `dvi` trägt. Das bedeutet *device independent*, die Binärdatei ist noch nicht auf ein bestimmtes Ausgabegerät hin ausgerichtet. Verwendet man `pdflatex`, entfällt der Zwischenschritt, man erhält un-

mittelbar eine pdf-Datei. Mittels eines geräteabhängigen Treiberprogramms wird aus der dvi-Datei eine Datei mit der Kennung `bit` erzeugt, die mit einem Linux/UNIX-Kommando wie `cat` durch eine transparente Schnittstelle zum Drucker geschickt wird. Das Programm `dvips` erzeugt aus der dvi-Datei eine PostScript-Datei, die auf einem beliebigen PostScript-Drucker ausgegeben werden kann. Während des Übersetzens schreibt der Compiler mehrere Hilfsdateien – beispielsweise mit Einträgen für den Index oder das Inhaltsverzeichnis – die roh oder nach Bearbeitung durch ein externes Werkzeug dem Compiler beim nächsten Durchgang zugeführt werden. Da der Compiler die Quelle sequentiell durcharbeitet, sind unter Umständen mehrere Läufe erforderlich, bis alle Seitenangaben stimmen. Zu jedem Lauf werden die Hilfsdateien des vorangegangenen Laufs herangezogen.

Zum Schreiben von Privat- oder Geschäftsbriefen verwendet man im deutschen Sprachraum die Dokumentklasse *dinbrief*. Die Klassenbeschreibung heißt `dinbrief.cls` und liegt im Verzeichnis `/usr/share/texmf/tex/latex/dinbrief/`, die zugehörige Dokumentation im Verzeichnis `/usr/share/doc/texmf/latex/dinbrief/`. Im persönlichen Briefverzeichnis legt man sich eine Vorlage (Muster, Schablone, Template) folgender Art an:

```
\documentclass[12pt]{dinbrief}
\usepackage{ngerman, newcent}

% \input{briefkopf}          % sofern vorhanden
\address{\textbf{Dr.-Ing. Wulf Alex}\\
         Latexweg 999\\
         D-12345 Linuxhausen (Baden)\\
         Tel. 01234/567890\\
         Mobiltel. 0160/12345\\
         VoIP 032123456789\\
         Email alex-weingarten@t-online.de}
\backaddress{W. Alex, Latexweg 999, D-12345 Linuxhausen}
% \signature{W. Alex}        % nach Bedarf
\place{Linuxhausen (Baden)}

\begin{document}

\begin{letter}{Superputer GmbH\\
              Debianstra"se 111\\
              \textbf{01234 Linuxberg}}

\subject{Open-Source-Treffen 2006}
\nowindowrules
\opening{Sehr geehrte Damen und Herren,}

Hier folgt der Text.

\closing{Mit freundlichem Gru"s\\Ihr}

% \ps{Postscriptum}          % nach Bedarf
% \cc{Verteiler}             % nach Bedarf
```

```
% \encl{Anlagen}           % nach Bedarf
\end{letter}
\end{document}
```

Quelle 5.2. \LaTeX -Datei `dinbrief.tex` als Vorlage für Briefe

die man kopiert, um einen neuen Brief abzufassen. Nach dem Editieren der Kopie geht es wie mit jedem \LaTeX -Dokument weiter. Effektivitätsbewusste Faulpelze schreiben sich ein Makefile dazu:

```
# Makefile fuer Briefe, W. Alex 2003-01-10

SRC = @$          # vordefinierte make-Variable
LATEX = /usr/bin/latex
PLATEX = /usr/bin/pdflatex
DVIPS = /usr/bin/dvips

.DEFAULT :      # vordefiniertes Target
    ${PLATEX} ${SRC}.tex
    ${LATEX} ${SRC}.tex
    ${DVIPS} -o ${SRC}.ps ${SRC}

clean :
    rm -f *.aux *.log *.dvi
```

Quelle 5.3. Makefile zum Erzeugen von Briefen mittels \LaTeX

und rufen `make` mit dem Namen der Briefdatei ohne Kennung als Argument auf.

Abbildungen

\LaTeX bringt einige Befehle zum Zeichnen mit, die jedoch mehr Frust als Freude hervorrufen. Es gibt Zusatzprogramme (TeXCAD u. a.) zu \LaTeX , die es erleichtern, den Text durch Zeichnungen zu ergänzen. Außerdem kann man Grafiken bestimmter Formate – Encapsulated Postscript, jpeg-Fotos und andere – in den Text einbinden. So wurden die Zeichnungen im Buch mittels `xfig` erstellt, siehe Abschnitt 8.2 *Zeichnen* auf Seite 295, und mit GIMP in eps- und jpg-Dateien umgewandelt. Abbildungen gehören wie Tabellen zu den gleitenden Objekten, das heißt, \LaTeX kann sie im Text im Interesse der Seitenaufteilung verschieben. Dabei gelten einige Regeln:

- Eine Abbildung erscheint frühestens an der Stelle, an der sie im Text genannt wird; verschoben wird nur nach hinten.
- Die Reihenfolge der Abbildungen wird nicht verändert.
- Wenn sich \LaTeX gar nicht mehr zu helfen weiß, verschiebt es die Abbildungen an das Ende des Textes (Kapitels).

Bei Manuskripten mit vielen Abbildungen muss der Autor helfend eingreifen, indem er sie möglichst weit vorn in den Text einfügt oder etwas verkleinert. Zur Abbildung

gehört auch ihre Unterschrift (E: caption), deren Zeilenanzahl man unter Umständen verringern kann. Im vorliegenden Buch wird das Makropaket `graphicx` verwendet; der Code für das Einfügen einer Abbildung in den Text sieht folgendermaßen aus:

```
\begin{figure}
\centering
\includegraphics[width=\figwidth]{bilder/gimp}
\caption[Screenshot GIMP]{Screenshot des Grafikwerkzeugs GIMP}
\label{fig:gimp}
\end{figure}
```

Ohne zwingenden Grund gebe man in der ersten Zeile keine Position (`[htb]`) an, sondern lasse \LaTeX alle Freiheit. Die Variable `figwidth` wird in der persönlichen Style-Datei als 0.96-faches der Breite des Satzspiegels definiert. Die Kennung der Grafikdatei `bilder/gimp` ist nicht angegeben; \LaTeX sucht sich die passende Datei (`eps` oder `jpg`) heraus.

Tabellen

\LaTeX kennt mehrere Formen von Tabellen. Die einfachste ist die fest in den Text eingefügte Tabelle ohne Überschrift. Sie erscheint nicht im Tabellenverzeichnis und wird mittels der `tabular`-Umgebung erzeugt. Mit der `table`-Umgebung definiert man gleitende Tabellen mit Überschrift etc., die sich bei Hinzuziehen des `bigtabular`-Makros auch über mehrere Seiten erstrecken dürfen (im Gegensatz zu Abbildungen).

Formeln

Da \TeX ursprünglich entwickelt worden ist, um mathematische Texte zu setzen, ist der Formelsatz von \TeX und \LaTeX unübertroffen. Auch die Vielfalt an fertigen mathematischen Symbolen übertrifft die meisten Bedürfnisse. Im Anhang werden einige einfache und komplizierte Formeln als Quelle und Ausdruck wiedergegeben, die man als Vorlage für die eigene Arbeit nehmen kann. Hat man oft mit Formeln zu tun, legt man sich eine Sammlung an und erzeugt neue Formeln durch Kopieren und Editieren ähnlicher Formeln.

Will man mit \LaTeX gesetzte Formeln in Dokumente anderer Herkunft (Webseiten) übernehmen, geht das meist nur auf dem Weg der Umwandlung in eine Rastergrafik. Einen Weg dazu zeigt STEFAN LAGOTZKI auf <http://www.lagotzki.de/scripte/formeln/>. Er verwendet Ghostscript und ImageMagick, verpackt in ein Shellskript.

Register (Index)

Ein Register, Index oder Sachverzeichnis gehört zu jedem ordentlichen Sachtext von mehr als ein paar Seiten Umfang. Es erleichtert dem Leser den Zugang, vielleicht

noch mehr als das Inhaltsverzeichnis. Andererseits zählt das Zusammenstellen eines Registers zu den unbeliebtesten Pflichten. Der Autor – ohnehin schon am Ende seiner Kräfte und vier Wochen im Verzug – geht mit Widerwillen an diese Arbeit, die kein rechtes Erfolgserlebnis beschert. \LaTeX greift ihm auch hier unter die Arme.

Im einfachsten Fall schreibt man schon beim Abfassen des Textes vor jeden Absatz Zeilen folgender Art:

```
\index{Stichwort}
```

pro Zeile ein Eintrag. Bei langen Absätzen muss man solche Zeilen auch in die Absätze einfügen, aber stets in eigenen Zeilen, sonst findet man sie schlecht wieder. Die Stich- oder Schlagwörter werden immer in ihrer lexikalischen Grundform (Infinitiv, Nominativ Singular) genannt, sofern nicht die gebeugte Form einen eigenen Begriff bildet. Bei Personennamen wird der Nachname vorangestellt. Die Einträge dürfen nicht die Zeichen `!`, `@` oder `|` enthalten, weil diese im vorliegenden Zusammenhang eine besondere Bedeutung haben. In die Präambel der Hauptdatei kommt eine Zeile:

```
\makeindex
```

die \LaTeX veranlasst, beim Durchgehen des Textes die `\index`-Zeilen samt Seitenzahlen in eine Hilfsdatei mit der Kennung `idx` herauszuschreiben. Weiterhin ist in der Präambel oder in einer gesonderten Datei das Paket *makeidx* einzubinden:

```
\usepackage{makeidx}
```

Das Paket stellt einige für das Registermachen benötigte Befehle bereit. An das Ende der Hauptdatei – meist unmittelbar vor `\end{document}` – kommt die Aufforderung zum Ausgeben des Index:

```
\printindex
```

Weiterhin ist nach ein bis drei \LaTeX -Durchläufen – wenn die Seitenzahlen feststehen – das Kommando:

```
joe@debian:~$ makeindex hauptdatei
```

aufzurufen, das die Aufbereitung der Hilfsdatei veranlasst. Abschliessend muss noch einmal `latex` bemüht werden, um den bearbeiteten Index in das Dokument einzufügen. Es empfiehlt sich, auch den Index Korrektur zu lesen, um Ungereimtheiten zu beseitigen.

Der Index lässt sich in mehrfacher Hinsicht ausgestalten. Zu Einträgen können Untereinträge erzeugt werden:

```
\index{Datei!Besitzer}
```

Das Ergebnis ist auf den letzten Seiten des Buches anzuschauen. \LaTeX erlaubt sogar Unter-Untereinträge, aber die sind selten erforderlich und ergeben ein unruhiges Schriftbild. Das Einsortieren eines Eintrags an anderer Stelle erzwingt ein Klammersaffe:


```
\index{hostname@\textit{hostname}}
\index{Aland@\AA}land}
```

Im ersten Fall erscheint der Eintrag in kursiver Schrift (Italics) an der richtigen Stelle, obwohl die Zeichenkette mit einem \LaTeX -Befehl beginnt, im zweiten wird die Inselgruppe unter A eingeordnet und nicht unter dem im deutschen Alphabet unbekannten Buchstaben Å und auch nicht wie im Schwedischen üblich hinter Z.

Verweise auf Synonyme oder Erklärungen erzeugt man mittels Eintragungen folgender Art:

```
\index{Computer|see{Rechner}}
\index{IP|see{Internet Protocol}}
```

Das Wörtchen *see* lässt sich durch eine Zeile in der Präambel umdefinieren:

```
\renewcommand*{\seename}{\textbf{$\rightarrow$}}
```

Hat man viele Verweise, packt man sie zweckmäßig in eine eigene Datei namens `verweise.tex`, die man in der Präambel mittels

```
\input{verweise}
```

einbindet. Für das vorliegende Buch haben wir sogar zwischen Synonymen und Abkürzungen unterschieden.

Will man das Register aufteilen in einen Personen- und einen Sachindex, ist das mühelos machbar. Man bindet dazu in der Präambel statt des Paketes `makeidx` das Paket `index` ein und streicht die Zeile `\makeindex`. Ferner definiert man:

```
\newindex{per}{pdx}{pnd}{Personenindex}
\newindex{sac}{sdx}{snd}{Sachindex}
\newcommand{pindex}[1]{\index[per]{#1}}
\newcommand{sindex}[1]{\index[sac]{#1}}
```

Damit stehen die Kommandos `\pindex{}` und `\sindex{}` zum Benennen der jeweiligen Einträge zur Verfügung. Wir haben jetzt zwei Hilfsdateien aufzubereiten:

```
joe@debian:~$ makeindex -o hauptdatei.pnd
hauptdatei.pdx
```

```
joe@debian:~$ makeindex -o hauptdatei.snd
hauptdatei.sdx
```

und müssen am Ende des Dokuments auch zwei Indexdateien einbinden:

```
\printindex[per]
\printindex[sac]
```

Alles Übrige wie gehabt. Natürlich lassen sich auch noch ein `Kommandoindex` und ein `Ortsindex` erzeugen. Der Begeisterung sind kaum Grenzen gezogen.

Fehlersuche in L^AT_EX-Dateien

T_EX/L^AT_EX ist nicht fehlertolerant, sondern gnadenlos. Bei der kleinsten Unstimmigkeit bricht die Übersetzung ab. Andererseits unterlaufen beim Schreiben Tipp- und Syntaxfehler, besonders wenn Mitternacht vorüber ist. Die erste Empfehlung lautet, die zu korrigierenden Textbrocken nicht zu groß werden zu lassen. Dazu unterteilt man einen großen Text in Dateien wie im Abschnitt 5.15 *Organisation eines Textprojekts* auf Seite 239 beschrieben. Die häufigsten Fehler sind:

- fehlende, überzählige oder falsche Klammern,
- Verwechseln von Schräg- und Gegenschragstrich,
- Metazeichen nicht gequotet, wenn man sie buchstäblich haben will,
- Vertippen bei Kommandos (ergibt ungültige Kommandos).

Manche Fehler machen sich erst viele Zeilen hinter ihrem Entstehungsort bemerkbar, spätestens am Ende des Dokuments. Tippfehler im eigentlichen Text interessieren L^AT_EX nicht, dafür sind Rechtschreibprüfer wie `ispell` zuständig.

Syntaxprüfer für T_EX/L^AT_EX-Texte sind `lacheck` und `chktex`, vergleichbar `lint` für C-Programme. Es macht Sinn, beide zu benutzen. Sie arbeiten schneller als der Übersetzer `latex` und warnen auch, wenn ihnen etwas verdächtig erscheint, ohne dass es gleich falsch zu sein braucht. Beide können einzelne Abschnitte prüfen, brauchen also nicht immer das gesamte Dokument. Schreibt man deutsche Umlaute in der Form "a, ist es angebracht, diesbezügliche Warnungen von `chktex` per Option zu unterdrücken und auch gleich noch die mittlerweile überflüssigen Warnungen wegen fehlender Italic-Korrektur:

```
joe@debian:~$ chktex -n18 -n6 latexdatei
```

Ihre Grenze erreichen die Werkzeuge bei komplizierten, verschachtelten Makros.

Zum Untersuchen des eigentlichen Textes auf Schreibfehler dienen Rechtschreibprüfer (E: spellchecker) wie `ispell`, siehe Abschnitt 5.13.4 *Rechtschreibung prüfen* auf Seite 231. Das Kommando kennt einen T_EX/L^AT_EX-Modus, in dem es offensichtliche L^AT_EX-Kommandos überliest. Wenn es auf ein deutsches Wörterbuch eingestellt ist, stört es sich verständlicherweise an englischen Wörtern, an manchen Schreibweisen von Sonderzeichen sowie an dem Einfügen von V zum Verhindern von Ligaturen in Wörtern wie *auffordern*. Die von `ispell` beanstandeten Wörter lassen sich verbessern, übergehen oder in ein persönliches Wörterbuch aufnehmen. Insgesamt gesehen ist der Einsatz von `ispell` anzuraten. Ein paar L^AT_EX-Fehler findet der Prüfer nebenbei, weshalb man ihn vor einem Syntaxprüfer einsetzen sollte. Zum Verhindern falscher L^AT_EX-Ligaturen in deutschen Texten dient das kleine Werkzeug `rmligs-german`. Man könnte sich einen Beautifier vorstellen, der sprachenbezogen ein L^AT_EX-Manuskript auf typografische Unschönheiten abklopft. Der müsste aber auch das Dickicht der Makros durchforsten, was nicht trivial ist.

Ferner kopiert man zweckmäßig die Hauptdatei (`main.tex` oder ähnlich) in eine Datei `test.tex` und fügt der Präambel eine Zeile folgender Art hinzu:

```
\includeonly{kapitel3/kap}
```

Die Zeile bezieht sich auf eine Zeile im Hauptteil:

```
\include{kapitel3/kap}
```

mit der eine Kapiteldatei eingebunden wird. Der `\includeonly`-Befehl führt dazu, dass trotz mehrerer in das Werk einzubindender Kapitel für Testzwecke nur jeweils ein Kapitel herangezogen wird. In der Datei `test.tex` kann man auch weitere Dinge ausprobieren, ohne die Hauptdatei zu ändern oder zu beschädigen. Beispielsweise kann man das Paket `syntonly` einbinden und im Vorspann mit dem Befehl `\syntonly` bewirken, dass nur die Syntax überprüft wird. Erscheint diese fehlerfrei, kommentiert man den Befehl aus und übersetzt.

Der \LaTeX -Übersetzer schreibt Meldungen auf den Bildschirm und gleichzeitig in eine Datei `main.log` oder `test.log`. Diese bricht im Fehlerfall mit einer Meldung ab, die nicht immer hilfreich ist, aber wenigstens den Ort des Fehlers eingrenzt. Oft reicht es aus, sich mit einem Editor die Umgebung des Fehlers anzusehen, um ihn zu finden. Die temporäre Reparatur des Fehlers aus dem Übersetzer heraus ist in der Regel nicht zu empfehlen.

Ist die den Fehler enthaltende Datei groß, lassen sich Teile daraus auskommentieren. Einzelnen Zeilen stellt man ein Prozentzeichen voran, ganze Abschnitte rahmt man mit `\iffalse` und `\fi` ein, um sie von der Bearbeitung durch den Übersetzer auszuschließen. `\iffalse` ist eigentlich ein \TeX -Kommando zum Basteln von Verzweigungen, erfüllt jedoch hier genau unseren Wunsch. In verzweifelten Fällen haben wir die fehlerhafte Datei auf diese Weise schrittweise halbiert und den Fehler mittels einer Art von binärer Suche eingekreist.

KDE Integrated \LaTeX Environment (kile)

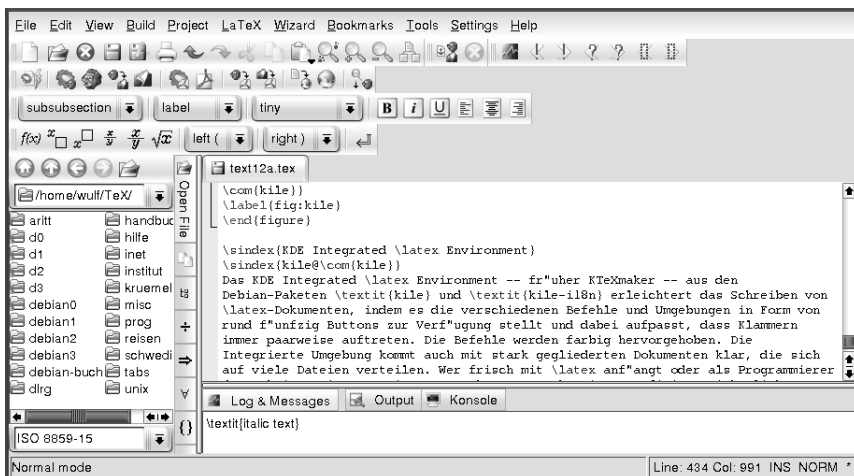


Abbildung 5.13. Screenshot des KDE Integrated \LaTeX Environments kile

Das KDE Integrated \LaTeX Environment – früher K \TeX maker – aus den Debian-Paketen *kile* und *kile-i18n* erleichtert das Schreiben von \LaTeX -Dokumenten, indem es die verschiedenen Befehle und \LaTeX -Umgebungen in Form von rund fünfzig Buttons zur Verfügung stellt und dabei aufpasst, dass Klammern immer paarweise vorkommen. Befehle werden farbig hervorgehoben. Die Fehlersuche wird dadurch erleichtert, dass man gleichzeitig die Protokolldatei des Übersetzerlaufs und den Quelltext vor Augen hat. Die integrierte Umgebung kommt auch mit stark gegliederten Dokumenten klar, die sich auf viele Dateien verteilen. Wer frisch mit \LaTeX anfängt oder als Programmierer das Arbeiten mit Integrierten Umgebungen gewohnt ist, profitiert von dem Werkzeug. Abbildung 5.13 zeigt ein Beispiel, und zwar genau den vorliegenden Abschnitt, der versuchsshalber unter *kile* getippt wurde. Ein bereits laufendes großes Textprojekt auf *kile* umzurüsten, ist weniger zu empfehlen. *kile* ist kein WYSIWYG-Editor, hierzu siehe Abschnitt 5.6.4 *LyX* auf Seite 201.

Metafont

Metafont ist ein Software-System zum Erzeugen von Fonts für \TeX / \LaTeX . Da der normale Benutzer hoffentlich nicht auf den Gedanken verfällt, eigene Fonts gestalten zu wollen, bräuchte Metafont ihn nicht zu interessieren. Aber man kann mit Metafont Logos, exotische Symbole oder allgemein gesprochen komplexe grafische Pixelbilder (Bitmaps) erzeugen, und das kann schon einmal gefragt sein. Eine Gebrauchsanleitung für Metafont würde den Rahmen des Buches sprengen. Die Metafont-Bibel (zwei Bände) stammt von DONALD E. KNUTH, eine kurze, deutschsprachige Einführung gibt HELMUT KOPKA in einem seiner \LaTeX -Bücher (die viel zur Verbreitung von \LaTeX im deutschsprachigen Raum beigetragen haben).

5.12 Schreiben für das Web

5.12.1 Was ist Hypertext?

Bei Hypertext und der Hypertext Markup Language (HTML) geht es auch um das Formatieren von Texten oder allgemeiner von Hypermedia-Dokumenten, aber es kommt noch etwas hinzu. Hypertexte enthalten Hyperlinks, kurz Links (F: *lien*) genannt und nicht mit den Links im Dateisystem zu verwechseln. Das sind Verweise, die elektronisch auswertbar sind, sodass man ohne Suchen und Blättern zu anderen Hypertexten weitergeführt wird. Man kann die Hyperlinks als aktive Querverweise bezeichnen. Der Begriff wurde Anfang der sechziger Jahre von TED NELSON geprägt. Auf Papier erfüllen Fußnoten, Literatursammlungen, Register, Querverweise und Konkordanzen einen ähnlichen Zweck, ohne elektronischen Komfort. Im ersten Kapitel war von WALLENSTEIN die Rede. Von diesem Stichwort könnten Verweise auf das Schauspiel von FRIEDRICH SCHILLER, die Werke von ALFRED DÖBLIN, RICARDA HUCH, PETER ENGLUND oder auf die Biografie von GOLO MANN führen, die die jeweiligen Texte auf den Bildschirm bringen, in SCHILLERS Fall sogar mit einem Film. In den jeweiligen Werken wären wieder Verweise enthalten, die auf

Essays zur Reichsidee oder zur Rolle Böhmens in Europa lenken. Leseratten würden vielleicht auf dem Alexanderplatz in Berlin landen oder bei einem anderen Vertreter der schreibfreudigen Familie MANN. Von dort könnte es nach Frankreich, Indien, Ägypten, in die USA oder die Schweiz weitergehen. Vielleicht findet man auch Bemerkungen zum Verhältnis zwischen Literatur und Politik. Beim Nachschlagen in Enzyklopädien gerät man manchmal ins ziellose Schmökern. Mit Hypertext ist das noch viel, viel schlimmer. So ist jede Information eingebettet in ein Gespinnst oder Netz von Beziehungen zu anderen Informationen, und wir kommen zum World Wide Web (WWW, W3).

5.12.2 Hypertext Markup Language (HTML)

Mit dem Web entstand der Bedarf an einer Sprache für Hyperdokumente. \LaTeX konnte den Bedarf nicht decken, da es für die Ausgabe auf Papier geschaffen wurde. Also ersannen TIMOTHY BERNERS-LEE und ROBERT CAILLIAU 1989 am CERN in Genf neben den Uniform Resource Locators (URL) und dem Hypertext Transfer Protocol (HTTP) auch noch die Hypertext Markup Language (HTML), in der Hyperdokumente – Webseiten – geschrieben werden. Aktuell ist Version 4.01, die Nachfolgerin XHTML ist jedoch schon da, siehe Abschnitt 5.12.4 *Extensible Markup Language* auf Seite 227. HTML erreicht für Druckerausgaben nicht die Leistung von \LaTeX . Beiden Sprachen gemeinsam ist das Konzept der Trennung von Inhalt, Struktur und Darstellung. HTML-Quellen kommen mit dem ASCII-Zeichensatz aus. Sie können externe Grafik- und Sound-Dateien verschiedener Formate einbinden.

Zum Lesen von Hypertexten braucht man HTML-Browser wie Amaya, Netscape, Mozilla, Firefox, Galeon, Opera, Mosaic, Konqueror oder den MS Internet-Explorer. Leider halten sich die wenigsten Browser an den gültigen HTML-Standard. Sie erkennen nicht alle standardkonformen HTML-Konstrukte und bringen eigene Vorstellungen mit. Wenn man HTML-Dokumente für die Öffentlichkeit schreibt, sollte man daher seine Erzeugnisse mit verschiedenen Browsern betrachten und überdies mit mehreren Werkzeugen testen.

Ein einfaches HTML-Dokument, das bei weitem nicht alle Möglichkeiten von HTML ausreicht, ist schnell geschrieben:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN">

<HTML>

<HEAD>
<TITLE>Institut fuer Hoeheres WWW-Wesen</TITLE>
</HEAD>

<BODY BGCOLOR="#ffffff">

<IMG SRC="http://logowww.jpg" alt="logo">

<H3>
Fakult&auml;l;t f&uuml;r Internetwesen
```

```

</H3>

<HR>

Geb&auml;ude 30.70 <BR>
Telefon +49 721 608 9999 <BR>

<H4>
Leiter der Verwaltung
</H4>
Dipl.-Ing. Schorsch Meier

<H4>
Werkstattleiter
</H4>
Alois Wuchtbrumme

<HR>
Zur <A HREF="http://www.uka.de/">Universit&auml;t</A>
<HR>

http://www.mvm.uni-karlsruhe.de/index.html<BR>
J&uuml;ngste &Auml;nderung 2004-11-25
<A HREF="mailto:webmaster@mvm.uka.de">Webmaster</A>

</BODY>
</HTML>

```

Quelle 5.4 . Quelltext eines einfachen HTML-Dokuments (Webseite)

Die erste Zeile ist HTML-Kommentar, beschreibt den Typ des Dokuments und kann vom Checker `tidy` erzeugt werden, falls man sie nicht selbst abfasst. Das eigentliche Dokument wird durch `<HTML>` und `</HTML>` eingerahmt. In seinem Inneren finden sich die beiden Teile `<HEAD>` und `<BODY>`. Die Formatanweisungen `<H3>` usw. markieren Überschriften (Header). Sonderzeichen werden entweder durch eine Umschreibung (entity) wie `ä` oder durch die Nummer im Latin-1-Zeichensatz wie `ä` dargestellt. `
` ist ein erzwungener Zeilenumbruch (break), `<HR>` eine waagrechte Linie (horizontal ruler). Am Ende sollte jedes Dokument seinen Uniform Resource Locator (URL) enthalten, damit man es wiederfindet, sowie das Datum der jüngsten Änderung und die Email-Adresse des Verantwortlichen.

Zum Schreiben von HTML-Dokumenten (Webseiten) braucht man kein besonderes Programm; der einfachste Editor, der US-ASCII beherrscht, reicht aus. Einige Spezialeditoren können jedoch bei komplizierten Dokumenten die Arbeit erleichtern. Dazu zählen:

- amaya, der Referenz-Editor und -Browser des W3-Konsortiums, <http://www.w3.org/Amaya/>,
- bluefish, <http://bluefish.openoffice.nl/>,
- Phase5, <http://www.qhaut.de/>, setzt wine voraus,

- Arachnophilia, <http://www.arachnoid.com/>, setzt Java voraus,
- tkhtml, <http://www.hwaci.com/sw/tkhtml/>, setzt Tcl/Tk voraus,
- nvu, <http://nvu.com/>, abgeleitet aus dem Mozilla Composer und gepflegt von der Firma Linspire, unter der GPL,
- Netscape bzw. Mozilla Composer, <http://www.mozilla.org/>.

Hingegen sind die von Word, Access, Frontpage oder älteren Versionen des Composers erzeugten HTML-Dokumente oft nicht mit dem HTML-Standard konform und unnötig groß. Man sollte sie erst nach sorgfältiger Prüfung veröffentlichen. Die kommerziellen HTML-Editoren SoftQuad HotMetal, Macromedia Dreamweaver und Adobe GoLive sind nicht für Linux verfügbar.

Vor einem verbreiteten Fehler sei gewarnt. Die HTML-Kommandos beschreiben eine Struktur, nicht das Aussehen. Viele Kommandos lassen sich missbrauchen, was in manchen Zusammenhängen gut geht, in anderen nicht. Das `
`-Element erzeugt einen Zeilenumbruch, mehrere aufeinander folgende `
`-Elemente also Leerzeilen. Diese kann ich auch mittels mehrerer `<P>`-Elemente hervorrufen, die eigentlich zum Einrahmen von Absätzen oder Paragraphen gedacht sind. Abgesehen von der unterschiedlichen Syntax (Attribute etc.) kann ein Browser unter einem Paragraphen etwas anderes verstehen als einen Zeilenwechsel, man weiß das nie sicher. Beliebt ist der Missbrauch von Tabellen zur Darstellung von mehrspaltigem Text. Spätestens beim Lesen des Dokumentes mit einem zeilenweise arbeitenden Screen-Reader geht das voll daneben. Man nehme also immer das HTML-Element, das den eigenen Wunsch logisch genau wiedergibt, und nicht eines, das unter den gegenwärtigen, zufälligen Umständen den gewünschten Effekt auf dem Bildschirm erzeugt.

HTML-Dateien tragen die Kennung `html` oder `htm`, letztere in der auf Dateinamen nach dem 8.3-Muster beschränkten Welt (DOS und Nachfolger). Dokumente können Anweisungen an den Web-Server enthalten, die dieser bei einer Anfrage nach dem Dokument vor dem Senden der Antwort ausführt. Diese Server Parsed Documents oder Webseiten mit Server Side Includes tragen oft die Kennung `shtml`, aber das ist nicht zwingend. Beispiele sind Zähler oder bei jeder Anfrage aktualisierte Tabellen oder Datumsangaben. Im Gegensatz dazu stehen Dokumente mit Anweisungen an den Client (Browser).

Es ist dringend anzuraten, eine neue oder veränderte Webseite durch mindestens ein Prüfwerkzeug untersuchen (validieren) zu lassen. Das Kommando `tidy` ist ein einfach anzuwendender Syntaxchecker, der auf Wunsch auch gleich noch den Quelltext in eine ansprechende Form bringt (<http://www.w3.org/People/Raggett/tidy/> oder <http://sourceforge.net/projects/tidy/>). Der Aufruf:

```
joe@debian:~$ tidy -ium alex.html
```

rückt in der HTML-Datei `alex.html` die Zeilen ähnlich wie bei einer Programmquelle ein, schreibt alle Tags in Versalien und modifiziert die Originaldatei. Dabei werden kleine Fehler korrigiert und größere beanstandet. Eine zweite Möglichkeit besteht darin, die Webseite <http://validator.w3c.org/> um die Validierung zu bitten. Bei <http://jigsaw.w3.org/css-validator/>

kann man etwaige Cascading Style Sheets (CSS) untersuchen lassen. Schließlich prüft <http://bobby.watchfire.com/bobby/> Webseiten auf Zugänglichkeit. Das Werkzeug `linklint` stellt fest, ob Hyperlinks funktionieren (<http://www.linklint.org/>). Darüber hinaus sollte man sich sein Produkt mit mindestens zwei unterschiedlichen Browsern ansehen. Das kostet Zeit, aber nur so ist zu gewährleisten, dass die Mehrzahl der Surfer keine Schwierigkeiten mit der Seite haben. Und das wollen Sie doch, oder?

Wer sich näher mit der Gestaltung von HTML-Dokumenten und Webseiten befassen will, sollte unbedingt die Website <http://de.selfhtml.org/> von STEFAN MÜNZ besuchen. Dort liegt ein umfassendes und ausgereiftes Werk in deutscher Sprache zu dem Thema. Aktuell ist die Version 8.1; das zip-Archiv zum Herunterladen umfasst 8,5 MB. Band 1 der zweibändigen Buchausgabe ist 1300 Seiten stark. Kürzer ist die *HTML-Einführung* des durch seine \LaTeX -Kurzeinführung bekannten Autors HUBERT PARTL. Darüber hinaus sind im Netz weitere Anleitungen und die ausführliche, verbindliche Spezifikation von HTML verfügbar, diese bei <http://www.w3c.org/> aktuell in der Version 4.01.

5.12.3 Standard Generalized Markup Language (SGML)

Texte, Bilder und Tabellen werden von HTML gut unterstützt. Zum Schreiben von mathematischen Formeln sind zwar im HTML-Standard Wege vorgesehen, die sich an \LaTeX anlehnen, die gängigen HTML-Browser geben jedoch die Formeln nicht wieder, sodass manche Autoren getrennte \LaTeX - und HTML-Fassungen ihrer Manuskripte herstellen (müssen). Aus dieser unbefriedigenden Lage schafft einen Ausweg die Standard Generalized Markup Language (SGML) nach ISO 8879, eine Sprache (Metasprache) zur Beschreibung von Sprachen wie HTML oder \LaTeX , deren Anfänge in den sechziger Jahren⁶ von CHARLES GOLDFARB bei IBM entwickelt wurden. Zugespißt lässt sich SGML als die Mutter aller Markup-Sprachen bezeichnen. Man verfasst einen Text mit einer Formatierungssprache (Markup Language) eigener Wahl und vielleicht sogar eigener Zucht. In einem zweiten Dokument namens Document Type Definition (DTD) beschreibt man dazu in SGML, was die Konstrukte der Markup Language bedeuten. Eine weiterverarbeitende Anwendung muss die Sprachdefinition verstehen, das heißt SGML beherrschen, und kann dann den Inhalt des Dokumentes verarbeiten. Ein SGML-Browser erzeugt aus Text und DTD nach Wunsch ASCII-, \LaTeX - oder HTML-Vorlagen für die jeweiligen Zwecke. Dieses allgemeine Konzept läuft proprietären WYSIWYG-Textprozessoren zuwider. Es ist zunächst aufwendiger, aber langfristig effektiver. Einzelheiten wie immer im Netz, Einstieg bei Wikipedia, Suchwort *SGML*, und <http://www.w3.org/MarkUp/SGML/>.

5.12.4 Extensible Markup Languages (XML, XHTML)

SGML ist für manche Aufgaben ein zu schweres Werkzeug. Andererseits möchte man gelegentlich mehr machen, als HTML erlaubt. Hier hilft die Extensible

⁶Die sechziger Jahre waren ein sehr fruchtbares Jahrzehnt. Viele Dinge, die heute in der Computerei Allgemeingut sind, wurden in dieser Zeit erdacht.

Markup Language (XML) mit Möglichkeiten, eigene Tags (Formatieranweisungen) zu definieren. Man könnte XML als eine abgespeckte SGML bezeichnen, als eine Teilmenge. Ehe man an XML oder SGML geht, sollte man ein einwandfreies HTML-Dokument und Style Sheets (CSS) schreiben können, was nicht bedeutet, alle Feinheiten von HTML zu beherrschen. Obwohl sich XML-Dokumente mit jedem Editor schreiben lassen, gibt es auch spezielle XML-Editoren wie *kxmleditor* aus dem KDE-Projekt. Bei <http://www.conglomerate.org/> entsteht ein XML-Editor, der unter der GPL verfügbar ist. Einen Umwandler von XML nach mehreren anderen Formaten enthält das Debian-Paket *xmlo*. Einzelheiten ebenfalls im Netz, Einstieg bei Wikipedia, Suchwort *Extensible Markup Language*, und <http://www.w3.org/XML/>. Auch STEFAN MÜNZ (siehe oben) erklärt XML. Bei allem, was mit XML zu tun hat, ist gegenwärtig viel im Fluss.

Die Extensible Hypertext Markup Language (XHTML) ist die Weiterentwicklung von HTML 4.01 und dabei, diese abzulösen. Die Sprache wird mittels XML anstelle von SGML beschrieben. Die Umwandlung von HTML-Dokumenten nach XHTML ist einfach; vor allem sind strengere syntaktische Vorgaben zu befolgen. Ein Web-Browser, der XHTML noch nicht versteht, zeigt XHTML-Dokumente wie gewöhnliche HTML-Dokumente an, sodass ein Übergang im Web gewährleistet ist. Ein deutschsprachiges Tutorial zu XHTML von JAN WINKLER liegt unter http://www.html-world.de/program/xhtml_ov.php.

5.12.5 Web-Entwicklungsumgebung (quanta)

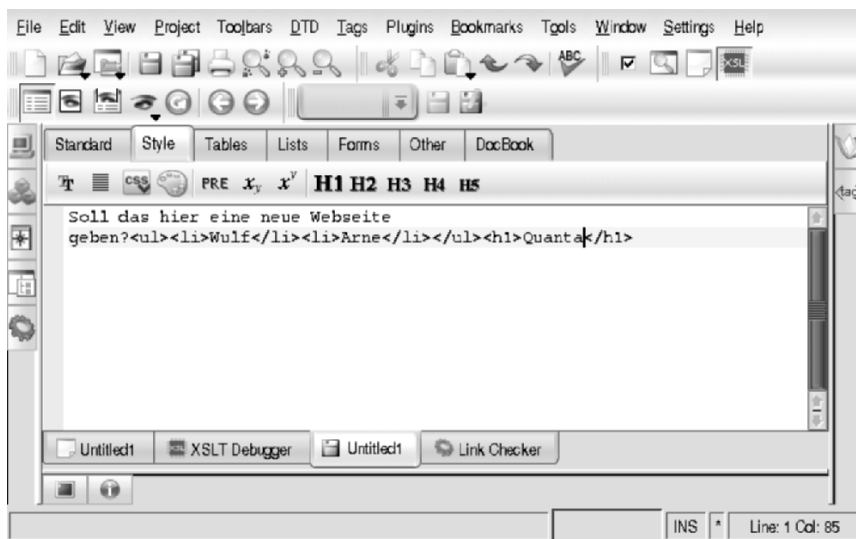


Abbildung 5.14. Screenshot der Web-Entwicklungsumgebung Quanta Plus aus dem KDE-Projekt, ein schwerer Hammer für umfangreiche Web-Projekte

Quanta Plus ist eine Web-Entwicklungsumgebung für KDE. Sie hat nichts mit kommerziellen Produkten ähnlichen Namens gemein. Die Manualseite hilft kaum weiter; die wesentliche Dokumentation liegt im DocBook-Format im Verzeichnis `usr/share/doc/kde/HTML/en/quantum/` und ist im KDE Help Center unter den *Application Manuals*, Punkt *Development* zu finden. Der Browser Konqueror sollte mit dem URL `help:/quantum/` die Dokumentation anzeigen, kam aber über das Auflisten des Verzeichnisses nicht hinaus, vermutlich weil ihm ein nicht näher bekanntes Plugin fehlte. Eine Konvertierung der DocBook-Dateien nach HTML oder PDF scheiterte an Syntaxfehlern. Die beste Hilfe findet man auf <http://quantum.kdewebdev.org/>.

Das Werkzeug ist gedacht zum Gestalten eines Projektes aus vielen Webseiten durch eine Gruppe von Entwicklern. Als einfacher HTML-Editor für den Einzelkämpfer im stillen Kämmerlein ist es zu umfangreich. Bevor die erste Webseite das Licht der Welt erblickt, ist das Projekt anzulegen, sind globale, projektbezogene und persönliche Vorlagen zu erstellen, Zugriffsrechte zu vergeben usw. Das übt man möglichst an einem kleinen Versuchsprojekt.

5.13 Ergänzende Textwerkzeuge

5.13.1 Tipptrainer (ktouch, tipptrainer, tuxtype)

Beim Arbeiten mit Tastaturen soll man alle Finger benutzen. Das Schreiben geht flotter, und die Last verteilt sich gleichmäßiger. Wer sich die Kunst des Schreibens mit zehn Fingern selbst beibringen oder auch nur üben will, greift zu einem Tipptrainer. Das KDE-Werkzeug KTouch erlaubt die Wahl einer Sprache und einer Tastatur und gibt dann Zeichenfolgen vor, die der Übende abschreiben soll, wobei jeder Fehler beanstandet und für die Statistik gezählt wird. Abbildung 5.15 zeigt das Aussehen. Eine Alternative enthält das Debian-Paket *tipptrainer*, zu Hause im PingoS-Projekt (<http://www.pingos.org/>). Kinder werden eher von dem Kommando und Debian-Paket *tuxtype* angesprochen, auf dessen Schirm Buchstabenfische vom Himmel fallen, leider nur mit englischem Text, ansonsten nett gemacht. Weitere Tipptrainer bietet Sourceforge unter dem Suchbegriff *typing tutor* an.

5.13.2 Vergleichen (diff, cmp, ed)

Für einzelne Aufgaben der Textverarbeitung gibt es Spezialwerkzeuge in Linux/UNIX. Häufig gebraucht werden das bereits erläuterte `grep` (1) (= global regular expression print). Mittels `diff` werden die alte und die neue Version einer Datei miteinander verglichen. Bei entsprechendem Aufruf wird eine dritte Datei erzeugt, die dem historischen Editor `ed` als Kommandoskript (ed-Skript) übergeben werden kann, sodass dieser aus der alten Version die neue erzeugt. Gebräuchlich zum Aktualisieren von Programmquellen. Schreiben Sie sich eine kleine Textdatei `alt`, stellen Sie eine Kopie namens `neu` davon her, verändern Sie diese und rufen Sie dann `diff` auf:



Abbildung 5.15. Screenshot des KDE-Tiptrainers KTouch

```
joe@debian:~$ diff -e alt neu > edscript
```

Fügen Sie mit einem beliebigen Editor am Ende des `edscript` zwei Zeilen mit den `ed`-Kommandos `w` und `q` (write und quit) hinzu. Dann rufen Sie den Editor `ed` mit dem Skript auf:

```
joe@debian:~$ ed - alt < edscript
```

Anschließend vergleichen Sie mit dem simplen Kommando `cmp` die beiden Versionen `alt` und `neu` auf Unterschiede:

```
joe@debian:~$ cmp alt neu
```

Durch den `ed`-Aufruf sollte die alte Version genau in die neue Version überführt worden sein, `cmp` meldet nichts. Das Werkzeug `diff3` vergleicht drei Textdateien zeilenweise.

5.13.3 Sortieren (sort)

Über das Sortieren sind Bücher geschrieben worden, ein bekanntes von DONALD E. KNUTH. Für einfache Sortieraufgaben steht das Werkzeug `sort` zur Verfügung. Per Option teilt man ihm mit, ob alphabetisch oder numerisch, aufsteigend oder absteigend und nach welchen Feldern ein Text zeilenweise sortiert werden soll. Das Kommando wird oft als Glied einer Pipe eingesetzt, siehe das Shellskript `frequenz` auf Seite 101.

Zur Übung schreiben wir eine unsortierte, zweiseitige Liste mit Familiennamen und Telefonnummern. Die Datei namens `liste` soll auch einige mehrfache (identische) Einträge enthalten, beliebig verstreut. Dann bearbeiten wir die Datei mit `sort` und schauen uns das Ergebnis auf dem Schirm an:

```
joe@debian:~$ sort liste

joe@debian:~$ sort -u liste

joe@debian:~$ sort -r liste

joe@debian:~$ sort --key=2 liste
```

Der erste Aufruf sortiert die Liste mit den Defaulteinstellungen, also die Zeilen alphabetisch aufsteigend. Der zweite macht dasselbe und wirft dabei mehrfache Zeilen raus (`u` = unique). Der dritte sortiert in umgekehrter Folge (absteigend, `r` = reverse), der vierte schließlich sortiert nach dem zweiten Feld (Telefonnummer), wobei die Feldzählung mit 1 beginnt und als Trennzeichen das Leerzeichen oder TAB angenommen wird. Man kann sogar angeben, ab welchem Zeichen im Feld der Sortierschlüssel beginnen soll, und wo er endet. Die zahlreichen Optionen erlauben die Anpassung an fast jede Sortieraufgabe. Das Locale sollte von `sort` beachtet werden.

5.13.4 Rechtschreibung prüfen (ispell, aspell, acheck)

Das alte UNIX-Werkzeug `spell` untersuchte Texte auf Verstöße gegen die nord-amerikanische Rechtschreibung. Das aus dem GNU-Projekt stammende jüngere Werkzeug `ispell` gibt sich international. Als Debian-Paket findet sich noch ein Werkzeug `aspell`, das behauptet, intelligenter zu sein als `ispell`, eine Kombination aus `ispell` und Metaphone. Beide bringen keine Wörterbücher mit; diese sind getrennt einzurichten, beispielsweise für Bretonisch oder Bengali. Auch ein deutsches Wörterbuch gibt es. Beim Korrigieren lernen sie mit Unterstützung des Benutzers neue Wörter und speichern sie in einem persönlichen Wörterbuch (`$HOME/.ispell_default` bzw. `$HOME/.aspell.de.pws`). Die persönlichen Wörterbücher lassen sich editieren; man sollte sich aber darauf beschränken, offensichtliche Fehleinträge zu löschen. Grafische Frontends gibt es auch, um überschüssige Prozessorkraft zu verbraten. Die Werkzeuge lassen sich in andere Programme wie Mail User Agents oder den Emacs einbinden, sodass sie aus diesen heraus aufgerufen werden können.

Falls `ispell` mit einem deutschen Wörterbuch als Default eingerichtet ist, untersucht der Aufruf:

```
joe@debian:~$ ispell kapitel5.tex
```

die Datei `kapitel5.tex` auf Verstöße gegen die deutsche Rechtschreibung, wobei das Programm auf Grund der Dateikennung `tex` automatisch von einer \LaTeX -Datei ausgeht und Formatanweisungen überliest. Hat man die Debian-Pakete `aspell` und `aspell-de` eingerichtet, so lässt sich mittels:

```
joe@debian:~$ aspell check --lang=german -t
kapitel5.tex
```

die gleiche Untersuchung wie vorstehend durchführen. `aspell` zeigt bei Korrekturvorschlägen etwas mehr Phantasie als `ispell`, kommt aber ohne weitere Konfiguration mit den deutschen Umlauten in \LaTeX nicht zurecht. Kein Rechtschreibprüfer erkennt Fehler, die das Verständnis eines Satzes erfordern (Maße vs. Masse, Tücher vs. Tuche, falscher Artikel, falsche Verbform, fehlendes Wort, falsche Wortfolge).

Das Werkzeug `acheck` zusammen mit dem Paket `acheck-rules` prüft die Lokalisation (Schreibweise von Datum etc.) eines Textes nach gegebenen Regeln und findet auch einige grammatische oder stilistische Fehler. Mit der Option `-s` zieht es gleichzeitig `aspell` zur Rechtschreibprüfung heran. Da sein Autor Franzose ist, gibt es auch einen französischen Regelsatz.

5.13.5 Weitere Werkzeuge (`cut`, `paste` usw.)

Weitere Werkzeuge, deren Syntax man im Manual nachliest, sollen hier nur tabellarisch aufgeführt werden:

- `bfs` big file scanner, untersucht große Textdateien auf Muster,
- `col` filtert Backspaces und Reverse Line Feeds heraus,
- `comm` common, vergleicht zwei sortierte Dateien auf gemeinsame Zeilen,
- `cut` schneidet Spalten aus Tabellen heraus,
- `expand/unexpand` wandelt Tabs in Leerzeichen um und umgekehrt,
- `fold` faltet lange Zeilen (bricht Zeilen um),
- `hyphen` findet Zeilen, die mit einem Trennstrich enden,
- `nl` number lines, nummeriert Zeilen,
- `paste` mischt Dateien zeilenweise,
- `ptx` permuted index, erzeugt einen permutierten Index,
- `rev` reverse, mu `nelieZ` `trhek`,
- `rmnl` remove newlines, entfernt leere Zeilen,
- `ssp` entfernt mehrfache leere Zeilen,
- `tr` translate, ersetzt Zeichen,
- `uniq` findet wiederholte Zeilen in einer sortierten Datei,
- `vis` zeigt eine Datei an, die unsichtbare Zeichen enthält,
- `wc` word counter, zählt Zeichen, Wörter, Zeilen.

Die Liste lässt sich durch eigene Werkzeuge beliebig erweitern. Das können Programme oder Shellskripte sein. Hier ein Beispiel zur Beantwortung einer zunächst anspruchsvoll erscheinenden Fragestellung mit einfachen Mitteln. Ein Sachtext soll nicht unnötig schwer zu lesen sein, die Sachzusammenhänge sind schwierig genug. Ein grobes Maß für die Lesbarkeit eines Textes ist die mittlere Satzlänge. Erfahrungsgemäß sind Werte von zehn bis zwölf Wörtern pro Satz für deutsche Texte zu empfehlen. Wie kann eine Pipe aus Linux/UNIX-Werkzeugen diesen Wert ermitteln? Schauen wir uns das Vorwort an. Als erstes wären die \LaTeX -Konstrukte herauszuwerfen. Hierfür gibt es ein Programm `detex` bzw. `delatex`, allerdings nicht

bei Debian, sondern als Tarball bei <http://www.cs.purdue.edu/homes/trinkle/detex/>. Leider brach die Übersetzung mit zwei Parserfehlern ab, die nicht auf die Schnelle behoben werden konnten. Auch ein Werkzeug `untex` soll es geben. Wir lassen die Konstrukte drin. Dann sind Zeilenwechsel durch Leerzeichen zu ersetzen sowie Ziffern und einige Satzzeichen zu entfernen – Werkzeug `tr`. Schließlich muss jeder Satz in einer eigenen Zeile stehen. Wir ersetzen dazu alle Punkte durch Line Feeds. Ein Fehler entsteht dadurch, dass Punkte nicht nur ein Satzende markieren, aber bei einem durchschnittlichen Text ist dieser Fehler gering. Schicken wir den so aufbereiteten Text durch das Werkzeug `wc`, so erhalten wir die Anzahl der Zeilen gleich Anzahl der Sätze, die Anzahl der Wörter (wobei ein Wort eine maximale Zeichenkette begrenzt durch Leerzeichen, Tabs oder Line Feeds ist) und die Anzahl der Zeichen im Text. Das Werkzeug `tee` ist nicht notwendig und schreibt nur den veränderten Text zur Kontrolle in eine temporäre Datei. Die Pipe sieht so aus:

```
cat $1 | tr '\012' '\040' | tr -d '[0-9],;\"() ' |
tr '.' '\012' | tee $1.tmp | wc
```

Quelle 5.5. Shellscript zur Stilanalyse

Die Anzahl der Wörter geteilt durch die Anzahl der Sätze liefert die mittlere Satzlänge. Die Anzahl der Zeichen durch die Anzahl der Wörter ergibt die mittlere Wortlänge, infolge der Leerzeichen am Wortende erhöht um 1. Auch das ist ein Stilmerkmal. Die Ergebnisse für das Vorwort sind 45 Sätze, 712 Wörter und 5197 Zeichen, also eine mittlere Satzlänge von 15,8 Wörtern und eine mittlere Wortlänge ohne Leerzeichen von 6,3 Zeichen pro Wort. Zählt man von Hand nach, kommt man auf 36 Sätze; 9 Punkte markieren nicht ein Satzende. Man müsste das Satzende genauer definieren. Die nächste Verbesserung des Verfahrens wäre, nicht nur die Mittelwerte, sondern auch die Streuungen zu bestimmen. Hierzu wäre der `awk` zu bemühen oder gleich ein C-Programm zu schreiben. Das Programm liefert nur Zahlen; ihre Bedeutung erhalten sie, indem man sie zu Erfahrungswerten in Beziehung setzt. Soweit sich Stil durch Zahlen kennzeichnen lässt, hilft der Rechner; wenn das Verständnis von Wörtern, Sätzen oder noch höheren Einheiten verlangt wird, ist heutige Software überfordert.

Es soll ein Linux/UNIX-Kommando `style` geben, das den Stil eines englischen Textes untersucht und Verbesserungen vorschlägt. Dagegen ist das Kommando `diplom`, das nach Eingabe eines Themas und einer Seitenanzahl eine Diplomarbeit schreibt – mit `ispell` und `style` geprüft – noch nicht ausgereift.

5.14 Verschlüsselung

5.14.1 Aufgaben der Verschlüsselung

Auf einem Linux/UNIX-System kann der Verwalter auf jede Datei zugreifen, auf MS Windows mit Einschränkungen auch. Das Netz ist mit einfachen Mitteln un-

auffällig abzuhören. Will man seine Daten vor Unbefugten schützen, hilft nur Verschlüsseln. Man darf aber nicht vergessen, dass bereits die Analyse des Datenverkehrs einer Quelle oder eines Ziels Informationen liefert. Wer ganz unbemerkt bleiben will, muss sich mehr einfallen lassen als nur eine Verschlüsselung.

Eng verwandt mit der Verschlüsselung (E: encryption, F: cryptage, chiffrement) ist die Authentifizierung oder Authentisierung (E: authentication, F: authentification). Diese Aufgabe behandeln wir im Abschnitt 11.3 *Electronic Mail* auf Seite 365, weil sie dort eine Rolle spielt. Hier geht es nur darum, einen Text oder auch andere Daten für Unbefugte unbrauchbar zu machen; für Befugte sollen sie natürlich brauchbar bleiben.

Das Ganze ist heute eine Wissenschaft und heißt Kryptologie. In den letzten Jahrzehnten hat sie einen stark mathematischen Einschlag bekommen. Sie bietet einen gewissen Unterhaltungswert, insbesondere die Kryptanalyse, der Versuch, Verschlüsselungen zu knacken.

Die zu verschlüsselnden Daten nennen wir Klartext (E: plain text, F: texte en clair), die verschlüsselten Daten Geheimtext (E: cipher text, F: cryptogramme).

5.14.2 Symmetrische Verfahren

Im einfachsten Fall wird jedes Zeichen des Klartextes nach einer Regel durch ein anderes Zeichen desselben Alphabets ersetzt. Die einfachste Regel ist die Verschiebung um eine feste Anzahl von Stellen im Alphabet, beispielsweise um +3 Stellen. Aus A (Zeichen Nr. 1) wird D (Zeichen Nr. 1 + 3). Dieses Verfahren soll CAIUS JULIUS CAESAR benutzt haben. Er vertraute auf die Dummheit seiner Gegner. Zum Entschlüsseln des Geheimtextes nimmt man dasselbe Verfahren mit −3 Stellen. Wählt man eine Verschiebung um 13 Stellen, so führt bei einem Alphabet mit 26 Zeichen eine Wiederholung der Verschlüsselung zum Klartext zurück. Dieses Verfahren ist unter dem Namen ROT13 bekannt und wird im Netz verwendet, um einen Text – beispielsweise die Auflösung eines Rätsels – zu verfremden. ROT26 gilt als die schwächste aller Verschlüsselungen des lateinischen Alphabets. Man kann die Verfahren raffinierter gestalten, indem man Zeichengruppen verschlüsselt, Blindzeichen unter den Geheimtext mischt, die Algorithmen wechselt usw.

Seit 1970 unterscheidet man zwei Gruppen von Verschlüsselungsverfahren:

- Symmetrische Verfahren (Private-Key-V.),
- Asymmetrische Verfahren (Public-Key-V.).

Dazu kommen für bestimmte Aufgaben noch Einweg-Hash-Verfahren. Symmetrische Verfahren arbeiten schneller als asymmetrische. Bei den symmetrischen Verfahren kennen Sender und Empfänger neben dem Algorithmus sowohl den Chiffrier- wie den Dechiffrierschlüssel. Beide Schlüssel sind identisch oder voneinander ableitbar. Da der Algorithmus kaum geheim zu halten ist, beruht die Sicherheit auf dem Schlüssel, der nicht zu simpel sein darf und geheim bleiben muss. Das Problem liegt darin, den Schlüssel vom Sender zum Empfänger zu schaffen. Das geht nur über einen vertrauenswürdigen Kanal, also nicht über Email. Treffen Sie Ihren Brieffreund gelegentlich bei Kaffee und Kuchen, können Sie ihm einen Zettel mit

dem Schlüssel zustecken. Wohnen Sie in Karlsruhe, Ihr Briefpartner in Fatmomakke, wird der Schlüsselaustausch aufwendiger. Ein weiteres Problem liegt in der Anzahl der benötigten Schlüssel beim Datenverkehr unter mehreren Beteiligten. Geht es nur darum, die eigenen Daten vor dem Superuser zu verbergen, ist kein Schlüsselaustausch nötig und daher ein symmetrisches Verfahren angebracht.

Die Verschlüsselung nach dem weit verbreiteten Data Encryption Standard (DES) gehört in diese Gruppe, zur Ver- und Entschlüsselung wird derselbe Schlüssel benutzt. DES wurde von IBM entwickelt und 1977 von der US-Regierung als Standard angenommen. Es gilt heute nicht mehr als sicher, Triple-DES ist besser. Weitere Mitglieder dieser Gruppe sind IDEA, Blowfish und CAST5.

Unter Linux/UNIX stehen ein Kommando `crypt` sowie eine gleichnamige C-Standardfunktion zur Verfügung, die ein nicht sehr ausgefeiltes symmetrisches Verfahren verwenden (DES oder MD5). Die Passwörter in `/etc/passwd` werden damit verschlüsselt. Jünger und vielseitiger ist `mcrypt` (<http://mcrypt.sourceforge.net/>). Man ver- und entschlüsselt mittels des Kommandos:

```
joe@debian:~$ crypt < eingabe > ausgabe
```

Das Kommando fragt nach einem Schlüssel. Dieser wird für beide Richtungen eingesetzt. Der Klartext ist erforderlichenfalls gesondert zu löschen (physikalisch, nicht nur logisch, das heißt zu überschreiben). Die Crypt Breaker's Workbench enthält alles Nötige, um diese Verschlüsselung zu knacken (<http://axion.physics.ubc.ca/cbw.html>).

5.14.3 Asymmetrische Verfahren

Die asymmetrischen Verfahren verwenden zum Verschlüsseln und Entschlüsseln zwei völlig verschiedene, nicht voneinander ableitbare Schlüssel. Benutzer A hat sich ein Paar zusammengehöriger Schlüssel gebastelt, den ersten zum Verschlüsseln, den zweiten zum Entschlüsseln, wie, werden wir noch sehen. Den ersten Schlüssel gibt er öffentlich bekannt, daher Public Key. Jeder kann ihn benutzen, zum Beispiel Benutzer B, der A eine vertrauliche Mail schicken möchte. Was einmal damit verschlüsselt ist, lässt sich nur noch mit dem zweiten Schlüssel entschlüsseln, und den hält Benutzer A geheim. Er teilt ihn niemandem mit, daher Private Key.

Jetzt kann es nur noch passieren, dass ein Benutzer C unter Missbrauch des Namens von B an A eine beleidigende Mail schickt und B darauf hin mit A Krach bekommt. Veröffentlicht A den Dechiffrierschlüssel und behält den Chiffrierschlüssel für sich, kann er chiffrierte Texte versenden, die jeder entschlüsseln und lesen kann, wobei die Texte nur von A chiffriert sein können. Das ist die Lösung des Authentifizierungs-Problems, auf das wir im Abschnitt 11.3 *Email, Listen* auf Seite 365 eingehen.

Wie kommt man nun zu einem derartigen Schlüsselpaar? Ein Weg beruht auf der Tatsache, dass man leicht zwei ganze Zahlen großer Länge miteinander multiplizieren kann, sogar ohne Rechner, während die Zerlegung einer großen Zahl (um die vierhundert dezimale Stellen entsprechend etwa 1000 Bits) in ihre Primfaktoren mit den heute bekannten Algorithmen und Rechnern aufwendig ist, jedenfalls

wenn gewisse Voraussetzungen eingehalten werden. RON RIVEST, ADI SHAMIR und LEONARD ADLEMAN haben auf diesem Gedanken aufbauend das verbreitete RSA-Verfahren entwickelt.

Man wähle zufällig zwei große Primzahlen p und q , zweckmäßig von annähernd gleicher Länge. Ihr Produkt sei $n = pq$. Weiter wähle man eine Zahl e so, dass e und $(p-1)(q-1)$ teilerfremd (relativ prim) zueinander sind. Eine vierte Zahl d berechne man aus:

$$d = e^{-1} \bmod ((p-1)(q-1)) \quad (5.1)$$

Die Zahlen e und n bilden den öffentlichen Schlüssel, die Zahl d ist der private, geheime Schlüssel. Die beiden Primzahlen p und q werden nicht weiter benötigt, müssen aber geheim bleiben (löschen).

Wir sehen den Klartext K als eine Folge von Ziffern an. Er wird in Blöcke K_i kleiner n aufgeteilt. Die Geheimnachricht G besteht aus Blöcken G_i , die sich nach

$$G_i = K_i^e \bmod n \quad (5.2)$$

berechnen. Zur Entschlüsselung berechnet man

$$K_i = G_i^d \bmod n \quad (5.3)$$

Einzelheiten und Begründung hierzu siehe die Bücher von FRIEDRICH L. BAUER oder BRUCE SCHNEIER, ersterer ein deutscher Computerpionier, letzterer ein bekannter Fachmann für Computer-Sicherheit. Nun ein Beispiel aus einem Buch von F. L. BAUER. Wir wählen einen Text aus lateinischen Buchstaben samt Zwischenraum und ersetzen die Zeichen durch die Nummern von 00 bis 26. Er bekommt folgendes Aussehen:

$$K = 051818011805000821 \dots \quad (5.4)$$

Wir wählen:

$$p = 47 \quad q = 59 \quad n = p * q = 2773 \quad (5.5)$$

und teilen den Klartext in vierziffrige Blöcke kleiner n auf:

$$K_1 = 0518 \quad K_2 = 1801 \quad K_3 = 1805 \dots \quad (5.6)$$

Zur Bestimmung von e berechnen wir:

$$(p-1)(q-1) = 46 * 58 = 2668 \quad (5.7)$$

Die Zahl 2668 hat die Teiler 2, 4, 23, 29, 46, 58, 92, 116, 667 und 1334. Für e wählen wir 17, teilerfremd zu 2668. Dann ergibt sich d zu:

$$d = 17^{-1} \bmod 2668 \quad (5.8)$$

Diese vielleicht unbekannte Schreibweise ist gleichbedeutend damit, ein Paar ganzer Zahlen d, x so zu bestimmen, dass die Gleichung:

$$d * 17 = 2668 * x + 1 \quad (5.9)$$

erfüllt ist. Die Zahl $d = 157$ ist eine Lösung mit $x = 1$. Gezielt ermittelt man Lösungen mittels des Erweiterten Euklidischen Algorithmus. Nun haben wir mit n , e und d alles, was wir brauchen und gehen ans Verschlüsseln:

$$G_1 = K_1^e \bmod n = 0518^{17} \bmod 2773 = 1787 \quad (5.10)$$

und entsprechend für die weiteren Blöcke. Gleiche Klartextblöcke ergeben gleiche Geheimtextblöcke, was bereits eine Schwäche ist. Zum Entschlüsseln berechnet man:

$$K_1 = G_1^d \bmod n = 1787^{157} \bmod 2773 = 518 \quad (5.11)$$

und so weiter. Die Arithmetik großer Ganzzahlen ist für Rechner kein Problem. Man kann sie sogar in Silizium gießen und erhält schnelle Chips zum Ver- und Entschlüsseln, ohne Software bemühen zu müssen. Da n und e öffentlich sind, könnte man durch Zerlegen von n in seine Primfaktoren leicht den privaten Schlüssel d ermitteln, aber das Zerlegen großer Zahlen ist nach heutigem Wissensstand aufwendig.

Es gibt weitere asymmetrische Verfahren wie das von TAHER ELGAMAL. Wird das Dokument symmetrisch verschlüsselt und der dazu erforderliche Schlüssel asymmetrisch verschlüsselt mitgeteilt, spricht man von hybriden Verfahren. Auf <http://www.rsa.com/> findet sich Material zur Vertiefung des Themas. Eine zehnteilige FAQ-Sammlung zur Kryptografie liegt im Netz.

5.14.4 Angriffe (Kryptanalyse)

Angriffe auf verschlüsselte Daten – wissenschaftlich als Kryptanalyse, sonst als Cracking bezeichnet – gehen möglichst von irgendwelchen bekannten oder vermuteten Zusammenhängen aus. Das kleinste Zipfelchen an Vorkenntnissen kann entscheidend sein⁷. Die Wahrscheinlichkeit, dass ein Benutzer seinen nur gering modifizierten Benutzernamen als Passwort verwendet, ist leider hoch. Damit fängt man an. Das Ausprobieren aller nur möglichen Schlüssel wird *Brute Force Attack* genannt und ist bei kurzen Schlüsseln dank Rechnerhilfe auch schnell von Erfolg gekrönt. Das Faktorisieren kleiner Zahlen ist ebenfalls kein Problem. Aber selbst bei großen Zahlen, die für einen einzelnen Rechner – auch wenn er zu den schnellsten gehört – eine praktisch unlösbare Aufgabe darstellen, kommt man in kurzer Zeit zum Ziel, wenn man die Leerlaufzeiten von einigen Tausend durchschnittlichen Rechnern für seinen Zweck einsetzen kann. Das ist ein organisatorisches Problem, kein mathematisches, und bereits gelöst, siehe <http://www.distributed.net/rc5/>. Das ganze Nachdenken über sichere Verschlüsselung erübrigt sich bei schlampigem Umgang mit Daten und Schlüsseln. Der Benutzer ist erfahrungsgemäß das größte Risiko.

5.14.5 Einrichten des GNU Privacy Guards (GnuPG)

Um mit dem Softwarepaket GNU Privacy Guard zu arbeiten, braucht man die Debian-Pakete *gnupg* und *gnupg-doc* (Verwalter). Unter `/usr/share/doc/`

⁷Beim Knacken von Enigma spielte eine Rolle, dass der Gegner wusste, ein Buchstabe wurde niemals durch sich selbst verschlüsselt.

gnupg/ und /usr/share/doc/gnupg-doc/ findet sich genügend Lesestoff für ein Wochenende, insbesondere das *GNU Privacy Handbook* in mehreren Sprachen – auch Deutsch – und Formaten, rund 60 Seiten. Um auszuprobieren, ob der Guard eingerichtet ist, ruft man das Manual mit `man gnupg` auf.

Als erstes erzeugen wir uns ein Schlüsselpaar mittels der Eingabe:

```
joe@debian:~$ gpg --gen-key
```

wobei wir die Fragen mit den Defaultwerten beantworten. Gegen Ende der Generierung wird nach dem bürgerlichen Namen des Benutzers gefragt, worauf etwas wie:

```
"Wulf Alex"
```

mit Anführungszeichen anzugeben ist. Dann will das Programm noch die Emailanschrift wissen:

```
alex-weingarten@t-online.de
```

und einen nicht zu langen Kommentar, beispielsweise ob es sich um die privaten oder dienstlichen Schlüssel handelt. Abschließend ist eine Passphrase – ein langes Passwort, etwas kürzer als ein Roman – einzugeben, die man sich unter Ausnutzung aller ASCII-Zeichenarten einfallen lässt, aufschreibt und an einem sicheren Ort verwahrt. Wir brauchen die Passphrase später unter anderem beim Verschlüsseln von Mails. Gelegentlich wird eine Passphrase auch Mantra genannt. Bei Erfolg erscheint eine Meldung, die man sich anschaut, aber nicht merkt. Im Verzeichnis `$HOME/.gnupg` ist das Ergebnis gespeichert, von dem man möglichst bald eine Sicherungskopie (Backup) zieht. Zugriffsrechte 700 für das Verzeichnis, 600 für die Dateien. Ein Verlust des privaten Schlüssels in `secring.gpg` führt dazu, dass man die Sicherheitsmechanismen nicht mehr benutzen kann, beispielsweise verschlüsselte Mails nicht mehr entziffern kann. Gelangt er in falsche Hände, stehen dem Missbrauch Tür und Tor offen. Den privaten Schlüssel und das unten erläuterte Widerrufszeug sollte man auf CD und auf Papier speichern und sicher aufbewahren. Nun können wir ein bisschen herumspielen:

```
joe@debian:~$ gpg --list-sigs
```

```
joe@debian:~$ gpg --list-secret-keys
```

```
joe@debian:~$ gpg --fingerprint
```

```
joe@debian:~$ gpg --edit-key "Wulf Alex"
```

```
joe@debian:~$ gpg --edit-key alex
```

Aus letzterem Kommando kommt man mit `<q>` (quit) heraus. Zur Weitergabe des öffentlichen Schlüssels ist dieser zu exportieren:

```
joe@debian:~$ gpg --export --armor
```

Den ausgegebenen Zeichensalat darf man jedermann anbieten, beispielsweise indem man ihn auf der eigenen Webseite veröffentlicht.

Umgekehrt brauchen wir die öffentlichen Schlüssel unserer Freunde, die samt unserem eigenen in der Datei `$HOME/.gnupg/pubring.gpg` (Schlüsselbund, E: key ring) gesammelt werden. Angenommen, wir hätten eine Datei namens `key.gpg` mit einem fremden öffentlichen Schlüssel bekommen, dann ist diese zu importieren:

```
joe@debian:~$ gpg --import key.gpg
```

Öffentliche Schlüssel lassen sich auch von einem Keyserver im Netz wie `http://wwwkeys.de.pgp.net/` herunterladen. Gibt man dort als Suchstring `peterh.ganten` ein, erhält man eine mehrzeilige Antwort. Die Zeilen mit anderen Zeitgenossen beglaubigen den voranstehenden Schlüssel und interessieren uns nicht im Einzelnen. Durch Anklicken eines Schlüssel-Identifiers bringt man den Schlüssel auf den Schirm, wo er uns nicht viel nützt. Mit dem Schlüssel-Identifier und dem Kommando:

```
joe@debian:~$ gpg --keyserver wwwkeys.de.pgp.net
--recv-keys identifier
```

importiert man den Schlüssel in die Datei `pubring.gpg`, den öffentlichen Schlüsselbund.

Will man seinen eigenen öffentlichen Schlüssel auf den genannten Keyserver hochladen, so erzeugt man vorsichtshalber noch ein Widerrufs-zertifikat:

```
joe@debian:~$ gpg --gen-revoke "Wulf Alex"> revoke
```

und fügt in der Datei `gpg.conf` gegen Ende folgende Zeile hinzu:

```
keyserver x-hkp://pgpkeys.pca.dfn.de
```

Dann verschickt man den Schlüssel mittels:

```
joe@debian:~$ gpg --send-keys
```

am einfachsten, bevor man fremde öffentliche Schlüssel heruntergeladen hat. Andernfalls ist der eigene Name in Anführungszeichen als Argument mitzugeben. Anschließend kann man den eigenen Schlüssel wie oben beschrieben suchen. Jetzt sind nur noch die Anwendungen von den neuen Möglichkeiten zu benachrichtigen. Ein Beispiel findet sich in Abschnitt 11.3 *Email, Listen* auf Seite 365.

5.15 Organisation eines Textprojektes

5.15.1 Verzeichnisstruktur

Einfache Texte wie Briefe oder Übungsblätter erfordern keine besondere Verzeichnisstruktur. Ihre Anzahl kann eine flache Verzeichnishierarchie zweckmäßig erscheinen lassen, mehr nicht. Ein Textprojekt (Diplomarbeit, Forschungsbericht, Buch)

hingegen ist nur mit einer durchdachten Verzeichnisstruktur im Griff zu behalten, zumal wenn mehrere Autoren beteiligt sind oder mehrere Auflagen zu erwarten sind. Damit Sie nicht unsere Fehler wiederholen, schildern wir ausführlich, wie wir ein Textprojekt strukturieren. Sie können von unserem Vorschlag abweichen, aber dann sollten Sie wissen, warum Sie es anders machen wollen. Der Aufbau einer solchen Struktur kostet am Anfang Zeit – etwa einen Arbeitstag – aber eine ordentliche Struktur spart insgesamt Zeit und hilft, Fehler zu vermeiden. Auch die Fehlersuche vereinfacht sich, wenn Text und Zutaten in handliche Portionen aufgeteilt sind. Wir nehmen an, dass mit \LaTeX gearbeitet wird, unter anderen Formatierern oder Textprozessoren bleiben die Überlegungen gleich.

Für das Textprojekt ist ein eigenes Verzeichnis anzulegen, eventuell sogar in Form eines Home-Verzeichnisses eines fiktiven Benutzers, unter dem die tatsächlichen Autoren zusammengefasst sind. In dieses Verzeichnis – nennen wir es `projekt` – wird für jedes Kapitel ein Unterverzeichnis angelegt, unabhängig von der Länge der Kapitel. Die Unterverzeichnisse erhalten kurze (4 bis 8 Zeichen), aussagekräftige Namen. Man kann die Verzeichnisse auch mit `kap1` bis `kapn` bezeichnen und aussagekräftige Namen dazu per Symlink erzeugen. Zu den Kapitelverzeichnissen kommen einige weitere Verzeichnisse hinzu:

- ein Verzeichnis namens `Haupt` oder `Main`, in dem sich übergeordnete Dateien finden,
- ein Verzeichnis `bilder` für Abbildungen,
- ein Verzeichnis `sonstig` für Vorwort und ähnliches Drumherum,
- bei Bedarf ein Verzeichnis `anhang`,
- ein Verzeichnis `Mails` für das Projekt betreffende Mails,
- ein Verzeichnis `tmp` für temporäre Daten.

Das Bilder-Verzeichnis könnte man auch jeweils als Unterverzeichnis der Kapitelverzeichnisse anlegen, aber bei nicht zu zahlreichen Abbildungen erscheint obiger Vorschlag zweckmäßiger. Man hat es beim Bearbeiten der Bilddateien leichter.

In jedem Kapitelverzeichnis wird eine Kapiteldatei angelegt sowie für jeden Abschnitt (section) eine Abschnittsdatei, Namensgebung wie gehabt. Dazu kommt bei Bedarf jeweils ein Unterverzeichnis `problem` für das Problem-Management (Fehler, TODO-Listen etc.). Eine Abschnittsdatei könnte die Pfadbezeichnung `projekt/kap2/sec3.tex` tragen, wenn wir die phantasielose Namensgebung wählen. In diesem Fall ließe sich sogar die Struktur auf Vorrat halten und für jedes neue Projekt kopieren. Die Nummern sind Schall und Rauch; sie legen nicht die endgültige Reihenfolge fest.

Das Verzeichnis `Haupt` (besser wäre `0haupt`, damit das Verzeichnis beim Auflisten mittels `ls` auf jeden Fall an vorderster Stelle steht) enthält in erster Linie:

- die Hauptdatei des Projektes wie `projekt.tex`,
- eine Testdatei `test.tex`, ähnlich wie die Hauptdatei, zum Testen einzelner Kapitel,
- ein Makefile für das Projekt,

- eine \LaTeX -style-Datei für das Projekt (`projekt.sty`),
- weitere \LaTeX -style-Dateien, die eigens für das Projekt beschafft wurden und nicht in den allgemeinen \LaTeX -Verzeichnissen enthalten sind,
- eine Datei mit schwierig zu trennenden Wörtern (`trenn.tex`),
- bei Bedarf eine Datei mit Synonyma für den Index,
- für jedes Kapitelverzeichnis einen Symlink der Art `kap3 -> ../kap3`,
- nach einem `make`-Lauf einige temporäre \LaTeX -Hilfsdateien sowie das Ergebnis (`projekt.ps` und/oder `projekt.pdf`).

Die Symlinks sind erforderlich, weil die Kapiteldateien in der Hauptdatei mit Bezug auf die Hauptdatei eingebunden werden. Eine andere Lösung wäre, die Kapitelverzeichnisse als Unterverzeichnisse auf derselben Ebene wie die Hauptdatei anzulegen. Wir halten unseren Vorschlag für übersichtlicher.

5.15.2 Unentbehrlich: `make`

Werkzeuge wie `make` werden als Builder bezeichnet und fassen die Kommandos zusammen, die aus den Teilen das Endergebnis erzeugen. Hauptanwendung ist das Übersetzen (Compilieren) von Programmquellen, aber auch bei Textprojekten ist `make` beinahe unerlässlich. Einige Aufgaben könnte man auch mit Shell- oder Perl-Skripten bewältigen, aber `make` ist besser an die Aufgabe angepasst und in seinen Anfängen einfacher zu erlernen. Die Anweisungen für `make` stehen in einer Datei namens `Makefile` im Verzeichnis `Haupt`.

Sehen wir uns das `Makefile` an, das zur Erzeugung der Vorlage zum Buch verwendet wurde, etwas vereinfacht (PostScript-Versionen weggelassen, nur ein Index):

```
# Makefile fuer Debian-Buch
# W. Alex, 2004-10-05

##### Makros, je nach System

SHELL=/bin/sh
MAKE=/usr/bin/make
LATEX=/usr/bin/pdflatex
MAKEINDEX=/usr/bin/makeindex -c -g -l
CP=/bin/cp
GREP=/bin/grep
RM=/bin/rm -f
WWW=/var/www/debian/

##### Targets

all : test buch www
test : test.pdf
buch : buch.pdf
www : web

##### Testversion einzelner Kapitel erzeugen
```

```

# Fuenf LaTeX-Durchlaeufer erforderlich,
# damit Index im Inhaltsverzeichnis erscheint.

test.pdf :
    $(LATEX) test.tex
    $(LATEX) test.tex
    $(LATEX) test.tex
    $(MAKEINDEX) -s svind.ist -o test.ind test.idx
    $(LATEX) test.tex
    $(GREP) chapter test.toc > test.ovr
    $(LATEX) test.tex

##### Testversion aller Kapitel (Buch) erzeugen

buch.pdf :
    $(LATEX) buch.tex
    $(LATEX) buch.tex
    $(LATEX) buch.tex
    $(MAKEINDEX) -s svind.ist -o buch.ind buch.idx
    $(LATEX) buch.tex
    $(GREP) chapter buch.toc > buch.ovr
    $(LATEX) buch.tex

##### Ergebnis ins lokale Web stellen

web :
    $(CP) buch.pdf $(WWW)debian2.pdf

##### Aufräumen

xclean: clean cleanp

clean :
    $(RM) *.log
    $(RM) *.lof
    $(RM) *.idx

# usw., alle Hilfsdateien löschen

cleanp :
    $(RM) *.pdf

```

Quelle 5.6. Makefile zu vorliegendem Buch, etwas vereinfacht

Das Doppelkreuz leitet eine Kommentarzeile ein. Der erste Block enthält Makros, das sind Abkürzungen für Kommandos samt Optionen oder für Pfade, die nachfolgend verwendet und buchstäblich ersetzt werden. Das hat den Vorteil, dass man etwaige Änderungen nur am Anfang vorzunehmen braucht, wenn beispielsweise der Webspace woanders liegt. Das erste Ziel (target), hier und meist mit dem Namen *all*, wird erzeugt, wenn man *make* ohne Argument aufruft. Die Unterziele haben hier nur

wegen der Übersicht nochmals eigene Definitionen, man kann sie so auch leichter erweitern. Der dritte bzw. vierte Block erledigen die eigentliche Arbeit. Nach dem Ziel folgen die Kommandos, wobei jede Zeile mit einem TAB beginnen muss. Im fünften Block wird das Ergebnis `buch.pdf` in den lokalen Webpace kopiert. Der Rest sind Aufräumarbeiten, wobei das Ziel `clean` üblicherweise alle Hilfsdateien löscht, das Ziel `xclean` auch die Ergebnisse, sodass nur die Quelldateien übrig bleiben. Diese beiden Ziele werden nicht unter `all` genannt, da man vielleicht eine Hilfsdatei untersuchen will. Es ist gebräuchlich, das Löschen nur auf ausdrücklichen Wunsch durchzuführen. Ein solches Makefile ist schnell geschrieben, vor allem, wenn man schon eine Vorlage hat, und leicht zu ändern. Es erspart viel Mühe.

Mit Makefiles kann man auch andere Dinge treiben, beispielsweise die *Türme von Hanoi* spielen. Das Spiel ist auf Seite 424 erläutert. Hier ein Makefile, das einer Vorlage von AMIT SINGH (<http://www.kernelthread.com/hanoi/>) nachempfunden ist:

```
# Türme von Hanoi, Aufruf: make -f makehanoi N=4

ifndef $(N)
    N = 3
endif

ifndef $(INITIALIZED)
    MAKEFILE = makehanoi
    F = 1
    U = 2
    T = 3
    INITIALIZED = 1
endif

hanoi:

ifeq ($(N),1)
    @echo "$(F) -> $(T) "
else
    @$(MAKE) -s -f $(MAKEFILE) F=$(F) U=$(T) \
    T=$(U) N=$(shell expr $(N) - 1)
    @echo "$(F) -> $(T) "
    @$(MAKE) -s -f $(MAKEFILE) F=$(U) U=$(F) \
    T=$(T) N=$(shell expr $(N) - 1)
endif
```

Quelle 5.7. Makefile, das die *Türme von Hanoi* spielt

5.15.3 Versionsverwaltung (rcs, cvs, subversion, arch)

Revision Control System (RCS)

Wenn zu befürchten steht, dass sich ein Text (Manuskript, Programmquelle, Konfigurationsdatei, Webseite) über längere Zeit hindurch entwickelt, oder wenn mehrere

Autoren daran arbeiten, muss man Ordnung in dem Projekt halten. Außer der oben erwähnten Verzeichnisstruktur helfen dabei Versionsverwaltungen wie SCCS, RCS, CVS oder Subversion. Die vorgenannte Reihenfolge entspricht der historischen Entwicklung, der Leistungsfähigkeit und der Komplexität der Versionsverwaltungen. Am weitesten verbreitet ist gegenwärtig das Concurrent Versions System (CVS). Solche Systeme

- führen Buch über die Änderungen an den Dateien,
- ermöglichen, ältere Versionen wiederherzustellen, ohne dass diese vollständig gespeichert werden,
- verhindern oder harmonisieren gleichzeitige schreibende Zugriffe mehrerer Autoren auf dieselbe Datei.

Versionsverwaltungen können nicht:

- ein Projekt von sich aus sinnvoll gliedern (modellieren),
- Termine verfolgen,
- Fehler verfolgen (bug tracking),
- Dateien testen,
- die Kommunikation zwischen den Beteiligten ersetzen.

Sowie es um mehr als Texte oder Programme für den einmaligen Gebrauch geht, sollte man make und eine Versionsverwaltung einsetzen. Der anfängliche Mehraufwand – der sich bei kleinen Projekten in bescheidenen Grenzen hält – macht sich langfristig bezahlt. Geht bei einem Textprojekt die Fehlerverwaltung über das einfache Korrigieren hinaus – was bei Software die Regel ist – ist der Einsatz eines Bug Tracking Systems wie GNATS zu erwägen (<http://www.gnu.org/software/gnats/>). Im Umfeld von Webservern trifft man häufig auf den Begriff *Distributed Authoring and Versioning* (DAV, speziell WebDAV), der eine Erweiterung einer einfachen Versionskontrolle von Webdokumenten bezeichnet.

Die klassische Versionsverwaltung unter UNIX ist das Source Code Control System (SCCS), das aber durch das von WALTER F. TICHY entwickelte Revision Control System (RCS) so gut wie überall abgelöst worden ist. Das RCS aus dem GNU-Projekt steht in einem gleichnamigen Debian-Paket zur Verfügung, dazu gibt es noch ein Project Revision Control System (PRCS) als Frontend für ein Bündel von Werkzeugen zur Projektverwaltung und ab *sarge* ein Advanced Revision Control System im Paket *darcs*. Informationen finden sich bei <http://www.gnu.org/software/rcs/rcs.html> und <http://www.cs.purdue.edu/homes/Trinkle/RCS/>. Das *CVS-RCS HOWTO* ist gegenwärtig zwecks Bearbeitung zurückgezogen.

Nehmen wir an, Sie hätten in einem der Unterverzeichnisse eine Textdatei `datei.tex` mittels eines Editors erzeugt und gefüllt. Dann sind die nächsten Schritte:

- Mit dem Kommando `ci datei.tex` (check in) stellen Sie die Datei in das RCS ein. Dieses ergänzt die Datei um Versionsinformationen und macht eine nur lesbare (444) RCS-Datei namens `datei.tex,v` daraus. Die Kennung ist

durch ein Komma abgetrennt, ungewöhnlich. Die ursprüngliche Datei löschen Sie. Ab jetzt gehört die Datei dem RCS.

- Mit dem Kommando `co datei.tex` (check out, Dateiname ohne `v`) erzeugen Sie eine nur lesbare Kopie der RCS-Datei. Diese Kopie kann man mit allen Linux/UNIX-Werkzeugen betrachten oder bearbeiten, nur das Zurückstellen in RCS mittels `ci` verweigert das System.
- Mit dem Kommando `co -l datei.tex` – die Option bedeutet *lock* – holen Sie sich eine les- und schreibbare Kopie der RCS-Datei in das Arbeitsverzeichnis. Dabei wird die RCS-Datei für weitere Zugriffe dieser Art gesperrt, bis Sie Ihre Kopie wieder zurückgestellt haben. So wird verhindert, dass zwei Benutzer gleichzeitig die Datei verändern. Sie erinnern sich; Der Letzte gewinnt ansonsten. Ihre Kopie können Sie mit allen Linux/UNIX-Werkzeugen bearbeiten, nur umbenennen wäre ein schlechter Einfall.
- Beim Zurückstellen mittels `ci datei.tex` haben sie Gelegenheit, einen kurzen Kommentar in die Versionsinformationen zu schreiben, beispielsweise Art und Grund der Änderung.

Falls Sie sich mit `co -l datei.tex` eine Kopie zum Editieren geholt und damit gleichzeitig das Original für weitere Zugriffe zwecks Schreibens gesperrt haben, anschließend ihre Kopie mittels `rm` löschen, so haben Sie nichts mehr zum Zurückstellen. In diesem Fall lässt sich die Sperre mit `rcs -u datei.tex` aufheben. Auf die Linux/UNIX-Kommandos zur Dateiverwaltung sollte man verzichten und nur mit den RCS-Kommandos arbeiten. Der Einsatz von RCS verlangt von den Beteiligten Disziplin.

RCS speichert die jüngste Version vollständig und zu den vorangehenden Versionen nur die Unterschiede (Differenzen, Deltas). Auf diese Weise wird Speicherplatz gespart, ohne auf das Zurückgehen zu älteren Versionen verzichten zu müssen.

In den Textdateien kann man einige RCS-Variable unterbringen, beispielsweise im Kommentar, aber auch in der Definition von Zeichenketten-Konstanten. Die wichtigsten sind:

- `Id` Diesen Bezeichner ergänzt RCS um den Namen der RCS-Datei, die Versionsnummer, das Datum des Zurückschreibens, den Autor, einen Status (z. B. experimentell) und den Namen des Benutzers, der die Datei gesperrt (geloct) hatte. So stehen auch im Text die wichtigsten RCS-Informationen.
- `$Header$` wie vorstehend, nur dass der volle Pfad der Datei genannt wird, meist überflüssig.
- `Log` der beim Zurückschreiben als Kommentar eingegebene Text. Auf diese Weise lässt sich die Entwicklung der Datei verfolgen.

Da sich Autoren meist um das Protokollieren dieser Informationen drücken, bietet das RCS eine bequeme, mit einem sanften Zwang verbundene Möglichkeit, die Daten sogar direkt im Text (Manuskript, Quelle) festzuhalten. Das Ergebnis sieht bei einem in RCS eingestellten Makefile so aus (abgeschnitten):

```

# Makefile fuer Skriptum
# $Id: text15.tex,v 1.11 2005/08/21 18:30:40 wulf Exp $
# $Log: text15.tex,v $
# Revision 1.11  2005/08/21 18:30:40  wulf
# Endfassung
#
# Revision 1.10  2005/08/01 10:16:12  wulf
# 1. Korrektur fertig
#
# Revision 1.9   2005/07/18 20:06:27  wulf
# Zwischenstand
#
# Revision 1.8   2005/07/08 15:33:26  wulf
# Kapitel text fertig
#
# Revision 1.7   2005/05/21 20:43:44  wulf
# Zwischenstand, Kapitel text fertig
#
# Revision 1.6   2005/05/08 11:54:33  wulf
# Zwischenstand
#
# Revision 1.5   2005/04/24 20:56:39  wulf
# *** empty log message ***
#
# Revision 1.4   2005/03/26 17:05:35  wulf
# Aufgeraemt.
#
# Revision 1.3   2005/03/26 15:08:27  wulf
# Sektion Lokalisierung rausgeworfen.

```

In \LaTeX -Dateien muss die Entwicklungsgeschichte nach `\end{document}` kommen oder mittels `iffalse - fi` eingerahmt werden, da RCS nicht das \LaTeX -Kommentarzeichen (Prozentzeichen) kennt.

Das System beherrscht Verzweigungen, Releases, eigene Zugriffsrechte und andere Dinge, deren Erläuterung zu weit führen würde. Der durchschnittliche Benutzer schöpft die Möglichkeiten von RCS ebenso wenig aus wie die von `make` oder die der großen Editoren.

Concurrent Versions System (CVS)

Das RCS hat aus heutiger Sicht Mängel, die sich bei größeren, im Netz verteilten Projekten bemerkbar machen. Deshalb wurde das Concurrent Versions System (CVS) entwickelt, das weit verbreitet ist und sich vom RCS vor allem in drei Punkten unterscheidet:

- Es kann mit Verzeichnissen umgehen, nicht nur mit Dateien,
- es ist nach dem Client-Server-Modell aufgebaut; die Clients können auf anderen Rechnern laufen als der Server, wobei eine Verbindung nur für das Hoch- oder Herunterladen der Daten benötigt wird, nicht für das Arbeiten,
- es erlaubt, gleichzeitig mehrere Arbeitskopien zum Schreiben auszuchecken. Konflikte werden beim Zurückschreiben (Einchecken) entdeckt und teils automatisch, teils mit Hilfe der Autoren gelöst.

Da CVS ursprünglich auf RCS aufbaute, sind die Kenntnisse, die man beim Arbeiten mit RCS erworben hat, nicht verloren. CVS ist im gleichnamigen Debian-Paket in der Abteilung *Development* enthalten, wo auch ergänzende Pakete liegen. Weiteres findet sich bei <http://www.cvshome.de/> oder [.org/](http://www.cvshome.org/), darunter eine hundertseitige, englische Anleitung von PER CEDERQVIST. Ein Linux CVS-RCS-HOWTO gibt es beim TLDP. Ein Frontend (Client) mit grafischer Benutzeroberfläche ist LinCVS. Einzelheiten unter <http://www.lincvs.org/> und <http://lincvs.sourceforge.net/>. Die Einrichtung des Tarballs mit den Binaries für die Qt-Lib 3.3 auf einem PC unter *sarge* verlief mühelos. Das Anlegen eines CVS-Projektes erfordert etwas Nachdenken und Zeit, das Arbeiten damit ist einfach⁸

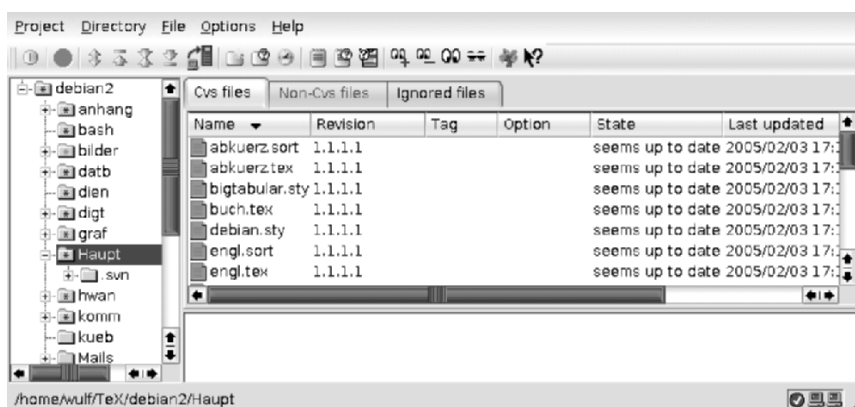


Abbildung 5.16. Screenshot der grafischen Oberfläche LinCVS für das Concurrent Versions System (CVS)

Vorausgesetzt, auf dem Server und den Arbeitsplätzen ist CVS eingerichtet, besteht der erste Schritt im Anlegen des Daten-Depots (repository) auf dem Server. Zu diesem Zeitpunkt sollte die Verzeichnisstruktur des Projektes grob festliegen, da Verschieben oder Umbenennen von Verzeichnissen oder Dateien zusätzlichen Aufwand bedeutet. Falls Sie mehrere Projekte unter CVS stellen wollen, ist meist ein Repository mit einem Verzeichnis je Projekt geschickter als ein eigenes Repository für jedes Projekt. Es kann vorkommen, dass ein Benutzer (Client) an mehreren, völlig getrennten CVS-Projekten auf verschiedenen Servern beteiligt ist. Das erfordert eine sorgfältige Konfiguration auf Seiten des Benutzers, insbesondere hinsichtlich der Authentifizierung. Den Anfang macht der Verwalter auf der Servermaschine:

```
debian:~# cvs -d /home/debian init
```

Damit wird ein Repository namens `/home/debian` angelegt; es könnte auch im `/var`-Zweig liegen. Dann richtet der Verwalter mittels des Kommandos `adduser`

⁸Das Debian-Buch von GANTEN + ALEX ist unter CVS entstanden. Der Server stand in Bremen, ein Client bei Karlsruhe.

einen Benutzer und eine Gruppe `debian` ein und übereignet ihnen mit dem Kommando `chown -R` das neue Verzeichnis samt Inhalt. Schließlich fügt er ebenfalls mittels `adduser` alle Benutzer des Repositories der Gruppe `debian` zu. Das Weitere sollte von einem der berechtigten Benutzer ausgehen.

Der begibt sich auf seinem Arbeitsplatz-Rechner in das Verzeichnis, welches das vorläufige Projekt beherbergt (die Unterverzeichnisse seines Projektes), räumt noch einmal auf und importiert die Verzeichnisse und etwaige Dateien in das CVS auf dem Server:

```
joe@debian:~$ cvs -d :ext:wulf@server:/home/debian
import debian2 Debian2 Start
```

Das CVS-Kommando importiert unter dem Namen des Benutzers `wulf` über das Netz (Zugriffsmethode `:ext:`) auf den Server `server` in das Wurzelverzeichnis `/home/debian` von CVS, neues Unterverzeichnis `debian2` alle Daten aus dem Arbeitsverzeichnis. Die beiden Argumente `Debian2` und `Start` spielen jetzt keine Rolle, müssen jedoch sein. Der Benutzer wird nach seinem Passwort auf dem Server und nach einem kurzen Kommentar gefragt. Dann beginnt die Übertragung der Daten. Anschließend findet man auf dem Server im Verzeichnis `/home/debian` neben dem mittels `init` angelegten Verzeichnis `CVSROOT` ein Verzeichnis `debian2` mit allen Daten.

Nun holen wir uns die Daten vom Server zurück auf den Arbeitsplatz. Dazu gehen wir ein Verzeichnis höher und benennen das Projektverzeichnis um (Löschen wäre verfrüht). Mittels des Kommandos:

```
joe@debian:~$ cvs -d :ext:wulf@server:/home/debian
checkout debian2
```

erzeugen wir lokal das eben in den Server importierte Verzeichnis `debian2` neu. Es sollte mit seiner umbenannten Vorlage übereinstimmen, zuzüglich eines Verwaltungsverzeichnisses `CVS` in jedem Verzeichnis und abzüglich etwaiger Symlinks, die erneut anzulegen sind. Ist das der Fall, löschen wir die umbenannte Vorlage. Wir arbeiten künftig nur noch mit der Arbeitskopie aus dem CVS-Repository auf dem Server. Das `checkout`-Kommando brauchen wir nur einmal.

Die Arbeit unter CVS beginnt mit dem Editieren einer beliebigen Datei. Sind wir der Meinung, dass sie reif zum Hochladen in das Repository ist, geben wir im Arbeitsverzeichnis, ohne Argument:

```
joe@debian:~$ cvs commit
```

(E: to commit = anvertrauen, einliefern). Die restlichen Daten ergänzt der lokale CVS-Client aus den Verwaltungsverzeichnissen. Rufen wir dann das Kommando:

```
joe@debian:~$ cvs status
```

auf, so erfahren wir den Zustand der Dateien aus unserem Arbeitsverzeichnis und sehen, dass die Revisionsnummer der eben editierten und hochgeladenen Datei um 1 hochgezählt worden ist. Ist unsere Arbeitskopie einer Datei älter als die Vorlage im Repository, lautet das Kommando:

```
joe@debian:~$ cvs update
```

Haben wir lokal eine Datei oder ein Verzeichnis neu angelegt und wollen sie in das CVS aufnehmen, sind zwei Kommandos erforderlich:

```
joe@debian:~$ cvs add dateiname
```

```
joe@debian:~$ cvs commit
```

Auf diese Weise lässt sich verhindern, dass lokal erzeugte pdf-Dateien oder L^AT_EX-Hilfsdateien den Weg in das Repository nehmen; wir addieren sie einfach nicht dazu. Ein anderer Weg mit Hilfe einer Datei `.cvsignore` wird unten beschrieben. Nach dem Löschen einer lokalen Datei lautet das Subkommando zum Entfernen aus dem Repository `remove`; es muss ebenfalls von einem `commit` begleitet werden. Die weiteren Möglichkeiten des CVS entnimmt man der Literatur aus dem Netz.

Die Versionsnummer einer Datei hängt davon ab, wie oft sie eingchecked wurde. Das führt dazu, dass die Nummern der Dateien auseinander laufen. Will man sich auf einen bestimmten Zustand des Projektes beziehen, müsste man für jede Datei die jeweilige Versionsnummer angeben. Diesem Missstand helfen symbolische Revisionen (Tags) ab. Man gibt – zweckmäßig im obersten Verzeichnis des Projektes – das Kommando:

```
joe@debian:~$ cvs tag Release-Feb-2005
```

Damit werden zu dem Namen der symbolischen Revision die jeweiligen Versionsnummern gespeichert. Man kann nun jederzeit auf den Zustand des gesamten Projektes zum Zeitpunkt der Vergabe des Tags zugreifen, beispielsweise um von dort eine Entwicklung abzuzweigen. Kommandos wie `checkout` kennen entsprechende Optionen. In ähnlicher Weise kann man sich auch auf einen bestimmten Zeitpunkt beziehen, ohne Tags angelegt zu haben.

Wer mit LinCVS arbeitet, findet ein Fenster wie in Abbildung 5.16 auf Seite 247 vor. In der obersten Zeile verbergen sich hinter den Worten Project, Directory, File, Options und Help Menüs mit den Kommandos, wie wir sie eben in der Kommandozeile eingegeben haben, und weiteren. Darunter folgt dasselbe mit Piktogrammen für die des Lesens Unkundigen. Links im Fenster wird die Verzeichnisstruktur dargestellt. Den Hauptteil des Fensters nimmt die Auflistung der Dateien im Arbeitsverzeichnis ein, eingeteilt in CVS-Dateien mit Revisions-Nummer, Tag, Status etc., Nicht-CVS-Dateien und von CVS zu ignorierende Dateien. Das Ganze lässt sich konfigurieren und macht einen aufgeräumten Eindruck; die Sprache ist Englisch.

Ein Benutzer kann in seinem Home-Verzeichnis eine Datei `.cvsrc` mit CVS-Kommandos anlegen, die immer wieder mit denselben Optionen aufgerufen werden, und sich so etwas Arbeit sparen. Ebenso kann er in einer Datei `.cvsignore` in seinem Home- oder jedem Unterverzeichnis Dateinamen angeben, die von den CVS-Kommandos `import`, `update` und `release` ignoriert werden sollen. Die Dateinamen dürfen die Jokerzeichen der Shell enthalten. Bei einem L^AT_EX-Projekt empfiehlt es sich, die zahlreichen temporären Hilfsdateien wie `*.aux` in `.cvsignore` aufzunehmen, unter Umständen auch die Enderzeugnisse im pdf- oder ps-Format. Bei einem Software-Projekt wären Objektdateien Kandidaten für diese Liste.

Subversion

Die Versionsverwaltung Subversion (SVN) behebt einige Schwächen von CVS und ist in ihrer Benutzung sehr ähnlich, sodass ein Umstieg leicht fällt. Es gibt auch ein Werkzeug `cvs2svn` zum Konvertieren eines Repositories. Die Vorteile von Subversion gegenüber CVS sind (mangels eigener Erfahrung aus der deutschen Wikipedia entnommen):

- Umbenennen oder Verschieben von Dateien oder Verzeichnissen ist direkt möglich,
- Zugriffsrechte können ebenfalls verwaltet werden,
- die Behandlung binärer Dateien ist erleichtert,
- Commits sind wie in Datenbanken atomar, das heißt, sie werden entweder ganz oder gar nicht durchgeführt,
- Revisionsnummer werden nicht dateiweise, sondern immer für das ganze Projekt erhöht (ob das in der Praxis ein Vorteil ist, bleibt auszuprobieren),
- der Zugriff über das Netz ist schneller und flexibler.

Als Nachteile stehen dem ein höherer Massenspeicherbedarf und eine höhere Komplexität gegenüber. Letztere trifft jedoch nur den Verwalter, nicht die Benutzer. Auch ist SVN noch nicht so verbreitet wie CVS, aber jeder fängt einmal klein an. Einzelheiten in der Wikipedia und bei <http://subversion.tigris.org/>. Ein Online-Buch über Subversion liegt auf <http://svnbook.red-bean.com/>.

Eine grafische Oberfläche für Subversion unter Verwendung der Qt-Bibliothek bieten die Debian-Pakete *esvn* und *esvn-doc*, beheimatet auf <http://esvn.umputun.com/>.

Die Software *websvn* (<http://websvn.tigtis.org/>) ist in PHP geschrieben und ermöglicht den Zugriff auf Subversion-Repositories. Man kann alles lesen oder herunterladen, was im Repository liegt, es ist aber keine grafische Oberfläche für Subversion.

SVK ist eine dezentralisierte Versionsverwaltung, die auf Subversion oder CVS aufsetzt und in Perl geschrieben ist. Näheres bei <http://svk.elixus.org/>. Das Projekt entwickelt sich noch lebhaft.

Arch

Das Versions-Kontrollsystem Arch (zur Zeit nur in *oldstable* enthalten) soll ebenfalls CVS ersetzen. Es arbeitet mit verteilten Repositories. Näheres bei <http://wiki.gnuarch.org/>. Interessant sind auf den Webseiten die Vergleiche mit anderen Versions-Kontrollen. Selbst wenn man bei CVS bleiben will, erfährt man hier Einiges über die Mitbewerber. CVS hat zweifellos Schwächen, aber die besseren Systeme sind noch ziemlich jung. Einen Vergleich mehrerer Versionsverwaltungen findet man auch unter <http://better-scm.berlios.de/comparison/comparison.html>.

Mathematische Werkzeuge

Dieses Kapitel stellt Werkzeuge zum Rechnen und zum Lösen mathematischer Aufgaben vor.

6.1 Grundbegriffe

In der Rechnerwelt wird streng zwischen Ziffern (Zeichen) und Zahlen unterschieden. Eine 5 kann eine Ziffer oder eine Zahl sein, entsprechend 12345 eine Ziffernkette (Zeichenkette, String) oder eine Zahl. Es gibt nur wenige Programme, die von sich aus entscheiden können, ob 12345 als Zeichenkette oder als Zahl aufzufassen ist. Mit der Ziffern- oder Zeichenkette *12345* kann ich alles machen, was ich auch mit der Zeichenkette *Karlsruhe* anstellen kann, aber nicht rechnen. Mit der Zahl *12345* kann ich rechnen. Telefonnummern, Kontonummern oder Postleitzahlen sind Zeichenketten, keine Zahlen. Es ist sinnlos, zwei Telefonnummern zu addieren. Der Unterschied macht sich unter anderem beim Sortieren bemerkbar, siehe Abschnitt 5.13 *Kleinere Textwerkzeuge* auf Seite 229.

Bei den Zahlen wird zwischen ganzen Zahlen und reellen Zahlen endlicher Länge (Genauigkeit) unterschieden. Brüche (rationale Zahlen) oder komplexe Zahlen verstehen nur wenige Programme. Für alle Zahlen gibt es einen größten, vom System handhabbaren Wert. In der Datei `/usr/include/limits.h` steht, dass auf einem System mit einer Breite des Maschinenwortes von 32 Bit vorzeichenlose Ganzzahlen (`unsigned long int`):

```
n <= 4294967295
```

sind, das sind gut 4 Milliarden. Der kleinste Wert für diesen Zahlentyp ist 0. Braucht man ein Vorzeichen, sinkt die obere Grenze auf die Hälfte und entsprechend die untere Grenze zu negativen Zahlen:

```
- 2147483648 <= n <= 2147483647
```

Weitere Werte stehen in `/usr/include/math.h`. Letzten Endes sind das die Grenzen des GNU-C-Compilers, der sich an den Standard ISO C99 hält.

Größere Zahlen werden in Exponentialschreibweise als Mantisse und Exponent zur Basis 10 dargestellt, beispielsweise:

```
x = 1.234567 E +307
```

zu lesen als 1.234567 mal 10 hoch +307. Entsprechend gibt es eine kleinste Zahl, beispielsweise:

```
x = 1.234567 E -308
```

die noch als von 0 verschieden angesehen wird. Die Genauigkeit dieser Zahlen wird bestimmt durch die Anzahl der Stellen der Mantisse (der Zahl vor dem E).

Man kann Programme schreiben, die diese Grenzen überschreiten und nur durch die Größe des Arbeitsspeichers behindert werden. Um uns nicht in den Abgründen der Numerischen Mathematik zu verlieren, merken wir uns nur, dass alle Zahlen im Rechner nach Größe und Genauigkeit begrenzt sind und ein Rechenergebnis unter Umständen beliebig falsch sein kann, vor allem nach langen Rechnungen¹. Abgesehen davon sind unsere Eingangsdaten auch fragwürdig.

In manchen Zusammenhängen ist ein Datum ein eigenes Format, weder Zeichenkette noch Zahl, vor allem in Datenbanken. Beim Umgang mit Kalenderdaten über längere Zeiträume sind auch die verschiedenen Kalenderreformen zu bedenken.

Die Fähigkeiten der nachfolgend vorgestellten Werkzeuge überlappen sich. Ein wissenschaftlicher Taschenrechner verfügt über einige Funktionen der höheren Werkzeuge, mit einer Tabellenkalkulation kann ich statistische Auswertungen durchführen, die Numerik-Werkzeuge beherrschen zumindest die numerischen Funktionen von Computeralgebrasystemen. In der Regel sind die einfacheren Werkzeuge schneller. Der Benutzer sollte sich eine ungefähre Vorstellung davon aneignen, wofür die Werkzeuge gedacht sind und welche für eine vorliegende Aufgabe am besten passen.

Das *Scientific Computing with Free Software on GNU/Linux HOWTO*, aktuell von 2004, deckt ein breites Themenspektrum ab, darunter *Mathematics Packages*, *Numerical Methods and Libraries* und *Linux in the laboratory*, geht jedoch kaum über eine Aufzählung hinaus.

6.2 Kleine numerische Werkzeuge (seq, pi, gbase, factor)

Das Werkzeug `seq` (sequence) aus dem Debian-Paket *shellutils* (*oldstable*) bzw. *coreutils* (*sarge*), Abteilung *Base Utilities*, schreibt nach dem Aufruf:

```
joe@debian:~$ seq MAX
```

eine Folge ganzer Zahlen von 1 bis MAX nach `stdout`, eine Zahl pro Zeile.

Das Werkzeug `pi` berechnet nach dem Aufruf:

¹Näheres bei U. KULISCH, *Memorandum über Computer, Arithmetik und Numerik*, Karlsruhe 1996

```
joe@debian:~$ pi n
```

die Zahl π auf ungefähr n Stellen (Default 100) und schreibt sie nach `stdout`. Mit der Option `- bibliography` nennt das Werkzeug zwei Bücher aus neuerer Zeit zu dem Thema, eine Suche im Web dürfte ergiebiger sein. Zugrunde liegt die Class Library for Numbers `libcln`. Weiteres zur Zahl π erfährt man beispielsweise bei <http://wasi.org/PI/> oder der Wikipedia.

Das einfache Werkzeug `gbase` dient der Umwandlung von Dual-, Oktal-, Dezimal- und Hexadezimalzahlen ineinander. Aufgerufen zeigt es vier Fenster für die vier Zahlenarten. Gibt man eine Zahl in eines der Fenster ein, werden sofort die anderen Zahlen berechnet. Eine Eingabe von 1 Milliarde dezimal brachte das Werkzeug nicht ins Schwitzen, nur die Dualzahl verlor ihre niedrigsten Stellen am rechten Fensterrand. Fehlerhafte Eingaben – die Ziffer 9 im Oktalfenster – nimmt das Werkzeug nicht an. Klein, hübsch und praktisch.

Die bereits erwähnten Debian-Pakete `shellutils` bzw. `coreutils` bringen ein Werkzeug namens `factor` zum Zerlegen von Zahlen in ihre Primfaktoren (faktorisieren) mit. Eine 18stellige Dezimalzahl wurde unverzüglich zerlegt, bei einer 36stelligen gab es Protest. Zum Knacken von kryptografischen Schlüsseln eignet sich das Werkzeug nicht.

6.3 Diagramme (gnuplot)

Das bewährte Werkzeug `gnuplot` dient zum Zeichnen (Plotten) von Diagrammen nach Wertetabellen oder Formeln. Es ist bei Debian in Form einiger Pakete erhältlich, auf <http://www.gnuplot.info/> zu Hause und mit dem GNU-Projekt weder verwandt noch verschwägert. Das Werkzeug lässt sich aus der Kommandozeile aufrufen oder durch ein Skript steuern und kommt ohne X11 aus. Ein Grundkurs liegt auf <http://userpage.fu-berlin.de/~voelker/gnuplotkurs/gnuplotkurs.html>, eine kurze Anleitung unter <http://www.we.fh-osnabrueck.de/fbwe/vorlesung/edv2/gplot/gplot.html>.

Die Ausgabe geht in eine Datei oder auf ein Gerät. Treiber für einige Geräte gehören dazu, ebenso die Möglichkeit, PostScript-, HPGL- oder \LaTeX -Dateien zu erzeugen. Hier ein kurzes Beispiel. Wir schreiben ein Skript namens `plotscript`:

```
set terminal latex          # Ausgabe im LaTeX-Format
set output "plot.tex"      # Ausgabe nach File plot.tex
plot sin(x)/x              # zu zeichnende Funktion
```

Quelle 6.1 . `gnuplot`-Skript zum Zeichnen der Funktion $y = (\sin x/x)$, Ausgabe im \LaTeX -Format in Datei `plot.tex`

und rufen `gnuplot` mit dem Skript als Argument auf:

```
joe@debian:~$ gnuplot plotscript
```

Interaktiv wären die Kommandos:

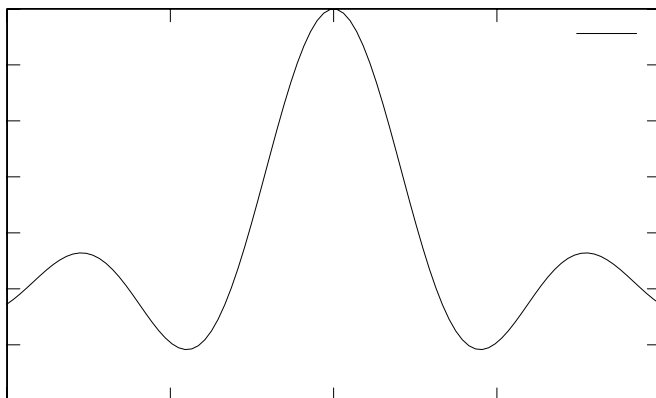


Abbildung 6.1. Diagramm von $(\sin x)/x$, erzeugt mit gnuplot

```
gnuplot
set terminal latex
set output "plot.tex"
plot sin(x)/x
quit
```

einzugeben. Für alle nicht genannten Parameter werden Default-Werte angenommen. Als Ausgabe erhalten wir eine \LaTeX -Picture-Umgebung, die sich mittels `\input` in ein \LaTeX -Dokument einbinden lässt, siehe Abbildung 6.1. Mit `set terminal jpeg` erhalten wir eine Ausgabe im jpg-Format. Abbildung 6.2 zeigt ein dreidimensionales Diagramm, das auf eine dreispaltige Tabelle mit Messwerten zurückgeht. Durch die Verkleinerung wird das Netz etwas verunziert, das Original hat die Größe DIN A4. Vor fünfzig Jahren hat man so etwas von Hand gezeichnet.

Außer kartesischen Koordinaten sind auch Polar-, Kugel- oder Zylinderkoordinaten möglich. Die Achsen können linear oder logarithmisch geteilt sein. Soweit sinnvoll werden reelle und komplexe Argumente verarbeitet.

6.4 Taschenrechner

6.4.1 Rechner für die Kommandozeile (bc, dc)

Der `bc` (basic calculator) ist ein interaktiver Rechner und eine kleine, interpretierte Programmiersprache zum Lösen einfacher numerischer Aufgaben. Der GNU-`bc` ist ein selbstständiges Programm, ansonsten dient der `bc` nur als Präprozessor für das Werkzeug `dc`. Der Rechner arbeitet mit Zahlen beliebiger Genauigkeit, welche die beiden Attribute *length* (Länge) und *scale* (Auflösung) besitzen. Unter *length* wird die gesamte Anzahl dezimaler Stellen – vor und nach dem Dezimaltrennzeichen –

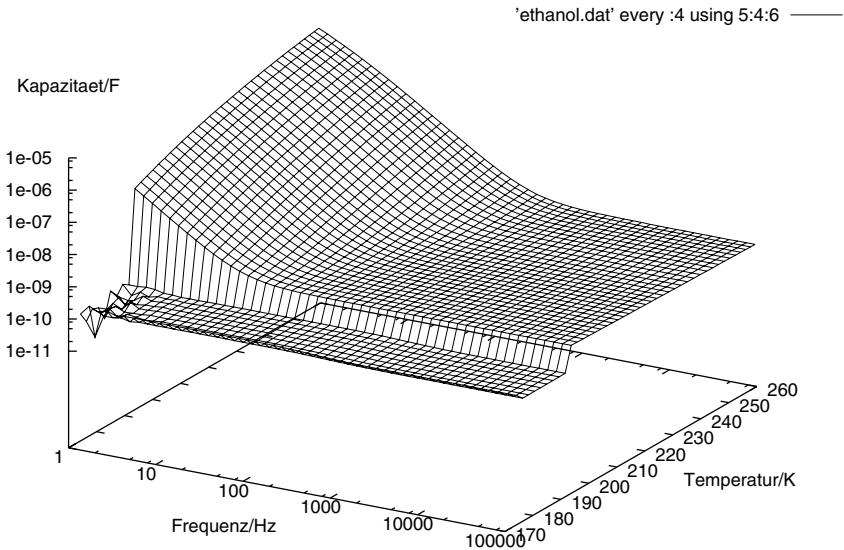


Abbildung 6.2. Dreidimensionaler Plot nach einer Tabelle von automatisch aufgenommenen Messwerten. Es geht um die Dielektrizitätskonstante von Äthanol bei Phasenumwandlungen in Abhängigkeit von der Frequenz.

verstanden, unter *scale* die Anzahl der dezimalen Stellen nach dem Dezimaltrennzeichen. Die Zahl 4711.099 hat eine Länge von 7 und eine Auflösung von 3 Stellen. Das Dezimaltrennzeichen ist stets der Punkt, auch in deutschsprachigen Umgebungen. Nach dem Aufruf von `bc` kann man beispielsweise folgende, mit `<cr>` abzuschließende Zeilen eingeben und erhält in der letzten Zeile das Ergebnis:

```
scale=10
sqrt((144*6)/32)
5.1961524227
quit
```

`sqrt()` ist eine Funktion, kein Operator wie `+` oder `/` (oder ein Wurzelzeichen). Ausdrücke werden wie gewohnt geklammert. Die Eingabe von `quit` beendet den Rechner.

Zur Umwandlung von Zahlen zu verschiedenen Basen dienen die `bc`-Variablen `ibase` (input base, Defaultwert 10, zulässig 2 bis 16) und `obase` (output base, Defaultwert 10, zulässig 2 bis zu einem Wert `BC_BASE_MAX`, siehe unten, auf der Testmaschine gleich 2147483647).

Folgende Zeilen bewirken, dass die Eingaben als Hexadezimalzahlen aufgefasst und als Dualzahlen ausgegeben werden:

```

obase=2
ibase=16
1A
11010
obase=5
1A
101
quit

```

In den letzten drei Zeilen ist die Ausgabebasis 5. Buchstaben müssen als Großbuchstaben eingegeben werden, Kleinbuchstaben kennzeichnen Variable. Die Buchstaben A bis F werden immer als Hexadezimalziffern verstanden. In obigem Beispiel kann man leicht in eine Denkfalle geraten. Wenn wir die Zeile `obase=5` eingeben, ist die Eingabebasis auf 16 gesetzt, die 5 ist eine hexadezimale Zahl. Wollen wir die Ausgabebasis auf 10 setzen, ist `obase=A` einzugeben, nicht `obase=10`. Im folgenden Beispiel sei der `bc` neu gestartet; wir wollen Dezimalzahlen in Sexagesimalzahlen (zur Basis 60) umrechnen, um uns mit den alten Babyloniern zu verständigen:

```

obase=60
69
01 09
71
01 11
1000
16 40
quit

```

Mit dem `bc`-Kommando `limits` erfährt man die Maximalwerte für einige Größen, darunter `BC_BASE_MAX`.

Der Rechner lässt sich vorteilhaft in Shellscripits und dergleichen verwenden, da die mathematischen Fähigkeiten der Shell eng begrenzt sind. Folgende Zeile erzeugt eine Shell-Umgebungsvariable `PI` mit dem entsprechenden Wert:

```
PI=$(echo "scale=12; 4*a(1)" | bc -l)
```

Mit der Option `-l` lädt der Rechner eine kleine mathematische Funktionsbibliothek, in der `a()` die arcustangens-Funktion bedeutet. Man kann auch eigene Funktionen definieren, beispielsweise zur rekursiven Berechnung von Fakultäten:

```

define f(x) {
    if (x <= 1) return(1);
    return(f(x - 1) * x); }
f(7)
5040

```

Nun ein Beispiel, dessen Idee bei AMIT SINGH (<http://www.kernelthread.com/hanoi/>) entlehnt ist. Wegen seines Umfangs schreibt man es zweckmäßig in eine Datei und gibt diese dem `bc` beim Aufruf als Argument mit. Es geht um ein Spiel, siehe Abschnitt 13.10 *Türme von Hanoi* auf Seite 424.

```

/*
 * nach http://hanoi.kernelthread.com
 */

define move(i, j) {
    print "move "
    print i
    print " -> "
    print j
    print "\n"
}

define play(n, h, d, o) {
    if (n > 0) {
        z = play(n-1, h, o, d)
        z = move(h, d)
        z = play(n-1, o, d, h)
    }
}

define hanoi(n) {
    z = play(n, 1, 3, 2)
}

```

Quelle 6.2. Script für den bc-Rechner zum Spielen der *Türme von Hanoi*

Im bc startet man das Spiel mittels `hanoi (n)`, wobei `n` die Anzahl der Scheiben ist.

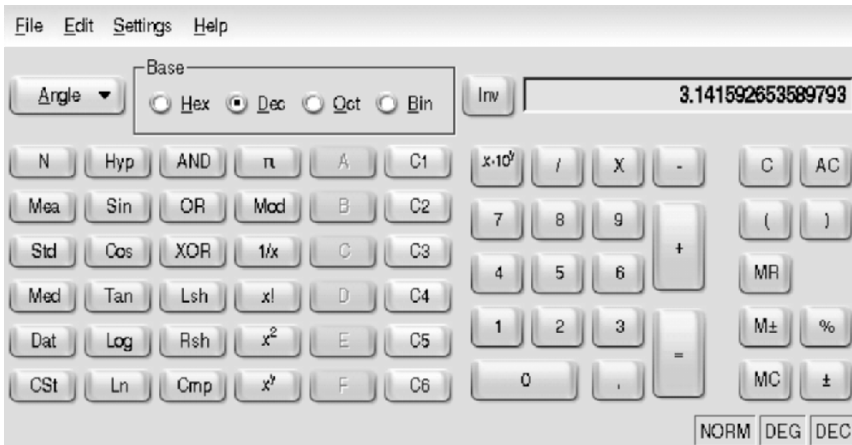


Abbildung 6.3. Screenshot KCalc, ein Taschenrechner für KDE, voll ausgebaut

Der `dc` (desk calculator) unterscheidet sich vom `bc` vor allem dadurch, dass er die Umgekehrte Polnische Notation (E: Reverse Polish Notation, RPN, F: notati-

on suffixiée) benutzt. Wer mit Taschenrechnern von Hewlett-Packard gearbeitet hat, kennt die Schreibweise. Sie ist dadurch gekennzeichnet, dass der Operator nach den Operanden eingegeben wird. Ferner muss man den `dc` mittels `p` ausdrücklich bitten, das Ergebnis anzuzeigen. Ein Beispiel:

```
8k
3
12
*
p
36
v
p
6.00000000
q
```

Die erste Zeile setzt die Auflösung auf 8 Stellen nach dem Komma (Punkt). Dann werden die Operanden 3 und 12 eingegeben, anschließend der Operator zum Multiplizieren. Mittels `p` erfährt man das Ergebnis. Der Operator `v` zieht die Quadratwurzel. `q` beendet den Rechner.

6.4.2 Ein Taschenrechner mit GUI (KCalc)

Wir kommen nun zu den Taschenrechnern, die auf dem Bildschirm auch wie solche aussehen. Alle setzen auf X11 auf. Statt Tasten anzutippen, klickt man mit der Maus auf die Darstellungen der Tasten. Wem die Umgekehrte Polnische Notation ans Herz gewachsen ist, der schaue sich die Debian-Pakete `calcoo` und `grpn` an; wir bleiben hier bei der algebraischen Schreibweise.

Als ersten Taschenrechner sehen wir uns `kcalc` aus dem KDE-Projekt an, der aus dem KDE-Menü über *Utilities* -> *Calculator* oder per Kommandozeile erreicht wird. Sein Funktionsumfang entspricht einem gehobenen Taschenrechner. Der Ausbaugrad lässt sich über den Menüpunkt *Settings* den jeweiligen Bedürfnissen anpassen. Abbildung 6.3 zeigt den maximalen Ausbau mit statistischen, trigonometrischen und logischen Funktionen. Durch Anklicken der Taste mit dem Fragezeichen bringt man das *KCalc Handbook* auf den Schirm. Die deutsche Ausgabe liegt auf <http://docs.kde.org/de/HEAD/kdeutils/kcalc/>. Um die Quadratwurzel zu ziehen, klickt man zuerst auf die INV-Taste (invers) und dann auf die Quadrat-Taste. Das Feld *NORM* rechts unten in der Ecke gehört zur Taste <INV>, mittels derer einige Funktionen durch ihre inverse Funktion ersetzt werden ($\sin \rightarrow \arcsin$).

6.4.3 Zwei Taschenrechner mit GUI (galculator)

Aus dem GNOME-Projekt stammt der `galculator`, der einen einfachen Standardmodus und einen wissenschaftlichen Modus mitbringt und sich zwischen algebraischer und Umgekehrter Polnischer Notation umschalten lässt.

Im wissenschaftlichen Modus gehen seine Funktionen etwas über den KCalc hinaus, andererseits bringt er keine Statistik-Funktionen mit. Die Hilfe beschränkt



Abbildung 6.4. Screenshot calculator, Standardaussehen

sich auf die Angabe der Version, mehr gibt es bei <http://calculator.sourceforge.net/>. Es lassen sich Formeln eingeben sowie benutzereigene Funktionen mit einer Variablen definieren. In der automatisch angelegten Datei \$HOME/.calculator werden persönliche Einstellungen gespeichert. Dorthin kann man auch mit einem Texteditor eigene Konstanten und Funktionen schreiben, eine Versuchung für einen Ingenieur.

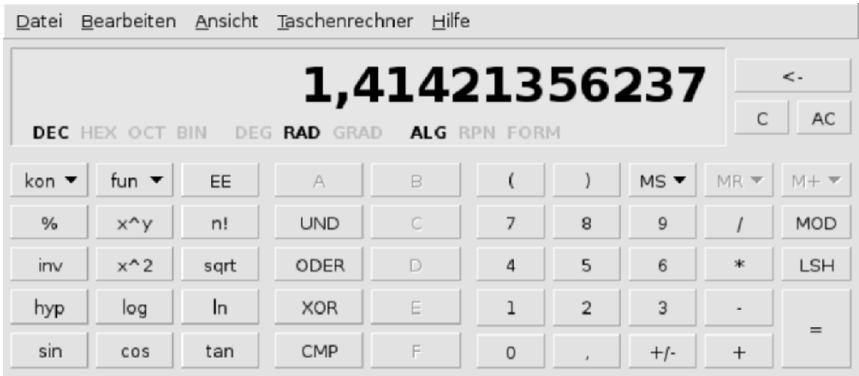


Abbildung 6.5. Screenshot calculator für wissenschaftliche Rechnungen

6.4.4 Drei Taschenrechner mit GUI (gcalctool)

Ebenfalls aus dem GNOME-Projekt stammt `gcalctool`, der drei Modi oder Ansichten beherrscht: einen Standard-Modus, einen Finanz-Modus und einen wissenschaftlichen Modus. Er arbeitet mit algebraischer Schreibweise und erlaubt benut-



Abbildung 6.6. Screenshot gcalc tool, Standardaussehen

zerdefinierte Konstanten und Funktionen. Als Hilfe kommt ein deutschsprachiges Handbuch auf den Schirm. Man kann eine Datei `$HOME/.gcalctoolrc` mit persönlichen Einstellungen anlegen.



Abbildung 6.7. Screenshot gcalc tool für kaufmännische Rechnungen

6.5 Zufallszahlen

Im Leben braucht man gelegentlich Folgen von Zufallszahlen, sei es für Simulationen, Probenahmen, Testzwecke, Passwörter, Schlüssel oder sei es für Spiele, seriöse oder leichtsinnige. Da ein Rechner grundsätzlich nicht würfelt, kann kein Programm



Abbildung 6.8. Screenshot gcalc tool für wissenschaftliche Rechnungen

Zufallszahlen erzeugen. Es gibt Hardware, die aus Rauschen oder aus dem radioaktiven Zerfall Zufallszahlen ableitet, aber den Aufwand leistet man sich selten. In den meisten Fällen reichen Folgen von Pseudo-Zufallszahlen, die zwar auf einem Algorithmus beruhen und bei Kenntnis aller Umstände reproduziert werden können, für einen bestimmten Zweck jedoch hinreichend zufällig erscheinen. DONALD E. KNUTH widmet dem Thema ein Kapitel im zweiten Band seines Lebenswerkes. Die deutsche Wikipedia bringt eine kurze Übersicht zu den Begriffen *Zufallszahl* und *Zufallszahlengenerator*. Unter <http://www.random.org/> finden sich weitere Hinweise.

Zu Linux/UNIX gehört eine C-Standardfunktion `rand()` oder `random()`, deren Beschreibung in der Sektion 3 des Manuals zu finden ist. Für einen Programmierer ist es eine Kleinigkeit, um diese Funktion herum ein Programm zu schreiben. Eine größere Auswahl an Generatoren bietet die Randlib (<ftp://odin.mda.uth.tmc.edu/pub/source/>). Braucht man nur einen Teil der Bits einer Pseudo-Zufallszahl, sind die oberen Bits zufälliger als die unteren. Will man mittels obiger Funktion `rand()` erzeugte Zufallszahlen z auf einen Bereich $0 \leq z < N$ begrenzen, so kann dies nach folgender Formel geschehen (aus dem FAQ der Newsgruppe *comp.lang.c*):

$$z = \frac{rand()}{RANDMAX/(N+1)} \quad \text{mit } N \ll RANDMAX$$

Darin ist *RANDMAX* die größte vom Generator erzeugte Zahl, deren Wert man in der Datei `/usr/include/stdlib.h` findet, beispielsweise 214748364.

Die Bash stellt eine Umgebungsvariable *RANDOM* mit Werten von 0 bis 32.767 zur Verfügung. Den zugehörigen einfachen Algorithmus nach dem Beispiel des ANSI-C-Standards findet man in der Quelldatei `variables.c` zur Bash². Für den Startwert (E: seed) des Zufallszahlengenerators wird als Default der Wert 1 genommen. Der Benutzer kann durch Setzen der Umgebungsvariablen *RANDOM* einen anderen Wert vorgeben:

```
joe@debian:~$ RANDOM=2
```

Sieht man sich die Zufallszahl an, erkennt man keinen Zusammenhang mit dem Startwert:

```
joe@debian:~$ echo $RANDOM
```

Derselbe Startwert führt jedoch immer zu derselben Zufallszahl. Nimmt man als Startwert die Anzahl der Sekunden auf der Linux/UNIX-Uhr (`date +%s`), so erhält man unterschiedliche Werte. Löscht man die Variable *RANDOM*, indem man ihr nichts zuweist, und definiert sie dann erneut, verliert sie ihre besondere Bedeutung und wird zu einer beliebigen, benutzerdefinierten Umgebungsvariablen.

Eine Suche nach dem Stichwort *random* in den Debian-Paketlisten fördert nur einige Werkzeuge zu Tage, um Webseiten oder Musikstücke in zufälliger Folge wiederzugeben, keine kritische Anwendung. Das Werkzeug *ent* (entropy) unterzieht Bit- oder Bytefolgen einigen statistischen Tests (Chi-Quadrat-Test und andere) auf ihre Eignung als Folge pseudo-zufälliger Muster.

6.6 Tabellenkalkulationen

6.6.1 Einfachst: *sc*

Eine Tabellenkalkulation (E: spreadsheet, F: *tableur*) ist ein gutes Modell für Vorgänge, die sich durch Zahlen, Datumsangaben oder Zeichenketten darstellen lassen, die in Spalten, Zeilen und gegebenenfalls Blättern geordnet sind, wobei manche Spalten, Zeilen oder Blätter sich aus anderen ergeben (Summen, Differenzen, Produkte ...). Im Gegensatz zu Datenbanken steht hier das Rechnen mit nicht zu großen Datenmengen im Vordergrund. Versuchsauswertungen, Haushaltspläne und Statistiken sind typische Anwendungen. Bei solchen Aufgaben lassen sich leicht Fragestellungen der Art *Was wäre, wenn ...* durchspielen. Eine Suche im Netz fördert etwa zwei Dutzend Tabellenkalkulationen für Linux zu Tage. Wir beschränken uns auf einen anspruchslosen Vertreter und die Werkzeuge aus den drei freien Office-Paketen. Der

²Hier zeigt sich wieder der Vorteil freier oder offener Software: Man kann in den Quellen nachlesen. Bei nicht-offener Software muss man glauben.

Urahne aller Tabellenkalkulationen ist VisiCalc von DAN BRICKLIN aus dem Jahr 1979, eine kommerzielle Alternative unter MS Windows ist MS Excel.

Wenn die mathematischen Fähigkeiten von Tabellenkalkulationen nicht ausreichen, muss man zu numerischen Werkzeugen (siehe Abschnitt 6.7 *Numerik* auf Seite 266), Statistik-Paketen (siehe Abschnitt 6.8 *Statistik* auf Seite 268) oder Computer-Algebra-Systemen (siehe Abschnitt 6.9 *Computeralgebra* auf Seite 269) greifen. Diese Werkzeuge setzen mathematische Kenntnisse und eine gewisse Einarbeitung voraus, während man mit einer Tabellenkalkulation den ersten Erfolg schon nach einer Viertelstunde verbuchen kann.

Das Kommando `sc` (spreadsheet calculator) startet eine einfache Tabellenkalkulation, die ohne X11 auskommt und mit einem Textbildschirm zufrieden ist. Damit ist sie auch für ältere PCs geeignet, die unter X11 in die Knie gehen. Die Kalkulation verwendet `vi`-ähnliche Kommandos und beruht wie dieser auf der `curses`-Bibliothek. Eine Maus wird nicht unterstützt. Konfigurieren lässt sich das Programm durch Dateien `.scrc` im Home-Verzeichnis und in Unterverzeichnissen. Außer einer ausführlichen Manual-Seite bringt das Programm ein kurzes Tutorial mit, das sich mittels:

```
joe@debian:~$ sc /usr/share/doc/sc/tutorial.sc
```

anzeigen lässt, sowie eine siebenseitige Quick Reference, die mittels:

```
joe@debian:~$ scqrf | lpr
```

ausgedruckt wird. Die Zellen können Zahlen, Zeichenketten oder Formeln enthalten, wobei die gängigen Funktionen mitgeliefert werden. Wenn man auf die Zusammenarbeit mit anderen Office-Werkzeugen keinen Wert legt, reicht `sc` für viele Aufgaben aus.

6.6.2 GNOME Gnumeric

Die Tabellenkalkulation von GNOME ist Gnumeric. Die Stärken des Werkzeugs liegen in der anpassungsfähigen Oberfläche, der Vielzahl von Dateiformaten (17 lesen, 13 schreiben) und der sorgfältigen Mathematik. Eine ausführliche Dokumentation ist über die Startseite des Projektes zu erreichen: <http://www.gnome.org/projects/gnumeric/>. Unter Debian beherrscht Gnumeric per Default Tabellen mit maximal 256 Zeilen und 65535 Spalten. Benötigt man andere Werte, geht man in die Quelldatei `src/gnumeric.h`, setzt die Variablen `SHEET_MAX_COLS` und `SHEET_MAX_ROWS` (gleich zu Anfang der Datei) auf die gewünschten Werte und übersetzt (kompiliert) das Programm neu³. Zum Jahreswechsel 2004/5 war Version 1.4.1 sowohl die jüngste stabile Version bei GNOME als auch die aktuelle Version in Debian *sarge*. Gnumeric ist Teil des GNOME Office-Paketes, siehe Abschnitt 12.3 *GNOME Office* auf Seite 410.

Das Werkzeug öffnet per Default ein Buch (E: workbook) mit drei Blättern (E: worksheet). Jedes Buch wird in jeweils einer Datei gespeichert. Ein Blatt enthält

³Noch ein Vorzug freier oder offener Software: der Benutzer kann die Software seinen Wünschen anpassen.

eine Tabelle, bestehend aus Zellen (E: cell), die in Zeilen (E: row) und Spalten (E: column) angeordnet sind. Die Anzahlen der Blätter, Zeilen und Spalten lassen sich anpassen. Das erste oder letzte Blatt eines Buches kann eine Zusammenfassung der übrigen Blätter enthalten, sodass sich beispielsweise Gesamtdarstellungen des Haushaltsplanes eines Projektes über mehrere Haushaltsjahre oder oder eine Zusammenfassung der Haushaltspläne der Abteilungen eines Vereins zu einem Plan des Gesamtvereins erstellen lassen. Wenn nichts anderes verlangt werden die Daten als XML-Text, komprimiert mittels `gzip`, abgespeichert.

The screenshot shows the Gnumeric spreadsheet interface. The menu bar includes 'Datei', 'Bearbeiten', 'Ansicht', 'Einfügen', 'Format', 'Werkzeuge', 'Daten', and 'Hilfe'. The toolbar contains various icons for file operations, editing, and formatting. The formula bar displays '=B2-B1'. The spreadsheet grid has columns labeled A through H and rows numbered 1 through 11. The data in the grid is as follows:

	A	B	C	D	E	F	G	H
1	0	2	2	0	0	0	0	0
2	1	4	2	4	4	0,11111	0,05556	0,11111
3	2	6	2	8	12	0,22222	0,11111	0,33333
4	3	7	1	6	18	0,16667	0,16667	0,5
5	4	8	1	6	24	0,16667	0,16667	0,66667
6	5	10	2	8	32	0,22222	0,11111	0,88889
7	6	12	2	4	36	0,11111	0,05556	1
8	7	14	2	0	36	0	0	1
9	8	20	6	0	36	0	0	1
10								
11								

The bottom status bar shows 'Summe=2'.

Abbildung 6.9. Screenshot der GNOME Tabellenkalkulation Gnumeric mit einer einfachen Versuchsauswertung

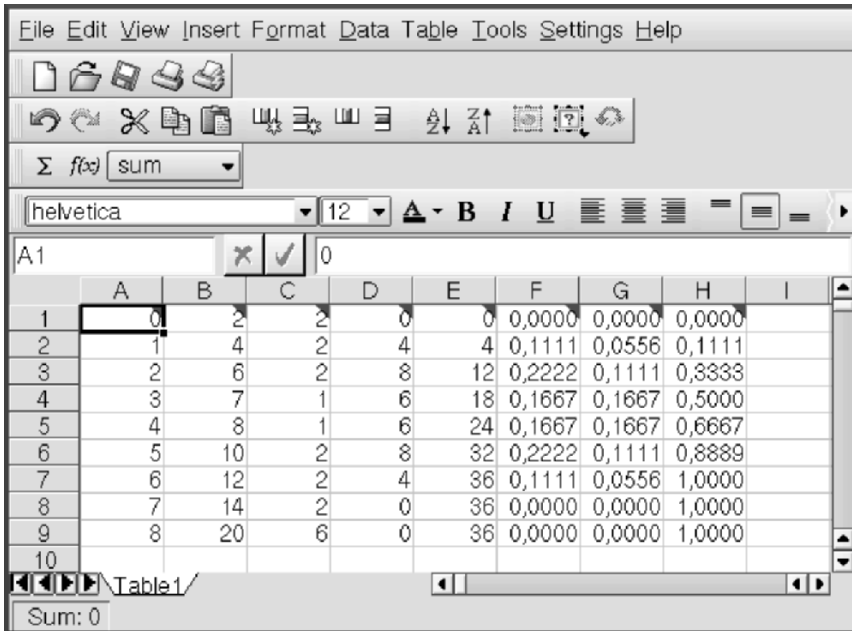
Der in Abbildung 6.9 gezeigte Screenshot stellt eine einfache Versuchsauswertung mit fiktiven Daten aus der Krümelkunde dar. Die Spalten enthalten als Kommentar ihre Bedeutung:

- Spalte A: laufender, bei 0 beginnender Zeilenindex
- Spalte B: obere Klassengrenze, die mit zur Klasse rechnet,
- Spalte C: Klassenbreite als Differenz der Klassengrenzen, wobei der erste Wert als Zahl eingegeben werden muss,
- Spalte D: Besetzungszahl
- Spalte E: kumulierte Besetzungszahl als Summe der Besetzungszahlen, wobei der erste Wert als Zahl eingegeben werden muss,
- Spalte F: relative Häufigkeit als Quotient aus D und dem Endwert von E,

- Spalte G: Verteilungsdichte als Quotient aus F und C,
- Spalte H: Verteilungssumme als Quotient aus E und dem Endwert von E.

Gnumeric hinterließ einen sehr guten Eindruck, ohne dass wir diese Meinung im Einzelnen begründen könnten.

6.6.3 KDE KSpread



	A	B	C	D	E	F	G	H	I
1	0	2	2	0	0	0,0000	0,0000	0,0000	
2	1	4	2	4	4	0,1111	0,0556	0,1111	
3	2	6	2	8	12	0,2222	0,1111	0,3333	
4	3	7	1	6	18	0,1667	0,1667	0,5000	
5	4	8	1	6	24	0,1667	0,1667	0,6667	
6	5	10	2	8	32	0,2222	0,1111	0,8889	
7	6	12	2	4	36	0,1111	0,0556	1,0000	
8	7	14	2	0	36	0,0000	0,0000	1,0000	
9	8	20	6	0	36	0,0000	0,0000	1,0000	
10									

Abbildung 6.10. Screenshot der KDE Tabellenkalkulation KSpread mit einer einfachen Versuchsauswertung

KSpread ist die Tabellenkalkulation von KDE und Bestandteil von KOffice. Ein deutschsprachiges Handbuch liegt unter <http://docs.kde.org/de/HEAD/koffice/kspread/>. KSpread kann Tabellen mit maximal je 32767 Zeilen und Spalten bearbeiten. Die Benutzung gestaltet sich ähnlich wie die von Gnumeric.

6.6.4 OpenOffice.org Calc

Die Tabellenkalkulation Calc von OpenOffice.org gibt es nicht als einzelnes Debian-Paket, sondern nur als Gesamtpaket *openoffice.org* oder bei <http://www.openoffice.org/> als komplettes Office-Paket, siehe Abschnitt 12.2 *OpenOffice.org* auf Seite 407.

	A	B	C	D	E	F	G
1	0	2	2	0	0	0	0
2	1	4	2	4	4	0,11	0,06
3	2	6	2	8	12	0,22	0,11
4	3	7	1	6	18	0,17	0,17
5	4	8	1	6	24	0,17	0,17
6	5	10	2	8	32	0,22	0,11
7	6	12	2	4	36	0,11	0,06
8	7	14	2	0	36	0	0
9	8	20	6	0	36	0	0
10							
11							
12							

Abbildung 6.11. Screenshot der Tabellenkalkulation Calc von OpenOffice.org

Die elementaren Funktionen sind gleich denen von Gnumeric oder Kspread. Eine sich auf das gesamte Office-Paket beziehende deutschsprachige Hilfe ist online verfügbar. Unter dem Punkt *Support* auf der Startseite des Projektes kann man sich zu weiteren Dokumentationen im pdf-Format durchklicken.

6.7 Numerik

6.7.1 Octave

Die hier genannten Werkzeuge dienen vor allem dem Zahlenrechnen und beherrschen auch Grafik zur Darstellung der Ergebnisse. Das Octave-System aus dem Debian-Paket *octave2.1* samt der Dokumentation aus *octave2.1-doc* ist ein interaktives Werkzeug und zugleich eine höhere, spezialisierte Programmiersprache zum Lösen von Aufgaben der Numerischen Mathematik. Dazu zählen:

- Reelle und komplexe Arithmetik,
- Trigonometrie,
- Matrizenrechnung,
- lineare Algebra,
- nichtlineare Gleichungssysteme,
- gewöhnliche Differentialgleichungen,
- numerische Integration,
- Optimierung,
- Statistik,
- Systemtheorie, Signalverarbeitung, Fast Fourier Transformation.

Das System stammt aus dem Chemieingenieurwesen, wo es für die Berechnung von Reaktoren eingesetzt wird. Es weist Ähnlichkeiten zu MatLab auf, die – unter Verzicht auf einige Vorzüge von Octave – per Konfiguration verstärkt werden können. Webseiten samt Dokumentation zu Octave finden sich unter <http://www.octave.org/> und <http://octave.sourceforge.net/>.

6.7.2 Euler

Das numerische Labor *euler* – benannt nach dem Mathematiker und Physiker LEONHARD EULER (1707 - 1783) – samt der Dokumentation im Paket *euler-doc* stellt interaktiv eine Vielzahl von Operationen mit reellen und komplexen Zahlen, mit Intervallen und Matrizen usw. bereit. Es weist ebenfalls Ähnlichkeiten mit MatLab auf. Eine Kommandoliste liegt unter `/usr/share/euler/help.txt`, eine ausführliche Beschreibung unter `/usr/share/doc/euler-doc/`. Zugehörige Webseiten finden sich bei <http://mathsrv.ku-eichstaett.de/MGF/homes/grothmann/euler/> und <http://euler.sourceforge.net/>.

6.7.3 Scilab

Nicht bei Debian, sondern beim Institut National de Recherche en Informatique et en Automatique (INRIA) bei Paris – dem ersten Internet-Teilnehmer in Frankreich – ist unter dem URL <http://scilabsoft.inria.fr/> das Numerik-Paket Scilab als Tarball von 20 MB mit Linux-Binaries oder mit den Quellen erhältlich. Eine Einrichtung der Binaries unter *sarge* im Verzeichnis `/usr/local/scilab/` verlief problemlos. Das Computeralgebrasystem MuPAD (Abschnitt 6.9.3 *MuPAD* auf Seite 270) hat Scilab integriert; zu Maple gibt es eine Schnittstelle.

Das Paket bringt Hunderte mathematischer Funktionen mit und lässt sich durch eigene Funktionen in C, Fortran oder weiteren Sprachen ergänzen. Es kann mehr als Plot-Programme, aber nicht so viel wie ein Computeralgebrasystem. Scilab beherrscht:

- zwei- oder dreidimensionale Grafiken, auch bewegt (animiert),
- Lineare Algebra,
- Polynome,
- rationale Funktionen,
- Simulationen,
- Gewöhnliche Differentialgleichungen,
- Differential-algebraische Gleichungen,
- Regeltechnik,
- Optimierung,
- Signalverarbeitung,
- Statistik.

und macht von mehreren externen Bibliotheken (Linpack u. a.) Gebrauch. Die Dokumentation ist zum größten Teil auf Englisch oder Französisch geschrieben; es gibt jedoch auf der Website eine ins Deutsche übersetzte Einführung in Scilab von BRUNO PINÇON im Umfang von 120 Seiten bzw. 1 MB als PDF-Datei.

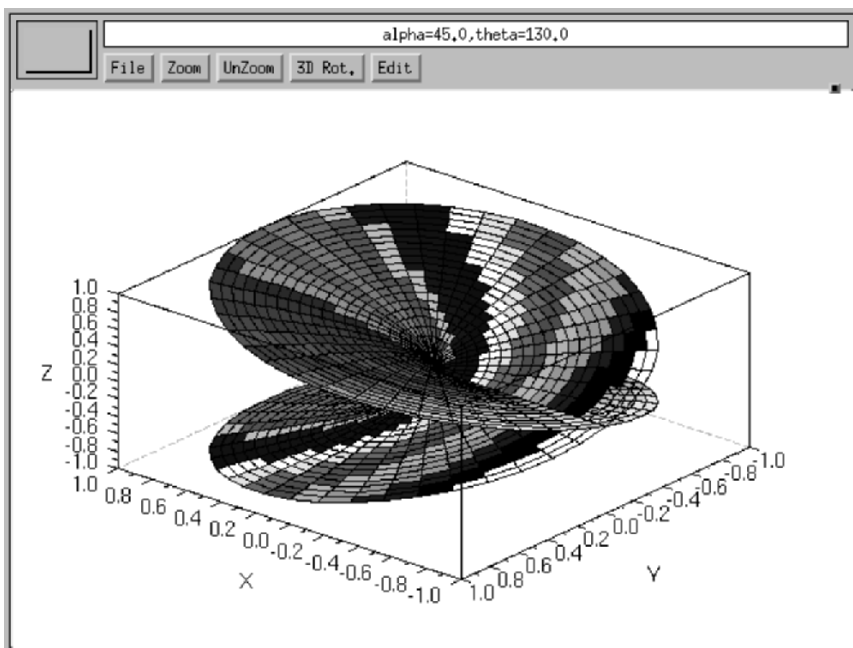


Abbildung 6.12. Screenshot des Numerik-Werkzeugs Scilab, hier die Demo zu einer RIEMANN-Fläche, auf dem Bildschirm farbig

6.8 Statistik (GNU R)

GNU R aus dem Debian-Paket *r-base* sowie dessen Begleitern in der Abteilung *Mathematics* ist der GNU-Klon des Statistik-Systems S, dessen Anfänge von JOHN CHAMBERS in den Bell Labs um 1975 entwickelt wurden. Es steht Statistik-Software wie SAS (<http://www.sas.com/>) oder SPSS (<http://www.spss.com/>) nahe. Eines der Debian-Pakete *r-doc-info*, *r-doc-pdf* oder *r-doc-html* sollte man auch gleich einrichten. R ist kein Werkzeug, sondern eine Maschinenversion, eine Programmiersprache und eine Umgebung für statistische Aufgaben und Datenvisualisierung. Es ersetzt kein Lehrbuch für Statistik⁴, trotz einer umfangreichen Dokumentation. Diese liegt vorwiegend unter `/usr/lib/R/doc/`. Zu S liefert <http://cm.bell-labs.com/cm/ms/departments/sia/doc/comp.html> viele Hinweise, zu R <http://www.r-project.org/>.

Abbildung 6.13 zeigt die Visualisierung von Messdaten, hier die Temperatur auf einem Vulkan, dargestellt in Form einer Landkarte mit Isothermen (Linien gleicher Temperatur). Das Beispiel ist eine Demo aus dem Paket. Wenn man GNU-R aus der Kommandozeile aufruft:

```
joe@debian:~$ R
```

⁴siehe Abschnitt N *Zum Weiterlesen* auf Seite 508.

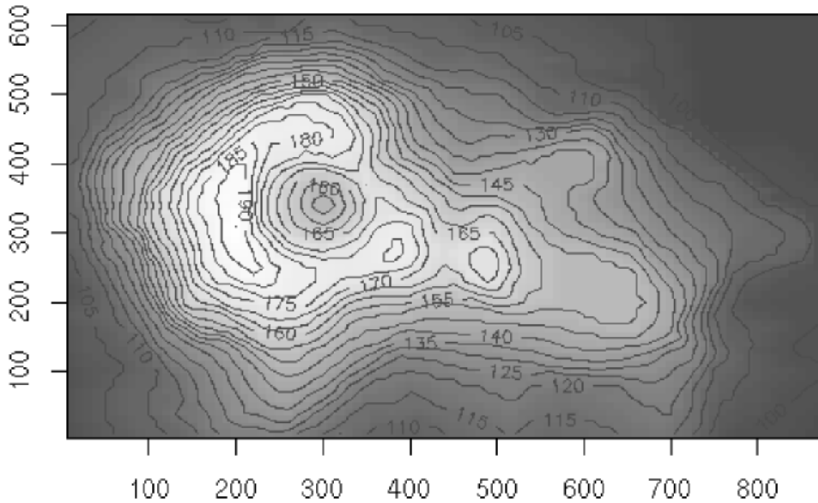


Abbildung 6.13. Screenshot einer mittels GNU-R dargestellten Temperaturverteilung auf einem Vulkan (mitgelieferte Demo). Die Farben reichen von Rot am Bildrand bis Weiß nahe des Zentrums. Die Linien sind Isothermen.

bekommt man einige Zeilen Text mit Vorschlägen für den Anfang und den R-Prompt auf den Schirm.

6.9 Computeralgebra

6.9.1 Was ist ein Computeralgebrasystem?

Ein Computeralgebrasystem (CAS) ist eine Software, die symbolische Mathematik (Buchstabenrechnung) beherrscht, neben der herkömmlichen Rechnung mit Zahlen beliebiger Länge (Genauigkeit). Grafik gehört auch dazu. Weit verbreitet sind die kommerziellen Computeralgebrasysteme Maple und Mathematica. Maple wurde an der University of Waterloo, Kanada entwickelt und wird heute von der Waterloo Maple Inc. vertrieben (<http://www.maplesoft.com/>). Mathematica wurde von STEPHEN WOLFRAM geschaffen, damals Physikprofessor an der University of Illinois. Heute wird Mathematica von Wolfram Research vertrieben (<http://www.wolfram.com/>). Beide Systeme haben eine lange Entwicklung durchlaufen – seit der Mitte der achtziger Jahre – und sind gegen Bares auch für Linux erhältlich. Für Lehr- und Lernkräfte gibt es besondere Ausgaben und Lizenzen. Ein weiteres kommerzielles, auch für Linux verfügbares Computeralgebrasystem ist REDUCE (<http://www.reduce-algebra.com/>).

6.9.2 YACAS

YACAS – Yet Another Computer Algebra System – ist als gleichnamiges Debian-Paket sowie als Tarball bei <http://yacas.sourceforge.net/> oder <http://www.xs4all/~apinkus/> zu haben. Es ist (noch) klein, flexibel, leicht erweiterbar und lehnt sich in seiner Syntax an Mathematica an. YACAS lässt sich in einen Client und einen Server aufteilen, die auf verschiedenen Maschinen laufen. Im Paket *yacas-proteus* findet sich eine noch experimentelle Benutzeroberfläche, die mit dem Kommando `proteusworksheet` aufzurufen ist..

6.9.3 MuPAD

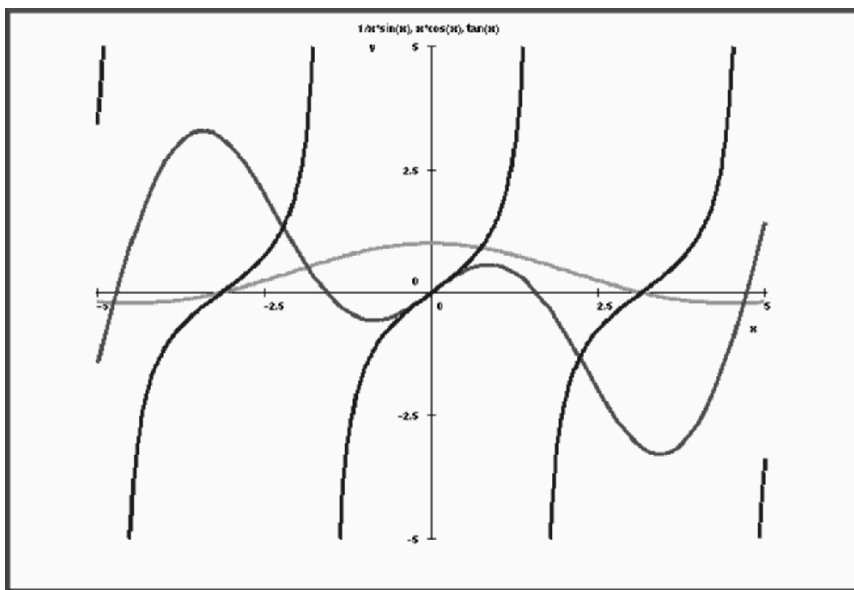


Abbildung 6.14. MuPAD 2D-Plot verschiedener Funktionen, auf dem Bildschirm durch Farben unterschieden

MuPAD (Multiprocessing Algebra Datatool) ist ein offenes Computeralgebrasystem der Universität Paderborn (<http://www.mupad.de/> oder <http://research.mupad.de/>). Für Lehre und Forschung sowie für den privaten Gebrauch ist es kostenfrei; man muss sich nur registrieren lassen, falls man die Software voll ausnutzen will. Es steht nicht als Debian-Paket zur Verfügung. Speziell mit MuPAD in Schulen befasst sich <http://mupad.zum.de/>. Von einem der genannten Server oder einem Spiegel lädt man sich die Version 2.5.3 (die kommerzielle Version Pro 3.1 ist nur für Windows erhältlich) für Linux, mit Scilab (siehe Abschnitt 6.7.3 *Scilab* auf Seite 267), als RPM-Paket herunter, etwa 30 MB groß. Dann richtet man als root das Paket ein:

```
debian:~# alien -i MuPADscilab-2.5.3-2.i386.rpm
```

Die MuPAD-Verzeichnisse gehen nach `/usr/local/`. Dort ist noch ein Symlink anzulegen:

```
debian:~# ln -s MuPADscilab-2.5.3 MuPAD
```

dann kann ein gewöhnlicher Benutzer anfangen:

```
joe@debian:~$ mupad
```

oder, falls X11 läuft:

```
joe@debian:~$ xmupad
```

Als erste Eingabe ist ein Fragezeichen nie verkehrt. In `xmupad` kann man den Help-Button anklicken. Das Beispiel in Abbildung 6.14 wurde durch Eingabe von:

```
plotfunc2d(sin(x)/x, x*cos(x), tan(x), x=-5..5, y=-5..5)
```

in `xmupad` erzeugt und durch das Unterprogramm VCam im png-Format exportiert.

6.9.4 Axiom

Axiom ist ein Computeralgebrasystem, das schon 1973 als kommerzielles Produkt unter dem Namen *Scratchpad* begann, seit 2004 aber frei und offen ist. Ab *sarge* ist es auch bei Debian in Form von 10 Paketen verfügbar. Es wurde entwickelt von der Numerical Algorithms Group (NAG, <http://www.nag.co.uk/>), die für ihre mathematischen Funktionsbibliotheken für Fortran und C bekannt ist. Beiträge lieferte auch die Firma IBM. Das System hat Schnittstellen zu Fortran und \LaTeX .

Die beiden wichtigsten Pakete sind *axiom* und *axiom-doc*. Letzteres richtet ein Buch von 1200 Seiten im PDF-Format und ein Tutorial im HTML-Format in `/usr/share/doc/axiom-doc/` ein. Es empfiehlt sich, mit dem Tutorial zu beginnen. Wir rufen Axiom aus der Kommandozeile auf:

```
joe@debian:~$ axiom
```

und bekommen nach der Überschrift und einigen Hinweisen den Prompt `->` des Werkzeugs. Dort geben wir nacheinander folgende Zeilen ein und schauen uns die jeweiligen Ergebnisse an (hier ausgelassen):

```
-> )summary
-> 5 + 6
-> 12 * 13
-> (3 + 4) * (5 + 6)
-> 6/4
-> sqrt(2.0)
-> sin 2.45
-> log 3.1
-> gcd(156,169)
-> lcm(156,169)
-> prime?(4711)
-> )quit
```

Den Rest lese man im Tutorial nach. Wir wenden uns dem nächsten CAS zu.

6.9.5 SINGULAR

Das Computeralgebrasystem SINGULAR wurde in der Universität Kaiserslautern entwickelt. Auf <http://www.singular.uni-kl.de/> steht es unter der GPL in Form zweier Tarbälle für Linux und andere Betriebssysteme zur Verfügung. Es deckt nicht ein so breites Aufgabengebiet ab wie die bekannteren Systeme; Polynome sind seine Stärke. Im Netz finden sich weitere, stärker spezialisierte Computeralgebrasysteme, die jedoch vertiefte mathematische Kenntnisse erfordern. Mit SINGULAR könnte man ohne diese wenigstens noch seine Steuererklärung ausrechnen, auch wenn das Overkill wäre. Die Website enthält einige anwendungsnahe Beispiele

Wer sich eingehender mit mathematischer Software befassen will, findet Informationen in reichlicher Menge bei:

- <http://directory.google.com/Top/Science/Math/Software/>,
- <http://gams.nist.gov/>,
- <http://math.haifa.ac.il/msoftware.html>,
- <http://mathematik.zum.de/> (Schulmathematik).

Datenbanken und andere Wissensspeicher

Hier werden einige Werkzeuge zum Bearbeiten strukturierter Daten sowie zwei unter Debian verfügbare Datenbanken erklärt.

7.1 Grundbegriffe

Wenn es darum geht, umfangreiche Datenmengen so abzulegen, dass eine größerer Benutzerkreis gezielt darauf zugreifen kann, stehen mehrere Modelle zur Verfügung:

- Datenbanken (E: data base system, F: base de données),
- Wikis,
- Verzeichnisdienste (E: directory service, F: service d'annuaire),
- Expertensysteme (E: expert system, F: système expert).

Vermutlich gibt es noch mehr Modelle, außerdem lassen sich die genannten Möglichkeiten feiner unterteilen. Welches Modell man wählt, hängt von der Art der Daten und dem gewünschten Zugriff ab. Teilweise sind die Modelle auch verflochten. So kann ein Expertensystem auf eine Datenbank zurückgreifen.

Nicht jede Ansammlung von Daten ist eine Datenbank, wenngleich mit dem Begriff großzügig umgegangen wird. Eine Datenbank enthält und verwaltet eine Menge ähnlicher Daten. Typische Anwendungsfälle sind:

- Listen von Mitgliedern, Kunden, Patienten,
- Ahnengalerien,
- Bücherkataloge,
- umfangreiche Literatur- oder Linksammlungen,
- Foto-, Dia-, Schallplatten- oder Briefmarkensammlungen,
- Inventarverzeichnisse,
- Buchhaltung, Lagerverwaltung,
- Ergebnisse von Wettkämpfen,
- Wörterbücher, Thesauri,
- Inhalte von Webseiten.

Mit den Daten werden vor allem folgende Operationen durchgeführt:

- Eingeben, Ändern, Löschen,
- Suchen,
- Auswählen,
- Verknüpfen,
- Sortieren.

Rechnen können Datenbanken auch, aber das ist nicht ihre Stärke. Jenseits der Grundrechenarten, dem Wurzelziehen, Logarithmen, Winkelfunktionen, einfachen statistischen Rechnungen und dem Umgang mit Kalenderdaten ist Schluss. Bei manchen Aufgaben hat man die Wahl zwischen einer Datenbank und einer Tabellenkalkulation, siehe Abschnitt 6.6 *Tabellenkalkulationen* auf Seite 262. In begrenztem Umfang lässt sich eine Tabellenkalkulation als Datenbank missbrauchen und umgekehrt. Wenn das Rechnen im Vordergrund steht und nicht zu viele Daten in mannigfacher Weise voneinander abhängen, ist vermutlich eine Tabellenkalkulation das bessere Modell. Auch die Kombination beider Werkzeuge ist manchmal zweckmäßig: Die Datenbank liefert die bereits vorkonzentrierten Daten für die Tabellenkalkulation. Oder umgekehrt: Die Ergebnisse aus der Tabellenkalkulation werden, wenn sie stabil geworden sind, langfristig in einer Datenbank verwahrt.

Bei Datenbanken unterscheidet man mehrere Modelle. Am weitesten verbreitet sind relationale Datenbanken (Relational Data Base Management System, RDBMS). Sie stellen alle Daten in Form von Tabellen mit Zeilen und Spalten dar. In den Schnittpunkten von Zeilen und Spalten befinden sich die Felder mit den Daten. Die Tabellen sind vielfältig durch Relationen (Beziehungen) miteinander verknüpft. Das ist für viele Anwendungen eine gewohnte Darstellung. Andere Modelle sind hierarchische oder objektorientierte Datenbanken. Eine Datenbank gliedert sich in:

- die Datenbasis (data base, DB)
- die Datenverwaltung (data base management system, DBMS)

Ein Benutzer verkehrt immer nur mit dem DBMS. Wie die Datenbasis organisiert ist, geht ihn nichts an. Große Datenbanken bekommen ein rohes Stück Massenspeicher zugewiesen und legen darin ihren *Tablespace* an, etwas Ähnliches wie ein eigenes Dateisystem. Mittlere Datenbanken greifen auf ein Linux/UNIX-Dateisystem zurück und legen ihre Tabellen als Linux/UNIX-Dateien an. Zwischen DBMS und Benutzer ist oft noch eine Ebene zwischengeschaltet, die den Umgang mit der Datenbank benutzerfreundlich gestaltet. Im einfachsten Fall handelt es sich dabei um Shellskripts. Aufwendiger ist ein Webserver, um die Datenbank vollständig vor dem Benutzer zu verbergen, der nur mit seinem Browser arbeitet.

Bei Datenbanken spielt der Begriff der *Transaktion* eine Rolle. Er bedeutet, dass eine Eingabe, Änderung oder Löschung von Daten entweder ganz oder gar nicht durchgeführt wird. Damit wird bei Störungen verhindert, dass kranke Datensätze entstehen, eine wesentliche Voraussetzung für einen stabilen Betrieb. Das Musterbeispiel einer Transaktion ist eine Banküberweisung, bei der einem Konto ein Betrag belastet und einem anderen Konto gut geschrieben wird. Würde die Überweisung auf halber Strecke abgebrochen, gäbe das ein Chaos auf den Konten. Deshalb

wird eine Transaktion zunächst nur unter Vorbehalt durchgeführt und kann zurück genommen werden (rollback). Erst nach erfolgreichem Abschluss wird sie endgültig übernommen (commit). PostgreSQL, MySQL und selbstverständlich alle großen Datenbanken beherrschen Transaktionen.

Zum Dialog mit dem DBMS braucht der Benutzer eine Sprache. Weit verbreitet ist die *Structured Query Language*, kurz SQL. Das ist eine standardisierte deskriptive oder deklarative Programmiersprache, in der der Benutzer in gebrochenem Englisch beschreibt, was er als Ergebnis haben will, nicht wie das Programm zum Ziel gelangt. Die meisten Datenbanken beherrschen eine Untermenge des Standards und fügen eigene Erweiterungen hinzu. Die Datenbank PostgreSQL hält sich ziemlich eng an den Standard ANSI/ISO SQL92, die Datenbank MySQL weniger, ist aber wegen ihrer Geschwindigkeit im Umfeld von Web-Anwendungen beliebt. Die Anfänge von SQL sind einfach und werden bei der Datenbank PostgreSQL im Abschnitt 7.4 auf Seite 279 erläutert. SQL-Kommandos werden interaktiv oder häufiger zu Skripten gebündelt verabreicht.

SQL kennt als deskriptive Sprache einige Konstrukte wie Schleifen nicht, die Programmierer von prozeduralen Sprachen (Fortran, Cobol, Pascal, C) her gewohnt sind und die auch für manche Aufgaben benötigt werden. Deshalb ergänzen viele Datenbanken ihr SQL durch eine prozedurale Sprache wie PL/SQL (Oracle) oder PL/pgSQL (PostgreSQL). Damit ist es möglich, prozedurale Programme zu schreiben, die SQL-Anweisungen enthalten. Wer ein einfaches C-Programm schreiben kann, kommt bald damit zurecht. Viele Datenbanken bringen Schnittstellen (Funktions- oder Klassenbibliotheken) für gebräuchliche Programmiersprachen (C/C++, Perl) mit.

The screenshot shows a database search interface titled "BIBSUCHEN". It contains several input fields for search criteria:

- LFD: 230
- ART: BUC
- AUTOR1: Alex, W.
- AUTOR2: Malmqvist, K.
- AUTOR3: Nystroem, C.
- TITEL: Partikelteknologisk ordlista med definitioner Tysk-Engelsk-Sve
- SCHLW: Woerter Schwedisch Partikel
- JAHR: 1977
- REIHE: (empty)
- VERLAG: (empty)
- ISBN: (empty)
- BIB: MV
- RAUM: 201

At the bottom, there is a control panel with the following elements:

- A button labeled "Automatisch".
- A status bar showing "Count: 1" and a "v" icon.
- A "<Replace>" button.
- A row of buttons: Enter, Execute, Menu, List, hpterm, Help, Show, Exit/, Accept/.
- A second row of buttons: Query, Query, Keys, Cancel, Commit.

Abbildung 7.1. Screenshot einer Bildschirmmaske zum Eingeben, Suchen oder Ändern von Datenbanksätzen, erzeugt mit Oracle Forms

Ein Benutzer schreibt oder liest Daten einer Datenbank selten direkt mittels SQL-Befehlen. Auf dem Bildschirm arbeitet er mit Masken wie in Abbildung 7.1 dargestellt. Zur Erzeugung einer solchen Maske sind offensichtlich festzulegen:

- Koordinaten (Zeile, Spalte) der Feldanfänge,
- Länge der Felder (maximale Anzahl der Zeichen, die größer sein kann als die Anzahl der angezeigten Zeichen),
- Beschriftung,
- Typ der Felder (Zeichenkette, Zahl, Datum),
- Namen der Felder bzw. der zugehörigen Variablen (die meist von der Beschriftung abweichen).

Bei einer Eingabemaske wie abgebildet sind alle Felder bis auf die automatisch vergebene laufende Nummer zunächst leer und schreibbar. Bei einer Änderungsmaske sind zumindest die obligatorisch zu füllenden Felder besetzt und alle Felder mit Ausnahme der laufenden Nummer änderbar. Bei einer Suchmaske können in alle Felder zu suchende Zeichenketten eingegeben werden, kein Feld wird jedoch im Datensatz geändert. Diese Eigenschaften in Verbindung mit Defaultwerten und Privilegien bilden die Attribute eines Feldes. Es können auch als Eingabe in ein Feld nur Werte aus einer anderen Tabelle oder Werte in einem bestimmten Bereich zugelassen sein. Hinter einer Maske verbergen sich oft eine Menge von Beziehungen oder Bedingungen, die unsinnige Werte verhindern, zusätzlich zu den Bedingungen, die die Datenbank selbst stellt.

Auswertungen, die über eine einfache Suche hinausgehen, werden in Form von Berichten oder Reports auf dem Bildschirm oder auf Papier dargestellt. Den Kern von Reports bilden die `select`-Anweisungen in SQL-Skripten, aber es gehört etwas mehr dazu. Die Reports müssen drucktechnisch aufbereitet werden (Überschrift, Seitenzahlen, Papierformat, Kommentar). Eine häufige Kombination sind Shellskripte, die SQL-Skripte aufrufen. Das Schreiben von Masken oder Reports macht vor allem am Anfang einen beträchtlichen Teil der Arbeit des Datenbankverwalters aus. Später kopiert und editiert er vorhandenes Material. Ein entsprechend aufbereiteter Report kann einem Webserver als HTML-Dokument (Webseite) übergeben werden, wenn man eine direkte Verbindung vom Webserver zur Datenbank vermeiden will.

Zu Office-Anwendungen (siehe Abschnitt 12 *Office-Pakete* auf Seite 405) gehören unter anderem benutzerfreundliche Frontends zu Datenbanken. Damit schiebt sich zwar eine weitere Softwareschicht zwischen Benutzer und Daten, aber das Look and Feel wird unabhängig von der jeweiligen Datenbank, sodass auch mehrere unterschiedliche Datenbanken von einem Frontend aus benutzt werden können. Ein Datenbankverwalter zieht den direkten Kontakt vor.

Beim Betrieb einer relationalen Datenbank sollte man für vier Aufgaben Zeit zum Nachdenken einplanen:

- die Gestaltung (Definition) der Tabellen,
- die Gestaltung der Eingabemasken,
- das Schreiben der Skripte oder Programme für die häufigsten Auswertungen,
- das Backup.

Von der Gestaltung der Tabellen hängt alles Weitere ab, weil sie am Anfang stehen. Eine Tabelle lässt sich zwar leicht um unwesentliche Spalten erweitern, ein Umkrempeln der Tabelle bedeutet jedoch Schreiben einer neuen Tabelle und Export der Daten von der alten in die neue Tabelle. Die Gestaltung der Masken ergibt sich größtenteils aus der Definition der Tabellen, aber der Datenbankverwalter hat an die Ergonomie, an unerfahrene Benutzer und an die Aufteilung der Prüfungen auf Maske und Datenbank zu denken. Die Auswertungen sind nicht kritisch, weil sie an der Datenbasis nichts ändern. Geht eine Auswertung daneben, schreibt man sich eine bessere. Das Schreiben von Skripten für einfache Auswertungen lernt man in Minuten, aber man kann mit einer Datenbank auch raffinierte Dinge machen. Die Daten in einer Datenbank stellen oft einen beträchtlichen Wert dar und sind für manchen Betrieb lebenswichtig. Deshalb sollte der Datenbankverwalter einen Plan zum Sichern und Wiederherstellen der Datenbank entwickeln, ausprobieren und befolgen. Datenbanken bringen Werkzeuge für diesen Zweck mit.

Ein Wiki ist eine strukturierte Ansammlung von Webseiten, auf welche die Benutzer mittels eines Browsers sowohl lesend wie auch schreibend zugreifen. Die Mitgestaltung durch die Benutzer oder Leser ist das wesentliche Merkmal eines Wikis. Ein großes Beispiel ist die schon mehrmals erwähnte Wikipedia, aber auch eine kleine Firma oder ein örtlicher Verein können einen nicht-öffentlichen Wiki betreiben, um das gemeinsame Wissen in eine Form zu bringen und zugänglich zu machen.

Ein Verzeichnis im allgemeinen Sinn ist eine geordnete Sammlung gleichartiger Daten. Beispiele sind:

- Telefonbücher, geordnet nach Ort und Namen,
- Radioprogramme, geordnet nach Sender und Zeit,
- Bibliothekskataloge, geordnet nach Sachgebiet, Autorennamen und Erscheinungsjahr,
- Dateiverzeichnisse, geordnet nach Unterverzeichnis- und Dateinamen,
- Verzeichnisse von IP-Adressen und zugehörigen Hostnamen (`/etc/hosts`),
- Benutzerverzeichnisse, geordnet nach Domänen und Namen,
- Linksammlungen (Webkataloge), geordnet nach Sachgebiet, Sprache usw.

Ein Merkmal solcher Verzeichnisse ist der hierarchische Aufbau, wie wir ihn vom Linux/UNIX-Dateisystem her kennen. Damit ähnelt ein Verzeichnisdienst einer hierarchischen Datenbank. Zu jedem Eintrag gehören bestimmte Attribute, in einem Telefonbuch beispielsweise zu einem Namen die Anschrift und die Telefonnummer. Die Dienstleistung, der Service, besteht darin, einem Benutzer oder Programm Auskunft zu erteilen. Ein Verzeichnisdienst kann auf eine Datenbank zurückgreifen, seine Daten aber auch selbst halten. Die Größe der Verzeichnisdienste reicht von lokal bis weltumspannend. Ein auf das lokale Netz begrenzter Verzeichnisdienst ist der Network Information Service (NIS), ein weltumspannender der Domain Name Service (DNS).

Ein großer Verzeichnisdienst (E: directory service) wie DNS kann als eine im Netz verteilte hierarchische Datenbank angesehen werden. Tatsächlich werden auch beide Bezeichnungen verwendet. Ein oder mehrere Server halten die Daten bereit

und stellen sie anfragenden Clients zur Verfügung. Dazu gehört ein Protokoll. Typischerweise überwiegt die Anzahl der lesenden Zugriffe. Beispiele sind:

- Kommerziell:
 - Novell Directory Services.
 - MS Windows Active Directory,
 - Oracle Internet Directory,
- offen oder frei:
 - Network Information Service (NIS),
 - X.500, ein Standard, kein Produkt,
 - Lightweight Directory Access Protocol, ebenfalls ein Standard, realisiert beispielsweise in OpenLDAP.

Schon das `finger`-Kommando in Abschnitt 4.2.7 *Suchen nach Benutzern* auf Seite 130 in Verbindung mit dem `fingerd`-Dämon bietet einen bescheidenen Verzeichnisdienst.

Ein Expertensystem stellt das Wissen – was mehr als nur Fakten bedeutet – von Fachleuten (Experten) einer offenen oder geschlossenen Benutzergruppe zur Verfügung. Eine typische Anwendung sind Analysen oder Diagnosen. Eine Benutzer hat eine ihm unbekannte Pflanze vor sich oder ein Problem mit seinem Rechner. In Frage und Antwort mit einem Expertensystem gelangt er schließlich zum Namen der Pflanze oder zur Lösung des Problems. Das System kann aus der Rückmeldung über Erfolg oder Misserfolg lernen und so sein Wissen (Erfahrung) vermehren. Expertensysteme gehören in das etwas umstrittene Gebiet der Künstlichen Intelligenz (KI, E: artificial intelligence, AI, F: intelligence artificielle). Nach einem Höhepunkt um 1990 herum ist es in jetziger Zeit etwas ruhiger um Expertensysteme geworden. Die Wunderwaffe gegen Armut, Dummheit und Pestilenz sind sie nicht, aber bei manchen Aufgaben leisten sie handfeste Arbeit.

7.2 Einfache Datenbank mit `awk`, `grep` und `sort`

Eine einfache Datenbank im weitesten Sinn, die zur Verwaltung eines Vereins mit hundert Mitgliedern ausreicht, kann man sich mit Hilfe klassischer Linux/UNIX-Werkzeuge zusammenstellen. Dazu braucht man nicht auf eine richtige Datenbank mit SQL zurückzugreifen. Folgende Textwerkzeuge werden benötigt:

- ein Editor (`vi` oder ein anderer) zum Eingeben der Daten,
- ein Listengenerator (`awk`) zum Auswerten,
- ein Sortierer (`sort`),
- ein Suchwerkzeug (`grep`),
- ein Filter (`sed`) zum Bearbeiten von Ausgaben,

siehe die jeweiligen Abschnitte im Kapitel 5 *Textverarbeitung* auf Seite 155. Mit ein paar Shell- oder Perl-Skripten verbunden lassen sich so Aufgaben wie das Schreiben von Mitgliederlisten, Briefen, Formularen, Mahnungen oder Urkunden bewältigen. Mit Hilfe der `curses`-Funktionen (siehe Abschnitt 1.3.1 *Kommandozeile, curses*-

Fenster auf Seite 9) können sogar einfache Bildschirmmasken zum Eingeben und Suchen gestaltet werden. Als Trennzeichen zwischen den Datenfeldern bietet sich das TAB-Zeichen an, aber auch das Komma ist gebräuchlich (comma separated values, CSV), als Trennzeichen zwischen den Datensätzen der Zeilenwechsel (newline).

7.3 Berkeley Database

Die Berkeley Database (Berkeley DB) ist keine fertige Datenbank, sondern ein Bausatz, eine Funktionsbibliothek, die von vielen Werkzeugen und Programmiersprachen verwendet wird. Die Quelle ist <http://www.sleepycat.com/>. In den Debian-Paketlisten sucht man am einfachsten nach *Berkeley* und *Database*. Im GNU-Projekt ist eine ähnliche Bibliothek unter dem Namen *gdbm* verfügbar. Bei Debian finden sich einige Pakete dazu, die in ihrem Namen die Zeichenkette *gdbm* enthalten, größtenteils erst ab *sarge*. Der gewöhnliche Benutzer hat mit dieser Software direkt nichts zu tun, sie wird von anderen Debian-Paketen verlangt.

7.4 PostgreSQL

7.4.1 Pakete und Dateien

Die objektrelationale¹ Datenbank PostgreSQL stammt ursprünglich aus der University of California at Berkeley wie manche andere UNIX-Software und ist auf <http://www.postgresql.org/> bzw. <http://www.postgresql.de/> beheimatet. Dort sind auch einige Bücher sowie Manuals und FAQs online verfügbar, darunter die *PostgreSQL 8.0.0 Documentation* im Umfang von 1300 Seiten oder 10 MB und eine deutschsprachige Einführung von 200 Seiten. Das *PostgreSQL HOWTO* ist zur Zeit zwecks Bearbeitung zurückgezogen. Die meisten Debian-PostgreSQL-Pakete sind in der Abteilung *Miscellaneous* zu finden:

- *postgresql*, die eigentliche Datenbank,
- *postgresql-client*, benutzerseitige Programme (Front-Ends),
- *postgresql-contrib*, verschiedene Hilfsprogramme,
- *postgresql-doc*, die Dokumentation,
- *odbc-postgresql*, aus der Abteilung *libs*, ODBC-Zugriff auf die Datenbank,
- *webmin-postgresql*, aus der Abteilung *admin*, Webmin-Zugriff auf die Datenbank,

Nach Einrichtung und Konfiguration sollte im Verzeichnis `/etc/init.d` eine Startdatei `postgresql` zu finden sein, die ein paar Kommentarzeilen für den Verwalter enthält. Im Verzeichnis `/etc/rc2.d` oder je nach Runlevel einem anderen sollte ein Link wie `S20postgresql` gelegt sein. Unter den Prozessen erscheinen drei mit der Zeichenkette `postgres` im Namen. Die Datenbank – der `postmaster`-Prozess – läuft. Folgende Dateien oder Verzeichnisse gehören zur Datenbank:

¹ ein Modewort, nicht zu ernst nehmen.

- `/etc/postgresql/` enthält die Konfigurationsdateien
- `/var/lib/postgres/data/` enthält die Datenbasis und weitere Dateien
- `/var/log/postgres.log` ist die Protokolldatei
- `/usr/lib/postgresql/` enthält den Programmcode und die Dokumentation

Die Datenbank-Prozesse gehören dem Datenbankverwalter namens `postgres`, nicht `root`. Das wäre ein Sicherheitsrisiko. De facto ist der Systemverwalter am Anfang aber auch der Datenbankverwalter.

7.4.2 Benutzer und Privilegien

Die Datenbank hat eine eigene, vom Betriebssystem unabhängige Benutzer- und Gruppenverwaltung. Dies ist erforderlich, weil die Rechte an den Daten weit feiner abgestuft sind als unter Linux/UNIX. Das Benutzerkonzept ist praktisch das gleiche wie unter Linux/UNIX, nur dass der Benutzer hier Rechte an Tabellen statt an Dateien hat. Es spricht nichts dagegen, für die Datenbank dieselben Benutzernamen wie auf dem System zu verwenden. Benutzer mit gleichen Rechten – Privilegien genannt – werden zu Gruppen zusammengefasst, um die Rechtevergabe zu vereinfachen. Statt in der Datei `/etc/passwd` sind die Benutzer in der Systemtabelle `pg_user` in der Datenbank aufgelistet.

Genau wie für ein Linux/UNIX-System gibt es auch für die Datenbank einen allmächtigen Superuser, der hier nicht `root`, sondern `postgres` heißt und zu einer eigenen, gleichnamigen Gruppe gehört. `Postgres` ist ein Linux/UNIX-Benutzer und daher von `root` anzulegen. Wenn die Datenbank läuft, ist seine vornehmste Aufgabe, weitere Datenbankbenutzer und -gruppen anzulegen (`create user`) und ihnen Rechte zu gewähren (`granting privileges`) oder auch wieder wegzunehmen (`revoking`). Rechte besitzt ein Datenbank-Benutzer an:

- Tabellen,
- Ansichten (`view`),
- Funktionen,
- Sequenzen, das sind laufende Nummern.

Die wichtigsten Rechte von insgesamt elf an einem Datenbankobjekt sind:

- auswählen, lesen (`select`),
- ein- oder hinzufügen, schreiben (`insert`),
- ändern/löschen (`update`, `delete`)

Der Schöpfer einer Tabelle ist zunächst und auch oft auf Dauer ihr Besitzer und verfügt über alle Rechte an ihr, nicht anders als bei der Erzeugung einer Datei. Er kann anderen Datenbankbenutzern Rechte einräumen:

```
joe@debian:~$ GRANT SELECT ON buercher TO GROUP leser;
```

Er kann auch ausgewählten Benutzern das Recht übertragen, ihrerseits Rechte weiterzugeben, was in einem Linux/UNIX-Dateisystem nicht vorkommt.

Da die Datenbank in Form von Dateien auf einem Linux/UNIX-System läuft, könnte root dem Datenbankverwalter ins Handwerk pfuschen. Das gäbe mächtig Ärger. Der Systemverwalter hat nach der ersten Einrichtung der Datenbank und ihres Superusers nichts mehr in ihr verloren.

7.4.3 Tabellen

Der erste Schritt eines Datenbank-Benutzers mit ausreichenden Rechten ist das Anlegen einer Tabelle. Hierzu muss man sich überlegen, welche Spalten oder Felder die Tabelle umfassen soll. Zwei Gesichtspunkte dabei sind:

- Keine Information doppelt (redundant) ablegen, weil das Speicherplatz kostet und nur zu Zweifeln führt, welche Information gültig ist. Damit sind nicht Backups gemeint, von denen hat man selten zu viele.
- Änderungen erleichtern. Beispielsweise ist in einer Personentabelle damit zu rechnen, dass eine Person ihren Familiennamen wechselt. Kommt der Familienname an mehreren Stellen vor, ist die Änderung erschwert. Der Name ist an einer Stelle maßgeblich (autoritativ) abzulegen, an weiteren Stellen sind nur Bezüge (Referenzen) darauf zu verwenden.

Bei einem Bücherkatalog ist an Autor, Titel, Verlag, Erscheinungsjahr, ISBN, Anschaffungsdatum und eine formlose Bemerkung zu denken. Jedes Feld gehört einem aus rund drei Dutzend Typen an. Damit lautet die SQL-Anweisung zur Erzeugung der Tabelle:

```
create table buecher
(autor   varchar(32),
 titel   varchar(255) NOT NULL,
 verlag  varchar(64),
 jahr    char(4),
 isbn    char(13),
 andat   date,
 bemerk text);
```

Die Zeilen werden eingegeben, nachdem man mit dem Kommando `psql` ein interaktives Terminal zur Kommunikation per Kommandozeile geöffnet hat. Einzelheiten sind mittels `man psql` oder in der Dokumentation zur Datenbank nachzulesen. Komfortablere Werkzeuge (Front-Ends) sind `pgaccess` (<http://www.pgaccess.org/>, verlangt Tcl/Tk) und `pgadmin` (<http://www.pgadmin.org/>, unter dem Paketnamen `pgadmin3` bei Debian erst in *unstable*).

Die Felder Autor, Titel und Verlag sind Zeichenketten variabler Länge, deren maximale Länge durch die Zahl in Klammern festgelegt wird. Das Feld Titel darf nicht leer gelassen werden, ein Buch ohne Titel ist wie ein Mensch ohne Namen. Eine andere Bedingung (E: constraint, F: contrainte) wäre, dass ein Feldinhalt einmalig sein muss, was bei laufenden Nummern der Fall ist. Die Felder Jahr und ISBN sind Zeichenketten fester Länge. Das Jahr besteht zwar aus Ziffern, mit dem Feld wird aber nicht gerechnet, sodass hier ein Zahlentyp ebenso unsinnig wäre wie bei Telefonnummern. Das Datumsformat nimmt Werte von 4713 vor Chr. bis 32757

nach Chr. an, das Y10K-Problem ist bereits gelöst, vergleiche den RFC 2550 vom 01-04-01999. Das Feld für Bemerkungen hat einen Texttyp beliebiger Länge. Im Unterschied zu den anderen Zeichentypen kann nach dem Inhalt dieses Feldes nicht sortiert oder gesucht werden.

Nach Erzeugung der Tabelle werden Datensätze eingegeben, in der Praxis mit Hilfe von Skripten, nicht mit einzelnen Kommandozeilen:

```
insert into buecher
values ('Ganten, P. H.', 'Debian GNU/Linux',
       'Springer, Berlin, Heidelberg',
       '2000', '3-540-65841-6', '2003-02-01',
       'Neue Auflage erscheint 2004');
```

Die Daten werden mit SQL-Anweisungen folgender Art ausgewertet:

```
select * from buecher
where autor = 'Ganten, P. H.'
order by andat;
```

Die Anweisung veranlasst die Datenbank, alle Spalten außer den Systemspalten der Tabelle `buecher` auszugeben, in denen im Feld `autor` die Zeichenkette `Ganten, P. H.` steht, und zwar nach dem Anschaffungsdatum geordnet. Die Möglichkeiten bei der `select`-Anweisung gehen viel weiter; hier kann nur ein Eindruck vermittelt werden.

Aus Spalten und Zeilen wirklich vorhandener Tabellen werden scheinbare Tabellen (Ansichten, Views) erzeugt, die sich dem abfragenden Benutzer wie jede andere Tabelle darbieten. Hinter einer View lassen sich komplexe `select`-Anweisungen verstecken, die erst bei Bedarf ausgeführt werden. Da die View nicht wirklich existiert, können in ihr keine Zeilen eingefügt oder gelöscht werden. Views sind eine sehr praktische Einrichtung.

Der Umfang der Datenbank ist unbegrenzt, eine Tabelle darf 16 Terabyte groß werden, sie kann maximal 1600 Spalten mit je maximal 1 GB umfassen. Large Objects (Bilder, Sound und dergleichen) bis zu einer Größe von je 2 GB werden innerhalb der Datenbank, aber außerhalb der zugehörigen Tabelle gespeichert. Damit ließe sich die pdf-Datei des vorliegenden Buches (rund 12 MB) leicht als Teil eines Datensatzes unterbringen.

7.4.4 Data Base Connectivity

Will man eine Datenbank im Netz verwenden (Anwendung und Datenbank auf getrennten Maschinen), gibt es vier Wege:

- Die Datenbank selbst bietet Werkzeuge an, die nach dem Client-Server-Modell arbeiten. Der Weg ist datenbank-spezifisch. Jede bessere Datenbank beherrscht diese Technik.
- Man arbeitet unter X11. Datenbank-spezifisch, nicht wirklich elegant.
- Man verwendet Webseiten als benutzerseitige Schnittstelle. Datenbank-spezifisch, Webserver mit cgi-Skripts auf dem Datenbank-Host erforderlich.

- Die Anwendung (der Client) verwendet ODBC-Funktionen, die Datenbank ist ODBC-fähig (es existiert ein ODBC-Treiber zur jeweiligen DB). Dasselbe gilt für die Java Data Base Connectivity (JDBC). Ähnlich wie Nr. 1, jedoch datenbank-unabhängig.

PostgreSQL bringt eine JDBC-Schnittstelle (Java) mit, eine ODBC-Schnittstelle kommt von dritter Seite. Die *Open Data Base Connectivity* (ODBC) ist eine von Microsoft entwickelte Schnittstelle zwischen beliebigen Anwendungen und Datenbanken. Für die Programmierung der Anwendungen stehen ODBC-Funktionsbibliotheken zur Verfügung. Da sich die ODBC-Funktionen an den SQL-Standard halten, sind sie mit fast jeder gängigen Datenbank verwendbar. Bei einem Wechsel der Datenbank können die ODBC-Anwendungen im günstigsten Fall unverändert weiter benutzt werden. Häufig möchte man aus einer Anwendung mehrere unterschiedliche Datenbanken ansprechen, was ohne ODBC oder JDBC aufwendig wird.

Die vorstehenden Zeilen sollen dem Benutzer die Scheu vor einer Datenbank nehmen. Der Einstieg ist einfach, alle Möglichkeiten einer Datenbank nutzen nur wenige aus. An die Grenzen einer richtigen Datenbank – siehe oben – stößt man nicht so schnell wie bei kleinen Lösungen. Was man beim Betrieb einer Datenbank der Mittelklasse lernt, lässt sich zu 90 % auch auf große Datenbanken wie Oracle oder IBM DB2 übertragen, falls man beruflich damit zu tun hat. Um das Nachschlagen in der Dokumentation kommt niemand herum.

7.5 MySQL und MaxDB

Die MySQL-Pakete sind der Abteilung *Miscellaneous* der Debian-Distribution zu finden. Wer nicht von einer der großen Datenbanken (IBM DB2, Oracle etc.) her kommt, infolgedessen auch nicht an SQL gewohnt ist und die Datenbank vor allem zur Fütterung eines Webserver einsetzen will, ist mit MySQL gut bedient. Viele Webserver sind auf die Zusammenarbeit mit MySQL vorbereitet, wenngleich PostgreSQL aufholt. Kommerzielle Webhoster bieten fast immer MySQL an. MySQL ist vor allem schnell und auf Lesezugriffe hin optimiert. Die maximale Größe einer Tabelle beträgt gegenwärtig 8 Millionen Terabytes, abhängig vom Betriebssystem, aber das ist eine Zahl, die für die meisten Leser jenseits des Bedarfs und des Vorstellungsvermögens liegt.

Die Datenbank wird von der schwedischen Firma MySQL AB im schönen Uppsala gepflegt und steht unter der GPL. Die Schweden vertreiben auch die große Datenbank MaxDB ex SAP DB ex Adabas, ebenfalls unter der GPL verfügbar, bei Debian sowie als Tarball und als RPM-Paket beim Hersteller. Die Firma lebt von Dienstleistungen und vom Verkauf kommerzieller Lizenzen. Weitere Informationen und eine reichhaltige Dokumentation – darunter ein englisches MySQL-Manual von 1500 Seiten und ein gut lesbares deutsches Handbuch von 700 Seiten – finden sich bei <http://www.mysql.com/> bzw. <http://www.mysql.de/>. Die wichtigsten Debian-Pakete für MySQL sind:

- *mysql-server-4.1*, die eigentliche Datenbank,
- *mysql-admin* und *mysql-admin-common*, eine grafische Oberfläche für den Datenbank-Verwalter,
- *mysql-client-4.1* und *mysql-common-4.1*., benutzerseitige Programme,
- *mysql-doc-4.1*, die Dokumentation,
- *webmin-mysql*, Webmin-Zugriff auf die Datenbank.

Dazu kommen rund drei Dutzend weitere Pakete mit Benutzer-Frontends, für die Zusammenarbeit mit anderen Programmen oder Programmiersprachen sowie Bibliotheken. Insgesamt erscheint MySQL weiter entwickelt als PostgreSQL, was noch kein Qualitätsurteil bedeutet. Bei kleinen Datenmengen, wo Geschwindigkeit keine Rolle spielt, kann die engere Bindung an den SQL-Standard bei PostgreSQL ein Argument sein. MySQL verfügt über ODBC- und JDBC-Treiber. Der Rest ist nicht anders als bei jeder relationalen Datenbank und am einfachsten in dem erwähnten deutschen Handbuch nachzulesen.

7.6 Wikis

Ein Wiki (auch WikiWiki oder WikiWeb) ist ein Wissens-Management-Werkzeug im öffentlichen oder in einem geschlossenen Web, das allen Teilnehmern zum Lesen *und* Schreiben offen steht. Das erste Wiki wurde um 1995 von WARD CUNNINGHAM für die Entwicklung von Software geschaffen. Als Datenbank kann es nur im weitesten Sinn bezeichnet werden. Verwandtschaftliche Beziehungen bestehen zu Content Management Systemen, siehe Abschnitt 11.12 *Content Management* auf Seite 400. Kommunikation gehört auch dazu, aber nicht vorrangig wie in den Netnews. Wir haben mit den Wikis wieder ein Beispiel dafür, dass die neuen technischen Möglichkeiten des Netzes zu neuen Formen der Wissensspeicherung und -vermittlung führen, die nicht in die herkömmlichen Schubladen passen.

Ein Wiki eignet sich dazu, Wissen in Form von Text, ergänzt durch Grafik oder auch multimediale Zutaten (alles, was das Web verkraftet), einem Publikum zur Verfügung zu stellen und durch dieses Publikum erweitern zu lassen. Dazu muss irgendwo ein Wiki-Server laufen; als Client dient jeder Web-Browser. Wiki-Server (Wiki engines) finden sich in den Debian-Paketen *aswiki*, *didiwiki*, *ewiki*, *hiki*, *jsp-wiki*, *kwiki*, *moinmoin*, *moniwiki*, *twiki*, *usemodwiki*, *wiliki* sowie als Zusatz zu Apache, eGroupWare, Emacs, phpGroupWare und Zope. Betreibt man einen Apache-Webserver, bietet sich zum Ausprobieren der Apache-Miniwiki an, geschrieben in Perl; wer keinen Webserver im Keller hat, bekommt mit *didiwiki* ein kleines, selbstständiges Wiki einschließlich Webserver, geschrieben in der Programmiersprache C. Die bei Miniwiki mitgelieferte Seitenvorlage *template.html* verwendet JavaScript und missbraucht eine Tabelle zum Seitenlayout, siehe Abbildung 7.2. Man kann sie als Einstieg verwenden, sollte sie aber nicht als Vorbild ansehen. Zum Einrichten und Verwalten eines Wikis gehören HTML-Kenntnisse, zur Auszeichnung der Einträge wird ein stark vereinfachtes HTML verwendet, zu dessen Erlernen ein paar Vorlagen ausreichen. In der deutschen Wikipedia finden sich Anregungen unter dem Suchwort *Formatvorlage*. Auf einem kleinen Wiki kommt man mit weniger aus.



Abbildung 7.2. Screenshot des Apache-Zusatzes Miniwiki, Seite entwickelt aus der mitgelieferten Vorlage (template.html)

Ein Beispiel – und zwar das weltweit umfangreichste – ist die schon mehrmals erwähnte, im Jahr 2001 gegründete Wikipedia mit über einer Million Artikeln, Seiten oder Einträgen in 100 Sprachen, darunter natürlich auch ein Eintrag zum Begriff *Wiki*. Betrachten wir den Werdegang eines Eintrags. Anno 2003 fanden sich in der deutschen Wikipedia unter dem Begriff *Weingarten* nur zwei Erklärungen, nämlich zum einen der Weingarten als Rebanlage und zum anderen Weingarten als Stadt in Württemberg. Dem galt es abzuweichen. So kamen kurze Einträge zu weiteren Orten namens Weingarten hinzu. Ein erfahrener Wikipedianer gestaltete bald den Eintrag zu einer *Begriffsklärung* um, das ist eine Seite, auf der die verschiedenen Bedeutungen eines Wortes aufgelistet und zu Einzelseiten verlinkt sind, ein gebräuchliches und sinnvolles Vorgehen. Damit wurde eine Einzelseite zu *Weingarten (Baden)* fällig. Im Laufe der Jahre kamen weitere Orte zu der Begriffsklärung sowie mit Einzelseiten hinzu, auch Personen dieses Namens. Die Einzelseite zu *Weingarten (Baden)* gedieh dank der Beiträge anderer Autoren prächtig. Aus einem zarten Pflänzchen entwickelte sich so ein kräftiger Baum, zur Freude aller Beteiligten und zum Nutzen aller Leser. Der erste Beitrag des Weingartener Wikiwichtels galt übrigens dem *Lake Peipsi* in der englischen Wikipedia. Der Beitrag ist mittlerweile mehrfach ergänzt worden. Oft ist es so, dass erst einmal ein paar Sätze auf dem Schirm stehen müssen, dann finden sich andere, die ihre 2 Cent beitrugen. Schreiben darf jeder, auch ohne eingetragener Wikipedianer zu sein. Gegen Vandalen schützt eine Versionskontrolle mit der Möglichkeit der Wiederherstellung von Seiten. Ein so großes Wiki wie die Wikipedia bedarf der Pflege durch einige Redakteure, aber diese könnten nie alle Inhalte schaffen. Einige Wikis außerhalb der Computerei sind:

- JuraWiki <http://www.jurawiki.de/>,
- Wikitravel <http://www.wikitravel.org/>,
- das Wiki des Institutes für Bibliothekswissenschaften der Humboldt-Universität <http://wiki.ib.hu-berlin.de/>,
- Versicherungs-Wiki <http://versicherungs-wiki.de/>,
- Tolkien-Wiki <http://www.thetolkienwiki.org/>,
- ein Wiki mit britischem Humor <http://www.wikihumor.com/>.

Es gibt Firmen, die leere Wikis vermieten so wie andere Provider Webspaces oder Briefkästen.

7.7 Verzeichnisdienste

7.7.1 Whois-Verzeichnisdienst

Whois ist ein weltweiter Verzeichnisdienst, der Auskünfte über Domänen und IP-Adressen erteilt, ein Verwandter des Domain Name Systems (DNS). Beschrieben wird der Dienst im RFC 3912. Im Gegensatz zum DNS, das hauptsächlich für die Maschinen da ist, richtet sich Whois an menschliche Fragesteller. Eine Abfrage kann aus der Kommandozeile gestartet werden:

```
joe@debian:~$ whois -h whois.internic.net
gatekeeper.dec.com
```

oder auch:

```
joe@debian:~$ whois -h whois.denic.de
alex-weingarten.de
```

wobei nach der Option `-h` ein Whois-Server zu benennen ist. Eine Hilfe erhält man durch Eingabe eines Fragezeichens als zweitem Argument. Bequemer ist der Zugang über das Web. Einige Whois-Server sind:

- [whois.denic.de](http://www.denic.de/), <http://www.denic.de/>,
- [whois.ripe.net](http://www.ripe.net/), <http://www.ripe.net/>,
- [whois.internic.net](http://www.internic.net/whois.html), <http://www.internic.net/whois.html>.

Der Whois-Dienst ist aus Gründen des Datenschutzes umstritten. Die Auskünfte lassen sich leicht zum Spammen missbrauchen. Andererseits ist ein Netz ohne ein Minimum an Meta-Informationen kaum zu betreiben. Man stelle sich das Telefonnetz ohne Telefonbuch vor.

7.7.2 Lightweight Directory Access (LDAP)

Das Lightweight Directory Access Protocol (LDAP) ist nicht nur ein Protokoll zum Datenaustausch zwischen LDAP-Clients und LDAP-Servern, sondern auch ein Konzept zum Speichern und Vermitteln von Informationen über Objekte der realen Welt.

Es weist Ähnlichkeiten mit dem Domain Name System (DNS) im Internet auf, weniger mit einer relationalen Datenbank. Die Informationen sind hierarchisch strukturiert; das Protokoll ist auf Lesezugriffe hin optimiert. Unter Objekten (E: entity, F: entité) hat man sich Personen, Gruppen, Firmen, Länder, Dokumente, Rechner, Netze, Protokolle, etc. vorzustellen. Der Phantasie sind keine Grenzen gesetzt. Der Dienst ist im Prinzip global verteilt wie DNS, der Zugriff auf einen Server kann jedoch begrenzt werden. Ungeeignet ist der Dienst zum Verwalten dynamischer (schnell wechselnder) Informationen (Leihverkehr in Bibliotheken), weil dabei Schreibzugriffe einen beträchtlichen Teil der Arbeit ausmachen. Ursprünglich (1993) sollte LDAP nur einen einfachen Zugang zu dem schwergewichtigen OSI-X.500-Dienst schaffen; inzwischen ist aber ein eigenständiger Dienst im Internet daraus geworden, der das Zeug hat, verschiedene Verzeichnisdienste unter einem Zugang zu vereinen. Das Protokoll ist in RFC 2251 ff. definiert.

Die Debian-Distributionen enthalten im Paket *slapd* aus der Abteilung *net* den gleichnamigen Stand-alone-Server des OpenLDAP-Projektes (<http://www.openldap.org/>), ferner in weiteren Paketen aus verschiedenen Abteilungen Werkzeuge und Bibliotheken. An erster Stelle ist das Debian-Paket *ldap-utils* zu nennen, das man am besten gleich zusammen mit dem Server einrichtet, zusätzlich auf allen Arbeitsplätzen, die mit dem Server kommunizieren sollen. Das Paket bringt folgende Kommandos mit:

- `ldapadd` dasselbe wie `ldapmodify -a`,
- `ldapcompare` Vergleichen von Einträgen,
- `ldapdelete` Löschen von Einträgen,
- `ldapmodify` Hinzufügen neuer oder Ändern vorhandener Einträge,
- `ldapmodrdn` Ändern des *Relative Distinguished Name* (RDN) eines Eintrags, das heißt Umbenennen,
- `ldappasswd` Setzen oder Ändern des Passwortes eines LDAP-Benutzers,
- `ldapsearch` Suchen nach bestimmten Einträgen (häufigstes Kommando),
- `ldapwhoami` Hilfe bei Identitätskrisen.

Bei OpenLDAP selbst sind Tarballs verfügbar. Zwei Linux-LDAP-HOWTOs leisten Hilfe beim Einrichten eines lokalen LDAP-Servers (<http://www.tldp.org/HOWTO/LDAP-HOWTO/>, von 2004, und <http://www.linuxhaven.de/dlhp/HOWTO?DE-LDAP-HOWTO.html>, von 1999). Das Debian-Paket *gq* enthält einen gleichnamigen LDAP-Client, der GTK+ verwendet und mit schlichter Grafik daherkommt. Er ist noch nicht ganz wetterfest und gehört zu den wenigen Programmen, mit denen ich meine Sitzung (nicht den Rechner) zum Hängenbleiben bringen konnte. Zahlreiche Links zu LDAP finden sich unter <http://verzeichnisdienst.de/>.

Ein Kapitel des *OpenLDAP Administrator's Guide* von <http://www.openldap.org/> beinhaltet einen *Quick Start Guide*, dem die nachstehenden Zeilen folgen. Die Einrichtung des Servers aus dem Debian-Paket ist harmlos; einige einfache Fragen nach der lokalen Domäne und dem vorgesehenen Passwort des LDAP-Verwalters und dergleichen sind zu beantworten. An der von Debian vorgegebenen Konfiguration in `/etc/ldap/` schraubt man zunächst einmal nicht

herum. Die Einrichtung der *ldap-utils* ist trivial. Auf dem Server ist der *slapd* mit Hilfe des Startskripts aus */etc/init.d* zu starten, Erfolgskontrolle mit:

```
joe@debian:~$ ps -ef | grep slapd
```

Dann gibt man auf einem Arbeitsplatz im lokalen Netz eine Anfrage ein:

```
joe@debian:~$ ldapsearch -v -x -H ldap://server:389 -s
base '(objectclass=*)' namingContexts
```

Die Optionen und Argumente bedeuten:

- *-v* verbose, harmlos und für den Anfang immer richtig,
- *-x* einfache Authentifizierung (statt *Simple Authentication and Security Layer*, SASL),
- *-H . . .* URI des Hosts oder LDAP-Servers, für *server* ist dessen IP-Adresse oder Name im lokalen Netz einzusetzen; 389 ist der Default-Port,
- *-s base* scope, Geltungsbereich der Suche,
- *(objectclass=*)* alle Objektklassen anzeigen; in Hochkommas, damit sich die Shell nicht an dem Sternchen vergreift; ist Default und könnte daher entfallen,
- *namingContexts* welche Attribute angezeigt werden sollen.

Erhält man eine vernünftige, wenn auch nicht in allen Einzelheiten verständliche Antwort, funktionieren LDAP-Server und -Client. Wir fügen nun einen Eintrag hinzu und schreiben mit unserem Lieblingseditor eine Datei *neu.ldif* im *LDAP Data Interchange Format* (LDIF), das im RFC 2849 spezifiziert ist. Die Datei könnte so aussehen:

```
dn: dc=localdomain
objectclass: dcObject
objectclass: organization
o: Familie Alex
dc: localdomain

dn: cn=Manager, dc=localdomain
objectclass: organizationalRole
cn: Manager
```

Für *localdomain* ist der Name Ihrer lokalen Domäne einzusetzen, wie bei der Einrichtung des *slapd* angegeben. Statt *Familie Alex* sollten Sie den Namen Ihrer Organisation wählen. Der Rest ist vorgeben, die Bedeutungen sind im RFC oder im Guide nachzulesen. *dn* bedeutet *Distinguished Name*, der eindeutige Name eines LDAP-Objektes, zusammengesetzt beispielsweise aus Domain Components (dc). Dann geben wir hoffnungsvoll das Kommando:

```
joe@debian:~$ ldapadd -v -x -D
cn=admin,dc=localdomainw Passwort -f neu.ldif
```

Die Optionen und Argumente – soweit noch nicht erklärt – bedeuten:

- `-D ... Distinguished Name (Name)`, mit dem die Verbindung zum Server aufgebaut werden soll (schreiben darf vorläufig nur `admin`),
- `-w Passwort` Passwort des `admin`, wie bei der Einrichtung des `slapd` angegeben und hoffentlich einfallsreicher als *Passwort*,
- `-f ... File`, Datei, die wir soeben editiert haben.

Statt einer Datei könnte das Kommando auch von `stdin` lesen, aber für den Anfang und bei zahlreichen Eingaben ist der Weg über eine Datei sicherer. Nun testen wir unsere Eingabe:

```
joe@debian:~$ ldapsearch -x -v -H ldap://server -b
dc=localdomain
```

Mit der Option `-b` (searchbase) wird der Ausgangspunkt der Suche festgelegt. In der Antwort sollte man die Organisation wiederfinden. Wer schon mit Datenbanken gearbeitet hat, erkennt viele Konzepte wieder. Die Bezeichnungen und die Einzelheiten der Syntax muss man lernen.

Über ein SQL-Backend kann der LDAP-Server *slapd* auf SQL-Datenbanken zugreifen. Es ist schwierig zu entscheiden, welche Kombination aus Datenbank, LDAP-Server und Web-Server in einer gegebenen Situation die beste Lösung darstellt. Zum Glück kostet ein bisschen Experimentieren dank Debian und anderer freier Software kein Geld. Nach meinem Eindruck ist LDAP als Weiterentwicklung von `/etc/passwd` brauchbar, aber keine Datenbank für Alles. Schon die hierarchische Struktur passt nicht zu allen Datensammlungen. Eine nahe liegende und sinnvolle Anwendung von LDAP ist der Ersatz von NIS, was in den Bereich des Systemverwalters fällt. Hinweise dazu gibt der RFC 2307. Weitere Informationen und Open Source Software liegen auf <http://www.padl.com/>.

7.8 Expertensysteme (CLIPS)

Ein Expertensystem ist eine Software, die auf der Grundlage von gespeichertem Wissen Antworten auf Fragen oder Lösungen für Probleme findet. Ein botanisches Bestimmungsbuch (SCHMEIL-FITSCHEN, *Flora von Deutschland*) ist ein Expertensystem, dass mit Hilfe eines Entscheidungsbaums die Einordnung von Grünzeug ermöglicht. Während ein Baum bei botanischen Fragen nahe liegt, können bei anderen Problemen – Diagnose der Krankheiten von Lebewesen oder Maschinen – Sammlungen von Fällen oder Faustregeln bessere Ergebnisse zeitigen. Ein beliebtes Testobjekt sind einfache Spiele wie Mastermind, siehe Abschnitt 13.2 *Brettspiele*, Go auf Seite 414. Wir brauchen immer eine sachgebietsspezifische Wissensbasis sowie eine Maschine, die darauf operiert und Schlüsse zieht. Die ganze Thematik fällt in das Gebiet der *Künstlichen Intelligenz*, die noch nicht so ausgereift ist wie die *Natürliche Dummheit*. Einige URLs für den Einstieg in Expertensysteme sind:

- GNU/Linux AI & Alife HOWTO, z. B. auf <http://www.comnets.uni-bremen.de/doc/howtos/en/AI-Alife-HOWTO.html>,

- FAQ: Expert System Shells, z. B. auf <http://www.uni-giessen.de/faq/archive/ai-faq.expert.part1/>,
- Environment for Diagnostic Knowledge Systems, <http://www.d3web.de/> (Universität Würzburg),
- CLIPS (C Language Integrated Production System), <http://www.tnt.uni-hannover.de/soft/case/lang/clips/overview.html>,
- CLIPS – a Tool for Building Expert Systems, <http://www.ghg.net/clips/CLIPS.html>.

Der letzte URL führt auf die Startseite von CLIPS, das als gleichnamiges Debian-Paket unter *sarge* verfügbar ist. CLIPS wurde ursprünglich von der NASA entwickelt und ist verbreitet. Der Baukasten ist in C geschrieben; eine Implementation in Java von ERNEST J. FRIEDMAN-HILL heißt Jess und ist unbekannt verzogen.

Von CLIPS und Jess beeinflusst ist Lisa, geschrieben in Common LISP, zu Hause bei <http://lisa.sourceforge.net/> bzw. <http://sourceforge.net/projects/lisa/>, nicht bei Debian. Das gleichnamige Debian-Paket beinhaltet etwas anderes. Die Software kann als Tarball oder zip-Archiv heruntergeladen werden. Die Dokumentation ist noch mager.

Der Dritte in diesem Bunde ist EMA-XPS, beschrieben und erhältlich auf <http://www.iai.uni-wuppertal.de/EMA-XPS/>, eine hybride grafische Expertensystem-Shell, die X11 und Motif verwendet und eine ausführliche deutsche Anleitung mitbringt. Die genannten Systeme sind allesamt Werkzeugkästen, bestenfalls Shells, deren Wissensbasen noch zu füllen sind, genau wie eine Datenbank.

Das grafische Atelier

Dieses Kapitel führt in den Gebrauch einiger Werkzeuge zum Erzeugen und Bearbeiten von Grafiken und Fotos ein.

8.1 Grundbegriffe

8.1.1 Aufgaben

Der Mensch verarbeitet von klein auf grafische Daten (Bilder), den Umgang mit Zeichen erlernt er im Alter von einigen Jahren unter gewissen Mühen. Bei Rechnern ist es umgekehrt. Die ältesten Rechner haben Zeichen eingelesen und ausgegeben. Erst als die Hardware einen hohen Leistungsstand erreicht hatte, wurde Computergrafik volkstümlich. Wie schwer sich Rechner mit Grafik und Sound tun, erkennt man unter anderem daran, dass die automatische Indizierung von Grafikdateien (siehe Abschnitt 4.2.4 *Suchen nach binären Mustern in Dateien* auf Seite 125) immer noch Gegenstand der Forschung ist, während die Indizierung von Texten zum Alltag gehört. Die Aufgaben der grafischen Datenverarbeitung lassen sich in zwei Gruppen einteilen:

- die Erzeugung (Synthese) und anschließende Weiterverarbeitung grafischer Objekte (CAD, Finite Elemente, Simulationen ...),
- die Untersuchung (Analyse) und Bearbeitung grafischer Objekte, die außerhalb des Rechners entstanden sind (Fotoretusche, Mustererkennung, Bildanalyse ...).

Beide Gruppen lassen sich feiner unterteilen. Die Vielfalt grafischer Operationen ist weit größer als die der Operationen mit Texten. Entsprechend den unterschiedlichen Aufgaben sind zahlreiche Formate für Grafik-Dateien in Gebrauch. Dazu kommt wegen des Umfangs der Daten ein Interesse an Kompressionsverfahren, die – wie wir in Abschnitt 3.1.8 *Packen* auf Seite 82 gelesen haben – verlustfrei oder verlustbehaftet arbeiten können. Fast zwei Dutzend FAQs zu Themen der Verarbeitung grafischer

Daten findet man auf <http://www.faqs.org/faqs/graphics/>. Eine kurze Einführung in die Computergrafik bietet auch *SelfHTML* von STEFAN MÜNZ, bereits erwähnt im Abschnitt 5.12.2 *Hypertext Markup Language* ab Seite 224. Ein Besuch bei einschlägigen Hochschulinstituten wie dem Lehrstuhl Computergraphik der Universität des Saarlandes (<http://graphics.cs.uni-sb.de/>) kann auch lohnen, vor allem wenn es um spezielle Fragen geht.

8.1.2 Rastergrafik

Bei der Darstellung von Grafiken werden zwei Prinzipien verwendet: die Rastergrafik und die Vektorgrafik. Beide haben Schwächen und Stärken. Wir haben die Unterscheidung bereits in Abschnitt 5.1.2 *Fonts* auf Seite 162 kennen gelernt. Bei einer Rastergrafik werden die Objekte durch Muster aus Punkten dargestellt, die in einem rechtwinkligen Gitter angeordnet sind. Je mehr Punkte pro Längen- oder Flächeneinheit verwendet werden, desto feinere Einzelheiten lassen sich wiedergeben. Gebräuchliche Auflösungen sind:

- Bildschirme (CRT) 100 Punkte (Pixel) pro Zoll, das sind 40 pro cm,
- Laserdrucker 600 Punkte pro Zoll oder 240 pro cm,
- Flachbettscanner 1200 Punkte pro Zoll,
- Fotosatzbelichter, Trommelscanner 2000 bis 10000 Punkte pro Zoll.

Da viele Ausgabegeräte – Bildschirme, Drucker – nach dem Rasterprinzip arbeiten, verursachen Rastergrafiken bei der Ausgabe den geringsten Aufwand. Ein Nachteil ist die schlechte Skalierbarkeit. Wie kann ich ein Punktmuster bei gleicher Auflösung auf 80 % verkleinern oder auf 120 % vergrößern?

Eine Rastergrafik wird im einfachsten Fall beschrieben durch eine Matrix, deren Elemente Farbvektoren (Zahlengruppen) sind. Bei einer Schwarzweißgrafik bleibt von dem Farbvektor nur ein Bit übrig. Einige Raster-Dateiformate sind:

- Tagged Image File Format (TIFF),
- Graphics Interchange Format (GIF),
- Joint Photographic Experts Group (JPEG oder JPG),
- Portable Network Graphics (PNG).

8.1.3 Vektorgrafik

Eine Vektorgrafik beschreibt ein grafisches Objekt durch so genannte Primitive wie Punkt, Strecke, Kurvenabschnitt, Rechteck oder Ellipse, die mathematisch durch Zahlen und Gleichungen dargestellt werden. Eine Rastergrafik kann als Bitmap in eine Vektorgrafik eingebettet werden. Durch Ändern des Maßstabs lässt sich eine Vektorgrafik beliebig skalieren, ohne Informationen zu verlieren. Zur Ausgabe auf einem Rastergerät ist eine Vektorgrafik in eine Rastergrafik umzurechnen (Raster Image Processor, PostScript Engine), was für heutige Hardware keinen nennenswerten Aufwand bedeutet. Ein echtes Vektorgerät waren die Plotter früherer Zeiten, die Linien auf Papier zogen. Gebräuchliche Vektor-Dateiformate sind:

- PostScript (PS) und Encapsulated PostScript (EPS),
- Hewlett-Packard Graphic Language (HP-GL),
- Data Exchange Format (DXF) von AutoCAD,
- Scalable Vector Graphics (SVG).

Die Umwandlung von Rastergrafiken in Vektorgrafiken (vektorisieren, tracen) ist schwierig, aber möglich, umgekehrt geht es leichter (rastern). Das Vektorisieren wird oft in zwei Stufen aufgeteilt: Erzeugen der rohen, zusammenhanglosen Vektoren und Erkennen der zu einem Objekt gehörenden Vektoren. Zusätzliche Informationen über die Art der Vorlage (technische Zeichnung, Schaltplan, Landkarte) oder eine grafische Vorbehandlung der Abbildung (Kontrastschärfung, Setzen von Schwellen, Skelettierung) erleichtern die Arbeit und verbessern das Ergebnis. Ein allgemeiner Algorithmus zum Vektorisieren ist uns nicht bekannt. Der älteste Algorithmus zum Rastern geht auf JACK E. BRESENHAM im Jahr 1963 zurück. Er ermittelt diejenigen Rasterpunkte, die einer gegebenen Linie am nächsten liegen. Werkzeuge zum Umwandeln werden in Abschnitt 8.8.1 *Konverter* auf Seite 303 vorgestellt.

8.1.4 Farben

Mit Farbe wird die Welt schöner, aber auch komplizierter. Erste Voraussetzung ist annähernd weißes Licht¹, das – wie wir seit *Sir Isaac* wissen – eine Mischung von Licht aller Wellenlängen ist. Die Mischung wird durch die Farbtemperatur gekennzeichnet, die für gelblich-rötliches Kerzenlicht bei 1000 K (Kelvin) liegt, für neutrales Tageslicht bei 5000 bis 6000 K und für Sonnenlicht bei blauem Himmel im Hochgebirge über 10000 K. *Warmes* Licht hat also eine *niedrige* Farbtemperatur.

Die zweite Voraussetzung zum Farbsehen ist ein farbtüchtiger Lichtempfänger, unser Auge. Den dritten Beitrag zur Farbwahrnehmung liefert unsere biologische Datenverarbeitung, die schon in der Netzhaut beginnt und sich im Gehirn fortsetzt. Die Bemühungen, Farben zu messen (Colorimetrie), haben zu mehreren Farbmodellen geführt. Die wichtigsten Modelle gehen von drei Grundfarben aus, zu denen aus technischen Gründen gelegentlich noch Schwarz hinzukommt. Aus den drei oder vier Grundfarben werden alle Farben gemischt. Bei hochwertigen Drucken nimmt man gelegentlich noch weitere Farben hinzu, um schwierige Farbtöne (Haut) besser zu treffen.

Bei der additiven Farbmischung wird Licht dreier farbiger Lichtquellen überlagert oder addiert, sodass sich der Eindruck weißen Lichtes ergibt. Jeder Benutzer kennt das von Farb-Röhrenbildschirmen, auf denen weiß leuchtende Flächen dadurch entstehen, dass rote, grüne und blaue Bildpunkte gemeinsam leuchten. Mit einer starken Lupe sind die Bildpunkte einzeln zu erkennen. Rot, Grün und Blau sind die Grund- oder Primärfarben des Modells, das deshalb als RGB-Modell be-

¹Weißes Licht scheint es nicht zu geben, sondern nur die Normlichtart D65, die einem Tageslicht mit einer Farbtemperatur von 6500 K nahekommt. Einzelheiten in den DIN-Normen ab DIN 5030, in Physikbüchern unter dem Stichwort *Optik/Colorimetrie* oder bei der Physikalisch-Technischen Bundesanstalt <http://www.ptb.de/>.

zeichnet wird. Praktischerweise haben die englischen Namen der Farben dieselben Anfangsbuchstaben.

Entfernt man aus weißem Licht eine der Grundfarben oder mischt man nur zwei der Grundfarben, so erhält man die Farben Cyanblau (Blau + Grün oder Weiß – Rot), Magentarot (Blau + Rot oder Weiß – Grün) und Gelb (Yellow, Grün + Rot oder Weiß – Blau). Diese Farben sind die Grundfarben der subtraktiven Farbmischung. Beleuchten wir eine blaue Fläche mit weißem Licht, so schluckt die Fläche den roten und grünen Anteil, reflektiert wird nur Blau (Cyan). Da ein sattes Schwarz aus technischen Gründen auf diese Weise nicht hin zu bekommen ist, nimmt man im Vierfarbendruck noch Schwarz (E: Black) hinzu und gelangt zum CMYK-Farbmodell der subtraktiven Farbmischung. Das K steht für *key* wie Schlüsselfarbe.

Jetzt müssen noch die Farben durch Zahlen gemessen und in einem Koordinatensystem dargestellt werden, aber das führt hier zu weit. Dazu kommen noch Farbatlanten wie RAL (RAL 3000 = Feuerrot), Pantone (Pantone Matching System, PMS 280 = Marineblau) oder die von X11 verwendeten Namen (X Color Management System, #6495ED = cornflower blue). Das Umrechnen von beispielsweise einer Bildschirmfarbe auf eine Druckfarbe ist wegen der verschiedenen Modelle nicht trivial. Ein geübtes Auge unterscheidet rund eine Million Farbtöne. Mit einer Farbtiefe von 16 Bit entsprechend 65000 Farbtönen liegen wir nicht schlecht und mit 24 Bit entsprechend 16 Millionen Farbtönen auf der sicheren Seite. Als die Grafikkarten noch nicht so viel Speicher besaßen wie heute und man sich mit 256 Farben oder weniger begnügen musste, wurden Zwischenfarben durch Mischen von Pixeln aus den verfügbaren Farben gebildet. Aus hinreichendem Abstand betrachtet bemerkt das Auge die Einzelfarben nicht. Dieses *Dithering* genannte Verfahren wird auch heute noch beim Drucken verwendet, beispielsweise um mit den Farben Schwarz und Weiß Halbtonbilder (Graustufenbilder) zu erzeugen.

8.1.5 Rendern (Perspektive, Beleuchtung)

Unter Rendern von englisch *to render* = *wiedergeben* versteht man in der Computergrafik das Berechnen von mehr oder weniger realistisch wirkenden Bildern aus Rohdaten. Dazu sind vor allem zwei Forderungen zu erfüllen: eine richtige Perspektive und die Wirkung von Licht und Schatten.

Perspektive

Unter Perspektive wird die Abbildung räumlicher Objekte auf meist ebene Flächen verstanden, wobei das Gehirn wieder einen räumlichen Eindruck rekonstruiert. Dabei spielt die Geometrie (Darstellende G.) eine wichtige Rolle, aber auch Farbe und Auflösung tragen zum räumlichen Eindruck bei. Es gibt mehrere geometrisch korrekte Möglichkeiten der Perspektive. Global betrachtet ist auch die Kartenprojektion eine Art der Perspektive.

Beleuchtung

Ohne eine Lichtquelle könnten wir nur selbstleuchtende Objekte sehen, und die sind in der Minderheit. Selbst wenn wir uns auf die Sonne beschränken, so kann sie hoch oder nahe am Horizont stehen, vom Himmel brennen, leicht verschleiert oder gänzlich durch dichte Wolken abgeschirmt sein. Außer direktem Licht kann auch noch gespiegeltes oder gestreutes Licht zur Erleuchtung beitragen, besonders eindrucksvoll in hohen Breiten zu erleben, wenn die Sonne die ganze Nacht knapp unter dem Horizont steht, was bei klarem Himmel eine eigenartige, schattenlose Helle erzeugt. Zum direkten Licht gehört der Schatten, der für realistische Bilder unerlässlich ist.

Die Verfahren, die sich in der Computerei um Licht und Schatten kümmern, werden unter dem Begriff *Raytracing* (Strahlverfolgung, aber das deutsche Wort ist unüblich) zusammengefasst. Sie bedeuten für den Rechner Schwerarbeit, weshalb bei Flugsimulatoren und ähnlichen Echtzeitaufgaben oft auf Raytracing verzichtet wird. In Abschnitt 8.8.7 *Modellierer, Renderer* auf Seite 313 werden einige Raytracer für Linux kurz vorgestellt.

8.2 Zeichnen (xfig, inkscape, kseg, Z.u.L.)

Das Werkzeug `xfig` dient zum Anfertigen von Zeichnungen und setzt X11 sowie eine Maus oder eine Rollkugel voraus, möglichst mit drei Tasten. Zweckmäßig richtet man zugleich auch die Pakete `xfig-libs` und `xfig-doc` ein. Es bringt eine Manualseite und eine umfangreiche Dokumentation in `/usr/doc/share/xfig/` mit, größtenteils als html-Dateien. Dank des eigenen Übersetzungswerkzeugs `transfig` steht eine Vielzahl von Ausgabeformaten zur Verfügung. Der Wohnsitz der Werkzeuge ist <http://www.xfig.org/>.

Die Benutzung ist einfach. Man sollte sich die auf den Rändern der Zeichenfläche verteilten Buttons etc. ansehen, die eine kurze Erklärung aufleuchten lassen, wenn sich der Cursor über ihnen befindet. Freihandlinien gibt es nicht – `xfig` ist nicht für künstlerische Zeichnungen gedacht – jedoch Splines (glatte Kurven) durch vom Benutzer vorgegebene Punkte. Einfache Funktionen zur Längen- und Flächenmessung sind vorhanden, aber für eine normgerechte technische Zeichnung nicht ausreichend. Für diesen Zweck sind CAD-Programme besser, siehe Abschnitt 8.3 *CAD* auf Seite 297. Einfügen und Verschieben von Text ist möglich. Der Benutzer kann sich eine eigene Bibliothek von Zeichnungselementen anlegen.

Die Zeichnungen im Buch wurden mit folgenden Einstellungen angefertigt:

- mittleres Gitter (grid mode),
- Strichstärke (line width) 2 bis 4,
- Schrift Helvetica Bold, Größe 24 bis 30 Punkte,

Die Zeichnungen wurden im fig-Format gespeichert und zusätzlich im eps-Format exportiert. Aus den eps-Dateien wurden mittels GIMP jpg-Dateien erzeugt. Das

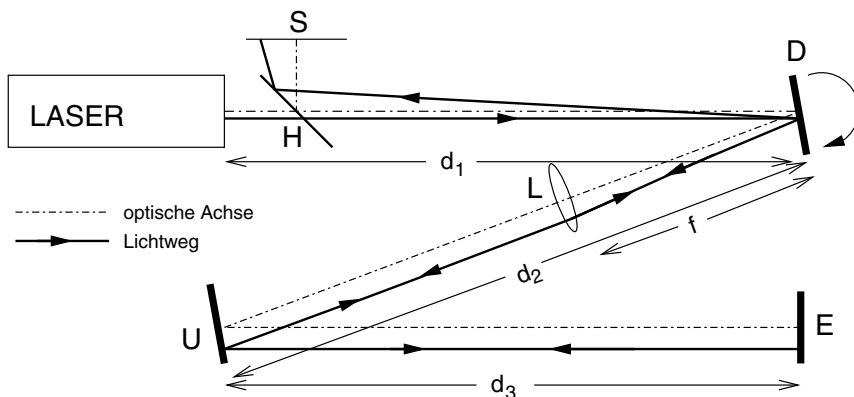


Abbildung 8.1. Skizze einer Versuchsanordnung, mittels xfig gezeichnet

Das \LaTeX -Kommando zum Einbinden der Abbildungen in den Text wird in Abschnitt 5.11.5 *Abbildungen* auf Seite 217 erläutert. Beim Einbinden wurden die Bilder auf das 0,7-fache verkleinert. Die feinste Linie soll im Endzustand nicht schmaler sein als 0,1 mm, eher breiter, und die kleinste Schrift nicht kleiner als im Textteil.

Ergänzend zu den Zeichnungen im Buch zeigt Abbildung 8.1 die mittels xfig gezeichnete Skizze einer Versuchsanordnung aus einem Physik-Praktikum. Mit dem Rechner geht es nicht unbedingt schneller als mit Bleistift und Papier, aber sowie aus einer Zeichnung weitere Zeichnungen abgeleitet werden, ist die rechnergestützte Arbeitsweise überlegen. Abgesehen davon lässt sich eine Datei leichter kopieren oder versenden als ein Blatt Papier.

Ein weiteres Werkzeug zum Zeichnen ist inkscape, ein Fork (Ableger) von sodipodi. Beide bezeichnen sich als Vektorgrafik-Editor und verwenden das Scalable-Vector-Graphics-Format (SVG). Die zugehörigen Webseiten sind unter <http://www.inkscape.org/> bzw. <http://www.sodipodi.com/> zu finden. Beide Werkzeuge sind bei Debian erhältlich, inkscape gibt es darüber hinaus als Tarball und als statisch gelinktes RPM-Paket auf der genannten Website. Der Funktionsumfang ist ähnlich wie bei xfig. Abbildung 8.2 zeigt einen Screenshot von inkscape mit ein paar einfachen Figuren.

Geht es um Geometrie und deren Verständnis, so bringt das Debian-Paket kseg (<http://www.mit.edu/~ibaran/>) alles mit, was man für die Euklidische Geometrie braucht. Das Programm verlangt die Qt-Bibliothek, hat aber nichts (mehr) mit dem KDE-Projekt zu tun. Das interaktive Geometrieprogramm *Zirkel und Lineal* gibt es nicht bei Debian, sondern bei <http://www.ku-eichstaett.de/MGF/homes/grothmann/java/zirkel/>. Es ist in Java geschrieben und läuft infolgedessen auf allen Rechnern, auf denen Java eingerichtet ist. Trotz eines erheblichen Leistungsumfanges kann es laut Aussage seines Autors RENÉ GROTHMANN im Unterricht von der Grundschule an bis zur Hochschule eingesetzt werden. Beide Programme stehen unter der GPL zur Verfügung.

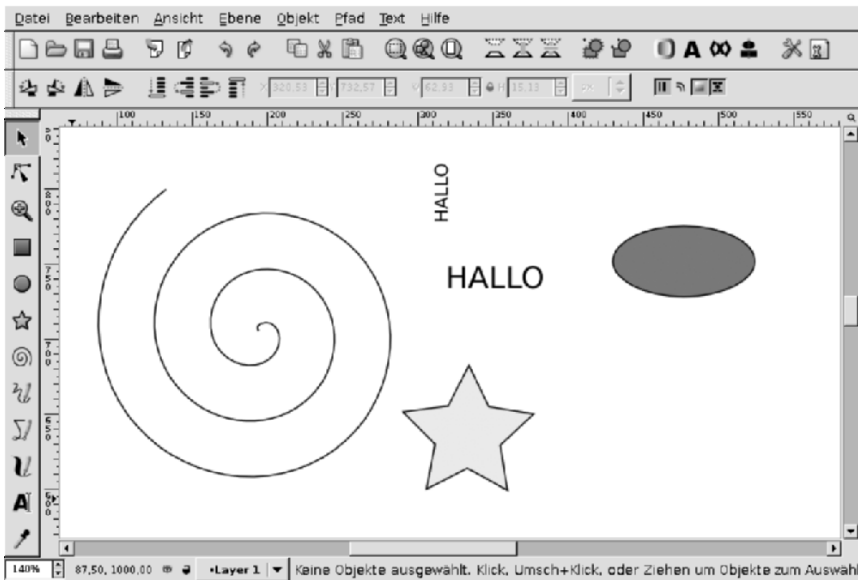


Abbildung 8.2. Screenshot des Vektorgrafik-Editors inkscape

8.3 CAD (qcad)

Der Marktführer bei Programmen zum Konstruieren am Bildschirm anstelle des Reißbretts ist AutoCAD, leider ökonomisch exklusiv. Bei einer Firma, die damit Geld verdient, ist das in Ordnung, aber will ich ehrenamtlich ein Gartentor zeichnen, muss ich mich nach etwas anderem umsehen. Die Auswahl an kostenfreier Software für diesen Zweck ist überschaubar. Bei Debian stehen die Pakete *qcad* samt der Dokumentation in *qcad-doc* und einigen Bibliotheken zur Verfügung. Die Heimat von QCad ist <http://www.qcad.org/> oder <http://www.ribbonsoft.com/de/qcad.html>. Die Manualseite verweist auf die ausführliche Dokumentation in `/usr/share/doc/qcad/html/`, wo ein *QCad User Reference Manual* liegt. Bei Qcad oder RibbonSoft ist das Reference Manual auch auf Deutsch oder im PDF-Format erhältlich, ausgedruckt 150 Seiten. CAM Expert aus derselben Schmiede ist eine kostenpflichtige Ergänzung zu QCad, kostenlose Demos werden angeboten. Ebenfalls bei RibbonSoft gibt es als Tarball das Werkzeug *vec2web*, das DXF-Dateien in mehrere gängige Rasterformate umwandelt.

QCad beschränkt sich auf zwei Dimensionen und verwendet wie AutoCAD das DXF-Dateiformat. Beim ersten Aufruf sind die Maßeinheit (mm) sowie die Sprache (German) festzulegen. Die Benutzung ist einfach, sofern man Grundkenntnisse im Konstruieren besitzt. Das Zeichenwerkzeug ist die Maus oder die Rollkugel mit mindestens zwei Tasten. Ungewohnt ist der schwarze Untergrund. Leider haben wir keinen Weg gefunden, das zu ändern. Im Druck wird ein weißer Untergrund verwendet.

- gEDA/gManager Projektmanager, der die Übersicht über alles hat,
- gerbv Betrachter für Gerber-Dateien (Leiterplattenentwurf),
- gnetlist Netzlisten-Generator, beherrscht ein Dutzend Formate,
- gnucape Schaltungsanalyse und -simulation, Alternative zu Berkeley Spice,
- gschem Hilfe beim Zeichnen von Schaltplänen (schematic capture),
- gsymcheck Symbolprüfer,
- GTKWave Untersuchung von Signalformen,
- gwave dito, aber anders,
- Icarus Verilog (Hardware Description Language) Compiler,
- ngspice Schaltungssimulator.
- PCB Printed Circuit Board, Leiterplattenentwurf,
- setup Grafisches Einrichtungswerkzeug (Installer),
- symbols Symbolbibliothek mit 1500 Symbolen,
- tcclspice Schaltungssimulator, der Tcl/Tk verwendet, ähnlich Berkeley Spice,
- utils mehrere Helferlein,
- vbs Verilog Behavioral Simulator.

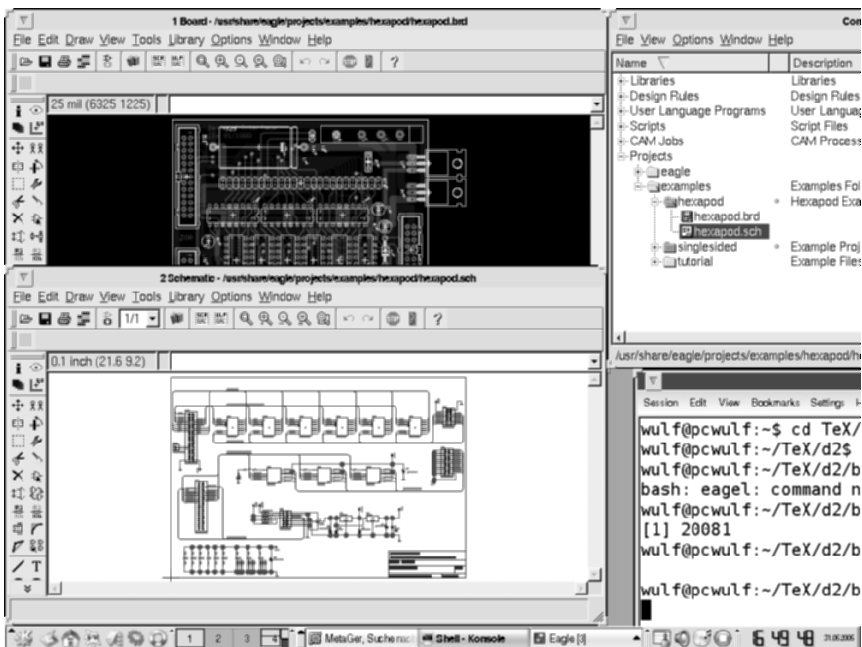


Abbildung 8.4. Screenshot des Leiterplatten-Editors EAGLE mit einer Demo; links oben die Leiterplatte, links unten das Schaltbild, rechts oben das Control Panel und rechts unten der Rest der Konsole. Ein Fall für mindestens zwei Bildschirme.

Mit den Debian-Paketen *eagle* und *eagle-data* aus der Non-free-Abteilung kommt ein *Easily Applicable Graphical Layout Editor* (EAGLE) zum Entwurf von

Leiterplatten, der Heimathafen ist <http://www.cadsoft.de/>. Bei der Einrichtung wird eine Lizenzdatei eingerichtet. Sofern man nicht Kunde des Herstellers ist und über einen Schlüssel verfügt, wählt man die Einrichtung als Freeware bzw. Light Edition. Kostenpflichtige Lizenzen gibt es in mehreren Stufen zu vertretbaren Preisen. Der Vorschlag für das Arbeitsverzeichnis von EAGLE ist verbesserungsfähig; man macht das später als gewöhnlicher Benutzer, nicht während der Einrichtung der Pakete. Mit den Paketen kommen zwei Beispiele, jeweils bestehend aus Schaltplan (Schematic) und Leiterplattenlayout (Board). Bei solchen Arbeiten wäre ein Rechner mit zwei oder drei Bildschirmen praktisch. Es wird eng auf dem Desktop, wie Abbildung 8.4 zeigt.

Chipmunks oder Streifenhörnchen sind eine Gattung der Hörnchen innerhalb der Nagetiere, in Nordamerika und der asiatischen Taiga heimisch. Das Chipmunk System (<http://www.cs.berkeley.edu/~lazzaro/chipmunk/>) ist eine Sammlung von Software-Werkzeugen, darunter Log, das in dem Debian-Paket *chipmunk-log* enthalten ist. Log besteht aus drei einzelnen Programmen:

- *chpmnk-analog* zur Simulation analoger Schaltungen,
- *chpmnk-diglog* zur Simulation digitaler Schaltungen,
- *chpmnk-logged* ein Gatter-Editor.

Von der genannten Berkeley-Site sind die weiteren Chipmunk-Werkzeuge im Quellcode und teilweise als Binaries erhältlich. Sie laufen auf einer Vielzahl von Systemen und sollten auf jedem Linux mit X11 ohne Schwierigkeiten zu übersetzen und zu gebrauchen sein.

8.5 Finite Elemente (FElT, TOCHNOG, FreeFem, FFEP)

Die Finite-Elemente-Methode (FEM) ist ein Verfahren der Numerischen Mathematik zur näherungsweisen Lösung von partiellen Differentialgleichungen mit Randbedingungen. Was allgemein formuliert für Laien fast unverständlich ist, wird an einem Beispiel sofort klar. Für Festigkeitsberechnungen eines nicht ganz einfachen Maschinenteils zerlegt man dieses gedanklich in viele kleine, geometrisch einfache Flächen oder Körper – die Elemente – und berechnet für diese die gesuchten Größen. An den Grenzen der Elemente müssen natürlich die berechneten Werte mit denen der Nachbarn übereinstimmen. Ein noch einfacheres Beispiel geht auf die Antike zurück. Wenn man ohne Kenntnis der Zahl π den Kreisumfang berechnen will, ersetzt man den Kreis durch ein Vieleck mit geraden Abschnitten und addiert deren Längen. Wegen der gewaltigen Zahlenmengen gewann die FEM erst mit dem Aufkommen der Computer an Boden, ab etwa 1960. Eine Linksammlung liegt auf <http://www.scicomp.uni-erlangen.de/SW/diffequ.html>.

Das Debian-Paket *felt* richtet ein FEM-System hauptsächlich für den Unterricht im Maschinenbau und Bauingenieurwesen ein. Seine Heimat ist <http://felt.sourceforge.net/>. FElT verfügt über drei Benutzerschnittstellen:

- die Kommandozeile,
- eine an MatLab angelehnte Skriptsprache,
- eine grafische, auf X11 aufbauende Oberfläche.

Für ein erstes Kennenlernen von FEM ist FELT gut geeignet, bei Aufgaben aus dem wirklichen Leben stößt man schnell an seine Grenzen.

TOCHNOG bei <http://tochnog.sourceforge.net/> ist ein freies Finite-Elemente-Programm unter der GPL für Linux und andere Systeme. Bei Debian finden sich die Pakete *tochnog* und *tochnog-doc*. Das Werkzeug kann aber auch vor Ort übersetzt werden und bringt mehrere umfangreiche Manuals mit. Beispiele aus der Mechanik, der Thermodynamik und dem Bauingenieurwesen erleichtern den Einstieg.

FreeFem ist eine Sprache zur Beschreibung von Aufgaben, die mittels FEM bearbeitet werden sollen (Systeme partieller Differentialgleichungen). Bei Debian sind mehrere Pakete dazu verfügbar; ihre Heimat ist <http://www.freefem.org/> in der Université Pierre et Marie Curie (UPMC) in Paris. Die Universität bietet die Möglichkeit an, FreeFem über das Web auf einem Debian-Cluster auszuprobieren. Man sollte sich allerdings mit FEM auskennen, die Site ist nicht der leichten Unterhaltung gewidmet.

Kein fertiges Werkzeug, sondern eine modular aufgebaute Bibliothek von C-Funktionen für GNU/Linux in Form eines Tarballs ist die *Free Finite Element Package* mit der Startseite <http://www.uni-essen.de/ingmath/ffep/>. Webüblich finden auch nicht-programmierende Benutzer dort weiterführende Links zur Ingenieurmathematik und zur Computergrafik.

8.6 Digitale Fotos (gphoto/gtkam, digikam, photoqc)

Digitale Kameras für Standbilder sind so populär geworden wie die traditionellen Kameras mit chemischer Bildaufzeichnung und werden diese bald überflügeln. Statt der Entwicklung und der Herstellung von Kopien (Vergrößerungen) hat man nun das Problem, die Bilder vom Speicher der Kamera in den Rechner zu bekommen. Angenommen, die Hardware passt zusammen, gibt es drei Wege:

- herstellereigene (proprietäre) Software, selten für Linux verfügbar,
- USB-Software, die den Kameraspeicher als einen externen Massenspeicher ansieht,
- USB-Software, die das Picture Transfer Protocol (PTP) verwirklicht und die Kamera als ein Still Image Device betrachtet.

Übersichten dazu sind auf <http://dplinux.org/> und auf <http://www.teaser.fr/~hfiguiere/linux/digicam.html> zu finden. Vor dem Erwerb einer digitalen Kamera, insbesondere einer älteren, sollte man die Softwarefrage klären. Von einem allgemein eingehaltenen Standard kann noch nicht die Rede sein. Das *USB Digital Camera HOWTO*, aktuell von 2002, hilft etwas weiter, geht nicht auf bestimmte Kameramodelle ein und könnte angesichts der raschen Entwicklung eine Bearbeitung vertragen. Dasselbe gilt vom *Kodak Digital Camera HOWTO*.

Das Debian-Paket *gphoto2* samt begleitenden Bibliotheken stellt eine Kommandozeile zum Übertragen der Daten von einer Digicam in den Rechner bereit. Der Heimathafen des Werkzeugs ist <http://www.gphoto.org/>. Wer es grafisch haben möchte, nimmt das Debian-Paket *gtkam* hinzu. Die Benutzung ist einfach.

Die Alternative des KDE-Projektes ist im Debian-Paket *digikam* enthalten, dessen Funktionen über die von *gphoto2* hinaus gehen. Das Werkzeug bezeichnet sich als *KDE Photo Management* und kann unter anderem Fotoalben verwalten. Wer nur seine Kamera auslesen will und seine Fotos mit einer Datenbank verwaltet, braucht die zusätzliche Funktionalität nicht. Eine dritte Möglichkeit bietet bei bestimmten Kamertypen das Debian-Paket *photopc* samt seiner grafischen Oberfläche *photok*. Wenn keines der genannten Werkzeuge imstande ist, die jeweilige Kamera auszulesen, kann man noch im Web nach kameraspezifischer Software suchen, allerdings mit geringer Aussicht auf Erfolg. Sind die Bilder erst einmal in der Kiste, stehen andere Werkzeuge wie GIMP für die weitere Verarbeitung bereit, siehe Abschnitt 8.8 *Bildverarbeitung* auf Seite 303.

8.7 Bildbetrachter (qiv, feh, xpcd, xv)

Zum Betrachten von Bildern – meist Fotos – in verschiedenen Grafik-Formaten mit begrenzten Möglichkeiten zur Beeinflussung der Bilder dienen Werkzeuge wie der unter X11 laufende Quick Image Viewer *qiv*. Ruft man ihn mit einigen Optionen auf:

```
joe@debian:~$ qiv -m -s -f *.jpg
```

so werden die jpg-Bilder in der Arbeitsdatei an die Größe des Bildschirms angepasst dargestellt und als Diaschau periodisch gewechselt. Weiteres bei <http://www.klografx.net/qiv/>.

Ein ähnliches Werkzeug ist *feh* (<http://www.linuxbrit.co.uk/feh/>). Der Aufruf:

```
joe@debian:~$ feh -r -t *
```

zeigt rekursiv im Arbeitsverzeichnis und darunter alle darstellbaren Bilder in Daumen nagelgröße (thumbnail) auf dem Bildschirm an. Durch Anklicken wird ein Bild in voller Größe dargestellt. Bilddateien können auch per FTP oder HTTP über das Netz geladen werden. Bei Debian findet sich etwa ein Dutzend solcher Bildbetrachter, wenn man die Paketlisten, Abteilung *Graphics*, nach dem Stichwort *viewer* absucht.

Die Debian-Pakete *xpcd* und *xpcd-svga* enthalten einige Werkzeuge zum Betrachten und Bearbeiten von Foto-CDs im pcd-Format (Kodak Photo Compact Disc). Die Fotos werden in mehreren Auflösungen komprimiert gespeichert, wobei eine CD etwa 100 Kleinbild-Fotos aufnehmen kann. Die Kodak Photo CD ist keine gewöhnliche CD, sondern ein Medium zum Aufbewahren von Fotos über lange Zeiten (<http://www.photocd.de/>). Es gibt Labors, die herkömmliche (analoge, chemische) Fotos, Dias oder Filme auf Kodak Photo CDs übertragen. Wegen der Kosten wendet sich die Kodak Photo CD eher an Berufsfotografen.

Der verbreitete Bildbetrachter `xv` von JOHN BRADLEY ist nicht als Debian-Paket erhältlich, sondern nur als Shareware im Quellcode (Tarball) bei <http://www.trilon.com/xv/xv.html>. Der Versuch, das Programm auf einem PC unter *sarge* zu übersetzen, schlug fehl. Die Empfehlung, den GNU-C-Compiler mit der Option `-traditional` aufzurufen, ließ sich nicht verwirklichen. Die Einrichtung des ebenfalls angebotenen RPM-Binärpaketes mittels:

```
debian:~# alien -i xv-3.10a-13.i386.rpm
```

ging ohne Beschwerden über die Bühne, und nachdem auch noch das Debian-Paket *libpng2* eingerichtet war, funktionierte der Betrachter, zumindest mit jpg-, gif- und tif-Dateien. In begrenztem Maße lassen sich mit `xv` auch Änderungen an den Bild-dateien vornehmen, aber – wie der Autor betont – in erster Linie ist das Werkzeug ein Betrachter.

Nicht bei Debian, sondern bei <http://mapivi.sourceforge.net/> gibt es unter der GPL den Fotobetrachter, -umwandler und -organizer MaPiVi (Martins Picture Viewer) von MARTIN HERRMANN, der sich besonders um die zu den Bildern gehörenden Metadaten (Exchangeable Image File Format, EXIF, und International Press Telecommunications Council/Information Interchange Model, IPTC/IIM) kümmert. Das Werkzeug beschränkt sich auf das jpeg-Format und bringt eine kurze, aktuelle Dokumentation auf Englisch mit.

8.8 Bildverarbeitung

8.8.1 Konverter (netpbm, autotrace)

Die Entwicklungsgeschichte von `netpbm` ist kurvenreich, nachzulesen bei <http://netpbm.alioth.debian.org/>. Die Startseite des in den achtziger Jahren von JEF POSKANZER begonnenen Projektes liegt unter <http://netpbm.sourceforge.net/>. Das gleichnamige Debian-Paket umfasst eine Sammlung von etwa 200 einfachen Werkzeugen vornehmlich zur Umwandlung von Dateiformaten.

Das Werkzeug `autotrace` wandelt Rastergrafiken (Bitmaps) in Vektorgrafiken um. Die Eigenheiten der beiden Grafikdarstellungen wurden in Abschnitt 8.1 *Grundbegriffe* auf Seite 291 erläutert. `autotrace` ist auf <http://autotrace.sourceforge.net/> zu Hause. Dort finden sich auch Links zu anderen Vektorgrafikprogrammen. Die wichtigsten Eingabeformate sind BMP, TGA, PNM, PPM, PGM und PBM, dazu jene Formate, die das Werkzeug ImageMagick versteht. Als Ausgabeformate stehen PostScript, SVG, XFIG, SWF, EMF, DXF, CGM, MIF, P2E und SK zur Verfügung.

Auch die nachfolgend besprochenen Werkzeuge zur Bearbeitung grafischer Daten (GIMP, ImageMagick) eignen sich zum Umwandeln von Formaten. Vermutlich werden sich die meisten Benutzer in eines dieser Universalwerkzeuge einarbeiten und so lange wie möglich dabei bleiben. Zu den Konvertern greift man dann nur, wenn entweder viele Dateien oder exotische Formate umzuwandeln sind.

8.8.2 Ein Werkzeugkoffer (GIMP)

Das GNU Image Manipulation Program *gimp* (The GIMP) ist ein umfangreiches Software-Paket zur Bearbeitung von Bildern, ähnlich wie das kommerzielle Produkt Adobe Photoshop, das in diesem Arbeitsgebiet Maßstäbe vorgibt. Über GIMP sind mehrere Bücher geschrieben worden, außerdem setzt das Ausreizen dieses Programms einige Kenntnisse aus dem grafischen Handwerk voraus, beginnend mit dem Fachvokabular. Zusätzlich zu GIMP selbst wird die Einrichtung der Pakete *gimp-help-common*, *gimp-help-de* und *gimp-helpbrowser* dringend angeraten. Die deutsche Übersetzung der Hilfe ist etwas holprig. Die Verteilung der Arbeitsgänge auf die verschiedenen Menüs hat sich im Lauf der Versionen geändert; man achte darauf, dass die Hilfe zur verwendeten Version passt. GIMP beachtet das Locale. Wir wollen uns drei typische Aufgaben bei der Bearbeitung von Fotos näher ansehen.

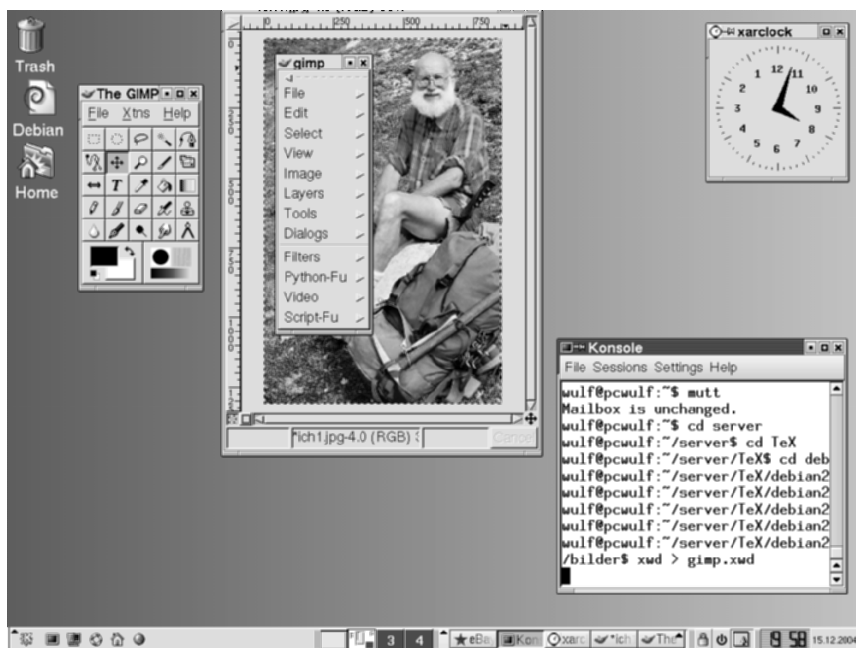


Abbildung 8.5. Screenshot des Bildbearbeitungsprogrammes GIMP bei der Bearbeitung eines Fotos. Links die Werkzeugkiste, in der Mitte das zu bearbeitende Bild. Die Uhr und das Fenster rechts unten gehören nicht zu GIMP.

Sehr oft will man aus einem digitalen oder digitalisierten Foto einen Ausschnitt herauschneiden. Dazu ruft man aus der Kommandozeile oder aus dem Hauptmenü *Graphik -> Bildeditor (GIMP)* das Werkzeug auf und bekommt das Grundfenster des GIMP wie in Abbildung 8.5 links gezeigt auf den Screen:

- Dort klickt man *Datei -> Öffnen* (File -> Open) an und wird zu einem zweiten Fenster geleitet, in dem man mit der linken Maustaste ein Verzeichnis und daraus eine Datei auswählt und öffnet. Ein gebräuchliches Dateiformat für Fotos ist JPG, Kennung `jpg` oder `jpeg`. GIMP versteht auch andere Formate, aber nicht alle Formate dieser Welt. Das zweite Fenster verschwindet, an seine Stelle tritt ein Fenster mit dem ausgewählten Bild wie in der Abbildung in der Mitte zu sehen.
- Von den im ersten Fenster angebotenen Werkzeugen wählt man das erste aus, es zeigt auf Wunsch ein Fähnchen mit dem Text *Einen rechteckigen Bereich wählen* (Select rectangular regions). Nun ziehen wir im Bild mit gedrückter linker Maustaste die Diagonale eines rechteckigen Bereiches, von links oben nach rechts unten oder andersherum. Lassen wir die Maustaste los, wird der Rahmen fixiert. Er besteht aus langsam vorankriechenden schwarzen und weißen Abschnitten, die als Wanderameisen (walking ants) bezeichnet werden.
- Um den Rahmen zu löschen, klickt man mit der linken Taste in den Bereich außerhalb des Rahmens.
- Ist man mit dem Ausschnitt zufrieden, klickt man mit der *rechten* Maustaste in den Ausschnitt, ein kleines Menü öffnet sich, wir wählen *Bearbeiten -> Kopieren* (Edit -> Copy), womit der Ausschnitt in eine Zwischenablage gespeichert wird. Alternativ aktiviert man das Menü aus der Menüleiste am oberen Fensterrand.
- Dann klicken wir mit der rechten Maustaste irgendwo in das Fenster und wählen *Bearbeiten -> Als neues Bild einfügen* (Edit -> Paste as new). Ein neues Fenster mit dem Ausschnitt öffnet sich, das keine Verbindung mehr mit dem ursprünglichen Fenster hat.
- Wir klicken mit der rechten Maustaste in das neue Bild und wählen *Datei -> Speichern* (File -> Save) bzw. *Speichern unter* (Save as), falls wir den Ausschnitt in eine neue Datei speichern wollen, was vermutlich der Fall ist.
- GIMP fragt nach dem Namen der neuen Datei und ersieht aus deren Kennung (`jpg`, `png`, `eps` ...), welches Dateiformat gewünscht wird. Unter Umständen folgen noch einige Hinweise und/oder Fragen wegen Export und dergleichen, wobei man die Vorschläge übernimmt, wenn man nichts Besseres weiß. Diese beiden Schritte wiederholt man für jedes gewünschte Ausgabeformat.
- Schließlich räumt man den Bildschirm auf, indem man alle nicht mehr benötigten Fenster schließt. Auch wenn man ein weiteres Bild bearbeiten möchte, sollte man aufräumen, sonst wird es irgendwann eng im Speicher. GIMP beendet man mit dem Menüpunkt *Datei -> Beenden* (File -> Quit) im ersten Fenster. Dabei werden alle vom GIMP erzeugten Fenster ebenfalls gelöscht.

Ist vorherzusehen, dass ein Foto ausschnittsweise verarbeitet wird, sollte man bei der Aufnahme und beim Speichern des Originals eine hohe Qualität anstreben (die Speicherplatz kostet). Herunterrechnen auf eine gerade noch ausreichende Qualität kann man ein Bild nachträglich.

Die zweite Aufgabe betrifft das Aufhellen einer zu dunklen Vorlage. Hier sind die Arbeitsgänge nicht so offensichtlich. Geht man über die Menüpunkte *Ebenen -> Farben -> Helligkeit, Kontrast* (Layer -> Colors -> Brightness, Contrast), so ist

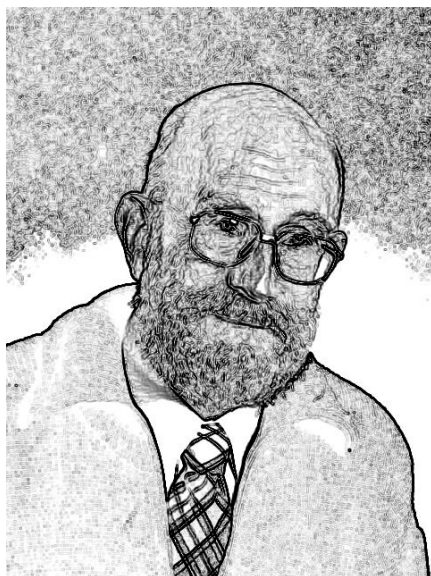


Abbildung 8.6. Strichzeichnung, mittels GIMP aus Foto erzeugt

das Ergebnis zwar heller, wirkt aber milchig. Die Wirkung der Einstellungen wird sofort angezeigt und lässt sich rücksetzen, man kann und sollte herumspielen (üben). Folgender Weg verspricht mehr Erfolg:

- Bild auswählen und öffnen wie oben beschrieben.
- Den Menüpunkt *Werkzeuge -> Farben -> Werte* (Tools -> Color Tools -> Levels) aufrufen. Unter der Überschrift *Quellwerte* (Input Levels) wird ein Histogramm gezeigt, das die relative Häufigkeit der Pixel über ihren Helligkeitswerten von Schwarz bis Weiß darstellt.
- Unterhalb des Histogramms befindet sich ein Band mit drei dreieckigen, verschiebbaren Knöpfen, die anfangs auf Schwarz, Mitte (Grau) und Weiß stehen. Verschiebt man das mittlere Dreieck nach links – etwa bis der zugehörige, mit *Gamma* bezeichnete Zahlenwert 1,5 erreicht – wird das Bild insgesamt heller, ohne dass ganz schwarze oder ganz weiße Stellen beeinflusst werden. Oft reicht das bereits aus.
- Bild abspeichern. Dabei ist ein neuer Name für die Datei zu vergeben, die alte bleibt erhalten.

Man kann auch die Ober- und Untergrenze, die Zielwerte sowie jeden der drei Farbkanaäle (Farbstich) verstellen und sieht sofort am Bild die Wirkung, wobei sich Bildschirm und Drucker durchaus unterschiedlich verhalten. Will man aus einem Farbfoto ein Schwarzweißfoto (Halbtonbild, Graustufenbild) machen, setzt man seine Farbsättigung auf null. Dahin führen mehrere Wege. Ein kurzer geht über die Menüpunkte *Ebene -> Farben -> Sättigung entfernen* (Layer -> Colors -> Desaturate). Da

die Nuancen der Änderungen am Bild im Buchdruck nicht deutlich herauskommen – vom Verlust der Farbe ganz zu schweigen – wird hier auf Bildbeispiele verzichtet.

Drittens sehen wir uns an, wie man aus einem Foto eine Strichzeichnung erzeugt. Ausgangspunkt ist das Farbfoto, das auf der Vortitelseite den Verfasser wiedergibt. Daraus wurde wie vorstehend erläutert ein Schwarzweißbild erzeugt und dessen Kontrast durch Drehen am Histogramm leicht verstärkt. Das Schwarzweißbild wurde mittels *Filter -> Kanten finden -> Kanten* (Filters -> Edge-Detect -> Edges) bearbeitet und schließlich das Ergebnis noch mittels *Ebene -> Farben -> Invertieren* (Layer -> Colors -> Invert) umgekehrt. Abbildung 8.6 zeigt das umgeschönte Endprodukt. An DÜRER reicht es nicht heran. Ein kantiges Objekt liefert bessere Resultate. GIMP bringt auch künstlerische Filter mit, die den Effekt von Kohlezeichnungen oder Ölgemälden hervorrufen. Man braucht Übung und Erfahrung, das heißt Zeit, um diese Tricks und Filter sinnvoll einzusetzen. Irgendetwas kommt immer heraus, aber der Anwender sollte bestimmen, was.



Abbildung 8.7. Mit GIMP auf alt getrimmtes Foto, auf dem Schirm in chamois, außerdem etwas aufgehellt, am Strand von Ahlbeck im Sommer 1938

Weitere Fähigkeiten des GIMP sind Beschriften, Drehen, Umkehren (Invertieren), Einfärben sowie zahlreiche mehr oder weniger künstlerische Effekte (Filter), wobei das Weichzeichnen älteren Fotografen bekannt vorkommt. Bei Por-

trätaufnahmen setzte man gern eine Duto-Scheibe vor das Objektiv, um Härten zu mildern. Wem die mitgelieferten Effekte nicht reichen, der kann sich mittels einer Skriptsprache eigene schreiben. Ein Beispiel für ein Skript ist der im Paket enthaltene Effekt *Altes Foto* aus dem Menü *Skript-Fu*, der mehrere Filter kombiniert und dessen Wirkung Abbildung 8.7 zeigt, auf dem Bildschirm leicht gelb getönt (chamois), lang ists her. Das Skript liegt im Verzeichnis `/usr/share/gimp/2.0/scripts/`. Auch Kaffeecken lassen sich erzeugen oder beseitigen. Über die Menüpunkte *Datei -> Zurücksetzen* (File -> Revert) werden Änderungen rückgängig gemacht. Ohnehin sollte man vorsichtshalber immer auf Kopien der Originaldatei arbeiten. Mit GIMP kann man sich lange beschäftigen.

8.8.3 Noch ein Koffer (ImageMagick)

Ähnliche Aufgaben wie GIMP erledigt ein bei Debian, aber auch im Netz unter einer GPL-ähnlichen Lizenz erhältlicher Werkzeugkoffer namens ImageMagick (<http://www.imagemagick.org/>). Auf der Website werden Links zu einigen Tutorials genannt. Die Werkzeuge sind aus der Kommandozeile aufzurufen und lassen sich sogar über eine Schnittstelle zu gängigen Programmiersprachen in eigene Programme einbauen. Die Einzelwerkzeuge – zu denen jeweils ein Manualeintrag gehört – sind:



Abbildung 8.8. Screenshot eines Kalenders, erzeugt mittels ImageMagick `import` und `convert`

- `convert` Dateiformate umwandeln und ein paar Dinge mehr,
- `identify` Eigenschaften eines Bildes bzw. einer Datei ermitteln,
- `mogrify` (ein Phantasiewort) verschiedene Transformationen (herunterrechnen, drehen, Farbverschiebungen usw.),
- `composite` zwei Bilder überlagern,
- `montage` mehrere Bilder überlagern,

- `compare` zwei verwandte Bilder vergleichen,
- `display` Bild darstellen, dabei einige Änderungen vornehmen (X11 erforderlich),
- `animate` eine Folge von Bildern wiedergeben (X11 erforderlich),
- `import` Bild (Screenshot) aus einem X11-Fenster in eine Datei schreiben, wie `xwd`,
- `conjure` ein Skript, geschrieben in der Magick Scripting Language, ausführen.

Abbildung 8.8 zeigt einen mit `import` erzeugten und mit `convert` umgewandelten Screenshot. Das Vorgehen war äußerst einfach:

```
joe@debian:~$ import
```

in das einzufangende Fenster klicken, anschließend in das benötigte Dateiformat umwandeln:

```
joe@debian:~$ convert magick.miff magick.eps
```

keine überflüssigen Fragen. Auf Grund der anderen Benutzungsweise ist ImageMagick nicht nur eine Alternative zu GIMP, sondern auch eine Ergänzung. Beide Koffer haben ihre Daseinsberechtigung.

8.8.4 Scannen (`scanimage`, `xscanimage`, `xsane`, `quiteinsane`)

Flachbettscanner verfügen zwar nur über eine geringe Auswahl an Schnittstellen (parallel, SCSI, USB), aber die Elektronik dahinter gehorcht unterschiedlichen Kommandos, sodass man für fast jedes Modell einen eigenen Treiber braucht. Jede Anwendung, die Daten von Scannern (F: `scanneur`) oder anderen Bilderfassungsgeräten einliest, müsste ihren Satz von Treibern mitbringen. Aus dieser Klemme hilft die SANE-Schnittstelle zwischen Treibern und Anwendungen, ein *Application Program Interface* (API). Die SANE-Software muss die Scanner kennen, die Anwendungen nur die SANE-Schnittstelle. SANE ist im Web durch <http://www.sane-project.org/> vertreten, mit Listen der unterstützten Hardware und der Anwendungen, die SANE verwenden. Eine ähnliche API ist TWAIN (<http://www.twain.org/>). Das *Scanner HOWTO*, aktuell von 2004, behandelt Hard- und Software.

Als Werkzeug für die Kommandozeile ist `scanimage` verbreitet, enthalten im Debian-Paket `sane-utils`. An Ausgabeformaten stehen TIFF, PBM, PGM und PPM zur Verfügung. Das Bild kann bereits beim Scannen beeinflusst werden, siehe Manual.

Grafische Frontends für SANE sind `xscanimage` aus dem Debian-Paket `sane`, `xsane` (braucht GTK+) und `quiteinsane` (braucht Qt). Die drei Werkzeuge lassen sich auch aus GIMP heraus als Plug-in starten, Menüpunkt *Datei -> Holen*. Sinnvoll, wenn man ohnehin mit GIMP arbeitet. Man scannt zuerst eine Vorschau (preview) ein, legt in dieser wie von GIMP gewohnt einen Bildausschnitt fest und startet dann den eigentlichen Lauf. Zum Scannen wählt man eine hohe Auflösung, da sich diese nachträglich herunter-, aber nicht heraufrechnen lässt.

Eingescannte Textseiten sind Grafik wie jede andere auch und müssen erst durch ein besonderes Werkzeug in Text umgewandelt werden, siehe Abschnitt 5.2.6 *Optical Character Recognition* auf Seite 173. Auf der Website von SANE wird ein Werkzeug *unpaper* zur Vorbehandlung der Scans angeboten, das den Erfolg der Umwandlung verbessert.

8.8.5 Panoramen (panotools, hugin)

Fotografische Panoramen gibt es seit altersher. Früher wurden sie entweder mit Spezialkameras aufgenommen oder mit Schere und Kleber aus Einzelaufnahmen zusammengefügt. Heute ersetzt Software die beiden Hilfsmittel und kann dabei noch einiges mehr leisten. Die Programme werden nach ihrer Hauptfunktion auch *Stitcher* (to stitch = nähen, sticken, heften) genannt. Die wichtigsten Aufgaben eines Stitchers sind:

- Zusammenfügen der Einzelbilder im Überlappungsbereich,
- Anpassen der Perspektive,
- Angleichen der Helligkeit und der Farben,
- Zuschneiden des fertigen Bildes.

Trotz leistungsfähiger Software gilt, dass gutes Ausgangsmaterial (Stativ, Drehung der Kamera um eine senkrechte Achse durch die optische Mitte, reichliche Überlappung, gleichmäßiges Licht) die Arbeit erleichtert und das Ergebnis verbessert. Der Rechner braucht vor allem Arbeitsspeicher, 1 GB ist kein Luxus.

Stitchen gehört nicht zu den Grundfunktionen von GIMP, aber es gibt ein *gimp-panorama-plugin* aus den Panorama Tools als Debian-Paket bei <http://people.debian.org/~jordens/debs-i386/gimp-pano\ldots>, siehe auch <http://bugbear.blackfish.org.uk/~bruno/panorama-tools/>. Eine Einrichtung auf einem System mit *sarge* und GIMP 2.2 verlief problemlos. Hinterher findet man eine Datei `/usr/lib/gimp/2.0/plugins/PanTools`, die die wichtigsten Funktionen zur Verfügung stellt. Die Datei `$HOME/.gimp-2.2` war noch zu erneuern, dann zeigte das GIMP-Menü *Filter* die Punkte *Pan Controls* und *Panorama Tools*. Erläuterungen in http://gimpforum.de/Grokking_the_Gimp/node70.html, der Online-Version des Buches *Grokking the Gimp* von CAREY BUNKS. Wer ohnehin mit GIMP arbeitet und einmal schnuppern will, ist mit dem Plug-in gut beraten.

Ein umfangreicherer Werkzeugkasten namens *Panorama Tools* stammt von HELMUT DERSCH und ist unter <http://webuser.fh-furtwangen.de/~dersch/> oder <http://www.all-in-one.ee/~dersch/> zu finden. Das Projekt ist auch bei Sourceforge unter <http://panotools.sourceforge.net/> erreichbar. Eine Benutzeroberfläche zu den Panorama Tools stellt das Paket *Hugin* dar, zu haben bei <http://hugin.sourceforge.net/>. Die Webseite von *Hugin* verweist auf einige weitere Werkzeuge wie *enblend* und *autopano*. Der Werkzeugkasten erfordert Einarbeitung und ist noch in lebhafter Entwicklung, bietet aber auch viel. Es kann unter anderem dazu verwendet werden, Teilscans

großer Vorlagen zu einem Ganzen zusammenzufügen. Eine Beschreibung von KATJA SOCHER, wie man mit *Hugin* und GIMP ein Panorama baut, ist unter <http://www.tuxgraphics.org/graphics/> zu lesen.

8.8.6 Warpen und Morphen (*morph*, *xmorph*, *gtkmorph*)

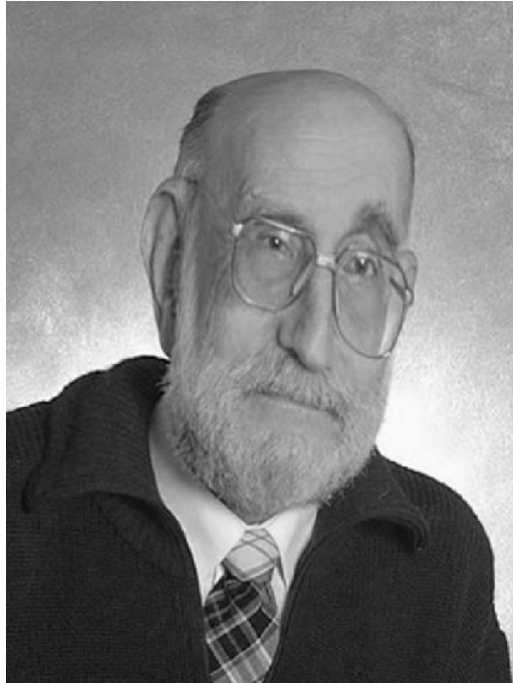


Abbildung 8.9. Gewarppte Kopie des Porträts auf der Vortitelseite

In der Computergrafik will man mitunter Bilder verzerren oder verfremden. Das Gesicht einer Person kann verzerrt (*warpen*) oder in mehreren Schritten in das einer anderen umgewandelt werden (*morphen*). Die Zwischenschritte lassen sich als Animation darstellen. Auch Schriftzüge werden gern gemorphen. Die Werkzeuge *morph*, *xmorph* und *gtkmorph* sind Frontends für dieselbe Bibliothek. *morph* wird aus der Kommandozeile benutzt, *xmorph* benötigt X11 und *gtkmorph* zusätzlich GTK+. Letzteres ist das vielseitigste Werkzeug, versteht mehrere Dateiformate und ist normalerweise zu bevorzugen. Die zugehörige Manualseite hilft nicht weit, ein Guide wird nach dem Aufruf im Fenster *resulting image* unter dem Menüpunkt *Help* angeboten. Der Heimathafen der drei Werkzeuge ist <http://xmorph.sourceforge.net/>, wo weitere Dokumentation herumliegt. Im Debian-Paket *gtkmorph-example* sind Beispiele zu *xmorph* und *gtkmorph* enthal-

ten. Die Werkzeuge bzw. die zu Grunde liegende Bibliothek beherrschen nur zwei Dimensionen.

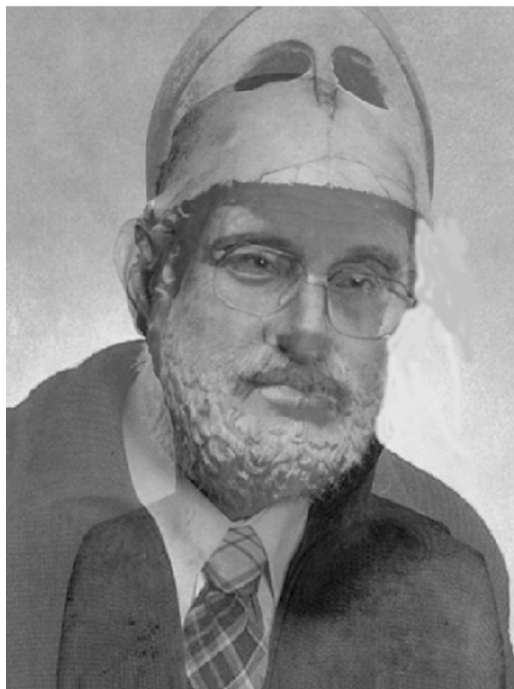


Abbildung 8.10. Aus dem Porträt auf der Vortitelseite und dem Foto einer Statue eines alten Griechen zusammengemorphtes Bild, ist noch nachzubearbeiten

Zum Warpen wird eine Bildvorlage gebraucht, über die das Werkzeug ein rechtwinkliges Gitter legt, in dem sich beliebige Punkte nach Auswahl des Menüpunktes *edit mesh* oben im Bildfenster verschieben lassen. Zieht man das Gitter enger zusammen, werden die zugehörigen Bildteile gedehnt. Das verzerrte Gitter lässt sich zwecks Wiederbenutzung speichern und laden. Abbildung 8.9 zeigt eine gewarperte Kopie des Porträts auf der Vortitelseite. Soll etwas Vernünftiges herauskommen, benötigt man Übung, da die Auswirkung der Verschiebung nicht sofort im Bild angezeigt wird, sondern erst nach Auswahl des Menüpunktes *do warp* im Bildfenster und einer kurzen Denkpause.

Das Morphen besteht im Berechnen mittlerer Werte aus zwei Ausgangsbildern. Das Ergebnis ist um so überzeugender, je ähnlicher die Ausgangsbilder sind. Bei Porträts heißt das, gleiche Größe (vor allem des Dreiecks Augen - Mund), gleiche Kopfhaltung, ruhiger Hintergrund und gleiche technische Parameter. Man muss sich immer vor Augen halten, dass das Programm rechnet, nicht denkt. In der Regel ist eine sorgfältige Vorbereitung der Ausgangsbilder mit GIMP anzuraten. Das Beispiel von Abbildung 8.10 lässt Wünsche offen, die mit etwas mehr Zeitaufwand zu er-

füllen gewesen wären. Das Werkzeug `gtkmorph` bietet an, das Morphen in einer wählbaren Anzahl von Schritten (Default: 10) durchzuführen und die Zwischenergebnisse in Form einer Animation darzustellen. Das hilft bei der Auswahl der optimalen Morphung. Alle Zwischenergebnisse werden im PNG-Format gespeichert und lassen sich mit GIMP nachbearbeiten.

8.8.7 Modellierer, Renderer (Ayam, Aqsis, povray, blender, yafray)

Die Erzeugung dreidimensionaler (perspektivischer) Szenen im Rechner erfordert drei Schritte mit jeweils eigenen Werkzeugen:

- das Herstellen der Objekte, Modellieren (modeling) genannt,
- das Bewegen der Objekte, Animieren (animation) genannt,
- das Darstellen der Objekte unter bestimmten Lichtverhältnissen, Rendern (rendering) genannt.

Das *3D Graphics Modelling and Rendering mini-HOWTO*, aktuell von 2001, beschreibt mit einfachen Worten die Einrichtung und Benutzung von Mops (Vorgänger von Ayam) und BMRT; wir bringen unten ausführlich ein ähnliches Beispiel. Beim Modellieren geht man von einfachen geometrischen Gebilden wie Kugeln, Zylindern, Kegeln oder Würfeln in einem Koordinatensystem aus. Wer einmal etwas von Darstellender Geometrie gehört hat, fühlt sich bei dieser Tätigkeit sofort zu Hause. Das Modell verhält sich zum Endergebnis etwa wie eine technische Zeichnung zum fertigen Maschinenteil. Ein Werkzeug zum Modellieren ist Ayam, wir benutzen es später als Beispiel. Ist unser Modell unbewegt, brauchen wir uns um die Animation nicht zu sorgen. Andernfalls müssen wir uns nach einem der dünn gesäten Animationswerkzeuge umsehen, bei Debian beispielsweise im Programm K-3D, Paket *k3d*, <http://k3d.sourceforge.net/>, enthalten. Wir befassen uns hier nicht näher mit Animationen. Das Rendern kann man sich wie das Fotografieren einer Szene vorstellen: Man hat Gegenstände, Lichtquellen und eine Kamera, alles im Raum verteilt, und am Ende kommt ein zweidimensionales Bild oder ein Filmstreifen heraus. Bei Renderern ist die Auswahl wieder größer, wir werden uns mit Aqsis – von Debian im gleichnamigen Paket verfügbar – beschäftigen. Zum Rendern gehört Licht; Werkzeuge, die auch Spiegelungen, Brechungen und dergleichen beherrschen – zum Preis eines hohen Rechenaufwandes – werden raytracing-fähige Renderer oder kurz Raytracer genannt. Die Renderer Aqsis und PRMan arbeiten nicht mit Raytracing; POV-Ray, Yafray und neuere Versionen von Blender beherrschen Raytracing. Es gibt auch Werkzeugkoffer mit Kombinationen von Werkzeugen für zwei oder alle drei Aufgaben wie *Blender*, bei Debian zu haben, die dem Benutzer Schnittstellenprobleme zwischen den Werkzeugen ersparen. Eine verbreitete Schnittstelle oder Spezifikation für den Datenverkehr zwischen modellierenden und darstellenden Programmen ist RenderMan (<http://www.renderman.org/>), geschaffen von der Firma Pixar (<http://www.pixar.com/>). Das zugehörige Dateiformat nennt sich RenderMan Interface Bytestream (RIB).

Wir sehen uns ein einfaches Beispiel genauer an und folgen dabei im Wesentlichen dem *Ayam Tutorial #1* von RANDOLPH SCHULTZ, ergänzt durch eigene Erfah-

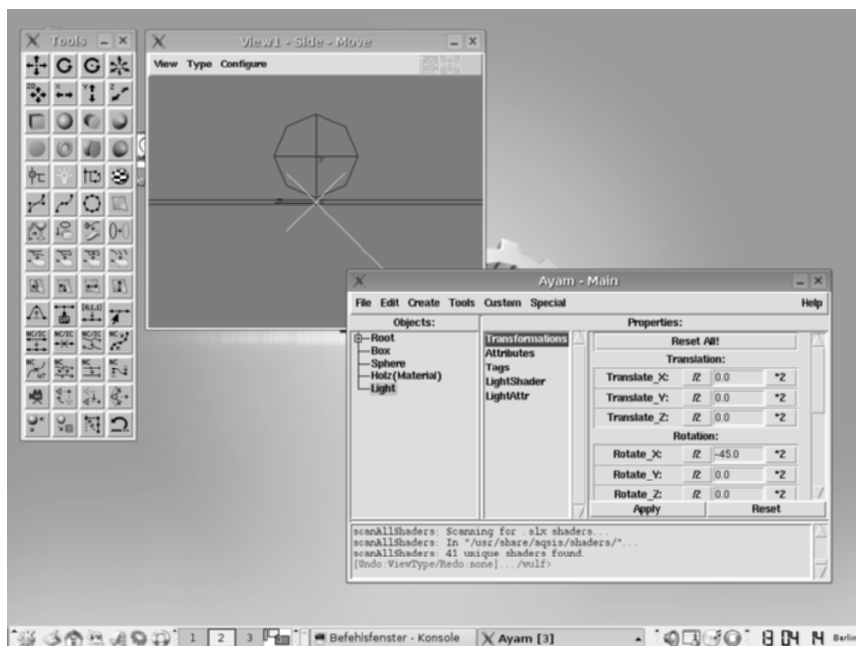


Abbildung 8.11. Screenshot des Bildschirms von Ayam, einem Modellierwerkzeug, mit drei Fenstern

runen. R. SCHULTZ verwendet und empfiehlt als Renderer die Blue Moon Rendering Tools (BMRT); diese sind jedoch nicht mehr im Netz aufzutreiben. Als Modellierwerkzeug setzen wir Ayam ein, als Renderer Aqsis. Vorbereitend sind folgende Schritte durchzuführen (Stand Mitte 2005, Ayam 1.8.2, Aqsis 1.0.0):

1. Einrichten des Debian-Paketes *aqsis*, als Verwalter, damit die Dateien an ihre Default-Standorte gelangen und von allen Benutzern verwendet werden können. Der Renderer kommt mit zwei Kollegen (*aqsis**) nach `/usr/bin/`, seine Shader (kleine Hilfsprogramme) nach `/usr/share/aqsis/shaders/`, Bibliotheken nach `/usr/lib/`. Der Heimathafen von Aqsis ist <http://sourceforge.net/projects/aqsis/>.
2. Einrichten des Tarballs *ayam1.8.2.linux.tar.gz* von <http://ayam.sourceforge.net/> oder <http://www.ayam3d.org/>. Das Programm samt Zubehör kommt nach `/usr/local/ayam/`.
3. Einrichten des Debian-Paketes *liblog4cpp-dev*, sofern noch nicht vorhanden.
4. Herunterladen der Datei *wood2.sl* von <http://www.renderman.org/> aus der Abteilung Shaders – BMRT Shaders mittels Browser in das Verzeichnis `/usr/share/aqsis/shaders/`.
5. Übersetzen der Shader-Quelldatei *wood2.sl* mit dem Aqsis-Compiler `/usr/bin/aqsl`. Hinterher sollte sich eine Datei *wood2.slx* mit den Zugriffsrechten 644 im Verzeichnis finden lassen.

- Die Umgebungsvariable `LD_LIBRARY_PATH` ist auf den Wert `/usr/local/ayam/lib` zu setzen bzw. um diesen Wert zu erweitern, zweckmäßig in `/etc/bash.bashrc`, damit alle Benutzer etwas davon haben. Ausloggen, wieder einloggen, testen mit `set | grep ayam`.

Verwendet man das Kommando `locate` zum Suchen von Dateien, empfiehlt es sich, nach dem Einrichten als Verwalter das Kommando `updatedb` aufzufufen. Man findet die Neuankömmlingen dann leichter.

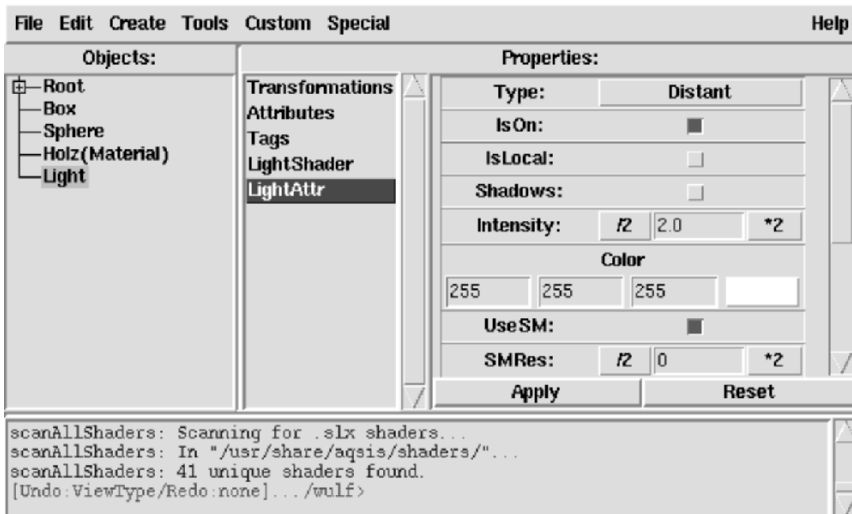


Abbildung 8.12. Screenshot des Fensters Ayam-Main

Die Datei `wood2.sl` ist ein Beispiel für einen Shader, geschrieben in der verbreiteten Shading Language (SL) von RenderMan. Solche Shader müssen mit einem zum Renderer gehörenden Werkzeug (hier `/usr/bin/aqsl`) übersetzt werden und können dann vom Renderer verarbeitet werden. Sie beschreiben – vereinfacht gesagt – optische Eigenschaften von Oberflächen oder allgemein Objekten, wie ihre Namen auch vermuten lassen. Im Netz kann man sich von mehreren Stellen solche Dateien holen, die Website von RenderMan ist ein guter Anfang. Ein Programmierer kann sich natürlich auch in die SL einarbeiten und Shader selbst entwerfen. Die Sprache zeigt Ähnlichkeiten zu C, ist aber nicht C.

Wenn Aqsis und Ayam eingerichtet sind, kann die Arbeit losgehen. Bei Aqsis gibt es nicht zu konfigurieren, Ayam ist zuerst mit den Shadern von Aqsis bekannt zu machen, dann erschaffen wir eine Kugel, die auf einer Platte liegt:

- Wir rufen aus der Kommandozeile Ayam auf und erhalten einen Splashscreen, der nach einigen Sekunden von allein verschwindet, sowie drei bleibende Fenster:

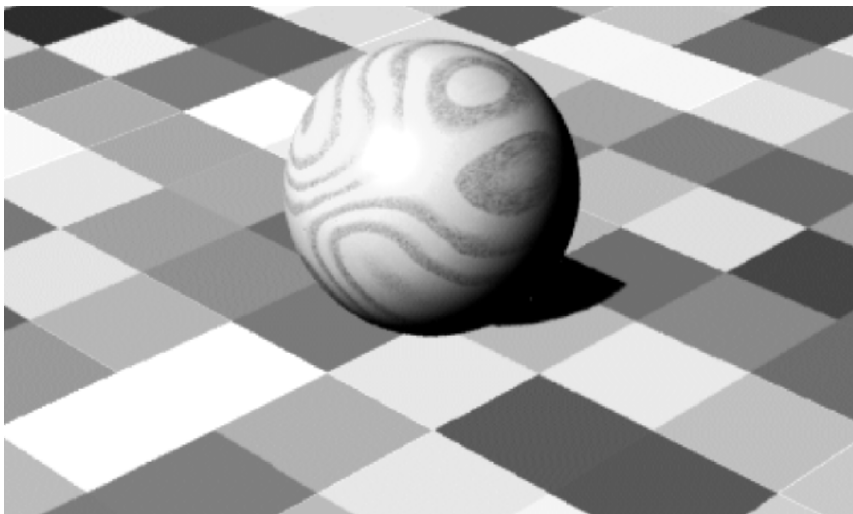


Abbildung 8.13. Screenshot eines Ayam-Modells (Kugel auf Platte), gerendert mit Aqsis, paralleler Lichteinfall. Der Renderer hat offenbar beim Schatten der Kugel auf halber Strecke die Lust verloren.

- Tools, mit 56 Buttons für Werkzeuge, ähnlich wie bei GIMP oder anderen Mal- und Zeichenprogrammen,
- Ayam-Main, seinerseits unterteilt in eine Menüleiste oben, links eine Objects-Liste, rechts eine zweigeteilte Properties-Box und unten eine Textleiste, in der das Werkzeug Meldungen ausgibt, die sich in der Log-Datei `/tmp/ay.log` wiederfinden,
- ein grafisches Fenster, View genannt, mit einer Menüleiste.

Wie üblich aktiviert man ein Fenster durch Linksklick in die Kopfleiste. Der Apply-Button ist das, was auf der Tastatur die `<cr>`-Taste ist: erst nach seiner Betätigung wird eine Eingabe wirksam. Abbildung 8.11 zeigt den ganzen Bildschirm, Abbildung 8.12 Ayam-Main.

2. Wird Ayam zum ersten Mal gestartet, existiert noch keine View. Wir erzeugen sie durch Anklicken des Menüpunktes *Create – View* in Ayam-Main.
3. In Ayam-Main wählen wir den Menüpunkt *File – Load Custom* und öffnen die Datei `/usr/local/ayam/bin/plugins/ayslx.so`, einen Parser (Leseprogramm) für `slx`-Dateien, wie sie Aqsis mitbringt. Unter Umständen muss man sich dazu durch ein paar Verzeichnisse hangeln. Erfolg oder Misserfolg wird in Ayam-Main unten verkündet. Tritt ein Problem wegen fehlender Bibliotheken auf, ermittelt man durch den Aufruf `ldd ayslx.so`, um welche Bibliothek es sich dreht, und lädt sie wie oben schon `liblog4cpp` nach. Dazu ist eventuell vorher in den Debian-Paketlisten nach dem Paket zu suchen, das die Bibliothek enthält.
4. Dann wählen wir in Ayam-Main den Menüpunkt *Edit – Preferences*, ein neues Fenster namens *Ayam Preferences* öffnet sich. Dort ist in der Zeile *Shaders*

das Shader-Verzeichnis einzutragen, also `/usr/share/agsis/shaders/`. Apply-Button betätigen. Anschließend ist der Button *Scan Shaders* anzuklicken. In dem unteren Teilfenster von Ayam-Main wird nach kurzer Zeit hoffentlich ein Erfolg gemeldet, andernfalls haben wir ein Problem. Wir prüfen auch die weiteren Angaben in dem Fenster und korrigieren sie erforderlichenfalls. Nach Anklicken des Buttons *OK* verschwindet das Preferences-Fenster.

5. Endlich werden wir schöpferisch tätig: Im Tools-Fenster klicken wir den Button mit einem Würfel an (dritte Zeile, erste Spalte, Fähnchen *create Box*). Im Views-Fenster taucht ein weißes Quadrat mit dem Mittelpunkt im Koordinatenursprung auf, zugleich in Ayam-Main unter den Objekten eine Box. In den zugehörigen Properties (Eigenschaften) wählen wir *Transformations* -> *Scale* und setzen diese auf die Werte $100 - 0.1 - 100$, um aus dem Würfel eine flache Unterlage (Tischplatte) zu machen. Nach Betätigen des Buttons *Apply* ändert sich die View entsprechend.
6. Auf gleiche Weise erzeugen wir durch Anklicken des Kugel-Buttons (dritte Zeile, zweite Spalte, Fähnchen *create Sphere*) im Tools-Fenster eine noch unvollkommene Kugel.
7. Wir schieben die Kugel nach oben, bis sie auf der Unterlage liegt, indem wir im Tools-Fenster den *move*-Button (erste Zeile, erste Spalte) anklicken und dann im View-Fenster einen beliebigen Punkt der Kugel mit gedrückter linker Maustaste verschieben.
8. Nun kommt ein Shader ins Spiel, um der Kugel bestimmte Materialeigenschaften zu verleihen. In Ayam-Main kontrollieren wir, dass unverändert die Kugel als Objekt ausgewählt ist, und wählen den Menüpunkt *Edit – Material* aus. Ein Fensterchen öffnet sich, wir geben dem Material den Namen *Holz*. Gleichzeitig erscheint *Holz* als neues Objekt in Ayam-Main. Aus dessen Properties wählen wir *Surface* (Oberfläche) aus und können dann einen Shader zuordnen (*set new Shader*). Wir suchen den wie oben beschrieben bei RenderMan kopierten und von uns mittels `ags1` übersetzten Shader *wood2* aus. Sie können auch *plastic* nehmen, aber das sieht nicht so schick aus. Den *Apply*-Button nicht vergessen. Für die Unterlage brauchen wir kein Material, als Default wird der Shader *matte* genommen. Wir können selbstverständlich auf gleichem Weg etwas Edleres wählen, beispielsweise *expensive* oder *cellnoisetest*.
9. Es werde Licht. Im Tools-Fenster klicken wir auf den Button mit der Glühbirne (fünfte Zeile, zweite Spalte, Fähnchen *create Light*). In Ayam-Main taucht ein neues Objekt auf, von dessen Eigenschaften wir *LightAttr* auswählen. Wir setzen den Typ von *Custom* auf *Distant*, aktivieren die Buttons *IsOn* (eine ausgeknipste Lichtquelle bringt uns nicht weiter) und *Use SM* (Shadow Maps verwenden), können die Intensität verdoppeln und vergessen den *Apply*-Button nicht.
10. Dann wählen wir *Transformations – Rotate_X* und setzen den Wert von 0 auf -45.0, *Apply*-button. Damit geben wir eine Richtung für das Licht vor.
11. Um den schrägen Lichteinfall in der View sehen zu können, müssen wir uns die Kugel von der Seite anschauen. Dazu wählen wir im View-Fenster den Menüpunkt *Type – Side* aus (bisher war es *Front*, also von vorn gesehen).

12. Eine gerenderte Vorschau erhalten wir – falls wir es wünschen – durch Anklicken des Menüpunktes *View – Quick Render* im View-Fenster. Es öffnet sich ein viertes Fenster mit einer alleinstehenden hölzernen Kugel.
13. Wir selektieren in Ayam-Main die Kugel (sphere) und wählen im View-Fenster den Menüpunkt *Configure – Zoom to Object*. Dadurch werden Größe und Lage der Kugel verbessert.
14. Wir nähern uns der Realität weiter an und klicken im View-Fenster auf den Menüpunkt *Type – Perspective*.
15. Wir erzeugen die Shadow Map durch Anklicken des Menüpunktes *View – create ShadowMaps* im View-Fenster.
16. Nun wählen wir im View-Fenster den Menüpunkt *View – Render*, um unsere Kugel endgültig zu rendern, was je nach Rechenleistung der Maschine ein paar Sekunden dauert.
17. Man kann jederzeit zurückgehen und beispielsweise das Material der Tischplatte oder den Winkel der Lichtquelle ändern. Dabei nicht den *Apply*-Button vergessen, sonst tut sich nichts.
18. Die ganze Pracht wird durch Anklicken des Menüpunktes *File – Exit* in Ayam-Main beendet (wie bei vielen Programmen).

Die Einstellungen (preferences) werden bei Ende der Arbeit in der Datei `$HOME/.ayamrc` abgelegt, wo sie nachgelesen und mit einem Texteditor editiert werden können, wenn man sie nicht in Ayam selbst ändern will. Abbildung 8.13 zeigt das Ergebnis. Bei der Erzeugung des Schattens hat der Renderer (Aqsis) auf halber Strecke die Lust verloren. Entweder stimmt eine Einstellung nicht – mea culpa – oder die aktuellen Versionen der beiden Werkzeuge verstehen sich noch nicht ganz. Ein zweiter Versuch mit einem Spotlight (ConeAngle = 20, Intensität = 300) zeitigte das erwartete Ergebnis, siehe Abbildung 8.14.

Der Renderer 3Delight (<http://www.3delight.com/>) ließ sich problemlos einrichten:

- *3Delight User's Manual* (PDF, HTML) herunterladen, ansehen (137 Seiten),
- Tarball mit der Software holen und auspacken,
- Datei `INSTALL` lesen, `./install` aufrufen,
- `source .3delight_bash` bzw. `source .3delight_csh` aufrufen, setzt Umgebungsvariable mit den Pfaden `/usr/local/3delight...`,
- Pfade für gewöhnliche Benutzer einrichten: Datei `.3delight_bash` nach `/etc/` kopieren, die Datei in `/etc/bash.bashrc` sourcen, Umgebungsvariable `PATH` erweitern,

war aber trotz Renderman-Kompatibilität nicht zur Zusammenarbeit mit Ayam zu bewegen, was an unterschiedlichen Compilern für Ayam und 3Delight liegen soll, siehe Ayam-FAQ. Ein Ausweg wäre das Herunterladen der Quellen und Übersetzen auf der lokalen Maschine. Als RenderMan-kompatible Renderer kommen weiterhin in Betracht:

- Pixie, <http://sourceforge.net/projects/pixie/> und <http://www.cs.berkeley.edu/~okan/Pixie/pixie.htm>, für X11,

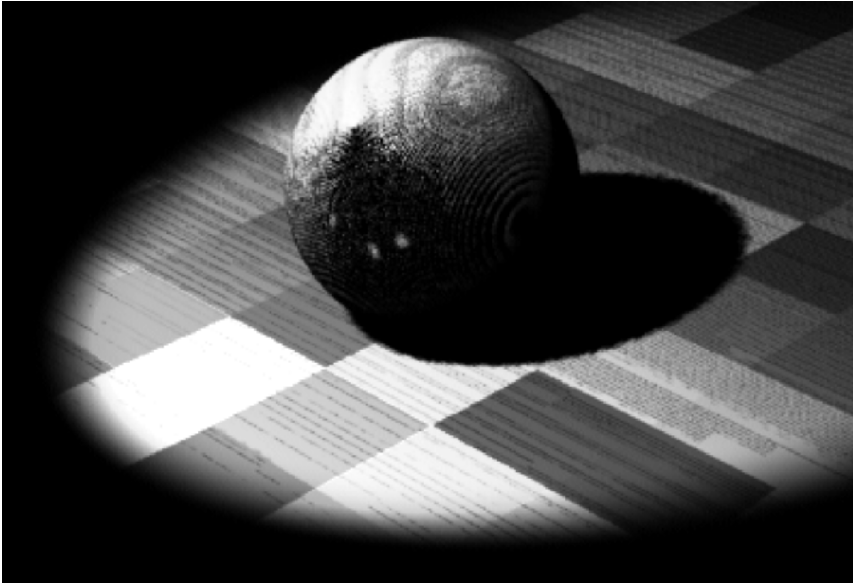


Abbildung 8.14. Screenshot eines Ayam-Modells (Kugel auf Platte), gerendert mit Aqsis, diesmal mit einem Spotlicht, in Ordnung.

- Angel, <http://www.dctsystems.co.uk/RenderMan/>
- sowie einige kommerzielle Produkte, darunter das Original, Pixars PhotoRealistic RenderMan (PRMan).

Hat man diesbezügliche Absichten, sollte man sich frühzeitig im Netz oder bei den Software-Autoren nach Erfahrungen erkundigen. Die Aufgabe ist komplex.

Für und bei Debian stehen drei raytracing-fähige Renderer zur Verfügung, alle drei mit eigenen Einträgen in der deutschen Wikipedia:

- Der Persistence Of Vision Raytracer (POV-Ray) aus der Non-free-Abteilung, ein reiner Renderer, <http://www.povray.org/>,
- Blender, zum Modellieren, Animieren und Rendern, <http://blender3d.org/> oder <http://www.blender.org/>, mit einer interessanten Anwendung (Open Movie) auf <http://orange.blender.org/>,
- YafRay (Yet another free ...), zum Animieren und Rendern, <http://www.yafaray.org/>.

POV-Ray und Blender sind zwei große, etablierte Werkzeuge, YafRay ist jünger und steht Blender nahe.

POV-Ray verarbeitet 3D-Modelle, die in einer besonderen Szenenbeschreibungssprache (scene description language, SDL) definiert sind. Die Sprache hat wie RenderMans Shading Language Ähnlichkeiten mit C/C++. Beherrscht man sie, kann man direkt Objekt- und Szenenbeschreibungen schreiben. Modellierwerkzeuge unter Linux mit Ausgabe für POV-Ray sind unter anderem:

- AC3D, <http://www.ac3d.org/>, kommerziell, aber bezahlbar,
- CYCAS, <http://www.cycas.de/>, teilkommerziell, insbesondere für Architekten geeignet,
- Art of Illusion, <http://www.artofillusion.com/>, in Java geschrieben, ein Werkzeug zum Modellieren, Animieren und Rendern, mit einem umfangreichen Handbuch (POV-Ray-Ausgabe?),
- Irit, <http://www.cs.technion.ac.il/~irit/>, mit einem Filter Irit2pov und ausführlicher Dokumentation,
- Sced, <http://freshmeat.net/projects/sced/>, macht keinen sehr lebendigen Eindruck, ist aber noch verfügbar,

und vermutlich weitere. Eine interessante Site ist auch <http://objects.povworld.org/> mit einer Sammlung fertiger Objekte. POV-Ray ist sehr leistungsfähig. Beispielsweise kann der Raytracer die Brechung des Lichtes durch ein Prisma darstellen. Er verlangt allerdings auch Rechenleistung und unterstützt daher verteiltes Rendern (Renderfarmen) mittels PVM.

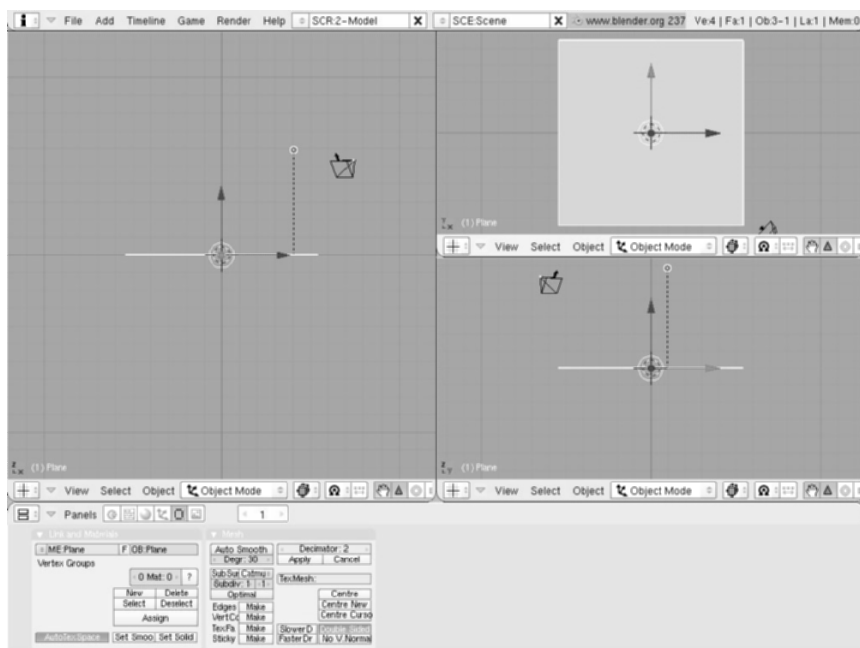


Abbildung 8.15. Screenshot des 3D-Werkzeugs Blender, am Anfang der Arbeit, nach der Erzeugung des Bühnenbodens. Für solche Arbeiten braucht man einen großen Bildschirm.

Blender ist – wie erwähnt – ein Komplettwerkzeug, ursprünglich entwickelt in einem professionellen Studio zum internen Gebrauch, nicht zum Verkauf oder zur freien Verbreitung. Benutzerfreundlichkeit war nie das vorrangige Ziel, sondern Leistung. Es kann als Debian-Paket oder als Tarball (statisch oder dynamisch gelinkt)

heruntergeladen werden. Beide Installationen verliefen problemlos; das Werkzeug besteht nur aus zwei Programmen und einem Verzeichnis für Plugins. Wir wollen ein kleines Beispiel durchexerzieren und folgen grob einer Anleitung von KATJA SOCHER, die auf <http://www.tuxgraphics.org/> einige Tutorials zu Blender, GIMP, ImageMagick und weiteren Werkzeugen veröffentlicht hat. Ein Bildschirm von 19 Zoll oder größer und ordentlich Arbeitsspeicher (512 MB oder mehr) sind kein Luxus bei derartigen Arbeiten. Blender greift sich den ganzen Bildschirm, nicht nur ein X11-Fenster. Deshalb mussten die Screenshots mit der zeitverzögerten Aufnahme von GIMP (`plug_in_screenshot`, *Datei -> Holen -> Screen Shot*) geschossen werden statt mit `xwd` wie sonst. Will man das Werkzeug unterwegs abbrechen, reicht ein Druck auf die Taste `<q>` (quit). Hat man sich rettungslos verkonfiguriert, löscht man das Verzeichnis `$HOME/.blender/` und beginnt von vorn. Die folgenden Schritte beziehen sich auf Version 2.37a, statisch gelinkt:

1. Start aus der Kommandozeile mit `blender` bzw. `blenderstatic`, falls kein Symlink angelegt wurde. Ein bildschirmfüllendes Fenster mit einem Splashscreen darin öffnet sich. Der Splashscreen verschwindet nach einer Mausbewegung oder Drücken der harmlosen Taste `<Shift>`. Wir lassen uns von der Vielzahl der Knöpfchen beeindrucken, aber nicht abschrecken.
2. Der Schirm besteht aus drei Fenstern oder Flächen (Area): am oberen Rand eine schmale Menüleiste (User Preferences Window), darunter die Arbeitsfläche (3d View) mit einer Menüleiste unten und am unteren Bildschirmrand eine breitere Leiste mit Knöpfchen (Buttons Window). In jedem Fenster linksaußen findet sich ein Window-Type-Button, der Auskunft über die Art des Fensters gibt und mit dem sich die Art wechseln lässt. Abbildung 8.15 lässt die Aufteilung im Zustand nach einigen Arbeitsschritten erkennen.
3. Auf der Arbeitsfläche existieren bereits ein Würfel und eine Leuchte. Links unten auf der Arbeitsfläche wird das für die nächste Aktion ausgewählte Objekt genannt, hier *Cube*. Da wir keine Verwendung für den Würfel haben, löschen wir ihn mittels der Taste `<x>` (extinguish). Die Rückfrage beantworten wir mit `<cr>`. Dann wählen wir mit der rechten Maustaste die Leuchte (kleiner gelber Kreis) aus und beseitigen sie ebenso. Übrig bleibt ein Gebilde aus schwarzen Strichen, die Kamera. Die brauchen wir.
4. Befindet sich der Mauscursor (Pfeil) auf einer Arbeitsfläche, so öffnet die Leertaste ein Menü mit häufig verwendeten Funktionen wie Hinzufügen (Add), Auswählen (Select) oder Editieren (Edit) von Objekten (Flächen, Körper, Text, Leuchten, Kamera ...). Die linke Maustaste verankert den 3d-Cursor (Fadenkreuz) an einer neuen Position. Dort erscheint das nächste hinzugefügte Objekt. Die mittlere Maustaste ermöglicht in manchen Zusammenhängen Rotationen oder Translationen. Mittels der rechten Maustaste lässt sich ein ausgewähltes Objekt auf eine neue Position ziehen, wo es mit der linken Maustaste verankert wird. Sie können das mit den vorgegebenen Objekten durchspielen, sollten danach aber noch einmal von vorn beginnen.
5. Da unsere Szene räumlich werden soll, unterteilen wir die Arbeitsfläche in drei Teilfenster, die wie eine technische Zeichnung unsere Szene von oben, von vorn

und von der Seite zeigen. Wir bewegen den Mauscursor (Pfeil) auf die Grenze zwischen Arbeitsfläche und oberer Menüleiste. Der Cursor wechselt sein Aussehen zu einen Doppelpfeil. Wir betätigen die mittlere Maustaste und erhalten ein kleines Menü, aus dem wir mit der linken Maustaste den Punkt *Split Area* auswählen. Ein senkrechte Trennlinie erscheint, die wir in die Bildmitte ziehen und dort durch einen Klick mit der linken Maustaste verankern. Auf gleiche Weise unterteilen wir die rechte Hälfte durch eine waagrechte Trennlinie. Falls die waagrechte Trennlinie in der linken Bildhälfte erscheint, bewegen wir den Mauscursor vor dem Linksklick nach rechts.

6. Die drei Teilfenster zeigen zunächst dieselbe Ansicht (von oben, Top). Wir wählen mit der linken Maustaste am unteren Teilfensterrand jeweils das Menü *View* und daraus im linken Teilfenster den Punkt *Front*, rechts oben *Top* (voreingestellt) und rechts unten *Side*. Damit haben wir alle drei Ansichten unserer Szene vor Augen. Die Ansichten sind miteinander verknüpft. eine Änderung in einer Ansicht wirkt sich auch in den beiden anderen aus.
7. Wir brauchen nun eine Kamera sowie eine Bühne mit Boden und Hintergrund und natürlich Licht. Die Kamera sollte bereits vorhanden sein, ein eigenartig aussehendes Gebilde aus schwarzen Strichen in jeder der drei Ansichten. Falls die Kamera fehlt, bauen wir sie gemäß Punkt 4, Hinzufügen. Hat aber noch Zeit.
8. Den Bühnenboden erzeugen wir in der Top-Ansicht (rechts oben), indem wir den Mauscursor in die Ansicht bewegen und dann die Leertaste drücken. Aus den auftauchenden Menüs wählen wir mit der linken Maustaste die Punkte *Add* -> *Mesh* -> *Plane*. Im Zentrum öffnet sich ein Quadrat mit gelben Eckpunkten, vergleiche Punkt 4, Hinzufügen. Wir könnten jeden der Punkte editieren, wollen das jedoch nicht und drücken deshalb <Tab>. Der Rand des Quadrats nimmt die Farbe Pink an. Nun drücken wir <s> (size) und bewegen den Mauscursor, bis das Quadrat noch etwas Abstand zum Rand hat. Mit der linken Maustaste fixieren wir die neue Größe. Abbildung 8.15 zeigt den zugehörigen Bildschirm.
9. In der Seitenansicht (rechts unten) bewegen wir den Bühnenboden mit der rechten Maustaste ein Stück nach unten und fixieren ihn dort mit der linken Maustaste. Versuchen Sie, sich an Hand der drei Ansichten die Szene aus Boden und Kamera räumlich vorzustellen. Die Kamera soll von rechts oben vorn auf die Bühne schauen.
10. Nun bekommt der Boden Farbe. Wir brauchen dazu das Buttons Window unten auf dem Bildschirm. Wir drücken <F5> (oder klicken mit der linken Maustaste im Buttons Window auf die Kugel, Fähnchen *Shading*). Aus dem neu erscheinenden Menü wählen wir mit der linken Maustaste *Material* -> *Add New*. Es tun sich neue Knöpfchen auf, darunter drei unscheinbare Schieber mit den Bezeichnungen R (rot), G (grün) und B (blau). Der Boden soll grün werden, also setzen wir mit der linken Maustaste den Rot- und den Blauschieber auf null. Gleichzeitig nimmt unser Boden im Top-Fenster (rechts oben) die neue Farbe an. In den beiden anderen Fenster ist nichts zu sehen, da der Boden dort nur ein Strich ist. Auf eigene Gefahr dürfen Sie auch eine andere Farbe mischen.
11. Wie in den Punkten 8 bis 10 dargestellt zimmern wir auch die Rückwand. Wir beginnen jedoch in der Frontansicht (links), da sich nur dort die Rückwand als

Fläche darbietet. Wir ziehen sie in der Seitenansicht mit der rechten Maustaste an den hinteren Rand des Bodens, fixieren sie mit der linken Maustaste und geben ihr die Farbe Blau.

12. Es werde Licht. Mit dem Mauszeiger auf der Frontansicht drücken wir gemäß Punkt 3 die Leertaste und wählen mit der linken Maustaste die Menüpunkte *Add -> Lamp -> Spot*. Auf der Bühne steht uns nun in allen drei Ansichten ein Spotlicht (Scheinwerfer) zur Verfügung. Wir positionieren es in zwei Ansichten so, dass es von vorn oben links auf das Zentrum der Bühne leuchtet. Die Schritte lassen sich beliebig wiederholen, bis alles stimmt:
 - a) Mit dem Mauscursor in der Seitenansicht drücken wir die Taste <g> und verschieben durch Bewegungen des Mausursors den Spot nach links oben, das heißt im Raum nach vorn oben. Fixieren durch Linksklick.
 - b) Mit dem Mauscursor in der Top-Ansicht drücken wir die Taste <g> und schieben den Spot nach rechts. Dann drücken wir die Taste <r> (rotate) und drehen die Lichtquelle, bis sie das Zentrum der Bühne ausleuchtet. Die zum Spot gehörenden Hilfslinien erleichtern die Ausrichtung. Voraussichtlich muss man ein paar Mal korrigieren.
13. Nachdem alles so gut geklappt hat, gönnen wir uns noch zwei Leuchten. Wir positionieren den 3d-Cursor (Fadenkreuz, Linksklick) über und hinter dem Bühnenzentrum und erzeugen dort ein Backlight mittels *Add -> Lamp -> Lamp*. Schließlich bringen wir hinter der Kamera auf gleiche Weise eine Leuchte zur Aufhellung der Schatten an. Vorher überlegen, in welcher Ansicht man zweckmäßig beginnt. Nachprüfen, wo die Leuchten hängen.
14. Die Helligkeit der Leuchten wird verändert, indem man im Buttons Window den Schieber *Energy* verstellt. Momentan nebensächlich.
15. Die Bühne ist fertig. Jetzt rendern wir sie, um einen Eindruck von unserer Schöpfung zu bekommen. Dazu drücken wir <F10> oder im Buttons Window das Knöpfchen *Scene*. Als Bildgröße wählen wir *SizeX:640* und *SizeY:480*, das reicht fürs Erste. Wir klicken im Buttons Window mit der linken Maustaste in den Button *Format -> SizeX*, ein roter Strich erscheint, der auf <Backspace>, Pfeil- und andere Tasten reagiert. Die Eingabe der Werte mit <cr> abschließen. Wollen wir Schatten sehen – wofür uns momentan Objekte fehlen – ist das Knöpfchen *Shado* zu drücken, seine Farbe wechselt zu Dunkelgrün (= ein). Schatten kosten beim Rendern Rechenzeit. Schließlich starten wir den Rendervorgang mittels <F12> oder des Knopfes *Render* im Buttons Window.
16. In einem neuen Fenster erblicken wir unsere Bühne. Die Taste <F11> schickt das Renderfenster in den Hintergrund. Nochmaliges Drücken von <F11> bringt es wieder in den Vordergrund, ohne erneut zu rechnen. Voraussichtlich gibt es anfangs einiges zu korrigieren, insbesondere die Kameraneigung, siehe oben beim Spot, Punkt 12.
17. Nun speichern wir die leere Bühne samt Beleuchtung ab. Wir wählen im User Preferences Window (ganz oben) per Linksklick den Menüpunkt *File -> Save as* und nehmen den Dateinamen *buehne.blend*. Die Kennung *.blend* ist zwingend. Wir verlassen Blender mittels <q> und prüfen, ob die Datei existiert und nicht leer ist. Zeit für eine Kaffeepause.

18. Wir starten Blender erneut und holen uns über den Menüpunkt *File -> Open* mit der mittleren Maustaste unsere vorhin abgespeicherte Bühne zurück.
19. In der Bühnenmitte erzeugen wir eine Kugel, indem wir den Mauscursor in die Frontansicht schieben, die Leertaste drücken und aus dem Menü *Add -> Mesh -> UVSphere* auswählen. In allen Ansichten erscheint die Projektion einer Kugel, ein Kreis. Vorschlag für die Anzahl der Segmente und Ringe annehmen. Editiermodus mit *<Tab>* beenden, Kugel wie unter Punkt 10 beschrieben nach Belieben färben. Mittels *<F9>* Editiermodus aufrufen und Button *Link and Materials -> Set smooth* einschalten, um die Oberfläche zu glätten. Kugel links vorn auf die Bühne positionieren, zwischendurch immer mal wieder rendern (*<F12>*).
20. Als zweites Objekt auf der Bühne erzeugen wir einen Würfel. Mit dem Mauscursor in der Frontansicht betätigen wir die Leertaste und wählen *Add -> Mesh -> Cube*. In allen drei Ansichten taucht ein Würfel auf samt seiner Bezeichnung jeweils links unten. Wir färben ihn wie unter Punkt 10 beschrieben gelb (*R = 1, G = 1, B = 0*).
21. Nun bringen wir den Würfel in eine hübsche Lage. Dazu verschieben und drehen wir ihn in allen drei Ansichten, bis von vorn und von der Kamera wenigstens zwei Seitenflächen zu erkennen sind. Wir beginnen mit dem Verschieben und rendern zwischendurch immer wieder, um das Ergebnis zu begutachten. Beim Drehen spielt das Drehzentrum eine Rolle. Wir können es über das Menü *Rotation/Scaling Pivot* in der Menüleiste am unteren Rand der Ansichten ändern. Voreingestellt ist das Zentrum des Objektes. Abbildung 8.16 zeigt ein Ergebnis von Laienhand. Da Blender kein Online-Renderer ist, sondern Modellieren und Rendern nacheinander durchführt, muss man zwischen Ändern des Modells in den Ansichten und Rendern hin- und herspringen. Um die Rechenzeit nicht unnötig hoch zu treiben, ist während des Ausprobierens die Qualität des gerenderten Bildes niedrig zu halten.
22. Wollen wir das gerenderte Bild speichern, drücken wir *<F10>*, falls noch nicht geschehen. Im Buttons Window geben wir links unter *Output* in der ersten Zeile (Fähnchen *Directory/name to save rendered Pics to*) den Namen der Bilddatei ohne Kennung ein. Dann tippen wir unter *ANIM* in das Feld *End*: den Wert 1 ein, wie unter Punkt 15 bereits die Bildgröße. Unser Kunstwerk wird dann als Animation mit einem einzigen Bild abgelegt. Unter *Format* sollte als Dateiformat *Jpeg* eingestellt sein. Nun betätigen wir den *ANIM*-Knopf und finden nach Verlassen des Blenders eine Datei mit dem vorgeschlagenen Namen, daran angehängt eine Ziffernfolge (0001) und die übliche Kennung. Mit GIMP kann das Bild angezeigt und weiter bearbeitet werden. Nicht vergessen, die ganze Szene wie unter Punkt 17 beschrieben für die Nachwelt zu retten. Wir sollten dann irgendwo zwei *blend*-Dateien und eine *jpg*-Datei finden.

Die Erläuterung war etwas ausführlicher als sonst, Werkzeuge wie Blender sind nicht ohne Anleitung zu handhaben. Viele Knöpfchen auf dem Bildschirm kennen wir noch gar nicht. Mangels Erfahrung geraten auch die ersten Ergebnisse nicht gerade glänzend. Falls Sie Appetit auf Mehr bekommen haben, sehen Sie sich auf <http://www.blender3d.org/> nach Tutorials und Vorlagen um.

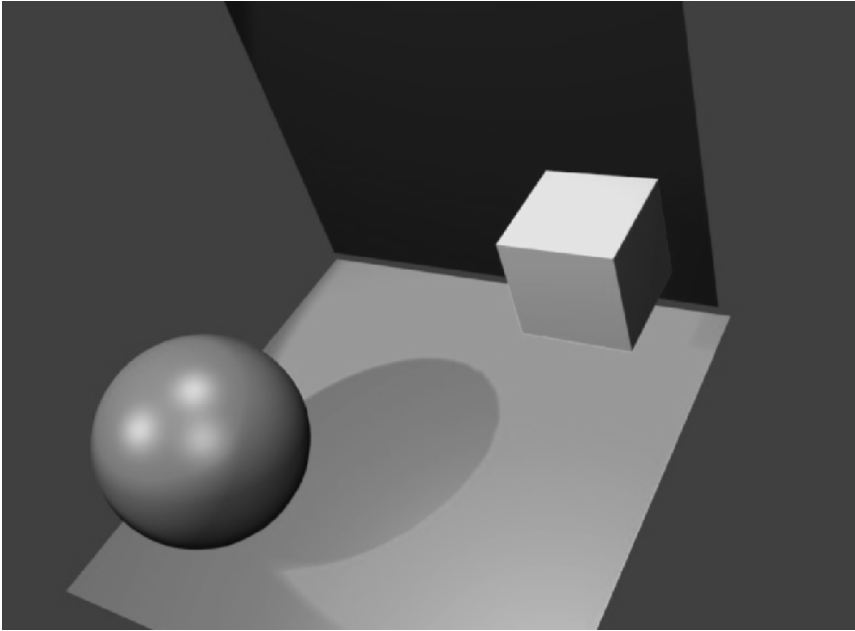


Abbildung 8.16. Ergebnis des 3D-Werkzeugs Blender, die gerenderte Bühne mit einer Kugel und einem Würfel. Kein Screenshot, sondern von Blender abgespeichert.

8.9 Virtuelle Welten

8.9.1 Was ist Virtuelle Realität?

Eine Fortführung von animierter Grafik ist die Virtuelle Realität (VR, E: virtual reality, F: réalité virtuelle). Darunter versteht man die rechnergestützte Erzeugung und Wiedergabe von Umgebungen oder Welten, die der Benutzer an Stelle der Realität wahrnimmt. Bekannte Vertreter sind manche Spiele und Simulatoren (Trainer). Der Aufwand wächst mit der verlangten Realitätsnähe. Der Benutzer taucht in die virtuelle Umgebung ein, was als Immersion bezeichnet wird und verschieden stark ausgeprägt sein kann. Zur VR gehört:

- Hardware, vor allem für die Mensch-Maschine-Schnittstelle,
- Echtzeitverhalten,
- Ansprechen mehrerer Sinne (Multimedia), mindestens Sehen und Hören,
- Interaktion des Benutzers mit der virtuellen Umgebung.

Die Mensch-Maschine-Schnittstelle soll den realen Verhältnissen nahe kommen. Ein Fahrzeug soll also nicht per Tastatur, sondern mit einem Lenk- oder Steuerrad dirigiert werden. Manchmal passt sich auch die Realität der Rechnerwelt an: Schiffe werden zunehmend mit einem Joystick gesteuert. Der visuelle Eindruck soll räumlich sein, wofür besondere Vorrichtungen wie Shutterbrillen oder Head-Mounted-Displays erforderlich sind. Das Echtzeitverhalten betrifft vor allem den Renderer,

dem nur wenig Zeit zur Darstellung eines geänderten Modells bleibt. Um Bewegungen ruckelfrei wiederzugeben, müssen 20 oder mehr Bilder pro Sekunde erzeugt werden. Die Interaktion des Benutzers mit der virtuellen Umgebung muss den gewohnten physikalischen Gesetzen gehorchen, sonst ist die VR als Simulator unbrauchbar und als Spiel gewöhnungsbedürftig. Das hört sich teuer an und ist es auch, aber bei Abstrichen an der Realitätsnähe reicht ein PC unter Debian GNU/Linux. Schließlich gab es schon unter DOS Flugsimulatoren.

8.9.2 Ein Anfang (OpenSG)

Das Debian-Paket *openscenegraph* sowie die Website <http://www.opensg.org/> bieten einen Einstieg. OpenSG ist kein vollständiges VR-System, sondern ein Echtzeit-Renderer unter Verwendung der OpenGL-Bibliothek. Die gesamte Szene wird in einem Graphen erfasst (scenegraph), der es ermöglicht, das Rendern auf die jeweils sichtbaren Objekte zu beschränken und so Zeit zu sparen. Als Ergänzung werden OpenTracker (<http://www.studierstube.org/opentracker>) und VR Juggler (<http://www.vrjuggler.org/>) empfohlen, beide nur als Tarball für Linux und andere Systeme erhältlich.

Zur Darstellung von interaktiven dreidimensionalen Szenen im Web dient die *Virtual Reality Markup Language* (VRML). Auf der Website <http://www.openvrml.org/> findet sich dazu Material, im Debian-Paket *openvrml-lookat* ein Betrachter namens *lookat*.

Das digitale Tonstudio

Hier werden einige Werkzeuge zur Aufnahme, Bearbeitung und Wiedergabe akustischer Daten (Schall, Sound) vorgestellt.

9.1 Grundbegriffe

9.1.1 Vom Schall zum Byte und zurück

Schall (E: sound, F: son) ist ein analoges Signal, nämlich der Schalldruck als Funktion von Ort und Zeit. Damit Schallsignale von einem Rechner verarbeitet werden können, sind sie zunächst in elektrische Signale zu wandeln, wofür Mikrofone verwendet werden, und die analogen elektrischen Signale zu digitalisieren, was Analog-Digital-Wandler bewerkstelligen. Die Digitalisierung kann nicht kontinuierlich geschehen, sondern nur in kurzen Zeitabständen, ähnlich wie bei Filmaufnahmen schnell hintereinander viele Stillbilder aufgenommen werden. Das Verfahren wird als Puls-Code-Modulation (PCM) bezeichnet, weil das analoge Signal in Pulse zerhackt wird, die binär codiert werden. Vom Rechner zum Ohr geht es den gleichen Weg zurück. Statt Mikrofonen stehen hier Lautsprecher, die im Prinzip genau so arbeiten und nur anders ausgelegt sind.

Will man ein analoges Signal verlustfrei digitalisieren, so muss gemäß dem Abtasttheorem (sampling theorem) die Abtastfrequenz mindestens das Doppelte der höchsten im analogen Signal enthaltenen Frequenz betragen. So kommt die Abtastfrequenz von 44,1 kHz für Audio-CDs zustande, denn das menschliche Ohr nimmt in jungen Jahren Frequenzen bis 20 kHz wahr. Das Abtasttheorem geht auf CLAUDE ELWOOD SHANNON zurück, den Vater des Bits. Bei Aufnahme und Wiedergabe muss dieselbe Abtastfrequenz verwendet werden.

Über einen einzigen akustischen Kanal lässt sich nur ein monophones Signal aufnehmen oder wiedergeben, ab zwei Kanälen ist Stereophonie (Raumklang) möglich. Ferner spielt der Wertebereich der Samples eine Rolle, also die Anzahl der Bits oder Bytes pro Abtastwert (Auflösung, precision). Je größer die Anzahl, desto feinere Unterschiede lassen sich aufzeichnen. Gebräuchlich sind 16 oder 24 Bits,

wie bei den Farben in der Grafik. Die Rohgröße einer Audiodatei ergibt sich als Produkt aus Abtastfrequenz, Dauer der Darbietung, Bits oder Bytes pro Sample und Anzahl der Kanäle. Solche rohen Audiodateien sind unhandlich. Sie belegen erheblich Speicherplatz und brauchen bei üblichen Netzanbindungen ein Mehrfaches der Wiedergabezeit zur Übertragung. Ein kleines Beispiel: Wir wollen 1 min Schall in CD-Qualität aufzeichnen, das heißt mit einer Abtastfrequenz von 44,1 kHz bei 16 Bit Auflösung auf zwei Kanälen (Stereo). Die Multiplikation führt auf eine Dateigröße von etwas mehr als 10 MB. Die Übertragung über ein Netz mit 1 Mbit/s (ADSL) dauert 85 s, also länger als der ursprüngliche Schall, was für Echtzeitanwendungen (Streaming) untragbar wäre. Aus diesem Grund werden Audiodaten meist komprimiert oder kodiert, wobei Eigenarten des menschlichen Gehörs ausgenutzt werden. Die entsprechende Hard- oder Software heißt Encoder, das Gegenstück am Ort der Wiedergabe Decoder. Die komprimierte Audiodatei hat nur ungefähr ein Zehntel der Größe der rohen Datei. Bei Abstrichen an der Qualität – was bei Sprache gut möglich ist – lassen sich die Daten noch stärker zusammenquetschen.

Zum Thema Sound sind mehrere Linux HOWTOs veröffentlicht worden, die alle bei The Linux Documentation Project herunterzuladen sind:

- Sound HOWTO, behandelt vor allem Soundkarten, aktuell von 2001,
- Sound Playing HOWTO, 1998, als Übersicht zu gebrauchen, viele Einzelheiten veraltet,
- MIDI HOWTO, 2002,
- MP3 HOWTO, 2001,
- Alsa Sound mini-HOWTO, 1999, veraltet, die ALSA-Webseiten sind vorzuziehen, siehe unten.

Ferner ist der *Linux Audio Users Guide* auf <http://www.djcj.org/> zu nennen, der neben eigener Dokumentation weitere HOWTOs und FAQs auflistet. Die *FAQs on Digital Signal Processing* liegen unter anderem auf <http://www.bdti.com/faq/>. Grundbegriffe der Elektroakustik werden in der deutschen Wikipedia erklärt.

Da zu einem voll ausgebauten digitalen Tonstudio mehrere Werkzeuge gehören und diese zusammenarbeiten müssen – was alles andere als selbstverständlich ist – wurde 2002 in der EU das Projekt *Agnula* (A GNU/Linux Audio Distribution) gestartet, das eine zweckmäßige Auswahl von Multimedia-Anwendungen vereint und auf <http://www.agnula.org/> zu finden ist. Die Debian-basierte Ausgabe nennt sich DeMuDi (Debian Multimedia Distribution) und ist auf <http://demudi.agnula.org/> zu Hause. Dort kann man sich ein vollständiges Debian-System herunterladen, das auf Multimedia-Bedürfnisse zugeschnitten ist.

9.1.2 Dateiformate

Bei Audiodaten besteht noch mehr als bei Grafikdaten das Bedürfnis, die Daten zu komprimieren, um Speicherplatz und Übertragungszeit zu sparen. Auch hier unterscheiden wir verlustlose und verlustbehaftete Kompression, siehe Abschnitt 3.1.8 *Packen* auf Seite 82. Entsprechend sind mehrere Dateiformate für Audiodaten gebräuchlich, hier eine Auswahl:

- ungepackte Daten:
 - AIFF, Audio Interchange File Format, ursprünglich von Apple, wenn AIFF-C, dann komprimiert,
 - RAW, rohe Audio-Daten ohne Header (headerless), die Aufnahmeparameter müssen für die Wiedergabe bekannt sein, sie stehen nicht in der Datei,
 - WAV, Waveform Audio Data, eine Untermenge des Resource Interchange File Formats (RIFF) von Microsoft,
- verlustfrei gepackte Daten:
 - FLAC, Free Lossless Audio Codec, eine verlustfreie Alternative zu MP3, <http://flac.sourceforge.net/>,
 - LPAC, Lossless Predictive Audio Compression, weiterentwickelt als MPEG-4 Audio Lossless Coding (ALS), <http://www.tnt.uni-hannover.de/project/mpeg/audio/>,
- verlustbehaftet gepackte Daten:
 - AU, Sun Microsystems Audio File, mehrere Varianten,
 - MP3, die Kurzform von ISO MPEG-1 Audio Layer 3. MPEG bedeutet *Moving Picture Experts Group* (<http://mpeg.chiariglione.org/mpeg/> und <http://www.mpeg.org/>). Das Format erlaubt eine hohe Kompression bei minimalen Abstrichen an der Qualität, ist patentiert und weit verbreitet,
 - Ogg Vorbis, patentfreie Alternative zu MP3,
 - AAC, Advanced Audio Coding, Nachfolger von MP3, ebenfalls von MPEG, der einige Schwächen des älteren Formats vermeidet. Wird beispielsweise verwendet von Apples iTunes Music Store.
 - RealAudio, von RealMedia, vorwiegend für Streaming Audio eingesetzt, also für die Wiedergabe von Audiodaten aus dem Netz während des Übertragens.

Die Manualseite zum Kommando `sox` listet weitere auf. Eine *Audio File Format FAQ* liegt unter <http://www.cnpbagwall.com/audio.html>. Der Baustein, der die Kompression und die Rückwandlung vornimmt, wird Codec (Coder – Decoder) genannt. Er kann in Hardware oder in Software verwirklicht werden.

Das Musical Instrument Digital Interface (MIDI) ist kein Format zur Speicherung akustischer Daten, sondern ein Standard zur Speicherung von Spielanweisungen (sequencing information) für Musikinstrumente, eine Art von Noten für elektronische Geräte, siehe Abschnitt 9.5 *Klangerzeugung* auf Seite 342.

Einen Mittelweg zwischen Audiodateien (digitalisierter Schall) und MIDI (Anweisungen) gehen die Module oder Mods. Sie enthalten von beidem etwas. Die digitalisierten Schallschnipsel werden Samples genannt. Die Anweisungen sagen, wann welches Sample auf welcher Spur wiedergegeben werden soll. Module sind in mehreren Formaten verbreitet. Zur Wiedergabe eignen sich viele bekannte Player wie `xmms`, unter Umständen erst nach Einrichtung eines Plugins, aber auch spezielle Modul-Player wie `mikmod` (<http://mikmod.raphnet.net/>). Wer ein Modul zusammenstellen (komponieren) will, braucht einen Tracker. Schließlich gibt es Konverter für verschiedene Modulformate und von

bzw. nach MIDI. Auf <http://linux-sound.org/mod.html> und <http://www.united-trackers.org/> findet man Näheres zu Modulen.

9.1.3 Soundsysteme (ALSA, OSS)

Die Soundsoftware von Linux gruppiert sich um zwei Soundsysteme, deren wesentlicher Bestandteil Treiber für Soundkarten sind:

- die *Advanced Linux Sound Architecture* (ALSA), zu finden auf <http://www.alsa-project.org/>, <http://alsa.opensrc.org/> und <http://wiki.debian.net/?ALSA>,
- das *Open Sound System* (OSS), <http://www.opensound.com/>.

Die Namen von Debian-Paketen für ALSA enthalten alle die besagte Zeichenkette. Man soll nicht beide Systeme durcheinander benutzen, obwohl es in Einzelfällen gut gehen kann. Seit Version 2.6 verwendet der Linux-Kern als Default ALSA. Wer eine Soundkarte erwerben will, sollte sich vorher umsehen, ob sie unterstützt wird. Das ist genau wie bei Grafikkarten. Im Debian-Paket *alsa-oss* findet sich ein Wrapper *aoss*, der für OSS geschriebenen Anwendungen so einwickelt, dass sie unter ALSA laufen, aber besser ist es allemal, wenn man ohne das Wickelpapier auskommt.

In einer Soundkarte ist ein Mischer enthalten, der über eine Software – meist mit grafischer Benutzeroberfläche – angesteuert wird, die sich ebenfalls Mischer oder Mixer nennt, siehe Abschnitt 9.2 *Mischer* auf Seite 331. Ein einfacher Mischer für die Kommandozeile ist *amixer*. Mit seiner Hilfe lassen sich schnell die Mischereinstellungen ermitteln oder ändern

```
joe@debian:~$ amixer | less
```

Will man CDs abspielen, müssen mindestens der CD-Eingang und der Master-Ausgang (Summe) eingeschaltet sein und oberhalb von null stehen.

Man tut gut daran, nicht mehr als ein Soundprogramm pro Soundkarte gleichzeitig zu starten, sonst kann es Streit im Rechner geben. Bei unerwarteten Problemen mit einem ehemals funktionierenden System hilft manchmal ein Blick auf die Taskbar oder in die Prozessliste. Ein Soundprogramm belegt das Audio-Device (meist `/dev/dsp`) exklusiv (blockiert es für andere Programme) und gibt es erst einige Sekunden nach Beendigung frei. Ein Ausweg ist Software, die sich zwischen Soundkarte und Audioanwendungen einschaltet und gleichzeitig mehrere Anwendungen bedient. Ein Beispiel ist der Enlightened Sound Daemon (Esound, ESD) aus dem Debian-Paket *esound* nebst Begleitern und mit dem Heimathafen <http://www.tux.org/~ricdude/Esound.html>. Ein weiterer Vertreter ist das JACK Audio Connection Kit aus dem Debian-Paket *jackit*. Das Kit kann mehr als der Esound, stellt aber auch höhere Anforderungen an die Hardware. Näheres siehe <http://jackit.sourceforge.net/>. Die Anwendungen müssen mitspielen und JACK-fähig sein, was zunehmend gegeben ist.

Die Zugriffsrechte auf das Audio-Device müssen dem Benutzer Lesen und Schreiben gestatten, was sich durch eine entsprechende Gruppenzugehörigkeit (Gruppe `audio`) erreichen lässt. Da die meisten Audioanwendungen Rechnungen

in Echtzeit erfordern, schadet es, wenn gleichzeitig andere rechenintensive Anwendungen arbeiten. Rechner mit mehreren Prozessoren haben da Vorteile. Überhaupt sollte die Hardware für Audio- und Videoarbeiten kräftig ausgelegt sein, da anders als bei Stillbildern nicht beliebig viel Zeit zur Verfügung steht. Wenn *Chopins* Minutenwalzer¹ sich über fünf Minuten hinquält, bleibt vom Genuss nichts mehr übrig. Das Ohr wird schon bei Verzögerungen von wenigen Millisekunden unruhig. Wer viel mit Audio oder anderen Echtzeitaufgaben zu tun hat, sollte sich zwei Rechner hinstellen: einen einfachen, zuverlässigen für die Alltagsarbeit wie Email, Terminkalender usw. und einen schnellen, hochgezüchteten für die Echtzeitaufgaben. Die Anforderungen sind zu unterschiedlich – angefangen bei der Konfiguration des Kerns – als dass sie optimal in einer Kiste vereint werden könnten. Aktuelle Versionen der Werkzeuge verwenden, ausprobieren und dabei bleiben, wenn sie funktionieren.

9.2 Mischer

Mischer (E: mixer, F: console de mixage) bilden das Zentrum jeder elektroakustischen Anlage. Man kann die Soundkarte als einen Mischer mit ein paar Zusatzfunktionen (Analog-Digital-Wandlung) auffassen. Soundkarte plus Mischer-Software machen im Rechner das, was im wirklichen Leben Aufgabe von realen Mischpulten ist: Sie nehmen Signale von verschiedenen Quellen entgegen, stellen ihre Lautstärke ein und vereinigen sie zu einem Ausgangssignal, das Ganze auf Wunsch in Stereo oder mehr. Das Ausgangssignal geht auf ein Speichermedium und/oder über einen Leistungsverstärker zu den Lautsprechern (Aktivboxen = Leistungsverstärker + Lautsprecher). Manche Mischer verfügen nur über einen Ausgang, können also nicht gleichzeitig über Lautsprecher wiedergeben und in eine Datei schreiben. Unter ALSA laufende Mischprogramme sind:

- `amixer`, für die Kommandozeile, aus dem Debian-Paket *alsa-utils*, zum Konfigurieren und Überprüfen, weniger für die laufende Arbeit, unterstützt mehrere Soundkarten gleichzeitig,
- `rexima`, ein *ncurses*-basierter Mischer,
- `alsamixergui`, ein grafisches Frontend für `alsamixer`, dieses enthalten im Debian-Paket *alsa-utils*, mit Unterstützung für mehrere Soundkarten gleichzeitig,
- `asmixer`, ein kleiner Mischer im Look and Feel des AfterStep Window Managers,
- `gamix`, ein ALSA-Mischer mit grafischer Oberfläche,
- `gnome-alsamixer`, ein ALSA-Mischer für GNOME,
- `KMix`, ein KDE-Mischer

und weitere aus der Abteilung *Sound* der Debian-Paketliste. Einige CD-Spieler enthalten auch einen Mischer.

¹Der Walzer dauert etwa zwei Minuten.

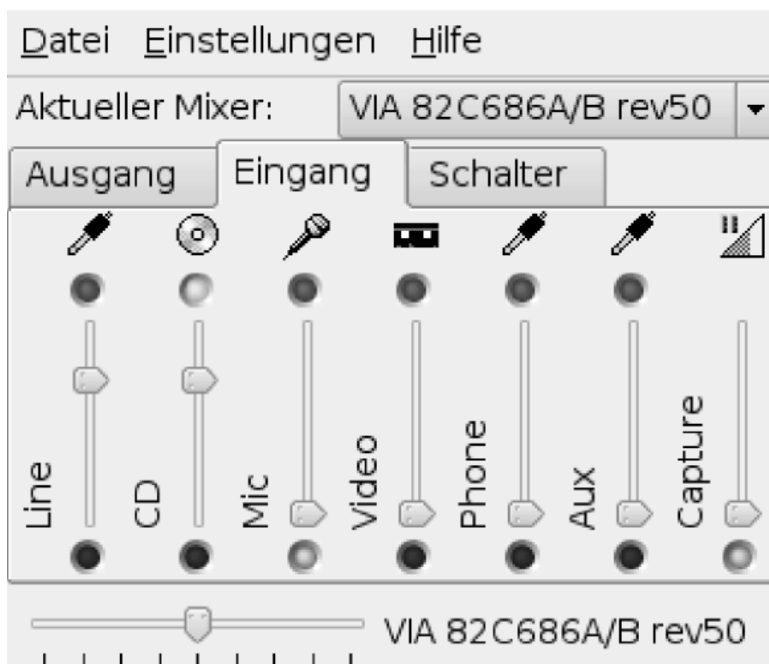


Abbildung 9.1. Screenshot des Audio-Mischers `kmix` aus dem KDE-Projekt

Abbildung 9.1 zeigt die Ansicht des Mischers KMix aus dem KDE-Projekt, und zwar das Eingangs-Mischpult. Unter der Menüleiste wird der aktuelle Mischerbaustein angezeigt für den Fall, dass mehrere verfügbar sind. Darunter folgt das Pult mit den Lautstärke-Einstellern für sieben Signalquellen und schließlich der Stereo-Balance-Einsteller (rechter - linker Kanal). Der rote Knopf unterhalb der Lautstärke-Einsteller legt ein Gerät als Aufnahmeziel fest, der grüne Knopf oberhalb schaltet einen Eingang vorübergehend stumm (mute). Grüne LEDs haben durchgehend etwas mit der Wiedergabe zu tun, rote mit der Aufnahme. Der PCM-Kanal (Puls Code Modulation) auf dem Ausgangspult wird von allen Geräten benötigt, die mit PCM-Audiodateien umgehen, wozu alle MP3-Spieler gehören. Mit den drei Karteireitern (Tabs) schaltet man auf das Ausgangs-Pult bzw. ein Pult mit einigen Schalter um. Ein Rechtsklick auf ein Gerätesymbol öffnet ein kleines Menü zu dem jeweiligen Gerät, in dem sich beispielsweise die Stereokanäle voneinander trennen lassen. Will man eine Musik-CD abspielen, reicht ein einfacher CD-Spieler (CD Player) ohne Lautstärke-Einsteller und das Mischpult, auf dem alle Werte eingestellt werden, in diesem Fall im Eingangspult mit dem Steller für die CD und auf dem Ausgangspult mit dem Steller für den Masterausgang (Summe). Es ist alles wie bei einem realen Mischverstärker. Höchstens vermisst man die Steller für die Höhen und Tiefen (Tonblende, Klangregler), weil der Klang bei den virtuellen Geräten mit anderen Werkzeugen (Equalizern) beeinflusst wird. Die anderen Mischer mit grafischer Oberfläche sehen ähnlich aus.

Als Dockapps für den Fenster-Manager WindowMaker (siehe Abschnitt 1.3.2 *Start von X11* auf Seite 13) kommen Mischer aus den Debian-Paketen *wmmixer* oder *mixer.app* (Aufruf `Mixer.app`) in Betracht. Beide laufen auch selbständig unter anderen Fenster-Managern.

9.3 Konverter, Denoiser, Audioeditoren

9.3.1 Encoder, Konverter (sox)

LAME ist ein quelloffenes MP3-Audiokompressionswerkzeug, allerdings kein fertiges zum Gebrauch durch den Endbenutzer, sondern ein Brocken C-Quellcode zum Experimentieren oder zum Einbau in andere Programme wie Audacity oder Xmed. Der Name war ursprünglich ein rekursives Akronym für *LAME ain't an MP3 Encoder*. Der Heimathafen der Software ist <http://lame.sourceforge.net/>. Sie steht unter der LGPL. Da die MP3-Kompression durch mehrere Patente geschützt ist, stehen der unbeschränkten kommerziellen Nutzung von LAME unter Umständen Lizenzfragen entgegen.

Infolge der Vielzahl von Dateiformaten für Audiodaten besteht ein Bedarf an Werkzeugen zur Formatumwandlung. Ein schlichtes, leistungsfähiges Werkzeug ist `sox` (Sound Exchange), zuhause auf <http://sox.sourceforge.net/>. Es wird aus der Kommandozeile mit der Quell- und der Zieldatei als Argumenten aufgerufen, wie ein Kopierprogramm:

```
joe@debian:~$ sox hallo.au hallo.ogg
```

SoX kennt mehr Dateiformate, als ein normaler Sterblicher jemals verwendet, aber die Unterstützung für MP3 ist nicht immer dabei oder auf Lesen begrenzt (Patentproblem). Um die auf Ihrer Installation unterstützten Formate zu ermitteln, geben sie `sox -h` ein. Das Werkzeug kann den Daten einige Effekte hinzufügen und bringt viele Optionen mit, siehe Manual. Als `soxmix` aufgerufen, mischt es zwei Dateien (Gesang, Instrument) in eine einzige Datei. Die beiden Werkzeuge `play` und `rec` sind einfache Frontends zu `sox` für die Wiedergabe bzw. Aufnahme.

Weitere Werkzeuge zum Umwandeln sind `mp3ogg`, `wav2cdr` sowie Audioeditoren, die in Abschnitt 9.3.3 *Audioeditoren* auf Seite 334 behandelt werden. Das Herausziehen von Audiodateien aus Audio-CDs wird als Rippen bezeichnet und in Abschnitt 9.4.4 *Ripper, Encoder* auf Seite 339 beschrieben.

9.3.2 Denoiser (gwc)

Geht es darum, ältere Schallplatten- oder Bandaufnahmen beim Digitalisieren von Störgeräuschen (Rauschen, Knistern, Knacken, Brummen) zu befreien, empfiehlt sich ein Werkzeug wie der *GNOME Wave Cleaner* aus dem Debian-Paket `gwc`, Aufruf `gnome_wavcleaner` (Symlink anlegen; die Manualseite wird davon natürlich nicht berührt, ist aber auch nahezu informationsfrei). Kurze Gebrauchshinweise erhält man mittels `gwc --usage`. Das Werkzeug kommt mit einer grafischen Oberfläche, die ein Hilfemenü enthält. Es zeigte sich beim Öffnen von Dateien

launenhaft. Auf <http://www.gnomefiles.org/> war das Werkzeug nicht zu finden, obwohl dort seitenweise Audiowerkzeuge aufgeführt sind.

Bei Sourceforge wird ein Projekt *Noisy*, ein Denoiser für Linux, gehostet, das sich seit 2001 in der Planung befindet. Ansonsten sieht es mager mit Denoisern (Dehisser, Declicker, Decrackler, Descratcher) für Linux aus. Einige Audioeditoren bringen entsprechende Plugins mit.

9.3.3 Audioeditoren (Audacity, ReZound, sweep usw.)



Abbildung 9.2. Screenshot des Audioeditors sweep beim Bearbeiten einer Datei mit Sprachaufzeichnungen

Ein Audioeditor ist ein allgemeines Werkzeug zur Bearbeitung von Audio-Daten. Anders als bei Texteditoren steht hier nicht das Erzeugen im Vordergrund, sondern das Bearbeiten. Grundsätzlich sollte man immer auf Kopien der Originaldateien arbeiten, manche Änderungen können nicht zurückgenommen werden. Gebräuchliche Funktionen sind:

- Aufnehmen,
- Wiedergeben,
- Schneiden, Einfügen und Kopieren,
- Speichern auf verschiedenen Medien, darunter CD/DVDs,
- Normalisieren, das heißt Einstellen auf eine Normallautstärke,
- Ändern von Tonlage (Pitch Shifting), Spieldauer (Time Stretching), Dynamik, Lautstärke, Frequenzgang (Equalizer), Stereobasisbreite, Nachhall,
- Einkreisen bestimmter Stellen (Scrubbing, wörtlich: Schrubben),
- Effekte wie Chorus, Echo, Flanger, Phaser, Wahwah, Reverse,
- Umwandeln von Dateiformaten.

Bei dieser Werkzeuggattung wird die Auswahl wieder größer. Einige der Editoren lassen sich als Audio-Workstation oder ausgewachsenes digitales Tonstudio bezeichnen:

- `ardour-gtk`, <http://ardour.org/>, die *Digital Audio Workstation* mit vielen Funktionen und im Debian-Paket mit GTK-Oberfläche, LADSPA-Schnittstelle für Plugins,
- Audacity, <http://audacity.de> und <http://audacity.sourceforge.net/>, mächtig, verbreitet, mit LADSPA-Schnittstelle, auch für andere Betriebssysteme verfügbar,
- GLAME GNU/Linux Audio Mechanics, <http://glame.sourceforge.net/>, zum GNOME-Desktop gehörig, LADSPA-Schnittstelle,
- Protux, <http://www.nongnu.org/protux/> und <http://freshmeat.net/projects/protux/>, nicht bei Debian, aber für GNU/Linux entwickelt und als Tarball verfügbar, benutzt die Qt-Bibliothek,
- gnusound, <http://www.gnu.org/software/gnusound/>, gehört ebenfalls zum GNOME-Desktop, mit LADSPA-Schnittstelle.
- ecawave, <http://www.wakkanet.fi/~kaiv/ecawave/>, braucht die Qt-Bibliothek, LADSPA-Schnittstelle, wird nicht mehr weiter gepflegt, verwandt mit dem Ecasound-Projekt desselben Autors,
- MiXViews, `mxv`, <http://www.create.ucsb.edu/~doug/htmls/MiXViews.html>, stark in der Signaldarstellung und -verarbeitung, eher für technisch interessierte Benutzer,
- ReZound, <http://rezound.sourceforge.net/>, Editor mittleren Umfangs, mit LADSPA, für Linux entwickelt,
- `snd`, <http://ccrma.stanford.edu/software/snd/>, aufgeteilt in mehrere Debian-Pakete, entfernt verwandt mit Emacs,
- `sweep`, zuhause auf <http://www.metadecks.org/software/sweep/> und <http://sweep.sourceforge.net/>, einfacher, genügsamer Editor mit LADSPA-Schnittstelle.

Eine umfangreiche Übersicht bietet <http://linux-sound.org/snded.html>. Kommerzielle Audioeditoren sind Adobe *Audition* oder *Wavelab* von Steinberg, beide nicht für Linux verfügbar.

Abbildung 9.2 vermittelt einen Eindruck von `sweep` beim Bearbeiten einer `ogg`-Datei mit Sprachaufzeichnungen, einem Editor, der einen Anfänger nicht durch zu viele Funktionen abschreckt. Er bringt eine LADSPA-Schnittstelle für Plugins mit, sodass er sich einfach um Effekte und Filter erweitern lässt. LADSPA bedeutet *Linux Audio Developer's Simple Plugin API* und wird auf <http://www.ladspa.org/> beschrieben. Solche Plugins sind unter anderem im Debian-Paket *swl-plugins* oder bei <http://www.plugin.org.uk/> zu finden.

9.4 Tonkonserven

9.4.1 Dateien (Zinf, Rhythmbox, AlsaPlayer, XMMS, ogg123)

Mit Audioeditoren kann man Audiodateien aufnehmen und wiedergeben, so wie man mit Texteditoren Textdateien lesen kann oder sich mit GIMP eine Grafikdatei anschauen, aber sie sind für diesen Zweck nicht optimal. Daher finden sich mehrere Werkzeuge zur Aufnahme und/oder Wiedergabe von Audiodateien und/oder CD/DVDs bei Debian. Eine zweite Aufgabe im Rahmen der Wiedergabe von Tonkonserven ist die Organisation großer Sammlungen, ein dritte das Herausziehen und Umcodieren einzelner Musikstücke aus CD/DVDs (Rippen). Video- oder Multimedia-Player können ebenfalls mit Audio-Daten umgehen, siehe Abschnitt 10.3 *Media Player* auf Seite 353.



Abbildung 9.3. Screenshot des Players zinf

Ein hübscher Player ist *zinf*, der Nachfolger von *freeamp* und dieser wieder eine Nachempfindung von *WinAmp*, siehe Abbildung 9.3. Näheres findet man auf <http://www.zinf.org/>. Er bietet in anderer Verpackung die gleichen Funktionen wie *noatun*. Mit den unscheinbaren Knöpfchen rechts oben kann man einen Equalizer öffnen oder den sichtbaren Teil von *Zinf* verkleinern. Im Debian-Paket *zinf-extra* sind einige zusätzliche Benutzerschnittstellen enthalten, in weiteren Paketen Plugins für ALSA, aRts und Esound.

Die GNOME Rhythmbox spielt Audiodateien ab und verwaltet sie. Im Hauptmenü des KDE-Desktops erscheint sie unter *Multimedia -> Musik-Player*. Als Quellen kennt sie lokale Sammlungen, Internetradio (Webradio) und einen Apple iPod, sofern vorhanden. In der Liste der Internet-Radiosender sind bereits einige URLs von Magnatune (<http://www.magnatune.com/>) und anderen eingetragen. Weitere sind auf <http://www.radiosites.de/> oder bei der deutschen Wikipedia, Stichwort *Internetradio-Anbieter*, nachzulesen. Ein Sender wird über die Menüpunkte *Musik -> Internet-Radiosender anlegen* eingetragen. Wenn ein Sender ausgewählt

ist, schaut man sich über die Menüpunkte *Musik -> Eigenschaften* die zugehörigen Eintragungen an und kann sie editieren. Wie es aussieht, versteht Rhythmbox nur HTTP, nicht aber das Realtime Streaming Protocol (RTSP), beispielsweise von Radio Sverige verwendet. In diesem Fall muss man auf `realplay` oder ähnliche Werkzeuge zurückgreifen, siehe Abschnitt 10.3.6 *Helix/Real Player* auf Seite 357.

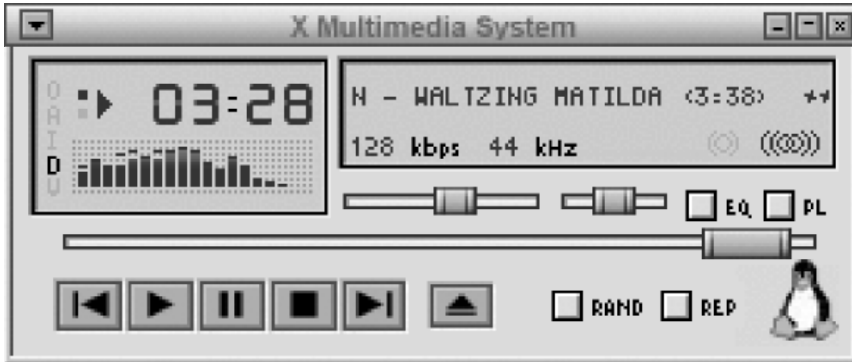


Abbildung 9.4. Screenshot des X MultiMedia Systems bei einer Audiowiedergabe, mit besonderer Skin

Der AlsaPlayer ist auf mehrere Debian-Pakete aufgeteilt, je nach Benutzeroberfläche und Ausgabe. Der Player spielt Musikstücke nach einer Liste ab, Queue genannt. Er verfügt über einen Schieber, mit dem sich die Abspielgeschwindigkeit ändern lässt, nicht zu verwechseln mit Pitch Shifting oder Time Stretch. Es wird einfach die Drehzahl der Schallplatte verstellt.

Das *X MultiMedia System*, Aufruf `xmms`, ist ein sehr vielseitiger, WinAmp-ähnlicher Player unter X11 und GTK+. Wir begegnen ihm noch einmal im nächsten Kapitel. Ein Mausklick auf den unauffälligen Knopf links oben öffnet ein Menü zur Auswahl der Dateien ebenso wie ein Rechtsklick auf eine neutrale Fläche. Auch dieser Player enthält einen Texteditor für Wiedergabelisten. Die zugehörige Manualseite hilft etwas, ausführlichere Informationen bekommt man bei <http://www.xmms.org/>. Die braucht man auch, wenn man das Werkzeug voll ausnutzen will; es gibt viele Plugins. Für den Player stehen mehrere Skins (Aussehen, skin = Haut) zur Auswahl.

Der ogg-Player `ogg123` aus dem Debian-Paket *vorbis-tools* und mit dem Heimathafen <http://www.xiph.org/> wird aus der Kommandozeile aufgerufen:

```
joe@debian:~$ ogg123 hallo.ogg
```

Ähnliche schlichte Werkzeuge gibt es auch für MP3-Dateien, beispielsweise den curses-basierten `mp3blaster`, der MP3- und ogg-Dateien abspielt, oder `mpg321`. Sie eignen sich zum Aufruf aus Skripten oder als Unterlage für eine grafische Benutzeroberfläche, können aber selbstverständlich auch ohne Zierrat benutzt werden.

Bei Debian oder Sourceforge finden sich noch zahlreiche andere Player für Linux, doch wir gehen weiter.

9.4.2 Musik-Server (mserv)

Ein Musik-Server wie `mserv` (<http://www.mserv.org/>) liest zunächst einmal Musikdateien (mp3, ogg) von seinem Massenspeicher. Dann verteilt er sie mittels TCP/IP-Protokollen an seine Clients im lokalen Netz, wobei er mehrere Clients mit unterschiedlichen Wünschen gleichzeitig bedient. Bei den Clients wird die Musik mit jeweils einem lokalen Player wiedergegeben. Der Dienst wird abgerundet durch ergänzende Informationen zu den Musikstücken, Statistiken etc. Der Server verfügt über ein Kommandozeilen- und ein Web-Interface und kann auch als Backend für einen Streaming Server wie Icecast dienen. Das Aufsetzen eines lokalen Streamingnetzes ist Sache des Verwalters.

9.4.3 CD/DVD-Spieler (WorkBone, xmcd, gnome-cd, KsCD)

Zur Wiedergabe von Audio-CD/DVDs im Laufwerk des Rechners stehen mehrere Werkzeuge zur Wahl:

- WorkBone, ein einfacher CD-Player ohne Grafik,
- Der Motif CD Audio Player `xmcd`, braucht X11 und bringt den `cda` für die Kommandozeile mit,
- der GNOME CD Player `gnome-cd` aus dem Debian-Paket *gnome-media*,
- der simple KDE CD Player `kscd` aus dem gleichnamigen Debian-Paket,
- die oben erläuterten Werkzeuge zur Wiedergabe von Dateien, teilweise nur mit zusätzlichen Plugins,
- die unten erläuterten Media Player.

und weitere. Falls Sie zunächst nichts hören, obwohl der Spieler einen Fortschritt anzeigt, ist vermutlich der Mischer falsch eingestellt, im Hauptmenü des KDE-Desktops hinter den Menüpunkten *Multimedia* -> *Lautstärkeregler* oder *Multimedia* -> *Soundmixer* verborgen. Der eventuell im Spieler vorhandene Lautstärkeregler besagt nicht alles.

Der `xmcd` fragte bei seiner Einrichtung ziemlich viel, war aber mit den teilweise vagen Antworten zufrieden. Er arbeitet mit mehreren Laufwerken oder Plattenwechslern im Rechner zusammen. Entsprechend konfiguriert bezieht er Informationen von einem Internet CD Database Server (CDDDB-Server) wie <http://www.freedb.org/>. Dazu berechnet er eine eindeutige ID der jeweiligen CD, schickt diese an den Server und erhält Informationen über Inhalt etc. der CD zurück, die er anzeigt. Diese Daten sind nicht auf der CD gespeichert. Eine solche Datenbasis kann auch lokal angelegt und gepflegt werden. Abbildung 9.5 zeigt Aussehen des Players. Es lässt sich durch Ziehen mit der Maus vergrößern und durch einen Knopf links oben auf die Basisfunktionen begrenzen. Die Konfiguration steht wie üblich systemweit in `/etc/xmcd/` und benutzerbezogen in `$HOME/.xmcdcfg/`. Ein



Abbildung 9.5. Screenshot des CD-Spielers xcd

Verwandter des xcd für die Kommandozeile ist cda. Das Werkzeug macht von den Konfigurationsdateien des xcd Gebrauch.

Der GNOME CD Player kommt schlicht daher, enthält aber alles Notwendige. Etwas üppiger wirkt sein Kollege vom KDE-Projekt, siehe Abbildung 9.6. Wie man sieht, arbeitet der Spieler per Default mit einer CD-Datenbank zusammen und holt auf Wunsch Informationen zum Künstler aus dem Web.



Abbildung 9.6. Screenshot des CD-Spielers KsCD

9.4.4 Ripper/Encoder (cdparanoia, abcde, jack, grip, eac)

Will man seine Musik-CD/DVDs auf dem Massenspeicher abspeichern, erfordert das zwei Schritte:

- das Herausziehen und Kopieren einzelner Musikstücke (ripen oder grabben),
- das Umcodieren in ein Sound-Dateiformat wie mp3 oder ogg.

Die gängigen Werkzeuge erledigen beide Schritte in einem Aufruf und können auch eine CD/DVD abspielen, ohne sie zu rippen. Umgekehrt können manche Player rippen. Beim Codieren nach MP3 sind aus Patentgründen gelegentlich zusätzliche Bibliotheken oder Plugins erforderlich. Einige Ripper arbeiten mit einer Titel-Datenbank wie Freedb (siehe oben) zusammen, in der Daten zu den Musikstücken gespeichert werden.

Der Ripper `cdparanoia` (<http://www.xiph.org/>) wird aus der Kommandozeile aufgerufen:

```
joe@debian:~$ cdparanoia 1-2
```

und beschränkt sich auf elementare Funktionen. Im Beispiel rippt er die Tracks 1 und 2 der CD und legt sie im Arbeitsverzeichnis in einer einzigen Datei `cdda.wav` ab, die nicht komprimiert und daher groß ist, abspielbar mit `zinf` oder anderen Playern. Das Werkzeug wird man nur nehmen, wenn man den Rippvorgang genau steuern will, andernfalls sind die nachfolgend beschriebenen Frontends bequemer. Bei Problemen legt man eine Audio-CD in das fragliche Laufwerk ein, ruft das Werkzeug mit einigen Optionen auf:

```
joe@debian:~$ cdparanoia -svQ
```

und versucht, aus der Ausgabe schlau zu werden. Da es als Backend für weitere Ripper/Kodierer dient, ist sein Funktionieren unerlässlich. Bei SCSI-Laufwerken muss auch das zugehörige generische SCSI-Gerät bekannt sein; obiger Aufruf kümmert sich darum. Es sieht so aus, als ob das Kommando das generische Gerät stets selbstständig ermittelt, falls es nicht angegeben wird. Wenn der Verwalter rippen kann, ein gewöhnlicher Benutzer jedoch nicht, liegt das höchstwahrscheinlich an Zugriffsrechten von Gerädateien.

Das Kommandozeilen-Werkzeug `abcde` (A Better CD Encoder) ist ein Frontend in Form eines Shellskriptes zu `cdparanoia` und `oggenc`, kann aber auch für andere Ripper oder Kodierer konfiguriert werden. Treffen die Default-Einstellungen zu, rippt man mit dem Aufruf:

```
joe@debian:~$ abcde
```

die im CD-Laufwerk befindliche Audio-CD und legt die einzelnen Musikstücke ogg-kodiert im Arbeitsverzeichnis ab. Einfacher geht es nicht. Die systemweite Konfiguration ist in `/etc/abcde.conf` zu finden. Will man sie an persönliche Wünsche anpassen, kopiert man die Datei unter dem Namen `.abcde.conf` in sein Homeverzeichnis und editiert sie.

Der Ripper/Encoder `jack` – benannt nach JACK THE RIPPER, einem mehrfachen Prostituiertenmörder in London Ende des 19. Jahrhunderts – ist ein zeichenbasiertes Frontend für Ripper wie `cdparanoia`. Er wird per Editor in der Datei `.jackrc` konfiguriert und legt seine Ergebnisse defaultmäßig als ogg-Dateien in einem Verzeichnis `$HOME/jack` ab. Jack the Ripper ist nicht zu verwechseln mit dem Jack Audio Connection Kit, zu Hause auf <http://jackit.sourceforge.net/>, das sich bemüht, unterschiedliche Audioanwendungen unter einen Hut zu bringen.

Der GNOME-Player, Ripper und Encoder (<http://www.nostatic.org/grip/>) versucht nicht, wie ein Gerät auszusehen, siehe Abbildung 9.7. Er legt die Musikdateien per Default in einem Unterverzeichnis des Verzeichnisses `$HOME/mp3/` ab, und zwar im ogg-Format. Von dort können sie beispielsweise mit `zinf` oder `xmcd` abgespielt oder mit `sox` umcodiert werden. Mit einem SCSI-DVD-Laufwerk, das von `cdparanoia` erkannt wurde, gab es in Verbindung



Abbildung 9.7. Screenshot des GNOME-Players/Rippers/Encoders grip

mit CDs, die der Datenbank unbekannt waren, Probleme; ein SCSI-CD-Brenner lief einwandfrei.

Das Werkzeug *Exact Audio Copy* (eac, <http://www.exactaudiocopy.de/>) ist ein Ripper, der besonderen Wert auf Fehlererkennung und -beseitigung legt. Leider gibt es ihn nur für andere Betriebssysteme, ein Fall für den Windows-Emulator wine. Eigene Erfahrungen liegen nicht vor.

9.4.5 Rekorder (brec, Sound Recorder, KRec, krecord)

Obwohl sich Audioeditoren zur Aufzeichnung von Sound gleich welcher Herkunft eignen, gibt es doch Werkzeuge speziell zu diesem Zweck, oft kombiniert mit Playern:

- brec und bplay, einfache Werkzeuge für die Kommandozeile, zum Aufnehmen und Wiedergeben ohne Kompression,
- der GNOME Sound Recorder aus dem Paket *gnome-media*,

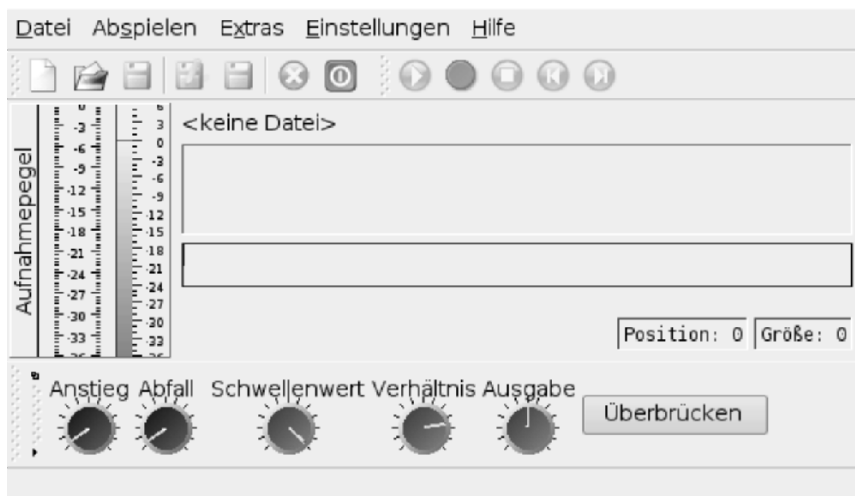


Abbildung 9.8. Screenshot des Werkzeugs Krec zur Aufnahme und Wiedergabe roher Audio-daten (wav-Format)

- KRec aus dem KDE-Projekt, siehe Abbildung 9.8,
- krecord aus dem KDE-Projekt, zum Aufnehmen und Wiedergeben von wav-Dateien

und weitere aus der Abteilung *Sound* der Debian-Paketliste. Im Vergleich zu einem Audioeditor sind die Rekorder wesentlich einfacher und anspruchsloser. Die Werkzeuge für die Kommandozeile lassen als Glied einer Pipe beispielsweise mit `sox` einsetzen und verstehen dann mehrere Dateiformate.

In Abbildung 9.8 sehen wir unter der Titelleiste eine Menüleiste und darunter eine Symboleiste für häufig gebrauchte Aktionen, die man auch aus der Menüleiste aufrufen könnte. Links befindet sich ein Schieber für den Aufnahmepegel, unten ein wahlweise dazuschaltbarer Kompressor, der besonders bei Sprachaufnahmen nützlich ist.

9.5 Klangerzeugung (MIDI)

Das *Musical Instrument Digital Interface* (MIDI) ist ein Protokoll, eine Schnittstelle zwischen Rechnern und digitalen Musikinstrumenten, und zwar für Aufzeichnung, Übertragung und Wiedergabe. Das Protokoll betrifft sowohl Hard- als auch Software. Der Datenfluss geht stets nur in eine Richtung (von *out* nach *in*), anders als beispielsweise bei einer Ethernet-Schnittstelle. Der Heimathafen der Spezifikation ist <http://www.midi.org/>, ein deutschsprachiger Guide von BERT MARX liegt auf <http://www.midiguide.de/>. Um es nochmal deutlich zu sagen: MIDI-Dateien sind *keine* Audio- oder Sound-Dateien, sondern mit Programmen zu vergleichen, die Steueranweisungen enthalten (welcher Ton ist wie lange wie laut von

welchem Instrument zu spielen), nur eben nicht für Rechner mit einem Fortran- oder C-Compiler, sondern für MIDI-Geräte. Bei der Wiedergabe mancher MIDI-Stücke wird der Hörer an alte walzen- oder scheibengesteuerte Pianos erinnert, wie wir sie auf Platten von SCOTT JOPLIN hören, korrekt und leblos.

Einige Soundkarten für PCs haben *auch* eine MIDI-Schnittstelle. Um festzustellen, ob Ihr Rechner MIDI unterstützt, lesen Sie entweder in der zugehörigen Dokumentation oder schauen nach, ob Ihr Linux-Kern entsprechende Module geladen hat:

```
joe@debian:~$ /sbin/lsmmod | less
```

Module wie snd-rawmidi, snd-seq-device oder snd-mpu401-uart lassen vermuten, dass MIDI eingerichtet ist.



Abbildung 9.9. Screenshot des MIDI-Werkzeugs TiMidity mit X11-Oberfläche

TiMidity (<http://www.onicos.com/staff/iz/timidity/> sowie <http://timidity.sourceforge.net/>) ist ein Synthesizer in Software, ein Konverter von MIDI (mid) zu WAVE (wav). Das Werkzeug ist ein Ausweg bei einfacheren Soundkarten ohne MIDI-Geräte in Hardware. Es spielt die wav-Dateien sofort ab oder speichert sie auf Platte, falls die entsprechende Option gesetzt ist:

```
joe@debian:~$ timidity -Ow cavatina.mid
```

Falls Sie nichts hören, werfen Sie einen Blick auf Ihr Mischpult: Master/Summe/Volume und PCM müssen eingeschaltet sein und oberhalb von null stehen. Die zahlreichen Optionen sind im Manual nachzulesen. Die Konfigurationsdatei von TiMidity liegt im Verzeichnis `/etc/timidity/` und ist überraschend kurz. Eine Alternative zu TiMidity ist im Debian-Paket *fluidsynth* samt Begleitern enthalten (<http://www.fluidsynth.org/>).

Damit TiMidity die MIDI-Anweisungen in Klänge umsetzen kann, braucht es Soundfonts, auch Patches² genannt, so wie ein Drucker Zeichenfonts braucht. Im Debian-Paket *freepats* oder bei <http://freepats.opensrc.org/> steht eine Sammlung im Umfang von 30 MB zur Verfügung. Nach Einrichtung des Paketes liegen die Patches in `/usr/share/midi/freepats/`. Die früher oft zitierten Eawpats von ERIC A. WELSH sind, soweit nicht Urheberrechte dagegen standen, in die Freepats eingegangen und kaum noch im Netz zu finden (<http://www.funet.fi/pub/unix/4.3bsd/NetBSD/packages/distfiles/>).

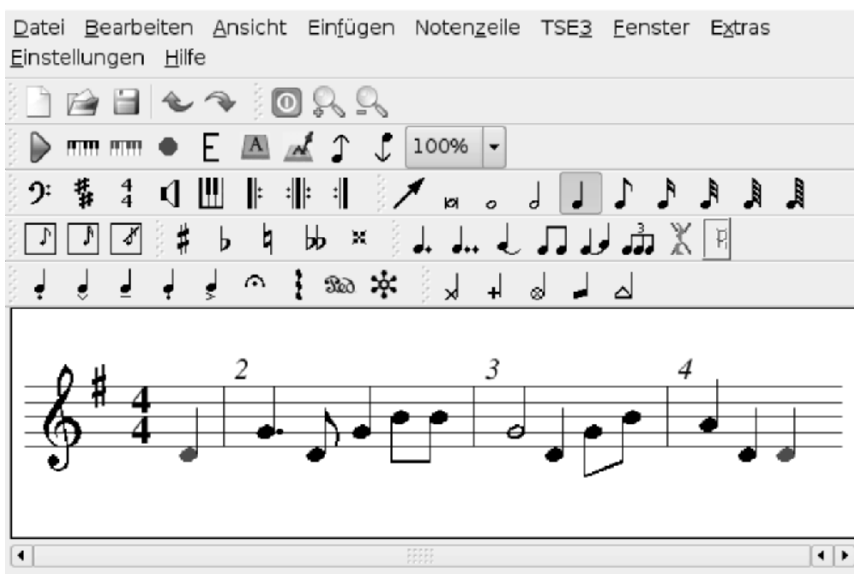


Abbildung 9.10. Screenshot des KDE Musikeditors NoteEdit

In dem Debian-Paket von TiMidity fehlen die TiMidity-Tools (*patinfo* usw.) sowie grafische Oberflächen (*xaw* oder *GTK+*). Das Paket *timidity-interfaces-extra* fügt einige Oberflächen hinzu. Der Aufruf:

```
joe@debian:~$ timidity -iatv
```

²Nicht zu verwechseln mit Patches zur Behebung von Softwaremängeln.

versieht das Werkzeug mit einem GUI gemäß Abbildung 9.9, sodass auch die Augen etwas von der Musik haben. Läuft TiMidity im ALSA Server Mode im Hintergrund:

```
joe@debian:~$ timidity -iA -Os &
```

so kann er von anderen Programmen als Synthesizer verwendet werden. Die Option `-iA` ruft TiMidity mit einem ALSA Sequencer Interface auf, die Option `-Os` lenkt die Ausgabe auf ein ALSA PCM Gerät. Mit dem Aufruf:

```
joe@debian:~$ pmidi -l
```

lassen wir uns die verfügbaren Ports (ALSA-Ausgänge) anzeigen, um sie bei einem Aufruf zwecks Wiedergabe einer midi-Datei als Option mitzugeben:

```
joe@debian:~$ pmidi -p 128:0 volgabo.a.mid
```

Wir gewinnen mit `pmidi` nicht viel, da TiMidity allein bereits in der Lage ist, midi-Dateien wiederzugeben. In gleicher Weise können wir KMid aus dem Hauptmenü des KDE-Desktops benutzen oder `playmidi`:

```
joe@debian:~$ playmidi -v -eD1 volgabo.a.mid
```

Die Option `-v` wie *verbose* ist nie verkehrt, die Option `eD1` – kein Zwischenraum – lenkt die Ausgabe auf das externe MIDI-Gerät D1, womit `/dev/snd/midiC0D1` gemeint ist. Eine Option `-t0.7` beschleunigt die Wolgaschiffer, ohne die Tonhöhen zu ändern (time stretching). Mit der Option `-p46` erzwingt man, dass alle Klänge harfenähnlich erklingen, vorausgesetzt der zugehörige Patch Nr. 46 ist vorhanden.

NoteEdit ist der KDE Musikeditor und auf <http://noteedit.berlios.de/> zu Hause. Mit seiner Hilfe setzt man Noten, speichert sie in verschiedenen Formaten ab und kann sie wieder einlesen. Unter den Formaten finden sich neben dem editoreigenen Format MIDI, MusicXML, ABC Music, PMX, MusiXTeX und LilyPond. Er importiert MIDI- und MusicXML-Dateien. Abbildung 9.10 zeigt den Editor bei der Arbeit. Wer sich mit Noten auskennt, kommt bald mit dem Editor zurecht. In Zusammenarbeit mit TiMidity, siehe oben, spielt er Kompositionen vor und hebt die jeweils ausgegebene Note rot hervor.

Die KDE-Anwendung Rosegarden4 ist ein mächtiger Musikeditor für alle, die Musik komponieren, aufnehmen, bearbeiten und wiedergeben wollen. Der Heimathafen des virtuellen Studios ist <http://www.rosegardenmusic.com/>. Es ist ein Noteneditor wie NoteEdit und ein Sequenzer für MIDI- und Audio-Dateien mit zahlreichen Funktionen, ähnlich wie Cubase von Steinberg für andere Betriebssysteme. Die Zusammenarbeit mit TiMidity klappte einwandfrei. Abbildung 9.11 zeigt Rosegarden4 beim Abspielen einer MIDI-Datei (Klavier solo). Im Noteneditor kann man sich zeitgleich die zugehörigen Noten vorführen lassen. Bei solchen Anwendungen sind Manualseiten überfordert. Der Menüpunkt *Hilfe* bringt *The Rosegarden Handbook* auf den Schirm. Weitere Musikedatoren sind Brahms (<http://brahms.sourceforge.net/>) und Muse (<http://muse.serverkommune.de/>). Nur ein Musiker kann eine begründete Auswahl treffen; uns machte Rosegarden den ausgereiftesten Eindruck, gefolgt von NoteEdit. Beide Werkzeuge liefen auf Anhieb und bieten eine Vielfalt von Funktionen.



Abbildung 9.11. Screenshot des KDE Musikeditors Rosegarden beim Abspielen einer MIDI-Datei. Gleichzeitige Anzeige der Noten im Noteneditor.

Das war ein erster Einblick in die MIDI-Welt. Das Linux MIDI-HOWTO führt weiter, wobei immer die Aktualität der Informationen zu beachten ist. Gerade bei Sound entwickeln sich Hard- und Software rasch weiter. Zudem ist Computersound ein Gebiet, das breit gefächerte Kenntnisse erfordert, die nicht in jedem Linux-Buch nachzulesen sind.

9.6 Spracherkennung (XVoice, sphinx)

Unter Spracherkennung (E: speech/voice recognition, SR oder automated speech recognition, ASR, F: reconnaissance vocale) versteht man die automatische Umsetzung von über ein Mikrofon eingegebener Sprache in eine Textdatei (dictation mode) oder in Kommandos an Anwendungen (command mode). Die natürliche Sprache spielt eine Rolle, Englisch hört sich anders an als Finnisch, der individuelle Sprecher samt seinen Unvollkommenheiten, das Vokabular, das der Rechner kennt, Umgebungsgeräusche und nicht zuletzt die Art des Mikrofons. Als Echtzeitaufgabe stellt die Spracherkennung Ansprüche an den Rechner. Anwendungen sind die Steuerung des Rechners oder anderer Geräte, wenn die Hände anderweitig beschäftigt sind oder fehlen, oder die Umsetzung von Stimme auf Text in Schreibbüros oder bei der Verkettung unterschiedlicher Kommunikationssysteme. Das *Speech Recognition HOWTO* (2002) vermittelt einen Einstieg in das Thema.

Beim Institut für Maschinelle Sprachverarbeitung der Universität Stuttgart (<http://www.ims.uni-stuttgart.de/>) findet der geneigte Leser weiterführendes Material.

XVoice (<http://xvoice.sourceforge.net/>) ist ein Werkzeug für X11 zur Spracherkennung, das zusätzlich die kommerzielle Software IBM ViaVoice for Linux benötigt. Alternativ wird Sphinx vorgeschlagen. Das Debian-Paket *sphinx2-bin* enthält außer Funktionen einige fertige Werkzeuge und Beispiele zur Spracherkennung in Echtzeit. Das Projekt ist auf <http://cmusphinx.sourceforge.net/> zu Hause und stammt aus der Carnegie Mellon University (<http://www.speech.cs.cmu.edu/>). Nach der Einrichtung startet man das Skript `sphinx2-test` und sieht sich die Ausgabe an. Ist alles OK, besteht Hoffnung, dass das Werkzeug läuft:

```
joe@debian:~$ sphinx2-test | tee sphinx.out
```

Das Werkzeug ist mager dokumentiert; am meisten steht in `/share/doc/sphinx2-bin/README.bin`. Man kommt nicht umhin, sich den *Sphinx-II User Guide* von Sourceforge zu holen, sinnvollerweise auch gleich *How to use a SR system*. Beide Dokumente sind keine Kurzanleitungen, sondern richten sich an Benutzer, die sich eingehend mit Spracherkennung befassen wollen. Bei Versuchen nicht vergessen, ein Mischpult bereit zu stellen und den Mikrofoneingang aufzudrehen.

9.7 Sprachsynthese, Screenreader

9.7.1 Einfache Beispiele (saydate, saytime, flite)

Bei der Spracherzeugung hat es der Rechner leichter als bei der Spracherkennung, weil das Ausgangsmaterial sauber definierte Daten sind und weil das Verständnis der Schallsignale beim Menschen liegt, der leichter lernt und sich schneller anpasst als eine Maschine. Einige Hinweise und Software zu dem Thema des Abschnitts bieten:

- das Institut für Kommunikationsforschung und Phonetik der Universität Bonn, <http://www.ikp.uni-bonn.de/dt/forsch/phonetik/>,
- die Königliche Technische Hochschule (KTH) Stockholm unter <http://www.speech.kth.se/>,
- die University of Edinburgh unter <http://www.cstr.ed.ac.uk/>.

Das sollte als Ausgangspunkt für eigene Recherchen ausreichen.

Ein Rechner mit der nötigen Hardware kann sprechen. Die Sprechweise ist gewöhnungsbedürftig, aber verständlich. Ein einfaches Beispiel sind die beiden kleinen Werkzeuge `saydate` und `saytime`. Aufgerufen nennen sie in englischer Sprache das Tagesdatum bzw. die Uhrzeit. Die dabei verwendeten Sound-Dateien liegen unter `/usr/share/saydate` und lassen sich durch eigene Erzeugnisse ersetzen. Die Dokumentation ist in `/usr/share/doc/saydate` untergebracht.

Das Kommando `flite` (festival lite) wandelt eine als Argument übergebene Zeichenkette oder Textdatei in eine wav-Datei um oder gibt sie als Sprache aus:


```
joe@debian:~$ flite "good morning old boy"
```

```
joe@debian:~$ flite hallo.txt
```

falls sich nicht ein anderes Audioprogramm `/dev/dsp` gekrallt hat. Englischer Text kommt verständlicher als deutscher, Zahlen werden auf Englisch gesprochen. Eine Dokumentation im pdf-Format liegt in `/usr/share/doc/flite/`. Die Heimat von `flite` ist <http://cmuflite.org/>. Zum Kennenlernen synthetischer Sprache ist das kleine, schnelle Werkzeug gut geeignet, an GERT WESTPHAL reicht es nicht heran. Das Kommando `eflite` wird in Verbindung mit Emacspeak behandelt.

9.7.2 Sprachsynthese (Epos, Festival, FreeTTS, MBROLA)

Epos ist ein Text-zu-Sprache-System (E: text to speech system, TTS system, F: synthétiseur de parole), das unabhängig von der jeweiligen Sprache (Deutsch, Englisch ...) ist. Es verteilt sich auf ein Debian-Paket mit den Werkzeugen und den erforderlichen Regeldateien sowie drei Pakete mit Sounddateien unterschiedlicher Herkunft, die nur bei Bedarf einzurichten sind. Sein Heimathafen ist <http://epos.ure.cas.cz/>, bei Sourceforge ist es unter <http://sourceforge.net/projects/epos/> zu finden.

Das System besteht aus einem im Hintergrund laufenden Server, mittels `epos` zu starten, und einer beliebigen Anzahl von Clients, die über TCP-Verbindungen und das Text-To-Speech Control Protocol (TTSCP) mit dem Server verkehren. Ein einfacher, mitgelieferter Client ist `say`. Der Aufruf:

```
joe@debian:~$ say "na shledanou"
```

führt zu der erwarteten Ausgabe, vorausgesetzt der Server läuft und konnte sich `/dev/dsp` reservieren. Zum Erlernen des Tschechischen reicht die Qualität der Aussprache nicht, da sind herkömmliche Schallkonserven überlegen. Deutsch kommt ähnlich gut oder schlecht. Der Server kennt eine Vielzahl von Optionen, `say` wenige. Die Dokumentation ist erfreulicherweise auf Englisch verfasst und ausführlich.

Festival ist ein weiteres Text-zu-Sprache-System, das unter Debian bereit liegt. Es stammt aus der University of Edinburgh (<http://www.cstr.ed.ac.uk/projects/festival/>) und eignet sich ebenfalls für mehrere natürliche Sprachen. Das Kommando `festival` kann als Kommandointerpreter oder im TTS-Modus aufgerufen werden:

```
joe@debian:~$ festival --tts textdatei
```

Ohne Argument liest es von `stdin` bis zu einem `<ctrl>-<d>`, immer unter der üblichen Voraussetzung, dass `/dev/dsp` nicht anderweitig belegt ist. Defaultsprache ist Englisch; Spanisch und Walisisch sind verfügbar. Deutsch wird infolgedessen mit einem deutlichen ASCII-Akzent wiedergegeben.

Das in Java geschriebene TTS-System *FreeTTS* gibt es nicht bei Debian, sondern auf <http://freetts.sourceforge.net/>. Es macht einen vielseitigen Eindruck und wird von einigen Screenreadern angesprochen.

Das MBROLA-Projekt (<http://tcts.fpms.ac.be/synthesis/>) stellt ein für den privaten Gebrauch kostenloses TTS-System für Linux und andere Betriebssysteme ins Netz (Freeware, nicht GPL, nicht bei Debian). Da es als Eingabe Anweisungen verlangt, benötigt es einen Präprozessor, der Text in Anweisungen übersetzt. Als Präprozessor kann *txt2pho* dienen, zu haben bei <http://www.ikp.uni-bonn.de/dt/forsch/phonetik/hadifix/Hadifix.html>. MBROLA stammt aus Mons in Belgien und soll sich gut für Deutsch und Französisch eignen. Eine online-Sprachprobe aus der Universität Bielefeld hörte sich jedenfalls besser an als ASCII-Deutsch.

Jetzt werden wir übermütig und wollen eine Texteingabe in eine ogg-Datei schreiben. Klar, dass wir eine Pipe brauchen:

```
joe@debian:~$ echo "rhabarber"| txt2pho | mbrola | sox
rhabarber.ogg
```

echo schreibt sein Argument nach *stdout*, *txt2pho* liest es und wandelt es in Anweisungen für MBROLA um, das als nächstes Glied der Pipe dran ist und eine wav-Datei erzeugt, die von *sox* in eine ogg-Datei konvertiert wird. So wie vorstehend geschrieben funktioniert die Pipe nicht, da fehlen einige Optionen. Die herauszufinden überlassen wir dem geneigten Leser. Falls das Ergebnis trotz richtiger Optionen nicht überzeugt, versuchen Sie es mal mit benannten Pipes (FIFO).

9.7.3 Screenreader (emacspeak, screader, uxdots)

Screenreader sind eine besondere Art von Programmen zur Spracherzeugung, ein Hilfsmittel für sehschwache oder blinde Benutzer. Sie lesen diesen den Text vor, der auf dem Bildschirm für die Augen dargestellt wird (*nicht* eine Textdatei). Man kann einen Screenreader mit einer Person vergleichen, die den Bildschirm abliest. Zur eigentlichen Sprachsynthese ziehen sie Werkzeuge aus dem vorigen Abschnitt heran. Daneben füttern sie eine Braillezeile. Ausführlicher werden Screenreader auf <http://www.matthias-haenel.de/wascreen.html> beschrieben. Der Adobe Reader bringt einen eigenen Screenreader unter dem Namen *Read Out Loud* mit. Wie weit der unter Linux funktioniert, ist noch auszuprobieren. Hinweise liefert <http://access.adobe.com/>. Bei Debian stehen folgende Screenreader zur Verfügung:

- *brltty*, ein Dämon, der auf die Textkonsole zugreift und verschiedene Braillezeilen ansteuert, keine Sprachausgabe, <http://dave.mielke.cc/brltty/>,
- *yasr*, greift ebenfalls auf die Textkonsole zu und arbeitet mit mehreren Synthesizern zusammen, auch mit einem Emacspeak-Server, der seinerseits weitere Synthesizer kennt, <http://yasr.sourceforge.net/>,

- Speakup, ein umfangreiches System, eine akustische Oberfläche, die mit einem Kernel-Patch beginnt und zahlreiche Module mitbringt, <http://www.linux-speakup.org/>,
- Gnopernicus aus dem GNOME-Projekt, daher für den grafischen GNOME-Desktop vorgesehen, für Sprachausgabe (Festival und andere) und Braillezeile (brltty), zuhause auf <http://www.baum.ro/gnopernicus.html>.

Wenn ein Hardware-Synthesizer am Rechner angeschlossen ist, kann Speakup bereits die Meldungen beim Systemstart vorlesen, bei einem Software-Synthesizer muss das System hochgefahren sein, ehe Speakup sich melden kann.

Der Alleskönner Emacs – siehe Abschnitt 5.4 *Universalgenie* auf Seite 188 – spricht auf Wunsch natürlich auch. Das Projekt Emacspeak liegt auf <http://emacspeak.sourceforge.net/> und bringt ein Manual aus dem Jahr 2002 mit. Zur weiteren Vorbereitung dienen *A Gentle Introduction to Emacspeak* von GARY L. MURPHY auf <http://emacspeak-guide.sourceforge.net/>, das *Installing Emacspeak HOWTO* von 2002, daher in Einzelheiten veraltet, und der *Emacspeak User's Guide*, gemäßigte 27 Seiten stark – vorausgesetzt Sie kennen den Emacs schon ein bisschen. Emacspeak ist kein Screenreader, sondern weiß alles, was momentan im Emacs geschieht. Die Ausgaben des Emacs erfolgen parallel optisch und akustisch. Die Entwickler versuchen, alle Fähigkeiten eines optischen Desktops auch akustisch darzustellen: Icons, Buttons usw.

Der Emacs sei eingerichtet und arbeitsfähig. Dann ist zuerst ein Sprach-Synthesizer (TTS-System) einzurichten, beispielsweise `eflite`. Es folgt der Zusatz Emacspeak selbst, der während der Konfiguration nach `eflite` oder einem anderen Synthesizer fragt. Zur wiederholten Konfiguration dient das Kommando `emacspeakconfig`. Wir rufen Emacspeak auf:

```
joe@debian:~$ emacspeak
```

und bekommen unter lebhaftem Gemurmel aus den Lautsprechern den Emacs auf den Schirm. Tippen wir nun Text ein, so werden jeder Buchstabe einzeln und beim Betätigen der Leertaste das letzte Wort gesprochen. Jetzt wünscht man sich noch eine deutsche Aussprache und ein paar Optionen zur Feinanpassung, aber das sind Dinge, die `eflite` mitbringen muss, und dem fehlt gegenwärtig sogar eine ordentliche Dokumentation. Man müsste sich mit den Quellen auseinander setzen und Stimmen schreiben, alles mit Arbeit verbunden.

Video und Fernsehen

Ein Rechner mit entsprechender Hardwareausstattung kann Filme oder Fernsehsendungen aufzeichnen, wiedergeben und bearbeiten. Das Kapitel zeigt einige Möglichkeiten.

10.1 Grundbegriffe

Wir kommen nun zu den im engeren Sinn multimedialen Anwendungen. Ohne genaue Definitionen zu bemühen – von denen es mehrere gibt – stellen wir uns multimediale Daten als eine Verbindung von Audio plus Video vor, als einen Tonfilm mit bescheidenen Möglichkeiten zur interaktiven Beeinflussung, das Ganze im Rechner natürlich digital. Noch kürzer: bunt, bewegt, laut. Über Auge und Ohr hinausgehende Sinne könnten auch angesprochen werden, das ist jedoch die Ausnahme. Anspruchsvolle Multimedia-Anwendungen fordern die heutige Hard- und Software bis an ihre Grenzen. Auch die Mensch-Maschine-Schnittstelle ist noch nicht am Ende ihrer Entwicklung. Es kommt vor, dass Sie einen neueren Kern brauchen und/oder knackfrische Software aus *testing* oder *unstable*, um bestimmte Anwendungen zu starten. Solche noch nicht bewährte Software richtet man ungern auf dem Rechner ein, der den Terminkalender oder die Kundenkontakte verwaltet. Multimedia wird vor allem eingesetzt:

- zum Lernen,
- zur Unterhaltung,
- zur Werbung,

wobei die Zwecke nahtlos ineinander übergehen. Ein gutes multimediales Werk zum Lernen ist unterhaltsam und wirbt zugleich für sein Thema. Ein Beispiel einer multimedialen Webseite ist <http://www.pippilotttirist.net/>.

Was uns als bewegtes Bild (Film, Fernsehen, Animation) erscheint, ist in Wirklichkeit eine rasche Folge ruhender Bilder (E: frame, F: cadre). Das gilt für das Dauemenkino ebenso wie für den Rechner. Die Bildfrequenz muss wenigstens 16 Bilder pro Sekunde betragen, damit wir den Eindruck einer flüssigen Bewegung haben.

Die anfallenden Datenmengen sind enorm, die für ihre Verarbeitung verfügbare Zeit ist kurz. Die Datenmenge lässt sich vermindern, indem man nur Differenzen zum vorangegangenen Bild überträgt und raffinierte Kompressionsverfahren einsetzt. Begleitender Sound muss synchron zu den Bildern wiedergegeben werden, und zwar über Stunden und auch über Pausen oder Sprünge hinweg. Das klappt nicht immer.

Da in der Unterhaltungsbranche viel Geld mitspielt, versucht die Industrie mit Patenten, Verschlüsselungen (Kopierschutz, Content Scrambling System, CSS) und technischen Tricks (Regional Playback Control, nichtöffentliche Standards) den Markt zu beherrschen. Ein Benutzer muss damit rechnen, dass nicht alles, was technisch möglich ist und sogar sinnvoll wäre, tatsächlich funktioniert und vom Gesetzgeber abgesegnet ist. Ein ehrlich erworbenes, standardkonformes DVD Video ist unter Umständen nicht auf dem heimischen DVD-Laufwerk abzuspielen, obwohl es technisch einwandfrei funktioniert. Die Rechtslage variiert von Staat zu Staat und von Jahr zu Jahr. IANAL.

Im Fernsehen werden aus Gründen, die in der Technik der frühen Jahre liegen, bei uns 50 Halbbilder pro Sekunde dargestellt. Dieses als Interleaving¹ bezeichnete Verfahren erlaubt die glattere Darstellung schneller Bewegungen und passt äußerst schlecht zur Videodarstellung im Rechner, der nur Vollbilder aufzeichnet und wiedergibt. Weitere Parameter sind Seitenverhältnis, Zeilen pro Bild und Pixelform. Die Farbräume stimmen natürlich auch nicht überein. Selbst beim Fernsehen gibt es mindestens drei Normen: PAL, SECAM und NTSC. Die Wiedergabe von analogem Fernsehen oder Fernsehaufzeichnungen über den Rechner geht nicht ohne Qualitätsverluste ab.

Um die Standardisierung der Dateiformate bemühen sich die *Moving Picture Experts Group* (MPEG, <http://www.chiariglione.org/mpeg/>) in der ISO und die *International Telecommunication Union* (ITU, <http://www.itu.int/home/>). Die MPEG-Standards tragen Bezeichnungen wie MPEG-1, die ITU-Standards heißen H.263 oder ähnlich. Der Standard MPEG-1 Audio Layer 3 ist als MP3 im Audibereich populär geworden. An einem DVD Video sind mehrere Standards beteiligt, siehe den Eintrag in der deutschen Wikipedia.

In den Debian-Paketlisten sind die Video-Anwendungen noch nicht in einer eigenen Gruppe versammelt, sondern unter *Graphics*. Das *DVD Playback HOWTO* von 2004 ist dem älteren *Linux DVD HOWTO* vorzuziehen, das sich noch mit Problemen befasst, die mittlerweile gelöst sind. Video4Linux (V4L, <http://www.video4linux.net/>) ist ein Projekt, das Software und Informationen bereit hält, meist als Hyperlinks auf entsprechende Webseiten. Linux TV (<http://www.linuxtv.org/>) ist dem Fernsehen unter Linux gewidmet, einschließlich Digital Video Broadcasting (DVB) und Video Disc Recording (VDR). Die Site hält Verbindung zu V4L. LinVDR (<http://linvdr.org/>) ist hilfreich, wenn es um Videoaufzeichnung auf DVDs unter Linux geht. Guides, FAQs, ein Glossar und weitere Informationen über DVD Audio/Video sind auf <http://www.doom9.org/> zu finden (jedoch nicht das Computerspiel ähnlichen Namens). Das Videokonferenz-

¹ Ein Interleaving gab es auch bei den Sektoren der ersten Festplatten im PC.

system GnomeMeeting wird in Abschnitt 11.8.2 *Internet-Telefonie* auf Seite 382 erwähnt.

10.2 Konverter, Videoeditoren

Zum Konvertieren verschiedener Datenformate dienen besondere Konverter oder allgemeine Videoeditoren, wie im Audibereich. An Konvertern sind verfügbar:

- XviD (<http://www.xvid.org/>) ist ein MPEG-4 Video Codec,
- Transcode (<http://www.transcoding.org/>) ist ein Werkzeugkasten mit mehreren Werkzeugen für die Kommandozeile, nicht bei Debian,
- mencoder, gehört zum Spieler MPlayer, siehe nächster Abschnitt, und versteht dieselben Formate wie dieser,
- ffmpeg, kommt zusammen mit einem Player namens ffplay und einem Streaming Server ffserver,
- MainActor von der Firma MainConcept (<http://www.mainconcept.de/>), kommerziell, aber gemäßigt im Preis.

Als Videoeditoren lassen sich folgende Softwarepakete bezeichnen:

- LinuxVideoStudio, eine Gtk-Oberfläche für die MJPEG-Tools von <http://prdownloads.sourceforge.net/mjpeg/>,
- ZS4, <http://www.zs4.net/>, nicht bei Debian, für den privaten Gebrauch kostenfrei,
- Avidemux, <http://avidemux.berlios.de/>, nicht bei Debian,
- kino, <http://kino.schirmacher.de/>, ein Editor für Digital Video mit Anklängen an den vi-Texteditor.

10.3 Media Player

10.3.1 VLC Media Player

Media-Player geben multimediale Schöpfungen wieder. Das schließt einfache Medien wie Audio ein. Mit einem Media-Player lassen sich sowohl eine Audio-Datei oder -CD als auch ein DVD Video oder eine Filmdatei abspielen. Die Player unterscheiden sich in der Auswahl von Formaten, die sie beherrschen, und in Zusatzfunktionen wie Schnappschuss oder Zeitlupe.

Der VLC Media Player (VLC = Video LAN Client, beheimatet auf <http://www.videolan.org/vlc/>) machte die wenigsten Schwierigkeiten. Nachdem eine Gerätedatei dvd als Link auf cdrom angelegt war:

```
debian:/dev# ln -s cdrom dvd
```

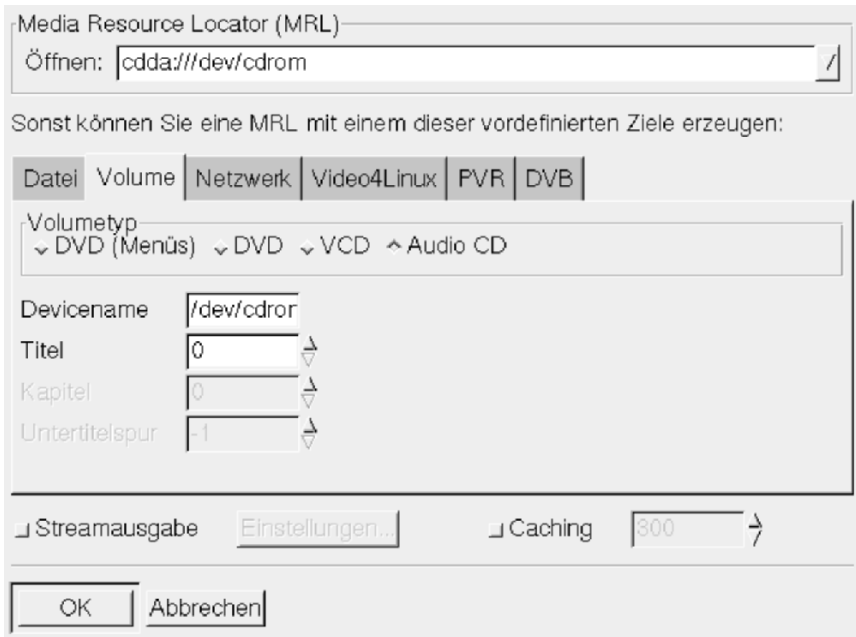


Abbildung 10.1. Screenshot des Auswahl Fensters des Media-Players VLC. Ausgewählt ist eine Audio-CD.

die Zugriffsrechte überprüft worden waren (666 root/cdrom) und noch eine nicht bei Debian erhältliche Bibliothek von einem Webserver als rpm-Paket heruntergeladen und mittels `alien -i` eingerichtet worden war, konnte das erste DVD Video mit Bild und Ton wiedergegeben werden. Falls Sie nichts hören: an den Mischer denken! Mit dem Kommando `svlc` wird ein ansprechenderes Frontend aufgerufen, der Rest bleibt wie beim Aufruf `vlc`. Abbildung 10.1 zeigt das Auswahlfenster des Players, wie es sich nach Anklicken der Menüpunkte *Datei -> Medium öffnen* darbietet. Hier ist eine Audio-CD im Gerät `/dev/cdrom` ausgewählt. Entsprechende Fenster ermöglichen den Zugriff auf Dateien, DVDs oder Quellen im Netz. Der Player kennt zahlreiche Formate und arbeitet auch als Streaming Server. Ein solcher Server spült Audio- oder Videodaten in das lokale Netz, wo sie von Streaming Clients aufgefangen und fast in Echtzeit (durch Puffern leicht verzögert) wiedergegeben werden.

10.3.2 Xine und Frontends (Xine, gxine, Kaffeine, Totem)

Xine (<http://xinehq.de/>) ist eine Bibliothek mit Funktionen für die Wiedergabe von Multimedia, auf der ein Dutzend Frontends oder Benutzeroberflächen aufsetzt: Xine-UI, Totem, Kaffeine, `gxine` (aus *unstable*), Sinek (aus *oldstable*), Opie, Pyxine und ein Plugin für aRts, um nur die bei Debian verfügbaren zu nennen. Im Prinzip sind sie alle ähnlich. Nach der Einrichtung von Xine sollte man das Pro-

gramm `xine-check` aufrufen, das Hinweise gibt, ob alles in Ordnung bzw. was noch zu tun ist. Abbildung 10.2 zeigt einen Schnappschuss aus einem mittels Totem abgespielten DVD Video. Die Menüpunkte zum Schnappschießen sind *Bearbeiten* -> *Bildschirmfoto*. Anschließend musste mit GIMP das Seitenverhältnis korrigiert werden, Menüpunkte *Bild* -> *Bild skalieren*, wobei die Bindung von Breite und Höhe aufzuheben war. In Kaffeine wird ein Schnappschuss im png-Format entweder durch die Tastenkombination `<ctrl>-<s>` oder durch Anklicken der Menüpunkte *Datei* -> *Schnappschuss speichern* erzeugt. Das Xine-User-Interface funktionierte ebenfalls problemlos. Kxine (<http://kxine.sourceforge.net/>) scheint verdorrt und durch Kaffeine abgelöst zu sein.



Abbildung 10.2. Schnappschuss aus einem mittels Totem wiedergegebenen DVD Video. Die Garnierung des Fensters fehlt – anders als bei einem Screenshot.

10.3.3 Mplayer, XMMS

Bei Debian gibt es nur die beiden MPlayer-Plugins *mozilla-mplayer* und *xmms-xmmpayer*, deren Einrichtung voraussetzt, dass der MPlayer (Movie Player) von <http://www.mplayerhq.hu/> heruntergeladen und eingerichtet worden ist. Dazu holt man sich die Quellen (sources), Codecs, Fonts und Skins und baut daraus den MPlayer zusammen. Etwas bequemer verläuft die Einrichtung eines inoffiziellen Debian-Paketes von CHRISTIAN MARILLAT. Die Datei `/etc/apt/sources.list` ist vom Verwalter um folgende Zeile zu erweitern:

```
deb ftp://ftp.nerim.net/debian-marillat/ sarge main
```


Dann aktualisiert man seine lokale Paketliste:

```
debian:~# apt-get update
```

und holt sich schließlich in gewohnter Weise das Paket:

```
debian:~# apt-get install mplayer-586
```

Die Manualseite ist recht ausführlich. Die Einrichtung der beiden oben genannten Debian-Pakete ging anschließend reibungslos über die Bühne. Wir testen den MPlayer zuerst mit einigen Audio-Dateien (wav, mp3, ogg):

```
joe@debian:~$ mplayer lilimarleen.mp3
```

und beobachten die Textausgabe des Players. Geht alles gut, wagen wir uns an ein DVD Video:

```
joe@debian:~$ mplayer dvd://1
```

Das funktionierte, allerdings manchmal erst im zweiten Anlauf. Vermutlich sind da einige Optionen suboptimal gesetzt. Wollen wir den MPlayer mit einer grafischen Oberfläche haben, rufen wir ihn als `gmplayer` auf. Nach einiger Konfigurationsarbeit klappte das auch.

Der MPlayer bringt ein Werkzeug namens `mencoder` zum Umcodieren von Filmen mit. Es beherrscht alle Formate, die auch der MPlayer kennt.

Das X MultiMedia System `xmms` begann mit dem Plugin, ein DVD Video abzuspielen, stieg aber nach einigen Minuten mit der Bemerkung aus, der Rechner sei zu langsam (1,3 GHz). Die Modularisierung stellt offenbar erhöhte Ansprüche.

10.3.4 Ogle

Ogle ist ein DVD-Player aus Göteborg (<http://www.dtek.chalmers.se/groups/dvd/>), der bei Debian in einer einfachen Fassung sowie in einer Version für CPUs mit MMX-Erweiterungen (praktisch alle neueren Typen) zu haben ist. Ebenso gibt es ein Debian-Paket mit einer GTK-Oberfläche. Die Einrichtung der Pakete `ogle-mmx` und `ogle-gui` verursachte keine Probleme; die Benutzung des Werkzeugs ist einfach, alle wichtigen Funktionen stehen während der Wiedergabe über ein eigenes Fenster mit Buttons zur Verfügung. Mit avi-Dateien kam der Player nicht klar.

10.3.5 Kaboodle, Noatun

Der Media-Player Kaboodle aus dem KDE-Projekt kommt schlank und schlicht daher und spielt Dateien ab. Erwartet man mehr Funktionalität, wendet man sich an seinen großen Bruder Noatun (<http://noatun.kde.org/>). Genau genommen ist er ein Frontend zum Backend *Analog Real-Time Synthesizer* (aRts) aus dem Debian-Paket `libarts1`, Heimathafen <http://www.arts-project.org/>. Um die Fähigkeiten des Players auszuschöpfen, sollte man zum Debian-Paket `noatun` auch die

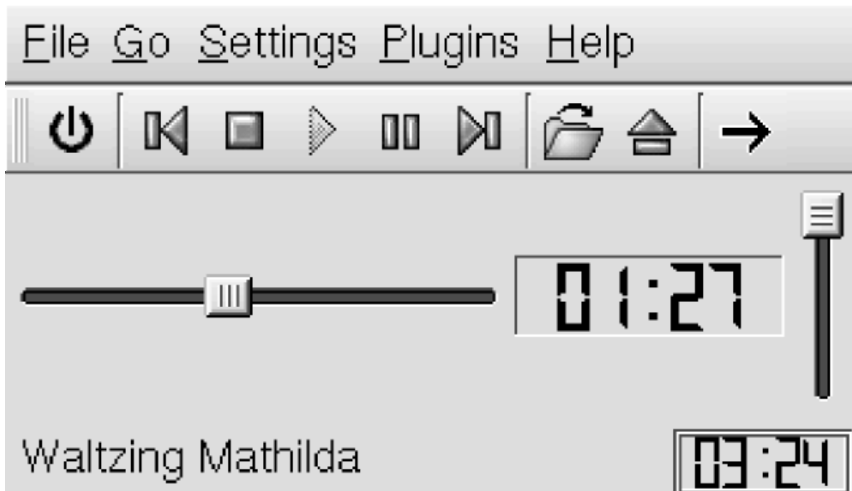


Abbildung 10.3. Screenshot des KDE-Media-Players *noatun* bei der Wiedergabe einer Audiodatei

Pakete *noatun-plugins* und *kdemultimedia* einrichten. Abbildung 10.3 zeigt sein Aussehen in der Defaultansicht; es lässt sich vielfältig konfigurieren. Wie die meisten Player bringt *noatun* eine Menüleiste, eine Anzeige des Spielfortschritts mit der Möglichkeit des Verstellens, eine Uhr, einen Steller für die Lautstärke sowie Knöpfe für Abspielen, Pause, Anhalten, schneller Vorlauf und schneller Rücklauf sowie Wiederholen mit. Der Spieler kann nach einer Wiedergabeliste (E: playlist, F: liste d'écoute) arbeiten. Über den Menüpunkt *Einstellungen* -> *Klangregler* bekommt man einen einfachen Equalizer auf den Schirm.

10.3.6 Helix/Real Player

Den Real Player gibt es nicht als Debian-Paket, sondern bei <http://www.real.com/linux/> zum Herunterladen. Die Standardversion ist kostenlos, die Plusversion kostet. Man sollte vor dem Klicken denken. Zu der Sitzung ziehen Sie am besten Ihren Rechtsbeistand hinzu, denn Sie müssen nicht nur bestätigen, dass Sie das Kleingedruckte gelesen haben und akzeptieren, sondern auch, dass Sie es verstanden haben. Bei der Einrichtung unter Linux wird zunächst eine Binärdatei heruntergeladen. Diese ist aufzurufen und besorgt den Rest. Anschließend steht ein Kommando *realplay* im Verzeichnis */usr/bin/* zum Aufruf bereit. Der Helix Player stammt aus derselben Schmiede und besteht ausschließlich aus offener Software.

Mit Sound-Dateien der Typen wav, mp3, ogg und rm bzw. ram hatte der Player keine Schwierigkeiten, auch die Zusammenarbeit mit dem Web-Browser Mozilla funktionierte. Beim Anklicken von CD-Clips (Hörproben) bei Amazon und anderen beanstandete der Player oft, dass sie mit einem veralteten, nicht mehr unterstützten Audio-Codec hergestellt seien, und verweigerte die Wiedergabe. Bei anderen Clips dagegen wurde sogar ein Film zum Sound dargeboten.

Will man mittels `realplay` einen Internet-Radiosender hören, ist unter dem Menüpunkt *Datei* -> *Adresse öffnen* dessen URL einzugeben. Wollen Sie das Programm 1 (P1) von Sveriges Radio hören, wäre dies:

```
rtsp://sr-rm.qbrick.com/broadcast/cluster/encoder/02038_p1.rm
```

also *nicht* der URL der Webseite `http://www.sr.se/`. Sie kommen jedoch per Klick von der Webseite auch zu dem Sender. In diesem Fall gibt Ihr Browser mittels Plugin die Sendungen wieder. Wer noch die Kommandozeile beherrscht, ruft das Kommando mit obigem URL als Argument auf oder schreibt den Aufruf in ein kleines Shellskript namens `sr-p1`. Nach einer kurzen Verzögerung, während der der Empfangspuffer gefüllt wird, meldet sich der Norden. Internetradio ist eine Alternative zum Kurzwellenempfang, mit anderen Arten von Qualitätsschwankungen. Während beim herkömmlichen Radio die Anzahl der Hörer keine Rolle spielt, muss der Internetradio-Server wie ein Webserver jeden Hörer einzeln bedienen. Es zeichnen sich aber Lösungen für größere Hörerkreise ab.

10.4 Fernsehen mit dem PC (xawtv, motv, tvtime, alevtd)

Die Elektronik eines PCs ist in einigen Teilen der eines Fernsehers ähnlich. Es fehlt nur der Hochfrequenzteil des Fernsehempfängers. Die unterschiedlichen Normen für die Bilderzeugung bereiten Schwierigkeiten, aber ansonsten liegt es nahe, mit Hilfe einer Einsteckkarte aus dem PC einen Fernseher zu machen (drahtlos, nicht über das Internet). Solche Karten gibt es, oft mit Radioteil (UKW/FM). Die Websites `http://video4linux.net/`, `http://linux.bytesex.org/` und `http://www.exploits.org/v4/` bieten einen Einstieg in das Thema. Digitales Fernsehen wird in Ansätzen von Linux unterstützt; Näheres dazu bei `http://linvdr.org/`. Vor der Anschaffung einer analogen oder digitalen TV-Karte unbedingt informieren, welche Karte unter welchen Voraussetzungen funktioniert.

Viele TV-Karten enthalten den Brooktree Video Decoder Chip Bt8x8, wobei das `x` für 4 oder 7 steht. Mit diesen Karten befasst sich das *BTTV HOWTO* von 2005, das man vor dem Erwerb einer TV-Karte lesen sollte. Hat man bereits eine Karte eingebaut, gibt das Kommando `lspci` Auskunft über den Chip. Finden sich in der Ausgabe Zeilen mit Bt848 oder Bt878, ist die Sache klar, und die Erfolgschancen steigen. Unter Umständen ist es erforderlich, den Audioausgang der TV-Karte über ein kurzes Kabel mit dem Audioeingang (Line, nicht Mikrofon) der Soundkarte zu verbinden. Das Audiosignal läuft dann durch den Mischer, siehe Abschnitt 9.2 *Mischer* auf Seite 331. Falls Sie nichts hören, sind dessen Einstellungen zu prüfen. Alternativ können Sie der TV-Karte auch ein eigenes Paar Aktivboxen spendieren. Mit dem Aufruf:

```
joe@debian:~$ dmesg | grep bttv
```

stellt man fest, ob ein passender Treiber geladen worden ist, bei *sarge* wahrscheinlich. Das Kommando:



Abbildung 10.4. Schnappschuss eines Fernsehbildes mittels der Grab-Image-Funktion (<J>) von xawtv

```
joe@debian:~$ ls -l /dev/video*
```

sollte eine nichtleere Menge von Videogeräten mit passenden Zugriffsrechten (660 root/video) anzeigen, was eine notwendige, jedoch nicht hinreichende Bedingung ist. Fernsehberechtigte Benutzer sind zusätzlich in die Gruppe *video* aufzunehmen, genau wie bei Audiogeräten.

Der nächste Schritt ist die Einrichtung von Anwendungen. Das Debian-Paket *xawtv* (X11 Athena Widgets Television) bringt neben anderen Werkzeugen eine gleichnamige Anwendung für X11 mit, die unter *sarge* und XFree86 Version 4.3.0 mit einer älteren TV-Karte von Hauppauge problemlos zusammenarbeitete. Die Empfangsqualität war nicht so gut wie bei einem Fernsehgerät am selben Kabel, aber bei einigen Sendern ausreichend. Abbildung 10.4 zeigt den Schnappschuss eines Fernsehbildes mittels der Grab-Image-Funktion des Werkzeugs; ein Screenshot wäre hier genauso nutzlos wie bei einem DVD-Video. Zum Testen geben wir:

```
joe@debian:~$ xawtv -hwscan
```

ein und schauen uns die Antwort auf etwaige Fehlermeldungen an. Wenn das Locale nicht von der Xlib unterstützt wird, lässt uns das kalt. Die Senderliste liegt in der Datei */etc/X11/xawtvrc*, wird mit dem Werkzeug *scantv* (Auswahl PAL, Europe-West) erzeugt und lässt sich mit jedem Editor ändern:

```
debian:~# scantv -n PAL -f europe-west -o datei
```

Ohne Dateioption schreibt das Werkzeug nach *stdout*, ohne die anderen Optionen erfragt es die erforderlichen Werte. Wie üblich kann ein Benutzer mit einer persönli-

chen Datei `$HOME/.xawtv` eigene Wünsche verwirklichen. Hören Sie nichts, steht vermutlich der Line-Eingang des Mixers auf der Soundkarte auf null.

Ein Linksklick in das Fernsehbild bringt die Senderliste hervor, ein Rechtsklick eine Liste von Optionen (Helligkeit, Kontrast, Bild speichern usw.). Per Mittelklick wird die Senderliste durchgezappt. Einzelnen Tasten sind ebenfalls Funktionen zugeordnet, beispielsweise schaltet die Taste `<f>` zwischen Vollbild und knapp postkartengroßem Bild hin und her. Wichtig ist vor allem die Taste `<q>`. Eine Weiterentwicklung von `xawtv` in Richtung benutzerfreundliche Oberfläche ist `motv`. Die Funktionalität ist die gleiche. Das Debian-Paket bringt auch den Teletext-Browser `mtt` mit, der aber im Gegensatz zum Fernseher nicht einwandfrei lief.



Abbildung 10.5. Schnappschuss eines Fernsehbildes mittels der Taste `<s>` aus dem Fernsehwerkzeug `tvtime`

Eine Alternative zu `xawtv` ist `tvtime` (<http://tvtime.net/>). Die Website betont und begründet die hohe Qualität des Fernsehers. Das Programm ließ sich problemlos einrichten, wobei als Norm PAL und als Frequenzbereich Europa anzugeben war. Auf das Hinzufügen von SUID-root zwecks Prioritätsverbesserung wurde verzichtet; mit SUID-root soll man aus Sicherheitsgründen zurückhaltend umgehen. Nachdem die Kanalsuche erfolgreich abgeschlossen war, konnte der Fernsehgenuss beginnen. Die Pfeiltasten rauf-runter wechseln den Kanal, die Pfeiltasten links-rechts stellen die Lautstärke ein. Zum Schnappschiessen – siehe Abbildung 10.5 – reicht die

Taste <s>, zum Ausschalten die Taste <q>. Alles Weitere steht im Manual. Konfigurationsverzeichnisse sind wie üblich /etc/tvtime/ und \$HOME/.tvtime/. Bei Beendigung schreibt das Programm einige nützliche Hinweise zur Verbesserung der Konfiguration auf den Schirm.

Das Debian-Paket *alevtd* mit dem Heimathafen <http://www.goron.de/~froese/> richtet einen lokalen Webserver ein, der auf Port 5654 Videotext-Seiten zur Verfügung stellt, falls auf dem Rechner zugleich ein Programm wie *xawtv* läuft. Die Kombination funktionierte teilweise. Vielleicht war die TV-Karte zu alt für Videotext. Zu konfigurieren gibt es an *alevtd* nicht viel. Das Paket *alevt* (ohne d) enthält ein Werkzeug, das die Seiten lokal auf den Schirm bringt, ebenfalls noch nicht ganz ohne Mängel.

10.5 Webkameras (webcam)

Webkameras (E: webcam, netcam, live cam F: webcaméra) sind Kameras – oft, aber nicht notwendig nur für Stillbilder – die ihre Bilder ins Web stellen. Dazu gibt es zwei Wege:

- einfache Kameras, die an einen Rechner angeschlossen sind, der die Software zur Abfrage (Treiber usw.) und den Webserver zur Übertragung ins Netz vorhält, Anschluss USB oder Firewire (IEEE 1394),
- aufwendigere Kameras (ab 100 EUR), die einen eingebauten Webserver mitbringen und daher selbst ein Knoten mit eigener IP-Adresse im Netz sind, Anschluss Ethernet.

Ein Vertreter der ersten Gattung ist die Logitech Quickcam Pro 4000 mit USB-Anschluss, Vertreter der zweiten sind die AXIS 205 (indoor) und die AXIS 211 (outdoor). Webcams sind alltäglich geworden. Viele touristisch reizvolle Plätze kann man sich im Web nahezu in Echtzeit anschauen, siehe beispielsweise <http://www.thunerseewind.ch/webcam/webcam.html>. Behörden in Finnland und Schweden überwachen den Zustand wichtiger Strecken mit Webcams, deren Bilder im Internet jedermann zugänglich sind, siehe <http://194.23.40.120/view/view.shtml> oder <http://www.finnra.fi/alk/english/>. Auch in Verbindung mit Wetterinformationen finden sich Webcams, siehe <http://www.weather.cs.uit.no/>. Die erste Webcam zeigte übrigens eine Kaffeemaschine in der Universität Cambridge, UK.

Bei <http://www.tldp.org/> liegt ein Webcam HOWTO, aktuell von 2005, das man lesen sollte, bevor man eine Kamera erwirbt. Es gibt bei einigen Kameratypen Treiberprobleme. Hier muss man genau auf den Typ achten: Logitech Quickcam sagt beispielsweise noch gar nichts, es kommt auf weitere Merkmale an. PHILIPP SCHAER veröffentlicht unter <http://www.schaer.de/webcam/tutorial.html> ein Webcam-Tutorial für Debian GNU/Linux. Entgegen der Angabe im Tutorial sind die Werkzeuge *webcam* und *webcamd* nicht mehr im Paket *xawtv* enthalten, sondern in eigenen Debian-Paketen. Die beiden *webcam*-Werkzeuge holen Bilder von einem Videogerät und und schaffen sie zum

Webserver. Weitere Hilfen findet man bei <http://www.aboutdebian.com/webcam.htm>, <http://linuxwiki.de/WebCam> und <http://webcam.sourceforge.net/>. Die Debian-Pakete *came*, *camstream*, *camserv* und *gqcam* enthalten Alternativen zu *webcam*.

Beim Beobachten öffentlich zugänglicher Plätze denke man daran, dass es nach Urheber- und Datenschutzrecht nicht gestattet ist, Bilder zu veröffentlichen, auf denen einzelne Personen zu identifizieren sind, es sei denn, diese hätten zugestimmt. Auf <http://remus.jura.uni-sb.de/faelle/> ist Genaueres nachzulesen. Will man sein Hab und Gut überwachen, hilft das Debian-Paket *motion*, das bei Änderungen im Bild Alarm schlägt, ein Bewegungsmelder.

Kommunikation

Hier werden verschiedene Wege zur Kommunikation mit anderen Benutzern beschrieben, vom einfachen Terminaldialog bis zur Groupware.

11.1 Grundbegriffe

Unter Kommunikation wollen wir den Gedankenwechsel zwischen Benutzern verstehen. Dass auch Maschinen untereinander oder mit Benutzern Informationen austauschen, ist zumindest nicht Thema dieses Kapitels. Die Kommunikationstechnik entwickelt sich seit der Etablierung des Internets rasch und vielfältig, beeinflusst nicht nur durch den technischen Fortschritt, sondern auch durch wirtschaftliche und politische Faktoren. Falls Sie den Überblick nur mit Mühe bewahren können, stehen Sie nicht allein. Letzten Endes laufen aber alle Dienste darauf hinaus, dass Bits oder Bytes von A nach B transportiert und vor Ort in Informationen der Menschenwelt umgesetzt werden bzw. umgekehrt. Der Rest sind Einzelheiten der Umsetzung und des Transports, die den Benutzer wenig interessieren.

Im einfachsten Fall liegen die Informationen oder Gedanken in Textform vor. Das stellt die geringsten Anforderungen an Hard- und Software und ist deshalb unter Linux/UNIX seit Anbeginn möglich. Erweiterte Zeichensätze kosten ein paar Bits mehr und erfordern eine entsprechende Vereinbarung zwischen Absender und Empfänger, sind aber nichts grundsätzlich Neues. Einige Netzdienste wie Email und News sind ursprünglich für Texte entwickelt worden und können mittlerweile auch andere Daten übertragen, brauchen das aber nicht in jedem Fall zu beherrschen. Die heutigen grafischen und akustischen Fähigkeiten der Rechner sowie Netze hoher Geschwindigkeit ermöglichen weltweite Videokonferenzen mit vielen Teilnehmern, sind jedoch nicht unbedingt trivial einzurichten und zu moderieren.

Bei einer Kommunikation mit maschinellen Hilfsmitteln unterscheiden wir:

- Einbahnstraßen wie Radio, Fernsehen, Durchsagen, die Message of the Day (/etc/motd) und das Web (größtenteils),
- Kommunikationswege mit Gegenverkehr wie Telefon, Email, Netnews, IRC.

Es gibt noch einen Mittelweg zwischen den beiden Möglichkeiten, nämlich die Einbahnstraße mit Richtungswechsel, im Funkverkehr Wechselsprechen genannt im Gegensatz zum Gegensprechen, wie wir es vom Telefon her kennen. Ferner gibt es Kommunikation:

- in Echtzeit (mit nicht spürbaren Verzögerungen der Nachrichten) wie Telefon, Radio, Internet Relay Chat (synchrone Kommunikation),
- mit Zeitversatz wie Email, News, Web, die nicht erfordert, dass der Empfänger zur selben Zeit wie der Sender am Gerät sitzt (asynchrone Kommunikation).

Neben diesen technischen Unterschieden spielt das Ausmaß an Öffentlichkeit der Nachrichten eine wichtige Rolle:

- Briefe, Telefonate, Mails gehen von einem Absender an einen genau bestimmten Empfänger, sie sind privat,
- Mitteilungen in Listen gehen von einem Absender an eine bestimmte oder unbestimmte Menge von Empfängern, je nach Konfiguration der Liste,
- Postings in News gehen ebenfalls von einem Absender an eine bestimmte oder häufiger an eine unbestimmbare Menge von Empfängern,
- Webseiten in öffentlich zugänglichen Webservern richten sich an eine unbestimmbare Menge von Empfängern.

Schickt man eine privat gemeinte Mail aus Versehen in eine Newsgruppe, kann das peinlich werden. So etwas kommt vor. Hier muss der Benutzer aufpassen, zumal einige Werkzeuge (Browser) die Unterschiede zwischen den einzelnen Netzdiensten vor dem Benutzer zu verbergen trachten. Der Benutzer soll sich nicht mit den Protokollen abgeben müssen, aber bei diesem letztgenannten Punkt muss er aufmerken und wissen, was er tut.

Zur Kommunikation zwischen zwei Partnern ist ein Protokoll erforderlich, ein Satz von Regeln, der bestimmt, wie der Austausch vor sich geht. Wenn Sie eine Person ansprechen, sagen Sie zuerst *Guten Tag*, *Griß Gott*, *Griiezzi* oder *Moin* oder sonst etwas Freundliches, um die Kommunikation in Gang zu setzen. Der Gegenüber sagt dann seinerseits etwas oder macht ein dummes Gesicht. Vielleicht versuchen Sie es dann auf Englisch oder Französisch, so lange bis die Verständigung klappt oder Sie aufgeben. Für das Ende des Gesprächs gelten ebenfalls Regeln. Verschiedene Situationen verlangen unterschiedliche Regeln. Unter Computern ist das genau so. Für das Internet sind die Regeln, die Protokolle in zahlreichen Requests For Comments (RFC) beschrieben.

Schließlich sei darauf hingewiesen, dass Datenverkehr im Internet abgehört, verfälscht oder umgeleitet werden kann. Die Gegenmaßnahmen lauten Verschlüsselung und Authentifizierung. In überschaubaren lokalen Netzen ist das nicht so dramatisch. Darüber hinaus hat das Eindringen des Kommerzes ins Internet dazu geführt, dass manche Netzdienste nicht mehr eine ungetrübte Freude sind und sowohl Verwalter wie Benutzer sich mit einem gewissen Aufwand vor unerbetenem Datenverkehr schützen müssen (Virens Scanner, Emailfilter (Spamfilter), Firewalls). Im Internet übertrifft das Volumen unerwünschter Mails (Spam, Unsolicited Bulk Email, UBE) bereits das der erwünschten und belegt spürbar Bandbreite.

Die fortgeschrittenen Werkzeuge zur Kommunikation beherrschen mehrere Funktionen und sind daher schwierig einzuordnen. Die nachstehenden Abschnitte dienen nur der groben Orientierung.

11.2 Direkter Dialog (`write`, `wall`, `talk`, `mlchat`)

Das alte Linux/UNIX-Werkzeug `write` ermöglicht, einem anderen Benutzer, der auf derselben Maschine eingeloggt ist (mit `who` ermitteln), eine Textnachricht auf seinen Bildschirm zu schreiben, vorausgesetzt, er hat das mit dem Kommando `mesg y` erlaubt bzw. nicht mit `mesg n` verboten. Der Gesprächspartner antwortet, indem er seinerseits `write` aufruft. Viele Benutzer stellen aber diese Möglichkeit ab, da es lästig ist, mitten in einer Eingabe plötzlich fremde Botschaften auf den Schirm zu bekommen. Dem Verwalter steht das Kommando `wall` (`write to all`) zur Verfügung, mit dem er alle angemeldeten Benutzer erreicht.

Etwas jünger ist das Kommando `talk` oder `ytalk`. Dieses Kommando erlaubt Verbindungen zu mehreren Benutzern im Netz gleichzeitig. Voraussetzung ist, dass auf jeder beteiligten Maschine ein `talk`-Dämon (`talkd` oder `ktalkd`) läuft, meist als Knecht des `inetd`. Ein Textbildschirm wird horizontal halbiert für Rede und Gegenrede, unter X11 werden dafür zwei Fenster eingerichtet. Mittels der Taste `<esc>` holt man sich ein einfaches Hilfemenü. Gibt man dann `<q>` ein, wird die Verbindung beendet.

Das Debian-Paket `mlchat` stellt ein Werkzeug bereit, mit dem man in einem lokalen Netz ohne Benutzung eines Servers (Peer-to-Peer) eine kleine Konferenz aufbauen kann, an der mehrere Benutzer ähnlich wie mit `talk` teilnehmen. Die mitgelieferte Dokumentation ist mager. Als zugehörige Webseite ist <http://savannah.nongnu.org/projects/cameleon/> anzusehen, auch nicht ergiebig. Das Programm funktionierte leider nur in einer Richtung, schade. Vielleicht hat ein Leser mehr Erfolg.

11.3 Email, Listen (mail usw.)

Electronic Mail oder Email (F: courrier électronique, courriel) ist ein alter und wichtiger Internetdienst, zuerst beschrieben im RFC 821 von 1982. Ursprünglich gedacht zum Versenden kurzer ASCII-Texte (Mails) werden heute über Email alle möglichen Dateien einschließlich Viren verschickt, aber für Sperrgut (riesige Dateien) ist Email nicht geeignet. Dafür ist FTP der bessere Weg. Ein Benutzer braucht nicht die Einzelheiten der Protokolle und des Versandwegs zu kennen – ebenso wenig wie bei der Gelben Post – aber er muss wissen, wie er eine ordnungsgemäße Mail abfasst und verschickt und wie er umgekehrt an seine Mails kommt. Im Internet werden Mails nicht zwischengelagert, sondern sofort befördert. Die Laufzeiten bewegen sich im Bereich von Sekunden bis wenige Minuten. Email ist ein offener und unsicherer Dienst: Mails können mitgelesen oder verfälscht werden, ungefähr wie Postkarten.

Es gibt Wege, die Mails zu sichern, nur erfordern sie etwas Aufwand. Die Konfiguration eines Emailsystems ist Aufgabe des Verwalters und hat ihre Tücken, hauptsächlich weil die Anforderungen an die Sicherheit laufend steigen.

Der Benutzer arbeitet mit einem Programm, das allgemein als Mail User Agent (MUA) bezeichnet wird. Mit diesem liest und schreibt er seine Mails. Eines der ältesten und einfachsten in der Linux/UNIX-Welt ist `mail`. Der schlichte Aufruf:

```
joe@debian:~$ mail
```

zeigt die im Briefkasten des Aufrufers lagernden Mails an. Die Eingabe eines Fragezeichens listet die Kommandos auf, die `mail` versteht. Die Eingabe `<t>` (type) schreibt die aktuelle Mail auf den Schirm, `<d>` (delete) löscht sie. Mit `<q>` (quit) verlässt man das Programm. In Shellskripten und `cron`-Tabellen spielt es noch eine Rolle, ebenso wie sein etwas leistungsfähigerer Nachfolger `mailx`, aber für den Normalgebrauch nimmt man bequemere Werkzeuge.

Lange Jahre hindurch war `elm` ein brauchbarer MUA, aber seine Entwicklung scheint eingeschlafen zu sein. `mutt` ist ebenfalls ein textbasierter MUA (kein X11 erforderlich), der alles Notwendige und nichts Überflüssiges bietet. Der Umstieg von `elm` auf `mutt` fällt leicht. `mutt` unterstützt die Multipurpose Internet Mail Extensions (MIME), Verschlüsselung mittels GPG oder PGP sowie das Abholen der Mails von einem Server mittels POP3 oder IMAP. Ein Benutzer passt `mutt` an seine persönlichen Wünsche mittels einer Datei `.muttrc` in seinem Home-Verzeichnis an:

```
set from="alex-weingarten@t-online.de"
set charset=us-ascii
set print_command="lp"
set print=ask-no
set print_decode
set smart_wrap
ignore x-keywords
ignore x-spam-status
ignore x-spam-record
ignore x-seen
alias verein alex-weingarten@t-online.de \
            vorstand@xxx.de \
            kasse@yyy.de
```

ausführlich auf den Manualseiten zu `mutt` und `muttrc` erklärt. Weiteres unter <http://www.mutt.org/>.

Wer seine Mails auf einer grafischen Oberfläche lesen und schreiben möchte, kann zu `sylpheed` (Debian-Pakete `sylpheed` und `sylpheed-doc`, <http://sylpheed.good-day.net/>, Abbildung 11.1), KDE `kmail` oder GNOME `balsa` (<http://balsa.gnome.org/>) greifen, die einen ähnlichen Leistungsumfang wie `mutt` bieten..

Der Email-Client und Newsreader Thunderbird stammt aus derselben Schmie-
de wie die Web-Browser Mozilla und Firefox, läuft aber unabhängig von diesen. Zweckmäßig richtet man die Debian-Pakete `mozilla-thunderbird` und `mozilla-thunderbird-locale-de` gemeinsam ein, weitere Pakete nach Bedarf. Das Werkzeug

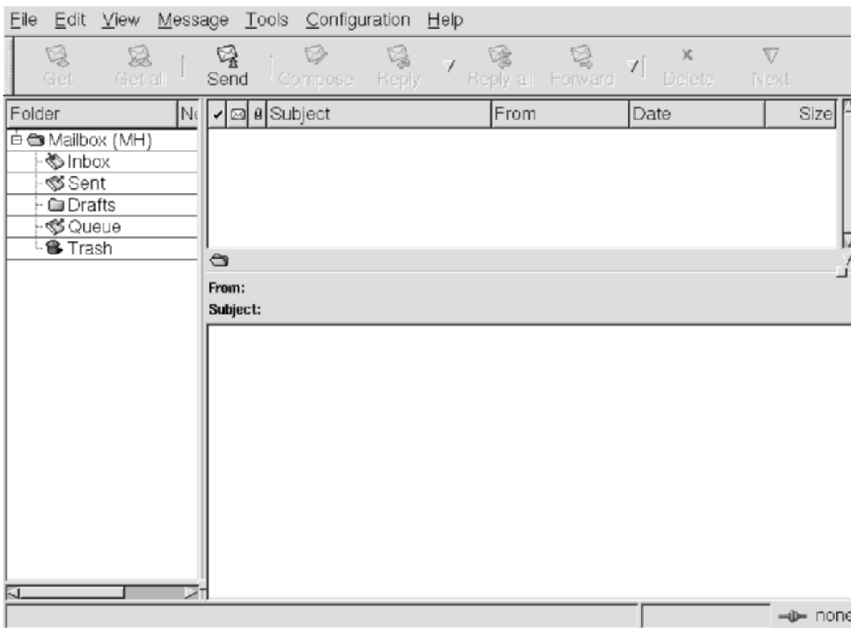


Abbildung 11.1. Screenshot des Mail User Agents sylpheed, ziemlich leer, da noch nicht konfiguriert

kommt Benutzern entgegen, die an MS Outlook gewöhnt sind, da es ähnliche Funktionen bietet:

- mehrere Konten, mehrere Benutzer,
- differenzierte Suche in Kopf und Rumpf,
- Virtuelle Ordner, Gruppierung,
- lernfähiges Spamfilter,
- elektronische Signatur, Verschlüsselung,
- abgestufte Darstellung von HTML-Mails,
- Abholen der Mails von POP3- oder IMAP-Servern.

Eine deutsche Rechtschreibprüfung fehlt wegen Lizenzproblemen, auf <http://www.thunderbird-mail.de/> werden aber Lösungen angeboten. Beim ersten Aufruf kann er Einstellungen usw. von Netscape oder Mozilla importieren. Ansonsten hilft der Konten-Assistent beim Anlegen des Emailkontos. Die tägliche Arbeit unterscheidet sich nicht von der mit anderen grafischen Email-Clients.

Die großen Web-Browser wie Mozilla oder Opera beherrschen neben dem Web auch Email und Netnews, der Trend geht jedoch zu schnellen, schlanken Browsern, die nur zum Surfen im Web dienen (Firefox beispielsweise). Eine Kombizange ist nicht immer besser als drei einzelne Spezialzangen.

Neben einem MUA kann ein emailender Benutzer noch einige ergänzende Werkzeuge gebrauchen. Das Filter procmail mit dem Stammhaus <http://www.>

`procmail.org/` kann eingehende Mails in verschiedene Postfächer (Verzeichnisse oder Dateien einschließlich `/dev/null`) einsortieren, an andere Benutzer weiterleiten (forwarden) oder abhängig vom Inhalt beliebige Programme starten. Es kann auch Spam (unerwünschter Werbemüll, Unsolicited Bulk Email, UBE, F: pollupostage) aussortieren, obwohl das schon zu einem früheren Zeitpunkt erfolgen sollte, nicht erst, wenn der Briefkasten schon zugestopft ist. `procmail` lohnt sich nicht, wenn man pro Tag eine Mail bekommt, aber wenn einem Benutzer die Mailflut bis zum Hals steht, sollte er mit seinem Verwalter über dieses Filter reden. Der Listenmanager `smartlist` setzt auf `procmail` auf und stammt aus demselben Haus; er hilft, wenn man mehrere Listen abonniert hat.

Das Debian-Paket *mpack* enthält Werkzeuge, um beliebige binäre Dateien in ein MIME-Format zu codieren und, falls gewünscht, auch gleich noch zu verschicken:

```
joe@debian:~$ mpack -s subjekt datei adresse
```

Der Empfänger decodiert die Mail mit dem Werkzeug `munpack` aus demselben Paket. Einen verwandten Coder/Decoder bringt das Debian-Paket *uudeview* mit.

Enthält eine Mail andere Daten als Text – sagen wir Sound – sind die Fähigkeiten der MUAs zur Darstellung schnell überfordert, sie müssen externe Helfer heranziehen. Der Datentyp einer Mail wird in ihrer Kopfzeile *Content-Type* angegeben, im einfachsten Fall als `text/plain`. Die in den RFC 2045 ff. festgelegten Datentypen werden im Rechner in den Dateien `/etc/mime.types` und `$HOME/.mime.types` aufgelistet, die zugehörigen Helfer in den Dateien `/etc/mailcap` und `$HOME/.mailcap`. Das Debian-Paket *metamail* stellt eine Grundausrüstung her und sollte von Verwalter systemweit eingerichtet werden. MUAs, die mit `mailcap` zusammenarbeiten, kommen dann auch mit Mails klar, die sie selbst nicht darstellen können.

Obwohl die MUAs selbst in der Lage sind, Mails per Post Office Protocol (POP, RFC 1939) oder Internet Message Access Protocol (IMAP, RFC 2060) von einem Mailserver abzuholen, ist es oft zweckmäßig, diese Aufgabe einem eigenen Programm zu übertragen. Die Debian-Pakete *fetchmail*, *fetchmail-ssl* und *fetchmailconf* enthalten alles Nötige. Wenn auch ein Benutzer `fetchmail` für sich allein einrichten kann, ist eine vom Verwalter eingerichtete systemweite Lösung sinnvoller. Aber man kann mit einer Einzellösung anfangen. Dazu hat ein Benutzer nach der Einrichtung der Pakete in seinem Home-Verzeichnis eine Datei `.fetchmailrc` folgender Art anzulegen, möglichst auf einem lokalen Server, der durchläuft:

```
set daemon 2000
set logfile /home/wulf/.fetchmail.log
poll pop.t-online.de proto pop3 user "alex-weingarten" \
                                password "?"
poll pop.uni-karlsruhe.de port 995 proto pop3 user "xyz" \
                                password "?" ssl
poll post.strato.de proto pop3 user "web@alex-weingarten.de" \
                                password "?"
```

Die erste Zeile bewirkt, dass `fetchmail` als Dämon im Hintergrund startet, unabhängig von der Sitzung weiterläuft und alle 2000 s aufwacht, um Mails zu holen.

Will man sichergehen, dass der Dämon nach Systemhalts wieder aufersteht, bittet man den cron-Dämon, ein- oder zweimal am Tag `fetchmail` zu starten. Die zweite Zeile teilt dem Dämon seine Protokolldatei mit, eine harmlose Sache. Die drei folgenden Zeilen sind wesentlich und müssen stimmen. Sie beauftragen den Dämon, Mails von den drei angegebenen Servern mittels des Protokolls `pop3` für den genannten Benutzer mit dem jeweiligen Passwort – hier durch ein Fragezeichen ersetzt – abzuholen. Die Passwörter stehen im Klartext in der Datei, weshalb diese die Zugriffsrechte 600 hat. Es empfiehlt sich auch, für solche Zwecke nicht sein Rechnerpasswort zu verwenden. Die Verbindung zur Universität Karlsruhe wird durch SSL geschützt, siehe Abschnitt 3.7 *Secure Sockets Layer, Transport Layer Security* auf Seite 112.

Webmail ist ein Weg, Mails mit Hilfe eines Web-Browsers zu lesen und zu schreiben. Das ist unterwegs gelegentlich praktisch, wenn man von einem Internetcafé aus seine Email erledigen möchte. Einen Web-Browser findet man dort immer. Mit dessen Hilfe geht man auf die Webmail-Seite seines heimatlichen Webservers, der um etwas Software für die Zusammenarbeit mit dem Emailsistem ergänzt sein muss. Beispielsweise kann ich von einem beliebigen Ort aus die Webmail-Seite des Rechenzentrums der Universität Karlsruhe aufrufen, mich mit Namen und Passwort anmelden und anschließend meine im Rechenzentrum lagernden Mails lesen, aber natürlich nicht die Mails, die für mich bei T-Online oder sonst wo liegen. Wollte ich auch das, müsste ich zu Hause, wo alle Mails zusammenkommen (via `fetchmail`), unseren Webserver um Webmail ergänzen und den Port 80 in der Firewall für beide Richtungen durchschalten. Das ist kein Hexenwerk, aber ein Loch mehr in der Firewall und in meinem Fall zu viel Aufwand für zu wenig Mail¹. Bei Debian finden sich einige Pakete für diesen Zweck, ebenso liefert eine Suche nach *webmail* bei Sourceforge mehrere Treffer. Die Einrichtung ist Sache des Verwalters, da mehrere Programme aufeinander abgestimmt werden müssen und Email ohnehin heikel ist.

Von einer persönlichen Antwort des Empfängers abgesehen gibt es keinen zuverlässigen Weg um festzustellen, ob eine Mail den Empfänger erreicht hat. Manche Absender versuchen, das mit einem Trick herauszufinden. Sie binden einen Hyperlink auf ein Bild oder dergleichen in die Mail ein. Wird das Bild vom Empfänger der Mail abgerufen, muss er sie geöffnet haben und darüber hinaus ziemlich leichtsinnig sein. Aktive Inhalte haben in einer Mail nichts verloren. Man weiß nie, was dahinter steckt. Bei Transportproblemen (unzustellbare Mail) versucht der absendende Mailserver wiederholt mehrere Tage lang, die Mail loszuwerden. Hat er keinen Erfolg oder weist der empfangende Mailserver die Mail ab, geht sie mit einer Fehlermeldung an den Absender zurück. Sind Sie vorübergehend abwesend (Urlaub), können Sie eine automatische Antwort erzeugen lassen – beispielsweise mittels `procmail` – aber davon ist abzuraten. Erstens könnten sich unwillkommene Zeitgenossen für Ihre Abwesenheit interessieren, zweitens haben Listen mit automatischen Antworten Probleme. Falls Sie dennoch so etwas einrichten wollen, suchen Sie im Netz nach *vacation* und *autorespond*. Der bessere Weg ist, dringende Mails als Kopie an einen

¹ Außerdem will ich im Urlaub meine Ruhe haben.

Vertreter weiter zu leiten oder für eine globale Erreichbarkeit zu sorgen (Webmail, VPN).

Eine Mail besteht aus einem Kopf (E: header, F: en-tête) und einem Körper (E: body, F: corps). Die Kopfzeilen dienen administrativen Zwecken und werden automatisch erzeugt, teilweise aus Eingaben des Benutzers. Der Körper enthält die eigentliche, vom Benutzer geschriebene Nachricht und wahlweise am Ende eine vom Benutzer geschaffene, automatisch angehängte Signatur. Ein Auszug aus dem Kopf einer echten Mail:

```
From andreas@ritters.de Fri Jan 21 18:29:10 2005
Envelope-to: wulf@localhost
Received: from pop.t-online.de [194.25.134.88]
by localhost with POP3 (fetchmail-6.2.5)
for wulf@localhost (single-drop); Fri, 21 Jan 2005 18:29:04
Subject: Aktuelle Version der Gliederung
To: alex-weingarten@t-online.de
Cc: andreas@ritters.de
Content-Type: text/plain
Content-Transfer-Encoding: 7bit
```

Die Zeilen bedeuten:

- Absender und Absendedatum,
- letzte (jüngste) Zustellanschrift (auf häuslichem Server),
- letzte (jüngste) Station des Versandweges; für jede Station erfolgt ein Received-Eintrag, der letzte zuoberst,
- Thema (Subject) der Mail, nach Angaben des Absenders, wichtig, weil der Empfänger auf Grund des Subjects entscheidet, ob er sich die Mail überhaupt ansieht oder ungelesen entsorgt,
- ursprüngliche Anschrift des Empfängers,
- Empfänger einer Kopie (carbon copy, CC) der Mail, hier der Absender selbst, was oft sinnvoll ist,
- Hinweise für den MUA, wie er die Mail zu decodieren hat.

Die Mail ist offenbar vor dem Absenden schon beim Empfänger eingetroffen, so schnell ist das Internet. Zum Body ist nicht viel zu sagen. Wer den Empfänger verärgern will, wählt eine ungebräuchliche Codierung. Dazu zählt auch HTML, von manchen Filtern grundsätzlich als spamverdächtig bewertet. Ich lese Mails mit einem MUA (mutt), nicht mit einem Web-Browser. Wer mir eine .doc-Datei schickt, bekommt die Antwort als \LaTeX -DVI-Datei, uuencodiert, wenn überhaupt. Die Signatur soll gemäß dem Internet Working Draft *Son-of-1036* von 1994 durch eine Zeile mit zwei Minuszeichen und einem Leerzeichen vom eigentlichen Inhalt der Mail abtrennen und nicht länger als vier Zeilen Text zu maximal 75 Zeichen sein. Gedacht ist die Signatur für Angaben zum Absender, die nicht in den Kopfzeilen stehen, beispielsweise die Telefonnummer, nicht zum Verkünden von Tiefsinn. Diese Signatur ist keine elektronische Unterschrift, die kommt gleich.

In einer Antwort (reply) auf eine Mail werden oft Teile der Mail zitiert. Die zitierten Zeilen sollen vor der Antwort stehen, da sie offenbar zum Verständnis der

Antwort beitragen. Man soll auch nicht mehr zitieren, als erforderlich ist. Die Anrede und die herzlichen Grüße brauchen nicht wiederholt zu werden.

Will man seine Mails signieren (elektronisch unterschreiben) und/oder verschlüsseln, so ist zunächst die entsprechende Software einzurichten, siehe Abschnitt 5.14.5 *Einrichten des GNU Privacy Guards* auf Seite 237, möglichst systemweit vom Verwalter. Wer unter Debian GNU/Linux mit dem MUA `mutt` arbeitet, braucht nichts weiter zu unternehmen. Nach dem Schreiben einer Mail, vor dem Abschicken, ruft man das `mutt`-Kommando `<p>` (privacy) auf, wird gefragt, was man will, gibt seine `gnupg`-Passphrase (Mantra) ein und schickt die Mail wie gewohnt auf die Reise. Zuerst sendet man sich selbst eine signierte Mail, dann eine signierte und verschlüsselte. Funktioniert das, wagt man sich an gute Freunde. Die Zusammenarbeit eines MUA mit `GnuPG` muss nicht immer so mühelos funktionieren wie bei `mutt`, aber die Dokumentationen zu `balsa` ab Version 2.2, `kmail` und `sylpheed` behaupten, dass ihre MUAs mit `GnuPG` zusammenarbeiten.

Unter Signatur wird hier eine aus der Mail und dem geheimen (privaten) Schlüssel des Absenders erzeugte Prüfsumme verstanden, deren Echtheit vom Empfänger mit dem öffentlichen Schlüssel des Absenders überprüft werden kann. Das ist eine andere Signatur als die vom MUA wahlweise an die Mail angehängte Signatur mit Angaben zum Absender wie Telefonnummer usw. Kann der Empfänger mit der Signatur nichts anfangen, weil sein MUA dafür zu dumm oder `GnuPG` oder eine vergleichbare Software nicht eingerichtet ist, ist das Lesen der Mail trotzdem möglich. Eine Signatur ist keine Verschlüsselung. Der Absender einer verschlüsselten Mail hingegen benötigt außer den Werkzeugen den öffentlichen Schlüssel des Empfängers. Damit verfügt der Empfänger zwangsläufig über die Einrichtungen und Kenntnisse zum Entschlüsseln.

Mailing-Listen (E: mailing list, F: liste de diffusion) oder kurz Listen sind ein Mechanismus innerhalb von Email, der es ermöglicht, eine Mail an viele Empfänger zu verteilen (Rundschreiben). In kleinem Maßstab kann das jeder Benutzer über Aliasnamen in der persönlichen Konfiguration seines MUAs erreichen, siehe die Datei `.muttrc` oben in diesem Abschnitt. Der Verwalter kann solche Aliasse systemweit einrichten. Größere Listen mit mehr Möglichkeiten erfordern ein Serverprogramm wie `minimalist`, `Majordomo` (veraltend) oder `Mailman`, <http://www.gnu.org/software/mailman/> und ebenfalls bei Debian zu haben. Ein Benutzer meldet sich bei einer Liste an (abonniert sie), empfängt alle Rundschreiben und darf selbst schreiben oder antworten. Es gibt öffentliche und geschlossene, moderierte und nicht-moderierte, lokale und globale Listen. Das Debian-Projekt betreibt mehrere Listen, siehe <http://www.debian.org/MailingLists/> oder <http://lists.debian.org/>. Als Debian-Listen-Server wird `smarthost` eingesetzt.

Bei den meisten Listen muss der Benutzer unterscheiden zwischen einer Mail an die Liste, das heißt an die Teilnehmer oder Abonnenten, und einer Mail an das Listenprogramm, betreffend administrative Fragen wie Abmelden oder Hilfe. Bei einer Antwort auf ein Rundschreiben muss er unterscheiden zwischen einer Mail an den Autor des Rundschreibens und einer Mail an die ganze Liste. Bei allem, was man ins Netz schickt, sollte man sich über den Empfängerkreis im Klaren sein.

11.4 Neuigkeiten: Netnews, Usenet (nn, gnus, knode, pan, tin)

Die Netnews (F: nouvelles Usenet), kurz News genannt, sind ein weiterer Kommunikationsdienst im Internet. Da der Dienst auch außerhalb des Internets vorkommt – ja sogar entstanden ist – lautet seine allgemeine Bezeichnung Usenet. Das Usenet ist keine Organisation, kein eigenes Netz, sondern die Menge der Personen und Rechner, welche die News schreiben, verteilen und lesen. Die Netnews sind die Schnittmenge aus Usenet und Internet. Am einfachsten stellt man sich die Netnews als eine Zeitung vor, die nur aus Leserbriefen besteht, keine Redaktion hat und fortlaufend erscheint. Ein anderes Bild zeigt das Usenet als eine Kneipe mit vielen Tischen, an denen sich die Gäste zwanglos zusammenfinden, um über bestimmte Themen zu diskutieren. Der Dienst ist überwiegend auf Text (ASCII) ausgerichtet.

Ein Benutzer braucht zur Teilnahme an den News eine Netzverbindung zu einem Newsserver wie `news.t-online.de` und lokal ein Clientprogramm, einen Newsreader. Einige Email-Clients und Web-Browser beherrschen ebenfalls das Network News Transport Protocol (NNTP) nach RFC 977. Die meisten Newsserver lassen nur einen begrenzten Teilnehmerkreis zu, der Newsserver der Universität Karlsruhe beispielsweise nur Angehörige der Universität. Wenn Ihr Provider keinen Newsserver vorhält, können Sie versuchen, bei <http://www.newsserverguide.de/>, <http://www.kuth.de/pns/> oder anderswo einen freien oder kommerziellen Server zu finden. Man bleibt in der Regel bei einem Server, da lokal Verwaltungsinformationen gehalten werden, die sich auf diesen Server beziehen. Als Clients kommen in Frage:

- textbasierte Reader: `nn`, `tin`, `trn` (non-free),
- X11-basierte Reader: `knews`, `gnus`,
- GNOME, GTK: `liferea`, `pan`,
- KDE, Qt: `knode`,
- Email- und News-Client: Thunderbird.

`gnus` nimmt eine Sonderstellung ein, weil es eigentlich ein Teil des XEmacs ist; das Debian-Paket bringt jedoch eine selbständige Version mit. Wegen seiner Mächtigkeit bezeichnet sich das Programm als *message reading laboratory*. Einzelheiten und Hilfe bei <http://www.gnus.org/>. Auch die bekannten Web-Browser beherrschen News, die Spezialwerkzeuge sind jedoch leistungsfähiger.

Wie eine gedruckte Zeitung sind die News in Rubriken gegliedert, die Newsgruppen (E: newsgroup, F: groupe de nouvelles) heißen. Bezeichnungen wie Area, Arena, Board, Brett, Echo, Forum, Konferenz, Round Table, Rubrik, Special Interest Group bezeichnen zwar etwas Ähnliches wie Newsgruppe, gehören jedoch nicht ins Internet. Es gibt:

- Internationale Gruppen, deren Sprache Englisch ist und die weltweit verbreitet sind,
- nationale Gruppen, die sich auf ein Sprachgebiet beschränken,
- lokale Gruppen, die sich auf eine Stadt wie Karlsruhe und deren Umland beschränken.

Große Server führen über 100.000 Gruppen. Eine von LEWIS S. EISEN zusammengestellte lange *Master List of Newsgroup Hierarchies* findet sich unter <http://www.magma.ca/~leisen/mlnh/>, eine Liste der deutschen Gruppen unter <http://www.dana.de/mod/>.

In einem lokalen Netz lässt sich ein eigener, kleiner Newsserver einrichten, der ausgewählte Gruppen aus dem Internet als Proxy und daneben auf das eigene LAN begrenzte Gruppen führt (*cnews*, *inn*, *leafnode*, *noffle*, *sn*). Die internationalen Gruppen gliedern sich in die *Großen Acht* (Big Eight):

- *comp* Computer Science, Informatik,
- *humanities* Geisteswissenschaften,
- *misc* Miscellaneous, Vermischtes,
- *news* Themen zu den News selbst,
- *rec* Recreation, Erholung, Freizeit,
- *sci* Science, Naturwissenschaften, Mathematik, Medizin,
- *soc* Society, Gesellschaftswissenschaften, Politik,
- *talk* Diskussionen aller Art.

Die *Großen Acht* sollten auf jedem allgemeinen Newsserver geführt werden. Dazu kommen alternative internationale Gruppen, die nicht überall verfügbar sind. Die Namen der nationalen Gruppen beginnen mit einem Kürzel: *de* für Deutschland, *fr* für Frankreich, *dk* für Dänemark. Bei den deutschsprachigen Gruppen richtet sich die nächste Gliederungsebene nach den *Großen Acht*. Die Namen deutschsprachiger Gruppen, die sich mit Informatik befassen, beginnen demgemäß mit *de.comp*. Je nach Bedarf setzt sich die Untergliederung noch über mehrere Ebenen fort. Die Namen lokaler Gruppen beginnen mit einem Ortskürzel, für Karlsruhe *ka*. Große Firmen wie Adobe oder IBM haben eigene Gruppen, die mit dem Firmennamen anfangen. Einige Beispiele:

- *news.newusers.questions* Questions and Answers for new users of Usenet,
- *soc.culture.nordic* kulturelle Fragen der nordeuropäischen Länder einschließlich Island, vorwiegend in Englisch,
- *alt.comics.asterix* Goscinny and Uderzo's brave gaulish warrior,
- *de.newusers.questions* deutschsprachige Gruppe für neue Teilnehmer an den News,
- *de.rec.fahrrad* deutschsprachige Gruppe, in der sich alles um das Fahrrad dreht,
- *france.alsace* La région Alsace,
- *it.eventi.papa2005* La successione papale,
- *thur.comp.linux* Thüringen, Linux-Forum,
- *ka.uni.rz.misc* Vermischtes aus dem Rechenzentrum der Universität Karlsruhe,
- *ibm.software.linux.ccl* befasst sich mit dem IBM Communication Controller for Linux,
- *linux.debian.announce* Official Announcements of the Debian Project (Read-Only)

Der Phantasie sind keine Grenzen gezogen. Einige Gruppen werden ad hoc zu einem aktuellen Thema eingerichtet und verschwinden nach einiger Zeit wieder, andere bestehen schon seit Jahrzehnten. In einigen erscheinen täglich mehrere hundert Zuschriften, Postings genannt, andere dümpeln vor sich hin. Einige sind moderiert, in anderen kann jeder schreiben, wie es ihm gefällt. In einigen ist der Ton locker, in anderen wird auf die Form geachtet. Ehe man selbst schreibt, sollte man einige Wochen gelesen haben. Man sollte auch nicht vor lauter Begeisterung zu viele Gruppen abonnieren, am Anfang vielleicht *de.newusers.questions* und noch zwei Gruppen aus anderen Gebieten. Die Auswahl lässt sich im Newsreader jederzeit ändern. Für Testzwecke stehen Gruppen wie *de.test* bereit. Dort lesen nur geduldige Programme (Reflektoren) mit, die Ihnen als Antwort Ihren eigenen Beitrag zurückschicken. Zunehmend sind Newsgruppen auch mit Seiten im Web vertreten, die Diskussionen finden aber nach wie vor in den Netnews statt.

Wenn Sie schreiben, beherzigen Sie folgende Punkte (ein Auszug aus der Netiquette und anderen Hilfen):

- Sie wenden sich an eine unbestimmbare Öffentlichkeit. Wenn Sie eine Person kritisieren, schreiben Sie nichts, was Sie nicht auch der Person in aller Öffentlichkeit sagen würden. Bleiben Sie sachlich und höflich.
- Schreiben Sie unter Ihrem richtigen Namen. Anonyme Zuschriften sind in den News ebenso wenig willkommen wie in einer Zeitung.
- Verwenden Sie für die News eine besondere Emailanschrift, nicht ihre gewöhnliche. Spamroboter grasen bevorzugt die News nach Emailanschriften ab.
- Achten Sie auf das Thema der Newsgruppe. Schicken Sie ein Posting nicht gleichzeitig an mehrere Newsgruppen (Crossposting).
- Antworten Sie nur in Ausnahmefällen gleichzeitig an die Newsgruppe und an die Emailanschrift eines Einsenders.
- Wenn Sie eine Frage in einer Newsgruppe veröffentlichen, erwarten Sie die Antwort ebenfalls in der Newsgruppe (damit auch andere etwas davon haben).

Die wichtigste Regel hat in weiser Voraussicht der Rektor der damaligen Universität Königsberg vor langer Zeit niedergeschrieben: *Handle so, dass die Maxime deines Willens jederzeit zugleich als Prinzip einer allgemeinen Gesetzgebung gelten könne*. Die Netiquetten werden regelmäßig in Newsgruppen wie *news.newusers.questions*, *news.answers* oder *de.newusers.questions* veröffentlicht. Eine hilfreiche Anleitung zum Fragen hat ERIC S. RAYMOND einmal verfasst; die deutsche Übersetzung der LUG Bozen liegt auf http://www.lugbz.org/documents/smart-question_de.html.

Wie verbreitet sich eine Zuschrift, ein Posting? Wenn Sie Ihrem Newsserver ein Posting übergeben, steht es sofort allen Benutzern dieses Servers zur Verfügung. Der Server nimmt von Zeit zu Zeit Kontakt mit Nachbarservern auf und tauscht mit diesen neue Postings aus. Das geht so weiter, bis das ganze Usenet durchdrungen ist. Beschleunigt wird das Verfahren dadurch, dass es so etwas wie übergeordnete Server gibt, die ein großes Gebiet versorgen. Die Laufzeit eines Postings in entlegene Winkel beträgt bis zu einigen Tagen.

Obwohl die Massenspeichergröße der Newsserver heute in Terabytes gerechnet wird, können die Server die News nicht unbegrenzt aufbewahren. Üblich ist eine Lebensdauer von wenigen Wochen. Danach sind die Postings verbreiteter Gruppen noch in Archiven zu finden, beispielsweise bei <http://groups.google.com/>. Naheliegenderweise sucht man mit der Suchmaschine Google in diesem Archiv, siehe Abschnitt 4.3.9 *Allgemeine Suchmaschinen* auf Seite 144.

Viele Postings enthalten Fragen. Manche Fragen wiederholen sich, weil immer wieder neue Teilnehmer dazu stoßen. Schließlich erbarmt sich jemand und stellt die häufigsten Fragen samt Antworten in einem Dokument zusammen, das regelmäßig in der zugehörigen Newsgruppe veröffentlicht wird. Diese Dokumente werden *Frequently Asked Questions* (FAQ, D: Fragen, Antworten, Quellen der Erleuchtung; F: Foire Aux Questions) genannt und sind eine wertvolle Hilfe beim Einarbeiten in ein neues Gebiet. Wir haben sie in Abschnitt 1.5 *Hilfen* auf Seite 36 kennen gelernt.

11.5 Internet Relay Chat (ircII, xchat, ksirc, zircon)

Früher trieben Leute, die Muße hatten, Konversation. Heute treiben Leute, die einen Internetanschluss haben, Kommunikation. Der Internet Relay Chat (F: service de clavardage IRC), kurz Chat oder IRC genannt, nach RFC 1459 ist ein textbasierter Kommunikationsweg, auf dem mehrere Teilnehmer in Echtzeit per Bildschirm und Tastatur miteinander diskutieren, ein globaler Dorftratsch sozusagen. Die Weiterentwicklung führt zu Videokonferenzen. Am besten funktioniert der Chat, wenn sich Leute regelmäßig treffen, die sich kennen. Der *Linux IRC mini-HOWTO*, aktuell von 2005, bereitet darauf vor.

Wie bei den News braucht ein Teilnehmer eine Netzverbindung zu einem IRC-Server und lokal einen IRC-Client. Es gibt öffentliche Server – Tendenz weniger werdend – und die Möglichkeit, im LAN einen kleinen, lokalen Server einzurichten (Debian-Paket *ircd* mit Varianten). Einige, teilweise beschränkt öffentliche Server sind:

- irc.fu-berlin.de mit Webseite <http://irc.fu-berlin.de/>,
- irc.belwue.de mit Webseite <http://irc.belwue.de/>,
- chat.t-online.de mit Webseite <http://community.t-online.de/>,
- [irc.oftc.net](http://www.oftc.net) mit Webseite <http://www.oftc.net/>,
- [irc.freenode.net](http://freenode.net) mit Webseite <http://freenode.net/>,
- [irc.debian.org](http://channel.debian.de) mit Webseite <http://channel.debian.de/>, wo eine Netiquette & HOWTO für den Kanal [#debian.de](http://channel.debian.de) liegt.

Auf den Webseiten geht es auch zu Informationen über den Internetdienst allgemein. An Clients stehen unter anderem folgende Debian-Pakete zur Verfügung:

- *cgiirc*, <http://cgiirc.sourceforge.net/>, zum Chatten mittels eines Web-Browsers,

- *ircii*, Kommando *ircII*, wird als klassisch bezeichnet, darauf aufbauend Bit-chX, <http://www.bitchx.org/>, dort Quellcode und Binaries als Tarball verfügbar,
- *epic4*, auf *ircII* aufbauend,
- *ksirc* aus dem KDE-Projekt,
- *kvirc* und *kvirc-doc*, ein grafischer Client,
- *lostirc* für X11,
- *pork*, ncurses-basiert, ähnlich wie *ircII*,
- *sirc*, Simple IRC, mit Perl-Skripten erweiterbar,
- *tinyirc*, schlanker als *ircII*, sonst ähnlich,
- *tirc*, mit Anklängen an den Editor *vi*,
- *tkirc*, auf Tcl/Tk aufbauend,
- *xchat*, für X11,
- *zircon*, mächtiger, nicht ganz einfach zu benutzender IRC-Client für X11.

Wer nur einmal schnuppern will, wählt einen schlanken Client, wer dagegen voll einzusteigen beabsichtigt, greift zu einem durch Skripte erweiterbaren oder von sich aus mächtigen Client.

Das Netzgeschwätz wird – ähnlich wie die News in Newsgruppen – in Gesprächskreise, Kränzchen oder Kanäle (E: channel, F: canal) eingeteilt, wobei nicht jeder Kanal auf jedem Server verfügbar ist. Ein Kanal wird einmal von einem Teilnehmer eingerichtet, der meist zugleich der Operator mit erweiterten Rechten ist, und bleibt bestehen, bis er gelöscht wird. In die lebhafteren Kanäle sind meist einige Teilnehmer unter ihren Nicknamen eingeloggt. Es gibt aber auch Kanäle, auf denen sich die Teilnehmer nur zu bestimmten Zeiten treffen.

Hat man das Debian-Paket *tinyirc* eingerichtet, so verbindet der Aufruf:

```
joe@debian:~$ tinyirc
```

den Benutzer mit dem Server *irc.debian.org*. Nachdem man einen eindeutigen Nicknamen gewählt hat, kann man das Kommando */list* eingeben, das die verfügbaren Kanäle auflistet. Mittels */join #debian.de* verbindet man sich mit dem Kanal *#debian.de* – bei entsprechender Reife eher mit *#debian.de.rentner* – und sollte die Beiträge der anderen Teilnehmer über seinen Bildschirm rauschen sehen. Was man selbst ohne führenden Schrägstrich eintippt, wird an die Runde verteilt, als erstes sinnvollerweise eine Begrüßung und Vorstellung. Die Runde verlässt man mit dem Kommando */quit*. Man muss das einmal erlebt haben. Vielleicht bleibt man dabei. Unterhaltsamer als manche Fernsehkanäle ist es allemal.

IRC-Nachrichten gehen von Haus aus unverschlüsselt über das Netz und sind daher nicht mehr zeitgemäß. Das Secure Internet Live Conferencing (SILC) arbeitet verschlüsselt und soll IRC ablösen, ist aber noch nicht verbreitet (<http://www.silcnet.org/>). Clients sind Gaim (im nächsten Abschnitt besprochen) und Silky (<http://silky.sourceforge.net/>, bei Debian-*unstable* erhältlich).

11.6 Instant Messaging (micq, imcom, kopete, gaim, jabber)

Instant Messaging (F: messagerie instantanée) ist eine Mischung aus talk, Internet Relay Chat (IRC), Email und File Transfer (FTP), deren Nutzen nicht offensichtlich, deren Erfolg aber um so deutlicher ist. Irgendwo mischt auch die Mobiltelefonie mit. Sicherheitsverantwortliche altern seit dem Aufkommen von IM deutlich schneller. Die IM-Dienste sind teilweise im Internet zu Hause (durch RFCs beschrieben, Jabber), teilweise sind sie zwar öffentlich und benutzen das Internet, folgen aber proprietären Protokollen (AOL Instant Messenger, Microsoft MSN Messenger), teilweise sind sie kommerziell und nur für den Einsatz in einem LAN oder Intranet gedacht (IBM Lotus Domino Messaging). Etwas Licht in den Nebel bringt die deutsche Wikipedia. Ein Unterschied zum IRC liegt darin, dass man sich nicht an vorgegebene Gesprächsrunden anschließt bzw. eine solche gründet, sondern mit einer selbst zusammengestellten Liste von Bekannten (Buddies) tratscht. Man kann aber auch einen Chatroom gestalten, der Ähnlichkeiten mit einem IRC-Channel aufweist. Auf jeden Fall hat der einzelne Benutzer mehr Einfluss auf seinen Freundeskreis und darauf, wer ihn als Freund bezeichnen darf. Wir beschränken uns auf für Debian GNU/Linux verfügbare, durch RFCs beschriebene Protokolle verwendenden Werkzeuge.

Der textbasierte IM-Client mICQ ist nicht bei Debian, aber als Debian-Paket bei <http://www.micq.org/> zu haben. Die Einrichtung mittels `dpkg -i` verlief ohne Zwischenfall. Dann liest man das Manual, Sektionen 1 und 7. Der erstmalige Aufruf:

```
joe@debian:~$ micq
```

legt wie üblich im Home-Verzeichnis ein Verzeichnis `.micq` an, in dem die Konfiguration und weitere Daten gespeichert werden. Der Client verlangt ein Passwort (nicht das wertvolle Systempasswort nehmen) und beantragt eine Benutzernummer bei `login.icq.com`. Geht der Versuch schief, löscht man das Verzeichnis `.micq` und fängt noch einmal an. Geht die Verbindung zum Internet über eine Firewall, muss diese den Verkehr durchlassen (Verwalter fragen). Innerhalb eines lokalen Netzes, in dem ein IM-Server (Debian-Paket *jabber*) werkelt, entfällt diese Hürde natürlich. IMcom ist ein in Python geschriebener, textbasierter Jabber-Client mit Ähnlichkeiten zu mICQ.

Beliebt ist der Jabber-Client Psi (*psi*), zu dem man noch das Debian-Paket *psi-translations* nehmen sollte. Der Client setzt X11 und Qt voraus. Einen Eindruck vermittelt Abbildung 11.2. Man meldet sich bei einem öffentlichen Jabber-Server wie `swissjabber.de` oder `jabber.ccc.de` an und ist damit für die Jabber-Welt unter einem URL (Jabber ID, JID) wie `walex@swissjabber.de` erreichbar. Dieser URL ist *keine* Emailanschrift, er kann nur von einem Jabber-Client verwendet werden. Die Jabber-Server verfügen über Gateways zu anderen IM-Welten wie AOL, teilweise mit eingeschränkter Durchlässigkeit. Es erleichtert den Start, wenn man schon den Jabber-URL eines Bekannten kennt und so gezielt eine Verbindung suchen kann. Die IM-Welt verwendet wie die anderen Dienste ein eigenes Vokabular, hinter dem sich aber nichts grundlegend Neues verbirgt. So ist ein Roster nicht nur eine Thüringer Rostbratwurst, sondern neuerdings auch eine Lis-



Abbildung 11.2. Screenshot des Instant-Messaging-Clients Psi beim Jabbern. Im oberen Fenster der bisherige Dialog, im unteren die nächste mit `<cr>` abzuschickende Nachricht.

te von Personen, Kontakten oder Buddies, im Englischen eigentlich ein Dienstplan. Wer bereits eine Anschriftenliste auf Papier oder im Rechner führt, kennt das und braucht sich nur an das Wort zu gewöhnen. Ein Jabberglossar findet man bei <http://web.amessage.info/Service/glossar> (ohne .html oder Schrägstrich am Ende).

Gaim setzt X11 und GTK+ voraus, ist modular aufgebaut (hat daher einige zusätzliche Pakete mit Plugins) und beherrscht mehrere IM-Protokolle, nämlich AIM, ICQ, Yahoo!, MSN, IRC, Jabber, Gadu Gadu und Zephyr. Wenn man AOL Instant Messaging (AIM) bevorzugt, ist er der Debian-Client der Wahl. Näheres bei <http://gaim.sourceforge.net/>. Gaim ist auch ein Client für Secure Internet Live Conferencing (SILC).

Kopete ist ebenfalls ein Multi-Protokoll-IM-Client und stammt aus dem KDE-Projekt. Näheres bei <http://www.kopete.org/>. Um diese Kommunikationsmöglichkeit zu beurteilen, hilft nur eines: Ausprobieren. Der Erfolg hängt stark von den Partnern ab, deshalb möglichst nicht als Einzelkämpfer starten.

Instant Messaging ist nicht zu verwechseln mit Text over IP (ToIP) nach RFC 2793. Dieser Dienst hat eher Ähnlichkeiten mit der Internet-Telefonie oder Voice over IP (VoIP), siehe <http://www.packetizer.com/voip/toip/>.

11.7 Präsentation (kpresenter, impress)

Eine Präsentation ist die Darstellung eines Themas vor einer Hörerschaft, ein im wesentlichen mündlicher Vortrag, unterstützt durch optische und/oder akustische Hilfsmittel, unter Umständen auch durch Geschmacksproben und dergleichen. Ein vielseitiges Hilfsmittel ist der Rechner in Verbindung mit einem Projektor (Beamer) und einer Lautsprecheranlage. Oft werden Auszüge aus dem Vortrag auf Papier (Handzettel, Handout) oder CD/DVD an die Teilnehmer verteilt; ersatzweise werden die Dokumente im Web veröffentlicht. Da nach einer abgewandelten Form des Gesetzes von EDWARD A. MURPHY, JR. die Technik genau dann versagt, wenn man sie am dringendsten braucht, zeugt es von Mut, sich bei einem Vortrag gänzlich auf die technischen Hilfen zu verlassen.

Für das Erstellen des Begleitmaterials benötigt man eigentlich keine besondere Software. Mit Texteditoren, Grafikprogrammen und Soundwerkzeugen lassen sich alle Wünsche verwirklichen. Zum bequemen Vorführen der Dokumente reicht ein einfaches Shellskript. Etwas aufwendiger ist eine Folge von Webseiten (HTML-Dokumenten), dargeboten von einem Web-Browser. Spezielle Werkzeuge für Präsentationen erleichtern die Arbeit, indem sie häufig vorkommende Teilaufgaben automatisieren und bestimmte Gestaltungsmerkmale vorgeben. Gleichzeitig verhindern sie einige Sünden wie zu kleine Schrift auf den projizierten Vorlagen. Bei der Vorführung erlauben sie ein einfaches Navigieren in der Sammlung. Eine Auswahl gängiger Präsentations-Werkzeuge:

- Adobe Illustrator
- MS PowerPoint
- Apple Keynote
- Harvard Graphics
- Lotus Freelance
- OpenOffice.org Impress
- KDE KPresenter (Teil von KOffice)
- GNOME ? (kein derartiges Werkzeug gefunden)

Nachteilig ist, dass sie die Gedanken des Autors auf Grund ihrer Vorgaben in bestimmte, sachlich nicht immer gerechtfertigte Bahnen leiten und den Zuhörer und -schauer mit Effekten vom Inhalt ablenken (was manchmal vielleicht gewünscht ist). Auf Grund der Verbreitung von PowerPoint reißen damit hergestellte Präsentationen mittlerweile niemanden mehr vom Hocker. Es werden mehrere Dateiformate verwendet; die meisten Werkzeuge können zumindest einige davon lesen.

An einem einfachen Beispiel mit Impress wollen wir uns den Werdegang des Begleitmaterials zu einem Vortrag veranschaulichen. Der Aufruf:

```
joe@debian:~$ ooimpress
```

bringt nach einigem Gerödel auf der Platte die erste Seite des Autopiloten (künftig: Assistent) auf den Schirm, auf der man sich zwischen einer noch leeren Präsentation, einer Vorlage oder einer bereits vorhandenen Datei entscheidet. Dann legt man Grundeinstellungen wie den Hintergrund und die Art des Diawechsels fest. Als Dia (E:



Abbildung 11.3. Screenshot aus einer Präsentation, erstellt mit OpenOffice.org Impress

slide, F: cliché) wird jeweils eine Seite oder Folie bezeichnet. Auf der fünften Seite des Autopiloten wird dem Autor eine siebeneteilige Gliederung des Vortrags nahe gelegt, die mit einer Titelseite beginnt – wogegen nichts einzuwenden ist – die aber im weiteren Verlauf nicht auf jeden Vortrag passt. Wer seinen Vortrag selbst gliedert, wählt besser die leere Präsentation.

Für die einzelnen Dias gibt es zwanzig Vorlagen zur Gestaltung, vom leeren Dia bis zu einem Dia mit Überschrift und vier Objekten, siehe Abbildung 11.3. Objekte sind Texte, Diagramme, Formeln, Bilder, Klänge oder dergleichen. Jedes Objekt kann mit optischen und/oder akustischen Effekten aus einer mitgelieferten oder selbst zusammengestellten Galerie verziert werden, außerdem mit einem Notizzettel für den Vortragenden. So fügt man Seite an Seite. Die Diafolge kann man sich handgesteuert oder automatisch vorführen lassen, ausdrucken, in acht Dateiformaten abspeichern oder in Formate wie pdf, html, eps und weitere exportieren. Eine fertige Präsentation lässt sich jederzeit ändern (editieren), nur vor der Titelseite kann keine Seite eingefügt werden, was einleuchtet. Für den Vortrag braucht man dann wieder Impress oder eine andere Software, die mit einem der Formate klarkommt. Als Minimum sollte ein Web-Browser verfügbar sein.

Der KPresenter sieht etwas anders aus als Impress und kam mit englischen Bezeichnungen, bietet aber die gleiche Funktionalität. Bei der Wiedergabe einer mit Impress erzeugten Diaschau hatte er kleinere Probleme mit den Effekten, die Bilder selbst wurden einwandfrei dargestellt,

11.8 Fax, Internet-Telefonie

11.8.1 Fax (hylafax, efax)

Telefonie und Fax sind zwei ursprünglich analoge Kommunikationsdienste, die das Telefonnetz benutzen, nicht das Internet. Man braucht daher einen Zugang vom Rechner zum Telefonnetz:

- lokal ein Modem, das zwischen einer seriellen Schnittstelle oder einer USB-Schnittstelle des Rechners und einem Telefonanschluss eingefügt und faxfähig ist, oder
- lokal eine ISDN-Karte im Rechner, die mit einem ISDN-Anschluss verbunden ist, oder
- ein Gateway (F: passerelle) im Internet, das Mails annimmt und als Faxe weiterleitet, beispielsweise <http://email4fax.de/>, <http://www.gnetx.com/>, <http://www.hostnet.de/faxgate.htm>, <http://www.efax-gate.de/> oder <http://fax.co.uk/> (allesamt kommerziell). Weitere finden Suchmaschinen nach Fütterung mit *email fax gateway*.

Die Verbindungen sind bidirektional, wenn nicht absichtlich anders konfiguriert. Lokal lassen sich wie üblich Einzellösungen oder Client-Server-Lösungen verwirklichen. Wenn in einem lokalen, per DSL und Firewall mit dem Internet verbundenen Netz auch nur ein Knoten über einen Telefonanschluss (analog oder ISDN) verfügt, muss dieser Zugang genau so sorgfältig gesichert werden wie der DSL-Zugang. Netzverwalter sehen solche Nebeneingänge nicht gern. Als Debian-Pakete stehen vor allem drei Hämmer unterschiedlichen Gewichtes zur Verfügung:

- *mgetty* mit mehreren *mgetty**-Paketen, eine kleine Lösung,
- *efax* aus dem gleichnamigen Debian-Paket oder aus dem Paket *efax-gtk*, das ein grafisches Frontend mitbringt, siehe <http://www.cce.com/efax/> und <http://efax-gtk.sourceforge.net/>, eine mittlere Lösung,
- HylaFAX mit den Paketen *hylafax-client*, *hylafax-server* und *hylafax-doc*, siehe auch <http://www.hylafax.org/>.

Die Einrichtung eines lokalen Fax-Gateways ist Aufgabe des Verwalters, der vor der Beschaffung von Hardware die Treiberfrage klären sollte. Eine Hilfe bietet die Webseite <http://freshmeat.net/projects/capi4hylafax/>. Zu HylaFAX kommt eine ausführliche Dokumentation mit. Das Kommando:

```
debian:~# faxsetup -client
```

konfiguriert und/oder verifiziert einen Hylafax-Client. Die Option *-server* leistet Gleiches für einen Server. Das Fax-Server-HOWTO befasst sich mit *efax*, stammt von 1997 und ist nicht sehr umfangreich.

Für den sendenden Benutzer stellt sich die Faxverbindung im einfachsten Fall wie ein weiterer Drucker dar. Eingehende Faxe werden entweder ausgedruckt oder als Mail-Anhang zugestellt. Bei lebhaftem Faxverkehr empfiehlt sich die Einrichtung eines Faxverzeichnisses parallel zum Mailverzeichnis, um die Dienste auseinander zu halten.

11.8.2 Internet-Telefonie (kphone, gnomemeeting, Asterisk)

Telefonieren über das Internet (IPTel), auch Voice over IP (VoIP, F: voix sur IP) genannt, obwohl es um mehr als nur Sprache geht (Video, Daten), ist eine Alternative zum althergebrachten Telefonnetz, sei es analog oder digital (ISDN). Im Internet werden die Daten in Form von Paketen aus Bits oder Bytes transportiert, nach den Regeln der TCP/IP-Protokolle. Die Endgeräte für VoIP müssen daher die akustischen oder sonstigen Signale digitalisieren, in netzgerechte Häppchen zerlegen und diese mit den von den Protokollen verlangten Zusatzinformationen garnieren. Ein Häppchen oder Paket enthält etwa 20 ms Sprache, Der Weg der Datenpakete verläuft nicht anders als bei Email oder FTP, es kommt nur die Erfordernis hinzu, dass die Verzögerungen unterhalb einer niedrigen Schwelle bleiben müssen (Größenordnung 100 ms), ein Problem, das auch bei den ersten transatlantischen Telefonkabeln zu lösen war. Hat man ohnehin einen Netzanschluss mit Volumentarif oder Flatrate, wird das Telefonieren innerhalb des Internets so billig wie das Versenden einer Mail, nämlich fast kostenfrei. Das *VoIP HOWTO*, aktuell von 2002, hilft beim Verständnis der Konzepte weiter, müsste aber an die gegenwärtige Hard- und Software angepasst werden. Neben Kostenvorteilen hat VoIP den Vorzug, dass nur noch ein Datenkabel (Ethernet) zum Arbeitsplatz führt, eventuell sogar nur eine drahtlose WLAN-Verbindung. Für einen alten Strippenzieher die Erfüllung eines Traums.



Abbildung 11.4. Drei Generationen von Telefonen, äußerlich kaum zu unterscheiden: links analoges Telefon mit TAE-Stecker, in der Mitte ISDN-Telefon mit RJ45-ISDN-Stecker, rechts Internet-Telefon mit RJ45-Ethernet-Anschluss

Die Aufgaben eines Endgerätes kann ein PC mit Soundkarte, Mikrofon, Lautsprecher und Software bewältigen (Softphone). Die technisch bessere Lösung ist

ein IP-Telefon, das einen selbständigen Netzknoten darstellt und per Ethernet oder WLAN an das lokale Netz angeschlossen ist. In Abbildung 11.4 sehen wir links ein einfaches analoges Telefon, in der Mitte ein ISDN-Telefon (Grundig) und rechts ein Internet-Telefon (Siemens) zum Anschluss als selbständiger Knoten an ein Intranet. Die wesentlichen Unterschiede sind in den Kästen verborgen, äußerlich weist der Stecker auf die Technik hin. Das Internet-Telefon wird entweder über ein Steckernetzteil oder über das Ethernet (Power over Ethernet) mit Strom versorgt. Auf dem Markt werden als Übergangslösung Adapter angeboten, um herkömmliche oder schnurlose, analoge Telefone mit dem Internet zu verbinden.

Zwei solcher Endgeräte können im Internet ohne weiteres miteinander telefonieren. Zur Verbindung mit dem herkömmlichen Telefonnetz bedarf es eines Gateways, wie es von Sipgate (<http://www.sipgate.de/>) oder T-Online betrieben wird. Wichtig ist, dass sich das Gate an die Protokolle des Internets (RFC 3261 *Session Initiation Protocol*, SIP) und/oder der International Telecommunication Union (ITU, <http://www.itu.int/>, vor allem die ITU-Empfehlung H.323 *Packet-based multimedia communications systems*) hält und nicht eigene Protokolle mitbringt, die den Benutzer in der Wahl von Hard- und Software sowie von Providern einschränken.

Mit der Internet-Telefonie ist die Frage nach der Adressierung einer Person oder eines Gerätes erneut in die Diskussion geraten. Ein Internet-Telefon kann in Deutschland entweder eine Ortsnetzzrufnummer wie ein Festnetztelefon haben oder eine Nationale Teilnehmerrufnummer (NTR), die wie eine Mobiltelefonnummer nicht ortsgebunden ist und mit der Ziffernfolge (0)32 beginnt, beispielsweise 032229002159. Keine deutsche Ortsnetzkennzahl beginnt mit 032, sodass es nicht zu Verwechslungen kommen kann. Beide Arten von Nummern können nomadisch benutzt werden (mit dem Benutzer auf Reise gehen).

Die Verbindung von Telefonnummern und Internet-Adressen hat ENUM (Telephone Number Mapping) zum Ziel. Unter einer Nummer (E.164-Nummer) sollen verschiedene, einem Benutzer gehörende Dienste wie Webseiten, Email, Fax, VoIP etc. zu erreichen sein. ENUM ermöglicht dem Inhaber einer Telefonnummer, mehrere Dienste unter dieser Nummer zu betreiben. Das setzt eine verteilte Datenbank ähnlich wie das Domain Name System (DNS) voraus, siehe RFC 2916. Dazu wird die Telefonnummer nach bestimmten Regeln in die Form eines DNS-Namens mit der Top-Level-Domäne *e164.arpa* übersetzt. Die Zuteilung einer VoIP-Nummer führt nicht automatisch zu einer ENUM-Eintragung, das sind zwei getrennte Vorgänge. Naheliegenderweise wird die oberste deutsche Datenbank beim DENIC (<http://www.denic.de/>) geführt, wo auch weitere Informationen angeboten werden. Da sowohl ENUM wie VoIP junge Entwicklungen sind, ist die Aktualität der Informationen zu beachten.

Das Werkzeug kphone mit der Startseite <http://www.wirlab.net/kphone/> ist ein SIP User Agent für Linux, der X11 voraussetzt und VoIP-Verbindungen herstellt. Die Dokumentation reicht für das Nötigste. Das Werkzeug gnomemeeting stellt ein Videokonferenzsystem ähnlich MS Netmeeting bereit und beherrscht natürlich auch VoIP (<http://www.gnomemeeting.org/>). Sein Dialpad sieht wie ein Mobiltelefon aus und lässt hoffen – passende Hardware,



Abbildung 11.5. Screenshot des Videokonferenzsystems gnomemeeting, hier das Dialpad zum Telefonieren

Treiber und Konfiguration vorausgesetzt – dass man damit wie gewohnt telefonieren kann, siehe Abbildung 11.5.

Die Asterisk-Software stellt unter Linux eine vielseitige Telefonanlage (PBX) bereit, die nicht nur Gespräche vermittelt und speichert, sondern auch als Call Center oder Mittelpunkt von Telefonkonferenzen arbeitet. Die Startseite ist <http://www.asterisk.org/>. Bei Debian besteht die Software aus 15 Paketen, was darauf hindeutet, dass sie eher für die Netze in Mittelbetrieben gedacht ist als für SOHO-Netze.

Einige hilfreiche Webseiten in Zusammenhang mit VoIP sind:

- <http://graphics.cs.uni-sb.de/VoIP/>,
- <http://www.packetizer.com/>,
- <http://www.openh323.org/>,
- <http://www.gnu.org/software/osip/osip.html/>,
- <http://www.packetizer.com/>,
- <http://www.sipforum.com/>,

- <http://www.voip-forum.com/>,
- <http://www.voip-info.de/>.

Das Telefonieren wird immer komfortabler, nur ein Problem wird wohl nie gelöst werden: die unterschiedliche Anordnung der Ziffern auf den Ziffernblöcken von Telefonen und Rechnern.

11.9 Vom Organizer zur Groupware

11.9.1 Übersicht

Ein Organizer aus Papier besteht aus einem Terminkalender, einer Anschriftenliste und einigen Blättern Papier, die ad libitum zu verwenden sind. Die elektronische Variante kann mehr, beispielsweise rechnen oder mittels GPS den Standort bestimmen, solange die Akkus mitmachen. Für ortsfeste Rechner gibt es Software, die diese Funktionalität bietet, in der Linux/UNIX-Welt seit alten Zeiten die Werkzeuge *calendar* (siehe Abschnitt 3.5 *Kalender, Uhr* auf Seite 107) und *awk* (siehe Abschnitt 7.2 *Einfache Datenbank mit awk, grep und sort* auf Seite 278). Verwaltet das Werkzeug auch noch Mails, Faxe und Telefonkontakte, gehört es zu den Personal Information Managern (PIM). Letzten Endes sind das Aufgaben für eine Datenbank, sodass der Hauptzweck des Managers darin besteht, dem Benutzer ein ergonomisches Frontend zu bieten. Arbeitet man sowohl ortsfest als auch transportabel, stellt sich die Frage, wie man die Daten in Einklang hält. Das erfordert Nachdenken und vor allem Disziplin, siehe Abschnitt 11.10 *Daten synchronisieren* auf Seite 396.

Erweitert man den Organizer auf die Benutzung durch eine Gruppe, so kommen einige Aufgaben vorwiegend kommunikativer Art hinzu, und wir gelangen zur Groupware, dem Wundermittel zur Effizienzsteigerung, Kostenoptimierung und Problembewältigung beim Teamwork. Groupware (E: collaboration software. F: collectif) ist Software zur Unterstützung der Zusammenarbeit in einer Arbeitsgruppe über zeitliche und/oder räumliche Abstände hinweg. Welche Funktionen dazu gehören, ist ebenso unbestimmt wie bei Office-Paketen. Es gibt einige obligatorische Grundfunktionen, der Rest schwankt. Zu den Grundfunktionen gehören:

- Email,
- Terminkalender,
- Adressbuch,
- Notizbuch.

Weitergehende Funktionen können sein:

- Projektverwaltung, Job Management, Zeiterfassung,
- Dokumentenverwaltung.
- Kontaktmanagement (Customer Relationship Management),
- Content/Site Management,
- Ressourcenverwaltung,
- Trouble Tickets, Request Tracker,

- Workflow Management,
- Foren, Wiki, Webserver, Chat.

Projektverwaltung oder -management ist selbst wieder eine zusammengesetzte Aufgabe, zu der man (nicht wir) eine zweisemestrige Vorlesung mit Praktikum veranstalten könnte. Im Umfeld von Groupware blühen weitere, meist englischsprachige Schlagworte wie *Computer Supported Cooperative Work* (CSCW) oder *Computer Coordinated Family Communication and Cooperation* (CCFCC). Früher sagte man Haustelefon dazu, noch früher reichte die kräftige Stimme des Familienoberhauptes, aber das ist undemokratisch und out. Einen Überblick findet man in der deutschen Wikipedia.

Bei kleinen Gruppen oder Projekten in einem lokalen Netz kann es das Einfachste sein, sich aus klassischen Linux/UNIX-Werkzeugen mit Hilfe von ein paar Shellskripten die benötigten Funktionen zusammenzustellen. Das Gruppenkonzept von Linux/UNIX unterstützt diese Arbeitsweise. Bei größeren Gruppen bietet eine Groupware-Lösung den Vorteil, dass alle Bausteine von vornherein zusammenpassen. Als einheitliches benutzerseitiges Frontend bietet sich ein Web-Browser an, den hat und kennt heute jeder. Es finden sich aber auch Client-Server-Lösungen mit groupware-spezifischen Clients, vor allem bei kommerzieller Software.

Das Zusammenführen verschiedener Informationen und Kommunikationswege unter einer gemeinsamen Oberfläche hat einiges für sich und die Vernetzung der Daten ebenfalls, aber man sollte sorgfältig überlegen, was man braucht, und die Sicherheit beachten, wozu in diesem Fall auch die Betriebssicherheit (Stromausfall, Plattencrash) zählt. Der Gedanke, dass ein Benutzer gar nicht mehr weiß, ob er auf eine Mail antwortet, in eine externe oder interne Newsgruppe postet, mit Kollegen oder Kunden jabbert oder ein Webformular ausfüllt, beunruhigt etwas. Die Software-Landschaft sieht gegenwärtig auch ziemlich bunt aus. Vielleicht zeichnet sich langfristig eine Lösung ab, die *alle* Daten in einer ordentlichen Datenbank mit standardisierter Schnittstelle ablegt, zu der die einzelnen Werkzeuge Frontends darstellen. Wenn als Werkzeug ein Web-Browser ausreicht, umso besser.

11.9.2 Notizbuch, Personal Information Manager (hnb, tina, Chandler, TuxCards, Kontact, openCRX)

Man kann kurze Notizen mit seinem Lieblingseditor in Dateien schreiben und diese in einem Unterverzeichnis sammeln. Beim Suchen nach einer Notiz oder allen Notizen zu einem Thema hilft `grep`. Wenn die Notizen (Dateien) mehr und mehr werden, wäre ein Ordnungsschema hilfreich, aber eine ausgewachsene Datenbank Overkill. Die Lücke füllen Notebook-Programme wie `hnb` oder die TuxCards. Beide Werkzeuge stellen ein hierarchisch gegliedertes Notizbuch bereit.

Das Programm `hnb` (hierarchical notebook) verwendet `curses`-Funktionen, kommt also ohne X11 aus. Die Einrichtung mittels `apt-get` noch unter *woody* verlief ohne Probleme. Das Erscheinungsbild ist schlicht, die Funktionalität reicht aus. Die Daten können als ASCII, HTML oder XML exportiert werden.

Das Werkzeug `tina` bezeichnet sich als Personal Information Manager und verwendet ebenfalls die `curses`-Funktionen. Das Manual gibt so gut wie nichts her; am

meisten erfährt man, indem man das Programm aufruft und ein Fragezeichen eingibt. Etwas mehr Hilfe wäre keine Verschwendung; ansonsten macht das Notizbuch einen vernünftigen, zweckgerichteten Eindruck.

Ein weiterer PIM ist *Chandler* von der Open Source Applications Foundation (<http://osafoundation.org/>). Die Software ist nicht als Debian-Paket zu haben, sondern nur als Tarball, und muss auf der eigenen Maschine übersetzt (compiliert) werden. Für Fedora Linux gibt es ein fertiges Binary. Das Werkzeug sieht sich als Nachfahre von Lotus Agenda aus den achtziger Jahren und verfolgt einige bemerkenswerte Ziele in Richtung Groupware.

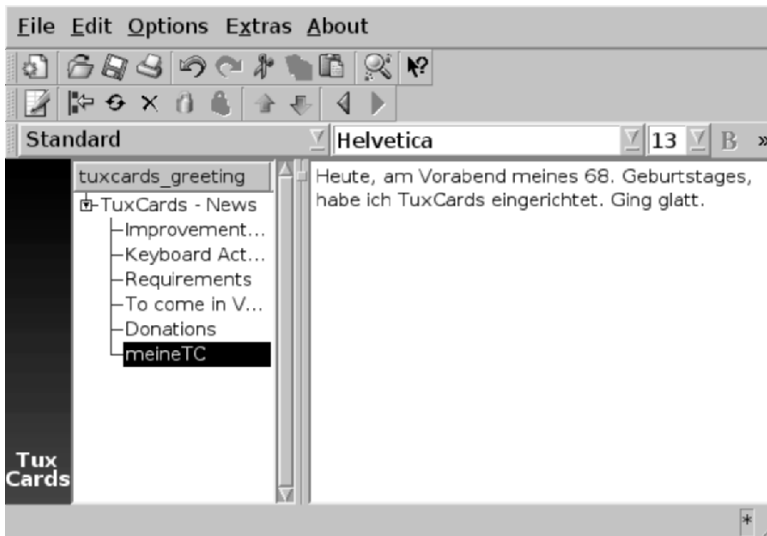


Abbildung 11.6. Screenshot des hierarchischen Notizbuches TuxCards

Die TuxCards sind den CueCards eines anderen Betriebssystems nachempfunden. Sie benötigen X11 sowie Qt und sind nicht bei Debian, aber als Debian-Paket bei <http://www.tuxcards.de/> zu bekommen. Das Paket ließ sich mittels `dpkg` unter *sarge* anstandslos einrichten, die Dateien liegen in `/usr/local/`. Eine Manualseite gibt es noch nicht, aber das Programm ist auch ohne Gebrauchsanleitung verständlich. Abbildung 11.6 zeigt einen Screenshot.

Das KDE-Werkzeug *kontakt* bezeichnet sich als *Integrierte Anwendung zur Verwaltung persönlicher Informationen* und vereint KDE-Einzelanwendungen für Email (KMail), Terminverwaltung (KOrganizer), Anschriften (KAddressBook), Notizen (KNotes) und News (KNode) unter einem Dach. Der Benutzer kann jedoch einzelne Dienste abwählen. Zusätzlich bringt *kontakt* einen Wetterbericht und einen Newsticker (KDE Dot News, Heise Online News und andere) mit. Der Wetterbericht liefert Zustandsmeldungen auswählbarer Orte, beispielsweise Straßburg, aber keine Vorhersagen. Dafür ist nach wie vor <http://www.dwd.de/> zuständig. Trotz

deutschem Locale redet `kontakt` Englisch. Insgesamt macht das Überwerkzeug einen aufgeräumten Eindruck.

Die KDE Personal Information Metapackage *kdepim* richtet etwas Ähnliches wie `kontakt` ein, bringt jedoch deutlich mehr Einzelwerkzeuge mit, darunter `kalarm`, eine Mischung aus `leave` und `cron`, `karm`, ein Time Tracker, und `kleopatra` zur Verwaltung von Sicherheits-Zertifikaten nach X.509.

Beschränken sich die Kontakte auf Kunden, wäre an ein Customer Relationship Management System (CRMS) wie `openCRX` zu denken, nicht bei Debian erhältlich, sondern als Tarball unter der BSD-Lizenz bei <http://www.opencrx.org/> oder <http://sourceforge.net/projects/opencrx/>. Derartige Werkzeuge beherrschen außer der Verwaltung der Kundendaten auch Sales Force Automation, Marketing Automation and Analysis, Customer Service, Budgeting, Forecasting, Controlling usw. Mit einem CRMS verkaufen sich Ihre Produkte von selbst, Sie können sich ganz seiner Pflege widmen.

11.9.3 Termine (`calendar`, `WebCalendar`, `KOrganizer`)

Das einfachste Werkzeug zur Terminverwaltung ist das alte Linux/UNIX-Kommando `calendar`. Es sucht im Arbeitsverzeichnis nach einer Datei namens `calendar`, die folgendermaßen aufgebaute Zeilen enthält:

```
#include calendar2
12/25 Weihnachten
```

Die erste Zeile bindet eine zweite Datei mit Terminen ein. Die zweite Zeile enthält ein Datum, ein Tabulatorzeichen(!) und beliebigen Text. Das Datum kann auf mehrere Weisen geschrieben sein. Es empfiehlt sich, zumindest zwei Dateien anzulegen: eine mit jährlich wiederkehrenden Terminen (Geburtstage) und eine andere mit einmaligen Terminen, die gelegentlich aufgeräumt wird. Man kann so eine kleine Hierarchie von systemweiten, gruppenweiten und persönlichen Terminen einrichten. Das Kommando `calendar` lässt der Verwalter jeden Morgen vom `cron`-Dämon für alle Benutzer ausführen:

```
05 06 * * * /usr/bin/calendar -a
```

Dann findet jeder Benutzer morgens in seiner Mail einen Gruß vom Reminder Service mit den Terminen des aktuellen und des nächsten Tages. Mit der Option `-l` lässt sich die Zeitspanne auf mehrere Tage ausdehnen, beispielsweise montags eine Wochenübersicht erzeugen. Das Kommando kann vielfältig konfiguriert werden, nur zwei Dinge kann es nicht: Termine selbst eintragen und Termine aufeinander abstimmen. Die Ausgabe ist schlicht gehalten.

Die Kalenderfunktion von `eGroupWare` basiert auf `WebCalendar` (<http://www.k5n.us/webcalendar.php>), einem Open-Source-Projekt, das auch bei <http://sourceforge.net> zu finden ist. Der `WebCalendar` benötigt eine der gängigen Datenbanken als Grundlage. Sein Funktionsumfang sprengt den Rahmen des Buches, unter anderem erkennt er Terminkonflikte. Zum Lösen der Konflikte empfiehlt er, der Sekretärin Schreibrechte im Kalender zu gewähren.

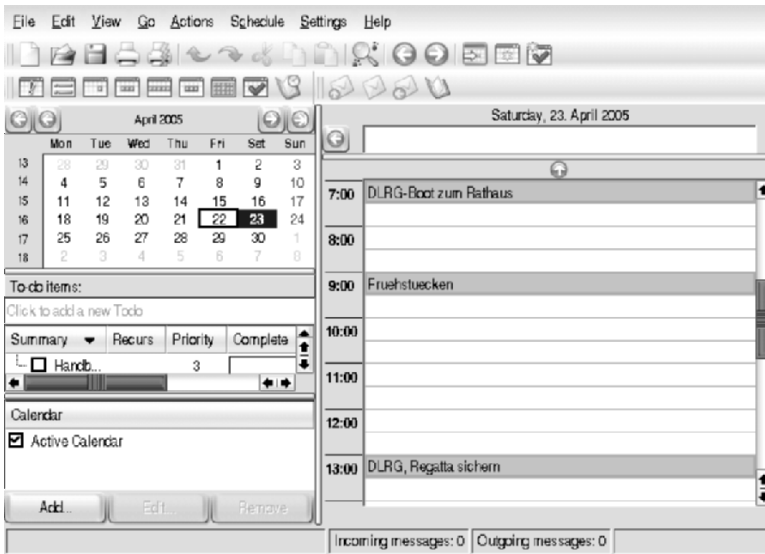


Abbildung 11.7. Screenshot des Personal Organizers KOrganizer

Der Personal Organizer *KOrganizer* verwaltet einen Terminkalender, siehe Abbildung 11.7. Er beherrscht das Personal-Data-Interchange-Format (PDI, Internet Mail Consortium, <http://www.imc.org/pdi/>) für elektronische Visitenkarten (vCard) und Terminabsprachen (iCalendar, vCalendar), die durch mehrere RFCs beschrieben werden und Standards im Internet darstellen. Eine Einführung in das Kalenderwesen des Internets gibt der RFC 3283.

11.9.4 Anschriften, Telefonnummern (XmAddressBook, KAddressBook)

Das Verwalten von Anschriften, Telefonnummern und ähnlichen Angaben zu Personen oder Sachen ist eine typische Aufgabe für Datenbanken oder Verzeichnisdienste. Groupware greift auch fast immer auf Datenbanken zurück und stellt nur Frontends bereit. Ein passives Verzeichnis ist nur eine halbe Sache, weshalb die hier zu erläuternden Werkzeuge zum Teil entweder mit Email- und Telefonprogrammen zusammenarbeiten oder deren Funktionen selbst übernehmen, womit wir beim Abschnitt 11.9.6 *Die Krönung: Groupware* auf Seite 393 wären.

Einfache Adressbücher finden sich unter *sarge* aber auch, in den üblichen Geschmacksrichtungen:

- *abook* arbeitet textorientiert mit curses-Funktionen,
- *xmaddressbook* setzt X11 voraus, verfügt aber auch über ein curses-Interface,
- *blackbook*, *dlume*, *gaby* und *rubrica* stammen aus dem GNOME-Projekt bzw. benötigen GTK,
- *kaddressbook* gehört zu KDE.

The screenshot shows a window titled 'XmAddressBook' with a menu bar containing 'File', 'Misc', and 'Help'. Below the menu bar is a toolbar with buttons for navigation (back, forward), search, goto, add, delete, update, and clear. The main area contains a form for a contact entry with the following fields:

- Name: Dr.-Ing. Wulf Alex
- Work Phone: 0160-8407076
- Fax Number: (empty)
- Home Phone: +49 7244 706516
- Company: Familie Alex
- Email Address: alex-weingarten@t-online.de
- Work Address: (empty)
- Home Address: Rieslingweg 14, 76356 Weingarten
- Date Updated: Sat 20:54:50 05/07/05
- Remarks: schreibt, dass die Tasten qualmen.

At the bottom, there is a status bar showing 'Current Entry: 3' and 'Total Number of Entries: 3'. A help button is also present with the text 'Help: Remarks or descriptive information'.

Abbildung 11.8. Screenshot des Adressbuches XmAddressBook

Das XmAddressBook speichert Anschriften und bietet Such-, Druck- sowie Email-Funktionen, bis auf die düsteren Farben ein angenehmes, sparsames, zweckmäßiges Werkzeug für den Benutzer, der nicht alle Bürotätigkeiten unter einer Oberfläche integriert haben will. Abbildung 11.8 zeigt das Aussehen eines Eintrags, vergleichbar einer Karteikarte. Das KAddressBook kommt eleganter daher und speichert wesentlich mehr Einzelheiten zu einer Person, bis hin zu einem Passbild und den geographischen Koordinaten des Schreibtisches. Biometrische Daten fehlen noch, aber der Benutzer kann eigene zusätzliche Felder definieren. Die Eingabe verteilt sich auf mehrere Fenster, den Screenshot einer Ausgabe zeigt Abbildung 11.9. Das Werkzeug kommt in die Nähe von Kontaktmanagern. Die anderen Werkzeuge bewegen sich zwischen diesen beiden.

11.9.5 Projekte (gnotime, kfocus, opensched, planner, GanttProject, gforge, maven)

Für große Software-Projekte gibt es mehrere Entwicklungsmodelle, die hier nicht diskutiert werden sollen. Auch bei umfangreichen Textprojekten mit vielen Autoren lassen sich diese Modelle anwenden, mit einigen Vereinfachungen. Ein Beispiel findet sich unter <http://www.v-modell-xt.de/>. Es geht immer darum, ei-

ne Mindestqualität unter Einhaltung von Zeit- und Kostenrahmen zu erreichen². Ein häufig benutztes Werkzeug zur Planung und Verfolgung von Projekten sind Gantt-Diagramme (Gantt chart, benannt nach dem Maschinenbauer und Unternehmensberater HENRY L. GANTT), die auf der horizontalen Achse die Zeit und auf der vertikalen Achse die Entwicklungsschritte des Projektes darstellen, den Start oben. Näheres in der deutschen Wikipedia unter dem Begriff *Gantt-Diagramm*. Ein weiteres Werkzeug bei der Projektverwaltung sind Netzpläne.

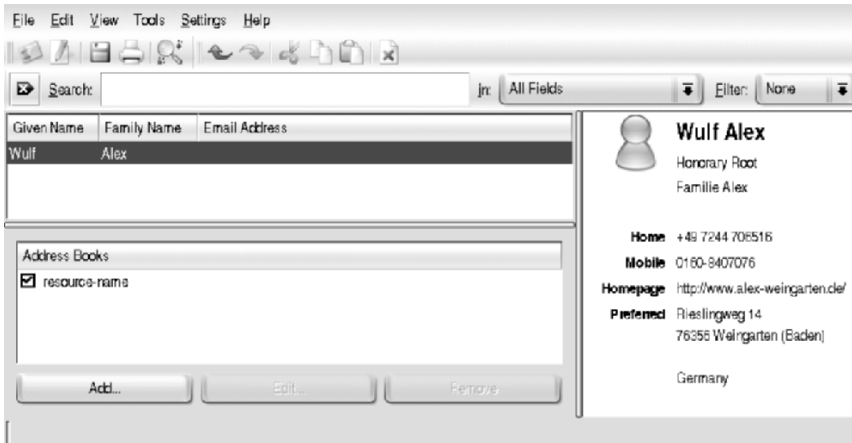


Abbildung 11.9. Screenshot der Ausgabe des KDE Adressbuchs KAddressBook

GNOME gnotime ist keine ausgewachsene Projektverwaltung, sondern dient nur der Verfolgung von Terminen und Aufzeichnung von Zeitverbräuchen (E: time tracker), ein quantitatives Tagebuch mit Vorausschau oder eine digitale Variante von MARCEL PROUSTS Roman *A la recherche du temps perdu* ohne dessen Unterhaltungswert. Ähnliche Funktionen bietet kfocuss aus dem KDE-Projekt, zu Hause unter <http://kfocuss.sourceforge.net/>.

Das einfache Werkzeug opensched benötigt eine ASCII-Textdatei mit einer Projektbeschreibung – das Format wird im Manual erklärt – und erzeugt daraus verschiedene Übersichten, Berichte, Tabellen und Diagramme. Arbeit an Wochenenden ist nicht vorgesehen; damit schied das Programm zur Planung des vorliegenden Buches aus. L^AT_EX und Encapsulated PostScript werden unterstützt. Außer bei Debian findet man es bei <http://opensched.sourceforge.net/>.

Das Werkzeug planner ist eine Projektverwaltung für GNOME und auf <http://planner.imendio.org/> beheimatet. Es geht zurück auf eine Software namens *Mr.Project*, die noch unter <http://mrproject.codefactory.se/> anzutreffen ist. Entsprechend seiner grafischen Ausrichtung macht planner einen moderneren Eindruck als opensched und ist vielseitiger in seinen Auswertungen. Abbildung 11.10 zeigt einen Screenshot. Links in dem dunklen Feld wählt

²Real Programmers don't believe in schedules.

man die Art der Darstellung aus: Gantt-Diagramm (wie dargestellt), Aufgaben, Ressourcen (Zeit, Kosten), Ressourcenverbrauch. Rechts in dem größeren Feld findet man beim Gantt-Diagramm unter einer waagrechten Zeitskala die Balken der Zeitspannen, bei den Aufgaben Einzelheiten wie Bearbeiter und Budget, bei den Ressourcen ähnliche Angaben und beim Ressourcenverbrauch wieder eine Zeitskala. Ein *Planner User Guide* kommt mit und lässt sich während der Arbeit einsehen.

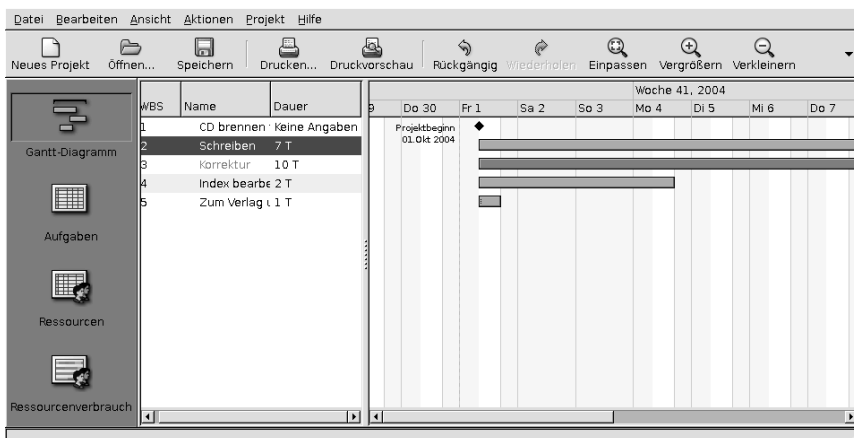


Abbildung 11.10. Screenshot der Projektverwaltung planner

Nicht bei Debian, sondern bei <http://ganttproject.sourceforge.net/> bekommt man die Software *GanttProject*, in Java geschrieben und als Multiplatform-zip-Archiv von 6 MB herunterzuladen. Die begleitende Dokumentation verspricht ein vielseitiges Werkzeug, wirklich eine Projektverwaltung; leider ließ es sich wegen einer Exception nicht ohne Nacharbeit zum Laufen bringen, obwohl Java in einer aktuellen Fassung eingerichtet war. Ein Java-Programmierer sollte wenig Mühe mit der Beseitigung des Hindernisses haben.

GForge ist ein Projekt unter der GPL zur Software-Entwicklung durch Arbeitsgruppen, geschrieben in PHP und bei <http://gforge.org/> zu Hause. Das gleichnamige Debian-Metapaket bezeichnet das Werkzeug als *Collaborative development tool* und erübrigt die Einrichtung von 10 Einzelpaketen. Das Werkzeug berücksichtigt den gesamten Lebenszyklus einer Software einschließlich Bug Tracking und Release Management. Für andere Projekte als Softwareentwicklung weniger geeignet.

Das *Apache Maven Project* (<http://maven.apache.org/>), nicht bei Debian, ist zunächst einmal wegen seiner Herkunft interessant. Wer den erfolgreichsten Webserver im Internet geschrieben hat, versteht etwas vom Programmieren. Auf seiner Webseite nennt sich Maven ein *software project management and comprehension tool*, also ein Werkzeug zur Projektverwaltung und zum Verständnis (Dokumentation, Change Logs, Software-Metriken, Distributionen etc.) von Soft-

ware. Es verwendet ein Objektmodell und empfiehlt sich als Alternative zu *Ant* (Debian-Paket und <http://ant.apache.org/>), das sich wiederum als Verbesserung gegenüber alten UNIX-Werkzeugen wie *make* sieht. Es kann als Tarball oder zip-Archiv heruntergeladen werden und dürfte ebenfalls auf die Entwicklung von Software beschränkt sein.

11.9.6 Die Krönung: Groupware

Freie (offene) Groupware-Projekte sind:

- eGroupWare (EGW, <http://www.egroupware.org/>),
- Evolution (<http://www.novell.com/products/evolution/>),
- Kolab (<http://www.kolab.org/>), ehemals Kroupware,
- OpenGroupware (<http://www.opengroupware.org/> und <http://www.skyrix.de/>),
- SuSE (Novell) Linux Openexchange Server (SLOX, <http://www.novell.com/products/openexchange/>),
- phpGroupWare (<http://www.phpgroupware.org/>),
- TUTOS (<http://www.tutos.org/> und <http://sourceforge.net/projects/tutos/>),
- open-EIS (Enterprise Information System, <http://www.open-eis.com/de/>).

und weitere. Einige kommerzielle Produkte zum Vergleich sind:

- IBM Lotus Notes,
- MS Exchange mit MS Outlook als Frontend,
- Novell GroupWise,
- Oracle Collaboration Suite,
- SAP Knowledge Management + Collaboration.

In Form von zwei Dutzend Debian-Paketen aus der Abteilung *Web Software* von *sarge* steht eGroupWare zur Verfügung, das wir uns näher ansehen wollen. Es ist in PHP geschrieben und zeichnet sich durch einen modularen Aufbau aus; man kann vorsichtig anfangen. Die *Daily Comics* sind weniger wichtig. eGroupWare braucht:

- nicht übersetzt (compiliert) zu werden, da es nur PHP, HTML und Grafiken enthält,
- eine Datenbank (MySQL, PostgreSQL oder andere),
- einen Webserver (Apache mit PHP),
- PHP 4.2 oder neuer,
- einen Mailserver (sendmail, exim, postfix).

Diese Voraussetzungen lassen sich mit jeder neueren Debian-Distribution erfüllen. Will man sofort alles haben, richtet man das Meta-Paket *egroupware* ein. Der häufigere Fall ist jedoch, dass eGroupWare in eine bestehende Struktur integriert werden soll. Da dies teilweise in die Gewohnheiten der Benutzer eingreift, verspricht ein schrittweises Vorgehen vermutlich mehr Erfolg als ein brutaler Wechsel. Folgende Pakete sind nach Bedarf einzurichten, grob geordnet nach deren Bedeutung:

- *egroupware-core*, Voraussetzung für alles Weitere,
- *egroupware-manual*, das Handbuch, nur benutzbar, wenn auf dem Rechner ein Apache-Webserver etc. laufen und *egroupware-core* eingerichtet ist, nicht allein stehend nur so zum Schnuppern geeignet. Dafür ist das *German User Manual* unter <http://www.egroupware.org/german-user-manual> richtig (ohne .html), trotz des englischen Namens deutschsprachig,
- *egroupware-addressbook*, Anschriftenverzeichnis, Entschuldigung, ein Kontakt-Manager, der auf eine SQL-Datenbank oder LDAP zurückgreift,
- *egroupware-bookmarks*, ein Bookmark-Manager,
- *egroupware-calendar*, Kalender mit der Fähigkeit, Verabredungen zu organisieren,
- *egroupware-forum*, web-basierte Tratschcke,
- *egroupware-messenger*, Nachrichtenaustausch zwischen den Teilnehmern,
- *egroupware-email*, *egroupware-emailadmin* und *egroupware-felamimail*, ein Email-System,
- *egroupware-ftp*, Dateiübertragung.
- *egroupware-filemanager*, noch ein Dateimanager,
- *egroupware-fudforum*, noch eine Tratschcke (<http://fudforum.org/>),
- *egroupware-headlines*, versorgt die Teilnehmer mit den neuesten Nachrichten aus dem Web und hält sie von der Arbeit ab,
- *egroupware-infolog*, verwaltet ToDos, Kundenkontakte, Telefonanrufe usw.
- *egroupware-jinn*, verteiltes Content-Management-System,
- *egroupware-ldap*, Werkzeuge zur Zusammenarbeit mit einem LDAP-Server,
- *egroupware-news-admin*, Werkzeuge zum Betrieb eines lokalen News-Dienstes,
- *egroupware-phpbrain*, ein Wissensbasis-System,
- *egroupware-polls*, ein Werkzeug für elektronische Umfragen und Abstimmungen, vorteilhaft in Verbindung mit einem Polling-Manager wie *edice* (<http://www.edice.cz/>) einzusetzen, der dem Benutzer lästige Entscheidungen abnimmt,
- *egroupware-projects*, eine Projektverwaltung,
- *egroupware-stocks*, eine Wertpapierverwaltung,
- *egroupware-tts*, Trouble-Ticket-System zur Unterstützung der Reparaturabteilung,
- *egroupware-wiki*, ein lokaler Wiki-Server,
- *egroupware-comic*, ein Comic-des-Tages-Server.

eGroupWare verfügt über eine Schnittstelle für anwendereigene Entwicklungen.

Evolution ist die Groupware Suite unter GNOME und stammt von Novell. Das gleichnamige Debian-Paket bringt das Wesentliche mit. Dazu gehören:

- Email (SMTP, POP, IMAP),
- Kalender (iCalendar),
- Aufgabenverwaltung,
- Kontakte (LDAP, vCard),
- Synchronisation von Datenträgern.

Die Unterstützung für das kommerzielle Produkt Novell GroupWise ist im Werden, die Zusammenarbeit mit MS Exchange soll bereits funktionieren. Vom Funktionsumfang her gleicht Evolution der KDE-Software *kontact*, vermehrt um die Synchronisation. Abbildung 11.11 zeigt den Kalender von Evolution; andere Bildschirme gehören zu Email, Kontakten oder Aufgaben. Nach Eingabe einiger persönlicher Daten (Zeitzone, Email-Adresse, Email-Server) ist das Werkzeug betriebsbereit. Die Benutzung sieht einfach aus. Unter dem Menüpunkt *Werkzeuge* -> *Einstellungen* -> *Verfügbarkeitsinformationen* kann man im lokalen oder öffentlichen Web täglich, wöchentlich oder nach Bedarf bekannt geben, wann man wo zu erreichen ist. Weitergehende Funktionen zur Förderung des Teamgeistes sind nicht offensichtlich.

phpGroupWare ist eine weitere bei Debian erhältliche Groupware, bestehend aus insgesamt 47 Paketen, also stark modularisiert und wie der Name vermuten lässt in PHP geschrieben. Der frühere Name lautete Webdistro. Die Groupware verwendet Web-Browser als Clients und setzt entsprechend auf der Servermaschine einen Webserver wie Apache voraus. Unter <http://de.TRYphpGroupWare.org/> ist ein deutscher Testserver zum Ausprobieren erreichbar.

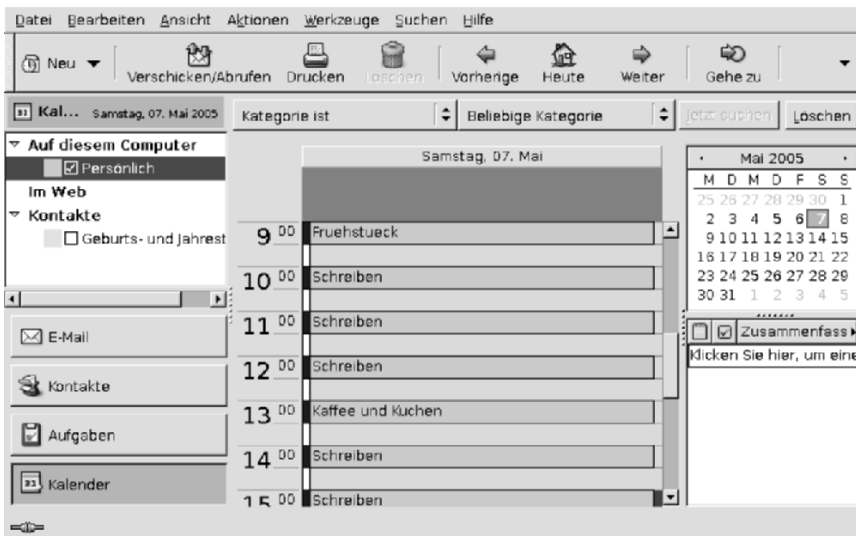


Abbildung 11.11. Screenshot der Groupware Evolution

Ebenfalls bei Debian erhältlich ist TUTOS, *The Ultimate Team Organization Software*. Das Werkzeug ist in einem einzigen Debian-Paket namens *tutos2* enthalten und wendet sich mehr an kleine Gruppen. Sein Funktionsumfang ist jedoch kaum geringer als bei den großen Groupware-Vertretern. TUTOS ist web-basiert und benötigt einen Webserver (Apache), eine Datenbank (PostgreSQL, MySQL oder andere) und PHP. Es unterstützt mehrere Sprachen, darunter Deutsch.

Nicht bei Debian, sondern nur als Tarball auf den jeweiligen Servern sind Kolab, SuSE Linux Openexchange Server, OpenGroupware und open-EIS zu haben. Kolab – aktuell Version 2 – ist aus einem Pilotprojekt einer deutschen Bundesbehörde entstanden und wird heute von einem Konsortium gepflegt, das ein herzliches Verhältnis zum KDE-Projekt erkennen lässt, ohne dazu zu gehören. Die Stärke von Kolab liegt darin, dass als Clients Web-Browser, KDE Kontact und MS Outlook gleichermaßen unterstützt werden. Es gibt zwar kein Debian-Paket, aber Hinweise für die Einrichtung unter *sarge*. Kolab arbeitet mit OpenLDAP, Postfix, Cyrus IMAP, Apache, ProFTPD und SASL zusammen.

Novells SuSE Linux Openexchange Server wendet sich an kleine und mittlere Unternehmen und bedient Web-Browser und MS Outlook als Clients. Große Unternehmen werden auf GroupWise verwiesen. Zusammen mit Evolution hat Novell drei Groupware-Pakete im Stall, was vermutlich kein Zustand auf Dauer sein wird.

OpenGroupware ist in Objective C geschrieben und verwendet Web-Browser als Clients – bei Bedarf auch MS Outlook – Apache als Webserver und PostgreSQL als Datenbank. Es verfügt über folgende Funktionen:

- Gruppenterminplanung,
- Betriebsmittelpflege (Räume, Geräte, Fahrzeuge ...),
- Adressbuch, Kontaktmanagement,
- Dokumentenmanagement,
- Aufgabenverwaltung (job management), Reiseabrechnung,
- Webmailer,
- Sprachunterstützung für Deutsch und Englisch, erweiterbar.

Die kommerzielle Ausgabe nennt sich Skyrix und enthält einige Werkzeuge mehr, darunter einen Load Balancer. Eine Zusammenarbeit mit OpenOffice.org wird angestrebt.

open-EIS will mehr sein als nur Groupware und bietet Funktionen von *Collaboration Platform* über *Groupware* bis *Zeit- und Kostenstellenverwaltung* an. Mit open-EIS können alle nötigen IT-Anwendungen geschaffen werden, die Sie für Ihren Unternehmenserfolg benötigen, schreibt der Hersteller.

Es ist schwierig bis unmöglich, eine klare Entscheidung für eine bestimmte Groupware zu treffen. Die Grundfunktionen beherrschen alle. Was darüber hinaus benötigt wird, hängt von der jeweiligen Situation ab. Diese Softwaregruppe ist zudem in lebhafter Entwicklung, sodass eine Entscheidung jährlich überprüft werden müsste.

11.10 Daten synchronisieren (unison, ksync, kitchensync, multisync, kandy)

Arbeitet ein Benutzer an mehr als einem Ort, möchte er vermutlich seine Daten an jedem Ort vorfinden. Eine Möglichkeit ist, die Daten in einen Laptop oder Personal

Digital Assistant (PDA) zu packen und durch die Gegend zu schleppen³ Oft ist ein mobiler Benutzer an einen Stammsitz mit ortsfestem Rechner gebunden, sodass sich das Problem ergibt, die Daten auf mobilem und ortsfestem Rechner abzugleichen (zu synchronisieren). Selbst in einem rein ortsfesten Netz mit Clients und Servern kann dieses Problem auftauchen. Welche Daten sind da authentisch, die auf dem Server oder die auf dem Client? Der Verwalter muss eine verbindliche Vorgehensweise festlegen. Es gibt – auch bei Debian – Werkzeuge, welche die Arbeit erleichtern. Einige sind Teil eines Personal Information Managers (PIM),

Das Werkzeug `unison` synchronisiert Verzeichnisse und Dateien, die auf verschiedenen Partitionen (sinnvollerweise auf verschiedenen Platten) oder auf verschiedenen Rechnern liegen, wobei Linux/UNIX und andere Betriebssysteme unterstützt werden. Es können also ein MS-Windows-Laptop und ein Linux/UNIX-Server miteinander in Einklang gebracht werden. Dabei wird in beiden Richtungen synchronisiert; es ist nicht erforderlich, den einen Rechner als Original und den anderen als Kopie anzusehen. Das kann zu Konflikten führen, wie wir sie von Versionskontrollsystemen her kennen, siehe Abschnitt 5.15.3 *Versionsverwaltung* auf Seite 243. Im einfachsten Fall ruft ein Benutzer das Werkzeug aus der Kommandozeile mit den beiden zu synchronisierenden Verzeichnissen als Argument auf, wobei die Reihenfolge keine Rolle spielt:

```
joe@debian:~$ unison verzeichnis1 verzeichnis2
```

Beim ersten Aufruf von `unison` für ein Verzeichnis werden identische Dateien nicht weiter beachtet, Dateien, die nur in einem Verzeichnis vorkommen, in das jeweils andere kopiert und verschiedene Dateien gleichen Namens als Konflikt behandelt und dem Benutzer die Entscheidung überlassen, welche Datei gelten soll. Obwohl die Benutzung von `unison` einfach und in der begleitenden Dokumentation gut erklärt ist, empfiehlt es sich, zuerst mit einem eigens zu diesem Zweck angelegten temporären Verzeichnis zu üben. Wer unbedingt eine grafische Oberfläche braucht, richtet das Debian-Paket `unison-gtk` zusätzlich ein. Das Manuskript zu dem vorliegenden Buch wurde teilweise mittels `unison` zwischen dem Arbeitsplatz und einem Backup-Server synchronisiert, zusätzlich zu CVS, das noch einige andere Aufgaben als nur ein Backup erledigt.

Die Werkzeuge `ksync` und `kitchensync` gehören beide zu dem KDE Personal Information Manager `kdepim`, sind aber auch einzeln benutzbar. `ksync` dient zum Abgleichen von Terminen, Lesezeichen (bookmark) und Anschriftenlisten. `kitchensync` soll das universelle KDE-Werkzeug zum Synchronisieren beliebiger Daten werden. Das entsprechende GNOME-Werkzeug `multisync` dient in erster Linie dem Abgleich von Terminen etc. zwischen Rechnern, PDAs und Mobiltelefonen, gehört also auch zu den PIM-Werkzeugen.

KDE `kandy` ist ein Werkzeug, um vom Rechner auf die in einem Mobiltelefon abgelegten Daten zuzugreifen, und ebenfalls Teil des PIM-Werkzeugkastens. Informationen liefert der Web-Browser und Datei-Manager `konqueror`, wenn man als

³Ich halte es für mittelalterlich, Hardware herumzutragen. Wo ich arbeite, hat die Hardware da zu sein, die Daten liefert das Netz. Ich habe ja auch nicht meinen eigenen Stromerzeuger im Gepäck. Mobiltelefone sind ein anderer Fall.

URL `help://kandy` eingibt, vorausgesetzt, das gleichnamige Debian-Paket ist eingerichtet.

11.11 Mindmapper und Jotter (FreeMind, gjots2, kjots, think, kdissert)

Eine Mindmap ist eine grafische Darstellung gedanklicher Zusammenhänge, im einfachsten Fall entworfen mittels Bleistift und Papier. Gegenüber einer linearen oder sequentiellen Aneinanderreihung von Gedanken – wie sie das Inhaltsverzeichnis des Buches darstellt – hat eine Mindmap den Vorteil zweier Dimensionen, ähnlich wie eine Tabellenkalkulation. Es lassen sich auch Zusammenhänge darstellen, die nicht als Baum strukturiert sind (Schleifen). Man gelangt schließlich zu semantischen Netzen oder allgemeinen Graphen mit Kreuz- und Querverbindungen. Die Elemente einer Mindmap können locker definiert sein, aber auch einer strengen Syntax genügen. Einfache Ausführungen, die nur zum geordneten Festhalten von Gedanken und Notizen dienen, werden auch als Jotter (Notizbuch, Zettelkasten) bezeichnet. Der Übergang zu Notizbüchern wie den TuxCards (Abschnitt 11.9.2 *Notizbuch, Personal Information Manager* auf Seite 386) oder Outline-Prozessoren wie VimOutliner (Abschnitt 5.3.11 *Outline Prozessor* auf Seite 187) ist fließend. Mindmaps werden zum Gewinnen, Sammeln und Strukturieren von Ideen eingesetzt, sowohl vom einzelnen Benutzer wie innerhalb einer Gruppe. In letzterem Fall dienen sie auch der Kommunikation. Näheres: Wikipedia.

Seit *sarge* stehen fünf Mindmapper oder Jotter zur Verfügung:

- freemind aus der *contrib*-Abteilung, zu Hause bei <http://freemind.sourceforge.net/>, setzt Java voraus,
- gjots2 aus dem GNOME-Projekt, zu Hause bei <http://bhepple.freeshell.org/gjots/>,
- kjots aus dem KDE-Projekt, Info bei <http://docs.kde.org/>,
- gnome-think aus dem GNOME-Projekt,
- kdissert aus dem KDE-Projekt, zu Hause bei <http://freehackers.org/~tnagy/kdissert/>.

Die beiden *jots* gehen zurück auf ein Werkzeug aus einem Paket (nicht bei Debian) namens *TkGoodStuff*, das für X11 unter Verwendung von Tcl/Tk einige nette Kleinigkeiten anbietet, Information dazu aus der *Linux Gazette* bei <http://www.ub.uni-stuttgart.de/lg/issue11/tkg.html>, Code bei <http://www.x.org/contrib/misdirected/>.

Der Mindmapper FreeMind braucht Java. Deshalb scheiterte die Einrichtung des gleichnamigen Debian-Paketes zunächst, die Abhängigkeiten waren nicht zu erfüllen. Erfolg hatte folgendes Vorgehen:

- Herunterladen und Einrichten der Java2 Platform Standard Edition für Linux von <http://java.sun.com/> als Self Extracting Binary, Umfang 16 MB,

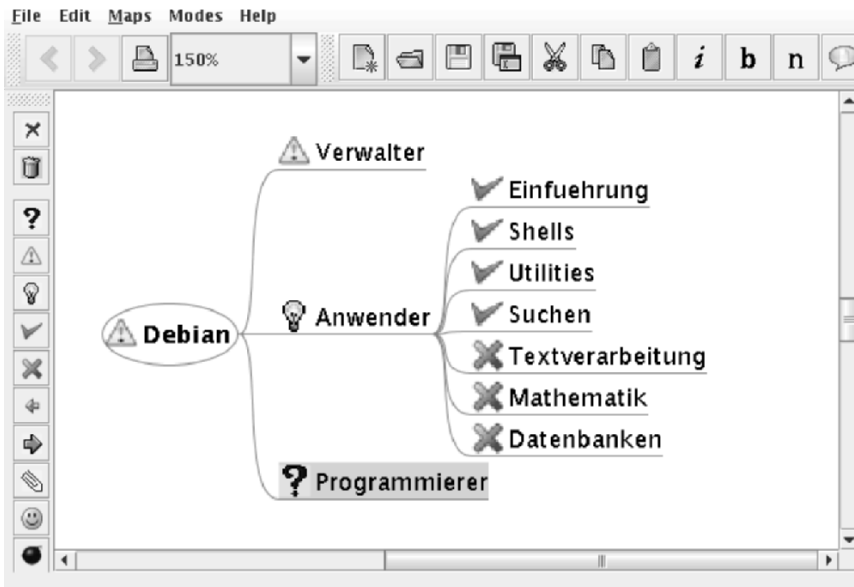


Abbildung 11.12. Screenshot des Mindmappers FreeMind

- Herunterladen und Einrichten von FreeMind (freemind.zip) von <http://freemind.sourceforge.net/>, Umfang 400 KB, alles in /usr/local/,
- Setzen der Zugriffsrechte (755),
- im Shellskript freemind.sh Ersetzen von java durch den absoluten Pfad /usr/local/jre1.5.0/_03/bin/java.

Danach kann ein gewöhnlicher Benutzer mit dem Aufruf:

```
joe@debian:~$ /usr/local/freemind.sh
```

den Mindmapper starten und benutzen. Blicke noch, den Aufruf in ein kleines Shellskript in /usr/local/bin/ zu verpacken oder die Pfade geschickter zu wählen, damit das Kommando freemind im Bereich der PATH-Variablen liegt. Abbildung 11.12 vermittelt einen Eindruck. Wir sehen einen Baum, Wurzel links, der die Struktur eines Buches über Debian andeutet. Die einzelnen Themen lassen sich durch hübsche Icons markieren, die ihre Wichtigkeit oder ihren Zustand markieren. Den Themen können Links beispielsweise auf Dateien zugeordnet werden, sodass man ein ganzes Projekt hinter der Mindmap unterbringen kann. Schließlich kann man die Mindmap nach HTML exportieren und auf einem Webserver den Mitarbeitern des Projektes vorlegen. Inwieweit eine Mindmap die Arbeit beschleunigt oder das Ergebnis verbessert, wäre zu untersuchen. Meinen Bleistift behalte ich vorläufig noch.

Der GNOME-Jotter gjots2 gibt sich etwas bescheidener als FreeMind. Er gestattet, Schlagworte und Notizen in einer Baumstruktur anzuordnen und kennt

mehrere Textformate sowie Verschlüsselung. Er versteht sich mit seinem ähnlich auftretenden Kollegen *kjots* vom KDE-Projekt, siehe Abbildung 11.13. Links im Fenster ist die Gesamtstruktur zu sehen, rechts der Inhalt der jeweils aktiven Seite. Das Werkzeug *gnome-think* (Symlink *think* anlegen) bezeichnet sich als Outliner und Organizer, ist den vorstehenden Jottern sehr ähnlich und könnte noch ein bisschen Entwicklung vertragen, um seinem Namen gerecht zu werden. Infolge seines schlichten Gemüts ist es ohne Hilfe benutzbar.

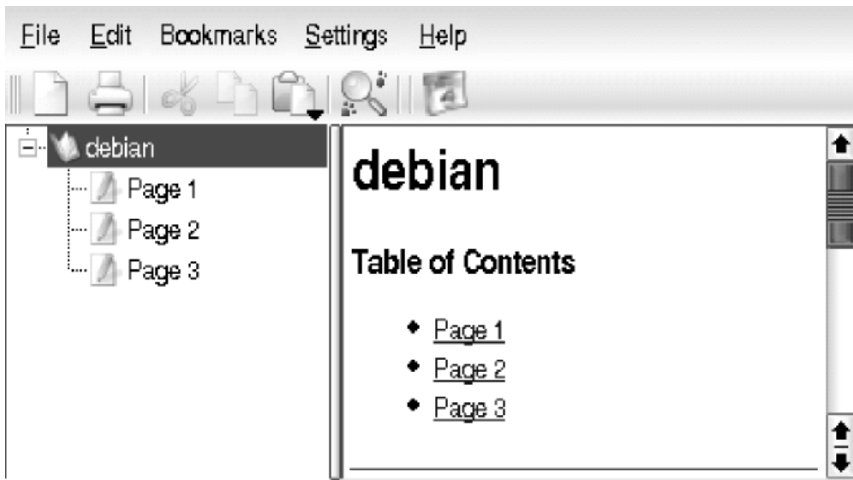


Abbildung 11.13. Screenshot des KDE-Jotters *kjots*

Das Werkzeug *kdissert* unterstützt die Erstellung von Mindmaps mit Texten, Bildern, Kommentaren und Links sowie die Erzeugung von Dokumenten wie Berichten, Präsentationen oder Webseiten aus den Mindmaps. Es arbeitet mit \LaTeX und OpenOffice.org zusammen und hinterlässt einen vielseitigeren Eindruck als die vorstehend genannten Jotter. Eine über die knappe Manualseite hinausgehende Information bekommt man über das KDE Help Center.

11.12 Content Management (OpenACS, Mambo, Zope, Plone, Typo3)

Content Management System (CMS, F: système de gestion de contenu) hört sich wichtig an, man ahnt wohl auch, dass damit irgendwo Ordnung gehalten werden soll, aber richtig klar wird einem Benutzer der Begriff erst nach längerer Praxis. Im Grunde geht es immer darum, Dokumente zu erzeugen, zu verwalten und meist, aber nicht nur per Web-Browser einem Benutzerkreis zugänglich zu machen. Handelt es sich bei den Dokumenten um Webseiten, spricht man von einem Web Content

Management System (WCMS), bleiben die Dokumente aller Art innerhalb einer Firma, wird das Biest als Enterprise Content Management System (ECMS) bezeichnet. Kleinere, auf Printmedien spezialisierte Content Management Systeme werden auch als Redaktionssysteme bezeichnet, aber der Übergang ist fließend. Für den Anfang ist ein Blick in die deutsche Wikipedia zu empfehlen, Begriffe *Content Management*, *Redaktionssystem*, *Application Server* und *WebDAV* samt Folgelinks. Das Thema berührt auch Versionskontrollsysteme wie CVS, siehe Abschnitt 5.15.3 *Versionsverwaltung* auf Seite 243, und Datenbanken, siehe Abschnitt 7 *Datenbanken und andere Wissensspeicher* auf Seite 273. Traditionell werden die Aufgaben mit einer Kombination aus Linux, Apache, MySQL und Perl/PHP/Python (LAMP) bewältigt, sowohl im Internet als auch in geschlossenen Netzen. Hier wollen wir uns kurz mit folgenden Werkzeugen befassen:

- OpenACS (Open Architecture Community System, OACS), ein Toolkit oder Rahmen (framework) zum Bau von Webanwendungen, nicht bei Debian, aber als Tarball unter der GPL bei <http://openacs.org/> zu haben. Bei Debian liegt nur eine einsame Dokumentation im Paket *openacs-doc*.
- Mambo, ein verbreitetes, einfaches Content Management System, <http://www.mamboserver.com/>,
- Z Object Publishing Environment (Zope), eine in Python geschriebene, objektorientierte Open Source Entwicklungsumgebung für dynamische, datenbankgestützte Internetanwendungen (Webanwendungsserver), unter Debian in einem gleichnamigen Paket sowie rund 100 ergänzenden Paketen verfügbar, zu Hause unter <http://www.zope.org/> oder <http://www.zope.de/>,
- Plone (benannt nach einer Band, ein Kunstwort ohne Bedeutung), ein in Python geschriebenes, auf Zope aufbauendes Content Management System, unter Debian in einem gleichnamigen Metapaket verfügbar und unter <http://plone.org/> sowie <http://sourceforge.net/projects/plone/> zu Hause,
- Nuxeo CPS, ein auf Zope aufbauendes Content Management System, als Debian-Paket bei <http://www.cps-project.org/> verfügbar,
- ZMS, bei Debian im Paket *zope-zms* enthalten, zu Hause auf <http://www.zms-publishing.com/>,
- ERP5, ein für Linux auf <http://www.erp5.org/> verfügbares, auf Zope aufbauendes Content Management System,
- TYPO3, ein auf PHP und MySQL aufbauendes Content Management System, das nicht bei Debian, aber unter der GPL bei <http://www.typo3.de/> zu haben ist,
- PostNuke, <http://www.postnuke.com/>.

Alle Werkzeuge mit Ausnahme von OpenACS, CPS, ZMS und ERP5 haben eigene Einträge in der deutschen Wikipedia. OpenACS wird in der englischen Wikipedia erläutert. Eine wichtige Anwendung ist die Erzeugung und Verwaltung von Webseiten auf großen Webservern, aber auch Dokumente beliebiger Art lassen sich mit einem Content Management System beispielsweise in einem Firmennetz verwalten. Für kleine Projekte oder Server lohnt sich der Aufwand nicht, die lassen sich mit klassischen Linux/UNIX-Mitteln beherrschen. Alle Content Management Systeme

trennen Inhalt, Struktur und Aufmachung (Design) deutlich voneinander. Wer WY-SIWYG gewohnt ist, muss umdenken. Wünschenswert ist die Trennung von Live-System (Webserver für Veröffentlichung) und Redaktionssystem (nur für den internen Gebrauch), da sich die Anforderungen stark unterscheiden.

OpenACS hat eine längere, wechselvolle Entwicklung hinter sich und erweckt einen ausgereiften Eindruck. Als Datenbank verwendet es primär PostgreSQL, als Skriptsprache Tcl. Eine größere Anwendung von OACS ist .LRN (mit Punkt als erstem Zeichen, <http://dotlrn.org/>), eine *Enterprise Elearning Platform*, entstanden am renommierten MIT. Unter einem Portal werden Kurs- und/oder Klassenverwaltung, Content Management, Kommunikation, Terminkalender und die Verwaltung von Hausaufgaben, Klausuren, Prüfungen etc. zusammengefasst. Die Universität Heidelberg organisiert einen Teil ihres Lehrbetriebs damit, aber die Software eignet sich für Schulen aller Art (K12 = Kindergarten bis 12. Klasse in den USA).

Mambo (Mambo Open Source, MOS) ist ein relativ einfaches Content Management System, das nicht bei Debian, aber unter der GPL als Tarball von dem vorstehend genannten Server zu haben ist. Es verwendet MySQL als Datenbank und PHP als Skriptsprache. Im deutschsprachigen Raum ist Mambo verbreitet, Anpassungen finden sich auf <http://mamboforge.net/projects/german/> und <http://www.mamboportal.de/>.



Abbildung 11.14. Screenshot des Content Management Systems Plone

Die Hauptanwendung von Zope ist der Bau von Werkzeugen für die Gestaltung von Webinhalten, insbesondere der Zugriff auf nahezu beliebige Programme

per Web-Browser. Dazu bringt die Software einen eigenen Webserver mit, kann aber auch unter einem Webserver wie Apache arbeiten. Ebenso verfügt Zope über eine eigene Objektdatenbank, lässt sich aber auch mit Datenbanken wie Oracle, MySQL oder PostgreSQL (siehe Abschnitt 7 *Datenbanken und andere Wissensspeicher* auf Seite 273) verbinden. Dank der hohen Modularität und der Erweiterbarkeit durch Python-Skripte lässt Zope keine Wünsche offen, erfordert jedoch Einarbeitung. Ein großer Teil der Funktionen von Zope lässt sich auch durch einzelne Werkzeuge erreichen, aber nicht mit einem einheitlichen Look and Feel und auch nicht mit einem durchgängigen Sicherheitskonzept. Wenn an den Inhalten auf einem Webserver mehrere Personen mitarbeiten, ein Teil der Daten dynamisch aus verschiedenen Quellen erzeugt wird und abgestufte Zugriffsrechte verlangt werden, ist es Zeit, sich Zope anzusehen. Auf der deutschen Website von Zope ist eine ausführliche, deutschsprachige Einführung verfügbar.

Auf Zope baut Plone auf, ein Content Management System, welches das Web benutzt, aber nicht die Erzeugung von Webseiten als Hauptziel hat, sondern ganz allgemein Dokumente oder Informationen (document publishing system). In Abbildung 11.14 ist beispielsweise eine Webseite eines mit Plone organisierten Buchprojektes zu sehen, an dem viele Autoren arbeiten. Plone stellt dabei mehr Kommunikationskanäle für die Autoren bereit als eine Versionskontrolle wie CVS. Der Häuptling des Projektes hat einige Arbeit mit den Vorbereitungen, aber dann können die Indianer mittels Web-Browser im Projekt spazieren gehen, Dateien veröffentlichen und miteinander plaudern. CPS, ZMS und ERP5 sind freie Alternativen zu Plone, die ebenfalls Zope zur Grundlage haben.. Einige kommerzielle CMS bauen ebenfalls auf Zope auf.

TYPO3 ist ein weiteres Content Management System und verfügt über eine breite Installationsbasis im Web, das seine wesentliche Anwendung darstellt. Es verwendet PHP als Programmiersprache, MySQL als Datenbank und ImageMagick als Grafikwerkzeug. Über eine Datenbank-Abstraktionsschicht lassen sich auch andere gängige Datenbanken ansprechen. Das Werkzeug beinhaltet eine ausgefeilte Rechteverwaltung und ein Projektmanagement, ist sehr mächtig und erfordert eine gewisse Einarbeitung und Pflege.

Nicht bei Debian, aber als Tarball unter der GPL ist PostNuke bei <http://www.postnuke.com/> erhältlich. Die in PHP geschriebene Software bezeichnet sich als *Community, Content, and Collaborative Management System* mit Hauptgewicht auf der Gestaltung von Websites. Es ist schwierig zu entscheiden, welches der Werkzeuge in einem konkreten Fall am besten geeignet ist, zumal ein Jahr später die Entscheidung anders ausfallen könnte. Die Programmiersprache – Python oder PHP – bietet sich als ein Kriterium an.

11.13 Sprachtrainer (klatin, kvoctrain)

Zum Kommunizieren gehört Sprache, im globalen Internet eine Vielzahl von Sprachen. Wenn es auch nur wenigen von uns gegeben ist, fünfzehn Sprachen – darunter Griechisch, Latein, Russisch und Arabisch – zu beherrschen wie weiland HEINRICH

SCHLIEMANN aus Mecklenburg, der Entdecker der Ruinen Trojas, so kann man als Netizen doch versuchen, nach und nach die Anfänge einiger Sprachen zu lernen. Mit 200 Wörtern und Redewendungen kommt man als Tourist schon weit, 1000 Wörter sind ein erreichbares Ziel, in seiner Muttersprache kennt ein Durchschnittseuropäer etwa 5000 Wörter, JOHANN WOLFGANG VON GOETHE soll 20.000 verwendet haben, WILLIAM SHAKESPEARE 50.000. Eine größere Enzyklopädie umfasst rund 250.000 Einträge, die deutsche Wikipedia hat diesen Wert überschritten, während die vorliegenden Zeilen aus der Feder strömen.

Beim ungeliebten Vokabellernen kann ein Rechner helfen, im einfachsten Fall dadurch, dass er sie immer wieder abfragt und eine Statistik darüber führt, welche Vokabeln schon beherrscht werden und welche nicht. Das obere Ende bilden multimediale Vokabeltrainer, die Aussprache, Grammatik und Situationen zu den Wörtern und Redewendungen (Phrasen) mitliefern. In einem guten Sprachtrainer steckt viel Arbeit. Die Grundlage könnte eine Datenbank liefern.

Das Debian-Paket *klatin* aus dem KDE-Projekt richtet einen Trainer für Latein ein, der X11 und Qt voraussetzt und drei Modi kennt: Vokabeln, Verbformen und Grammatik. Das Programm stellt seine Fragen auf Englisch, sodass ein deutscher Schüler gleich zwei Fremdsprachen auf einmal lernt. Für die Vokabeln wird ein Multiple-Choice-Verfahren mit vier Möglichkeiten angewendet, nicht unbedingt realitätsnah. Bei den Verbformen wird nach bestimmten Formen (3. Person Plural Präsens Indikativ Aktiv) eines gegebenen Verbs gefragt, die man eintippt. Der Grammatikmodus fragt nach dem Vokativ Plural von *puella* und ähnlichen Scherzen. Nach meinem Eindruck eignet sich der Trainer gut zum Nachprüfen der Kenntnisse, weniger zum Lernen. Non est ad astra mollis e terris via.

Einen nicht auf eine bestimmte Sprache festgelegten Vokabeltrainer ohne Grammatikteil nach der Art von Karteikärtchen (Lernkartei, flashcards) enthält das Debian-Paket *kvoctrain*, ebenfalls aus dem KDE-Projekt. Der Benutzer muss den Wortschatz selbst anlegen, wofür sich in einer Schule Gruppenarbeit (E: teamwork, F: travail d'équipe) anbietet. Wenn man sich mit der Beschränkung auf Vokabeln zufrieden gibt, stellt das Werkzeug einen brauchbaren Rahmen dar.

Als Tarball bei <http://linux.handorf-langenberg.de/> ist der Vokabeltrainer KVocabulary zu haben, der unter KDE läuft. Die elektronische Lernkartei *Pauker* ist in Java geschrieben, steht unter der GPL, läuft auch unter Linux und ist auf <http://pauker.sourceforge.net/> beheimatet. Sie eignet sich wie jede Lernkartei auch zum Lernen von Geschichtszahlen oder sonstigen Fakten. Pauker bringt vorgefertigte Lektionen für Englisch, Spanisch, Geografie und Mathematik mit, die man als Muster für eigene Karteikästen verwenden kann. Ansonsten sieht es im Netz dünn mit Sprach- oder Vokabeltrainern für Linux aus. Viele Vokabeltrainer halten es nicht einmal für nötig darauf hinzuweisen, dass sie nur unter einem anderen Betriebssystem laufen.

Office-Pakete

Das Kapitel bringt eine Übersicht verschiedener Office-Pakete, die für Debian GNU/Linux verfügbar sind. Einzelne Komponenten haben wir bereits in den vorangegangenen Kapiteln kennen gelernt.

12.1 Übersicht

Ein Office-Paket (F: suite bureaucratique) ist ein Bündel der wichtigsten im Bürobetrieb gebrauchten Werkzeuge. Drei Komponenten gehören unbestritten dazu:

- eine WYSIWYG-Textverarbeitung (Textprozessor),
- eine Tabellenkalkulation,
- eine kleine Datenbank oder zumindest ein Frontend für eine Datenbank.

Die vierte Komponente war früher ein Programm zur Datenübertragung über Modemstrecken und ist heute ein Werkzeug zum Erstellen von Präsentationen. Was darüber hinausgeht, ist beliebig. Wer nur ein Werkzeug zum Schreiben braucht, ist besser mit einem einzelnen Texteditor oder Textprozessor bedient – siehe Abschnitt 5 *Textverarbeitung* auf Seite 155 – ein Office-Paket wäre glatter Overkill und würde unnötig Ressourcen (Speicher, Prozessorzeit) belegen. Auch gehen die Fähigkeiten der Textkomponenten in den Office-Paketen weit über das hinaus, was ein durchschnittlicher Anwender braucht, mit entsprechendem Lernbedarf. Wer berufsmäßig schreibt, kann vielleicht eine Office-Textverarbeitung ausreizen.

Office-Pakete sollen gegenüber Einzelwerkzeugen den Vorteil haben, dass die Zusammenarbeit, der Datenaustausch zwischen den eigenen Komponenten reibungslos abläuft. Ein anderer Weg zu diesem Ziel führt über offene Datenformate. Ferner braucht sich der Benutzer nicht den Kopf über der Frage zu zerbrechen, welches Werkzeug für seine Arbeit das beste ist. Er erwirbt ein Office-Paket und damit basta. Marktführer in diesem Softwarebereich ist MS Office (<http://www.microsoft.com/office> oder <http://office.microsoft.com/>), das 1989 für den Macintosh und 1991 für PCs unter MS Windows herauskam.

Tabelle 12.1. Übersicht über die unter Debian GNU/Linux verfügbaren Office-Pakete im Vergleich zu MS Office

Aufgabe	GNOME	KDE	OpenOffice.org	Microsoft
Textverarbeitung	AbiWord	KWord	Writer	Word
Tabellenkalkulation	Gnumeric	KSpread	Calc	Excel
Datenbank	GNOME-DB	Kexi (Frontend)	Database User Tools	Access
Präsentation		KPresenter	Impress	PowerPoint
Bildverarbeitung		Krita	Draw	
Zeichnen		Karbon14	Draw	
Diagramme		Kivio, KChart	Draw	Visio
Formeleditor		KFormula		
HTML-Editor				FrontPage
Geschäftsberichte		Kugar		Publisher
Notizbuch				OneNote
Terminkalender				Outlook/Entourage
Email				Outlook/Entourage
Projektmanagement		KPlato		

Beim Lesen vorstehender Tabelle ist zu beachten, dass unter GNOME oder KDE weitere Werkzeuge für die genannten Aufgaben bereit stehen; sie sind nur nicht Bestandteil des jeweiligen Office-Paketes. Unter GNOME steht beispielsweise die leistungsfähige und vielbenutzte Bildbearbeitungssoftware GIMP zur Verfügung, ist aber nicht in GNOME Office eingebunden und wird das vielleicht nie. Da GIMP viele Datenformate beherrscht, ist seine Zusammenarbeit mit anderen Werkzeugen kein Problem. Auch Email lässt sich selbstverständlich ohne ein Office-Werkzeug senden und empfangen. MS Office 2003 enthält weitere, hier nicht aufgeführte Bestandteile.

Wenn man die drei für Debian verfügbaren Office-Pakete miteinander vergleicht, kann man den Eindruck gewinnen, dass OpenOffice.org am einheitlichsten ist, KOffice am vielfältigsten und das GNOME-Projekt von dem Office-Konzept nicht sehr begeistert ist, sondern mehr auf miteinander verträgliche Einzelwerkzeuge setzt.

12.2 OpenOffice.org

OpenOffice.org geht auf StarOffice zurück, eine Entwicklung der Firma Star Division Software, Hamburg, aus den achtziger Jahren, die unter MS Windows 3.* lief und später von Sun Microsystems übernommen wurde (<http://www.openoffice.org/> bzw. <http://www.sun.com/staroffice/>). Aus markenrechtlichen Gründen muss das Projekt auf den Anhang .org Wert legen. Es umfasst zur Zeit folgende Werkzeuge:

- Writer, eine Textverarbeitung,
- Calc, eine Tabellenkalkulation,
- Draw, ein Werkzeug zum Erstellen von Vektorgrafiken oder Diagrammen,
- Impress, eine Präsentationssoftware.

Die Werkzeuge gibt es nicht einzeln, sondern nur als komplettes Office-Paket, weil die Namen der einzelnen Werkzeuge nur Symlinks auf `openoffice` sind, das selbst ein Shellskript (Wrapper) zum Aufrufen von `/usr/lib/openoffice/program/soffice` ist, das wiederum ein Shellskript ist, welches schließlich das Binary `soffice.bin` startet. Die eigentliche Arbeit wird von einigen Dutzend Bibliotheken geleistet. Ab Debian GNU/Linux 3.1 (*sarge*) ist OpenOffice.org als Debian-Paket *openoffice.org*, Abteilung *Editors*, verfügbar. Alternativ kann die Software als Tarball von 80 MB Inhalt bei <http://www.openoffice.org/> oder einem Spiegel heruntergeladen werden. Die Einrichtung des Tarballs auf einem PC mittleren Alters unter *sarge* verlief problemlos und brauchte nur ihre Zeit. Man packt das Archiv in einem beliebigen Verzeichnis aus, geht in das dabei entstandene Unterverzeichnis `OO...` und ruft als root `./install` auf (kein `configure`, kein `make`). Die ganze Software liegt anschließend in `/usr/local/OpenOffice.org1.1.4`. In dessen Unterverzeichnis `program` finden sich die Startscripte für die Komponenten. Beim ersten Aufruf hat man dann noch ein paar intime Fragen zu beantworten, den Lizenzvertrag zu lesen und zu akzeptieren, dann geht es los. Die Version 2.0 befindet sich zur Zeit im Betastadium und ist als Tarball beim Hersteller

sowie bei Debian im Release *experimental* zu beziehen. Die hinzu gekommenen Fähigkeiten sind bei <http://de.openoffice.org/> nachzulesen; es sind viele Kleinigkeiten.



Abbildung 12.1. Screenshot des Begrüßungsbildschirms (Splashscreen) von OpenOffice.org

Abbildung 12.1 zeigt den Begrüßungsbildschirm (Aufmacher, E: splashscreen, splash page F: page de garde) des Office-Paketes. Es ist gleich, welches Werkzeug man aufruft. Wird viel mit dem Paket gearbeitet, ist ein Symlink `oo` auf `openoffice` zweckmäßig, der Name ist noch frei. Insgesamt macht OpenOffice.org einen ausgereiften Eindruck und soll sich gut mit anderen Office-Paketen vertragen.

StarOffice ist nach wie vor als kommerzielles, kostenpflichtiges Produkt von Sun Microsystems zu haben, auch für Linux. Von Einrichtungen der Lehre werden keine Lizenzgebühren verlangt, die Version Small Office/Home Office ist bezahlbar, die Enterprise Editions kosten etwas mehr, nach Anzahl der Benutzer gestaffelt. Von StarOffice 8 ist eine kostenfreie, auf 90 Tage begrenzte Evaluation Version für den Sommer 2005 angekündigt. Die Einrichtung der Evaluation Version von StarOffice 7 auf einem PC unter *sarge* verlief ohne Probleme:

- Herunterladen von <http://www.sun.com/>, ohne die Dokumentation rund 125 MB,
- Zugriffsrechte der `bin`-Datei auf 700 setzen,
- die `bin`-Datei als Kommando aufrufen, dabei als Verzeichnis `/usr/local/staroffice7/` angeben,

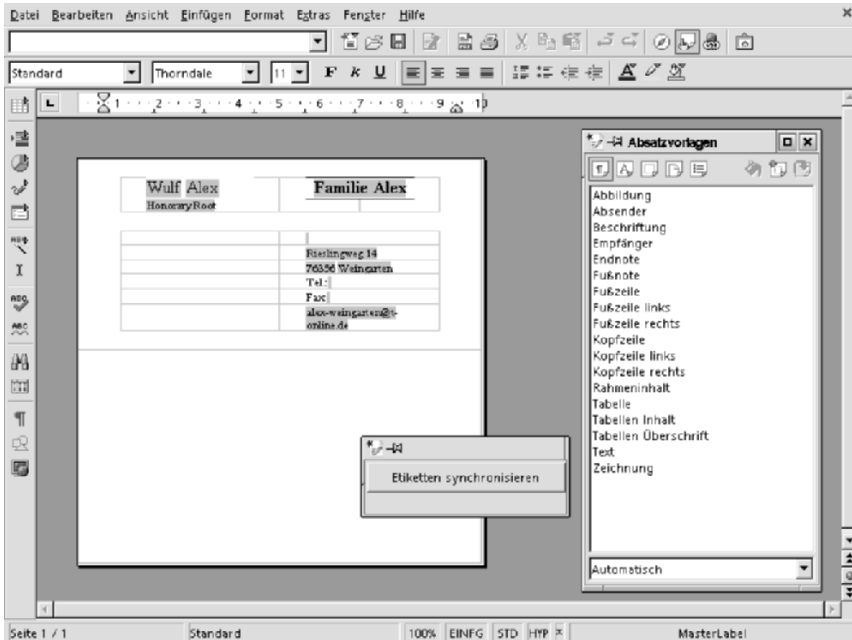


Abbildung 12.2. Screenshot der Testversion von StarOffice 7

- Java war zuvor unter `/usr/local/java/` eingerichtet worden und wurde gefunden, siehe Abschnitt 11.11 *Mindmapper und Jotter* auf Seite 398,
- die Datenbank Adabas wurde nicht gefunden und war auch nicht erforderlich,
- die Lizenz war zu akzeptieren, einige Fragen sollten beantwortet werden.

Dann konnte StarOffice von einem gewöhnlichen Benutzer aufgerufen werden:

```
joe@debian:~$ /usr/local/staroffice7/program/soffice
```

Abbildung 12.2 zeigt das Programm beim Entwerfen einer Visitenkarte; die Ähnlichkeit mit OpenOffice.org überrascht nicht. Nach dem erfolgreichen Test wurde StarOffice wieder vom Rechner entfernt, da ein Parallelbetrieb zu OpenOffice.org unsinnig wäre.

Vergleiche in der Vergangenheit haben einen leichten Vorsprung von StarOffice vor OpenOffice.org in der Behandlung (Import) von MS-Office-Dateien gezeigt. Wer auf einen intensiven Datenaustausch mit der MS-Office-Welt angewiesen ist, wird mit StarOffice vielleicht glücklicher als mit OpenOffice.org. Wer dagegen seine Büros einheitlich ausrüsten kann und nur gelegentlich gezwungen ist, ein MS-Office-Dokument zu öffnen, kommt mit OpenOffice.org gut aus. Mit der Version 2 von OpenOffice.org dürfte der Unterschied geringer werden.

Für StarOffice und OpenOffice.org lassen sich eigene Makros (kleine Programme) schreiben, um spezielle Wünsche zu erfüllen. Dafür steht ein Basic-Dialekt sowie voraussichtlich mit den neuen Versionen auch Java zur Verfügung. Ein *Program-*

mer's Guide ist als PDF-Dokument verfügbar; von der Startseite auf *Support* gehen und im gesamten Projekt unter *Documents & files* nach *Basic* suchen.

12.3 GNOME Office

GNOME Office (GO, <http://www.gnome.org/gnome-office/>) ist eine lockere Verbindung verschiedener, anerkannt leistungsfähiger Werkzeuge für Büroaufgaben aus der GNU-Sammlung. Die Komponenten sind nicht mit dem Ziel entwickelt worden, daraus ein Office-Paket zu schnüren, und sind auch heute noch weitgehend unabhängig voneinander. Die Zusammenarbeit der Komponenten ist aber ein erklärtes Ziel der Entwickler. Aus dieser lockeren Verbindung rührt, dass man das Paket nahezu beliebig erweitern kann, beispielsweise um die Bildbearbeitung GIMP oder den Terminplaner PIM. Gemäß der offiziellen Webseite gehören derzeit zu GNOME Office:

- Abiword, eine Textverarbeitung,
- Gnumeric, eine Tabellenkalkulation,
- GNOME-DB, ein Datenbank-Frontend, siehe Abbildung 12.3.

Abiword ist in Abschnitt 5.6 *WYSIWYG-Textprozessoren* auf Seite 199 vorgestellt worden, Gnumeric in Abschnitt 6.6 *Tabellenkalkulationen* auf Seite 262. GNOME-DB ist ein grafisches Frontend für die GNOME Data Access (GDA) Architektur, eine Schnittstelle zu verschiedenartigen Informationsquellen. Das Debian-Paket *gda-postgres* beispielsweise richtet eine GDA-Schnittstelle zur Datenbank PostgreSQL ein, siehe Abschnitt 7.4 *PostgreSQL* auf Seite 279. Aufgerufen wird das Frontend als:

```
joe@debian:~$ gnomedb -fe
```

Eine ausführliche Dokumentation findet man auf <http://www.gnome-db.org/>. Im GNOME-Lager ist keine große Begeisterung für das Office-Konzept zu erkennen – wofür es gute Gründe gibt – aber *Office* ist nun einmal zu einem Schlagwort geworden.

12.4 KOffice

Das Paket KOffice (<http://www.kde.de/koffice/> oder <http://www.koffice.org/>) besteht derzeit (Version 1.4.1) aus folgenden Komponenten:

- KWord, eine FrameMaker-ähnliche Textverarbeitung,
- KSpread, eine Tabellenkalkulation,
- KPresenter, eine Präsentationssoftware,
- Kivio, ein Werkzeug zum Zeichnen von Flussdiagrammen,
- Karbon14, ein Werkzeug zum Zeichnen von Vektorgrafiken,
- Krita, ein Bildbearbeitung ähnlich GIMP oder Adobe Photoshop,

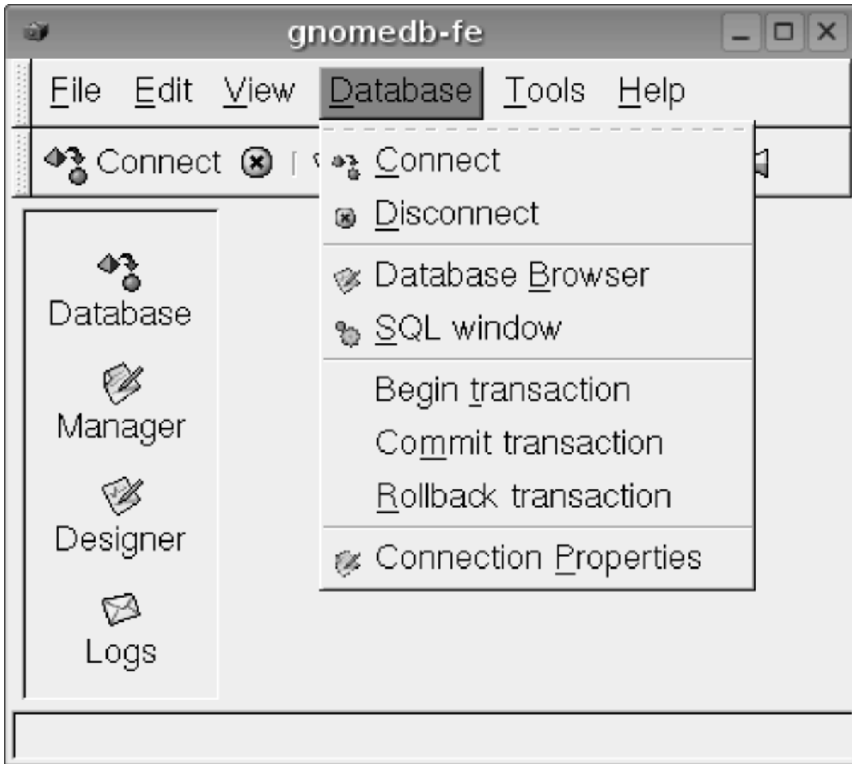


Abbildung 12.3. Screenshot von GNOME-DB, einem Datenbank-Frontend aus GNOME Office. Im Menü erkennen wir die üblichen Funktionen zum Verbinden mit der Datenbank und Trennen von ihr, zum Eingeben, Bestätigen oder Zurücknehmen einer Transaktion. Mit dem Designer legt man Tabellen und Views an.

- Kugar, ein Werkzeug zum Abfassen von Geschäftsberichten,
- KPlato, Projektmanagement und Planung,
- KChart, zum Erstellen von Diagrammen,
- Kexi, ein Frontend zu Datenbanken und -diensten, darunter MySQL und PostgreSQL, siehe Abbildung 12.4,
- KFormula, ein Formel-Editor.

Zu KOffice kommen laufend Werkzeuge hinzu, manche in einem noch ziemlich grünen Zustand. Es versteht sich, dass KOffice gut in den KDE-Desktop eingebunden ist. Die Aufteilung in viele kleine Werkzeuge ist einerseits schwierig zu überschauen, andererseits führt sie zu schnellen Programmen. Es ist zu hoffen, dass sich die drei Office-Pakete für Linux auf gemeinsame Datenformate hin bewegen, damit die Entscheidung für ein Paket nicht Werkzeuge aus anderen Paketen ausschließt und die Daten zwischen verschiedenen Rechnern austauschbar sind.

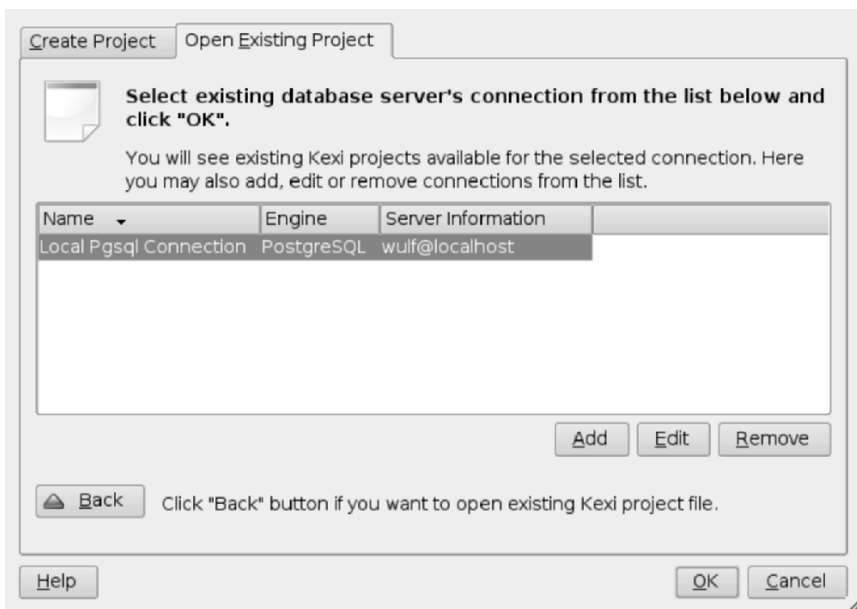


Abbildung 12.4. Screenshot des Datenbank-Frontends *kexi* aus dem KOffice-Paket

12.5 CrossOver Office

CrossOver Office von der Firma Codeweavers (<http://www.codeweavers.com/>) ist ein kommerzieller, aber dennoch preisgünstiger Emulator von MS Windows, der es ermöglicht, MS Office und einige weitere Anwendungen unter Linux zu betreiben. MS Windows wird nicht benötigt, nur die Anwendung (Office). Der Emulator kommt in drei Ausführungen. Die Standard Edition ist eine Software für den Einzelplatz, die Professional Edition ist multi-user-fähig, und die weniger interessante Server Edition ist für Linux Thin Clients gedacht. CrossOver Office ist kein vollständiger Windows-Emulator, sondern arbeitet nur mit einigen bestimmten Anwendungen zusammen. Im Zweifelsfall ist vorher zu klären, ob eine in Frage stehende Anwendung unterstützt wird. Codeweavers ist auch die Schmiede des freien Windows-Emulators *wine*, siehe Abschnitt 1.8.5 *Software für MS Windows oder MacOS* auf Seite 52.

Spiele und Spielereien

Hier werden einige unter Debian GNU/Linux verfügbare Computerspiele und mehr oder weniger ernsthafte Spielereien vorgestellt.

13.1 GNU Chess (gnuchess, xboard)

Spiele stellen hohe und vielseitige Anforderungen an Hard- und Software. Wer nicht bloß spielen will und sich für die Hintergründe interessiert, ist mit dem *Linux Gamer's HOWTO*, aktuell von 2004, gut bedient.

GNU Chess ist ein erprobtes Schachprogramm, aktuell in Version 5, das mit einem Textbildschirm (curses-Bibliothek) zufrieden ist. Es sieht aber hübscher aus, wenn man die grafische Benutzerschnittstelle *xboard* hinzunimmt, die auf X11 aufsetzt. Alternativ dazu die Debian-Pakete *eboard* plus *eboard-extras-pack1*. Man braucht die Debian-Pakete *gnuchess*, *gnuchess-book* und *xboard* bzw. *eboard* aus der *sarge*-Distribution. Aus einem Terminalfenster (xterm, Konsole, gnome-terminal) wird das Spiel folgendermaßen gestartet:

```
joe@debian:~$ xboard -size small
```

Anstatt *small* passt oft *mediocre* oder *average*, weitere Größen und Optionen siehe man *xboard*. Ein Handbuch zu GNU Chess liegt unter */usr/share/doc/gnuchess/MANUAL*. Man kann allein gegen die Maschine spielen, zu zweit auf einem Rechner oder im Netz. Es ist auch unterhaltsam zu beobachten, wie die Maschine gegen sich selbst spielt. Das Programm bietet vielfältige Möglichkeiten zum Mitschreiben, Analysieren und Üben.

Eine Alternative zu *gnuchess* enthält das Debian-Paket *crafty* aus dem Non-free-Bereich mit mehreren Paketen *crafty-books-** aus dem Contrib-Bereich. Neuigkeiten und Quellen bei <http://www.crafty-chess.com/>.

Weitere Informationen bei <http://www.gnu.org/software/chess/chess.html> und <http://www.tim-mann.org/chess.html>. Auf den Webseiten finden sich auch interessante Hinweise auf andere Schachprogramme. Zu Schach allgemein siehe <http://www.unix-ag.uni-kl.de/~chess/>,

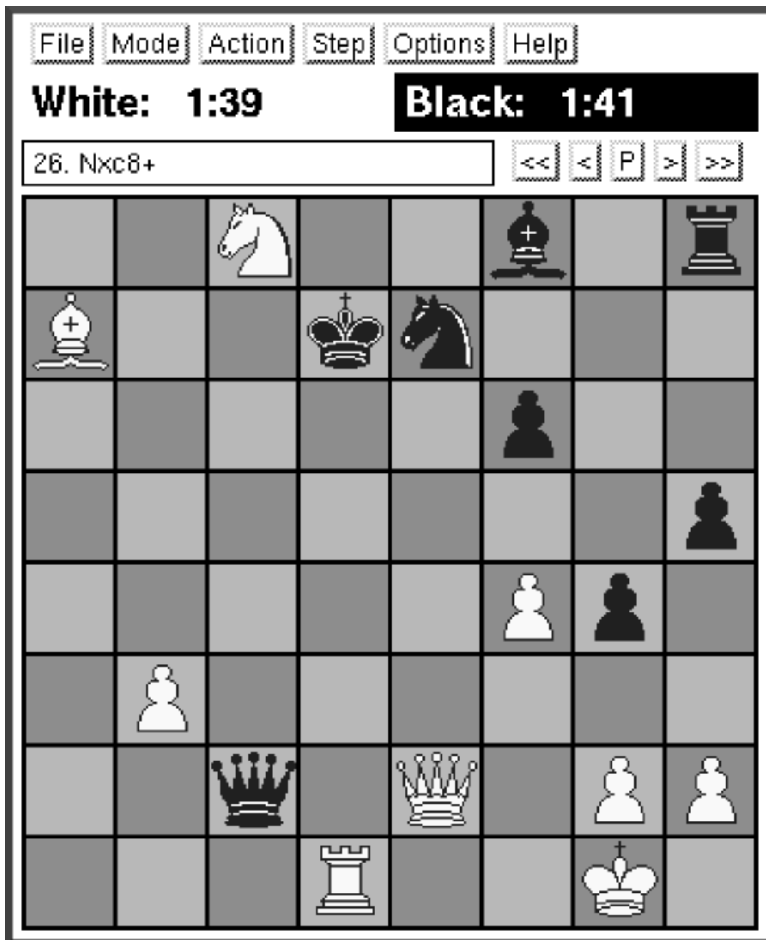


Abbildung 13.1. Screenshot von GNU Chess mit xboard

<http://www.schachbund.de/> und <http://www.fide.com/>, speziell zu Computerschach: <http://www.computerschach.de/>. Und falls Sie einmal einen richtigen Computer beim Schachspielen erleben wollen, ist <http://www.research.ibm.com/deepblue/> die passende Anschrift.

13.2 Brettspiele (gtkboard), Go (gnugo, cgoban)

Das Debian-Paket *gtkboard* bietet unter dem gleichnamigen Kommando eine Reihe von Brettspielen an, teils ausgereift, teils noch nicht spielbar. Abbildung 13.2 zeigt ein Puzzle aus dem Paket, Abbildung 13.3 den Schirm eines Mastermind-ähnlichen Spiels, bei dem die Verteilung der am rechten Rand gezeigten Farben hinter den am

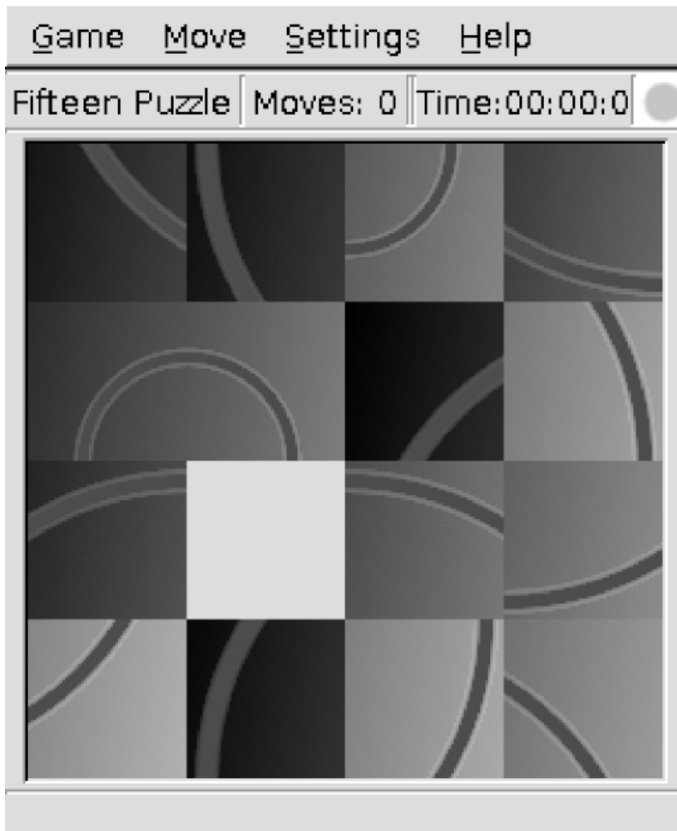


Abbildung 13.2. Screenshot eines Puzzles aus dem Debian-Paket *gtkboard*, auf dem Schirm farbig

oberen Rand schwarz dargestellten Kreisen zu raten ist. Die Spielereingabe ist am unteren Rand zu sehen. Ein ähnliches Spiel ist in dem Paket *gnomermind* enthalten.

GO ist ein altes strategisches Brettspiel aus China für zwei Spieler. Deutschsprachige Informationen findet man bei <http://www.degob.de/> und wie immer in der Wikipedia. Das Kommando `gnugo` startet ein GO-Spiel auf einem Textbildschirm (ohne X11). Nimmt man das Paket `cgoban` hinzu und hat X11 laufen, wird das Spiel hübscher.

13.3 Freeciv – ein Civilization-Klon

FreeCiv ist ein freier Klon des Spieles *Civilisation II* der Firma Microprose. Informationen finden sich in `/usr/share/doc/freeciv` und auf <http://www.freeciv.org/>. Es ist ein Strategiespiel, auch für mehrere Spieler über ein Netz.

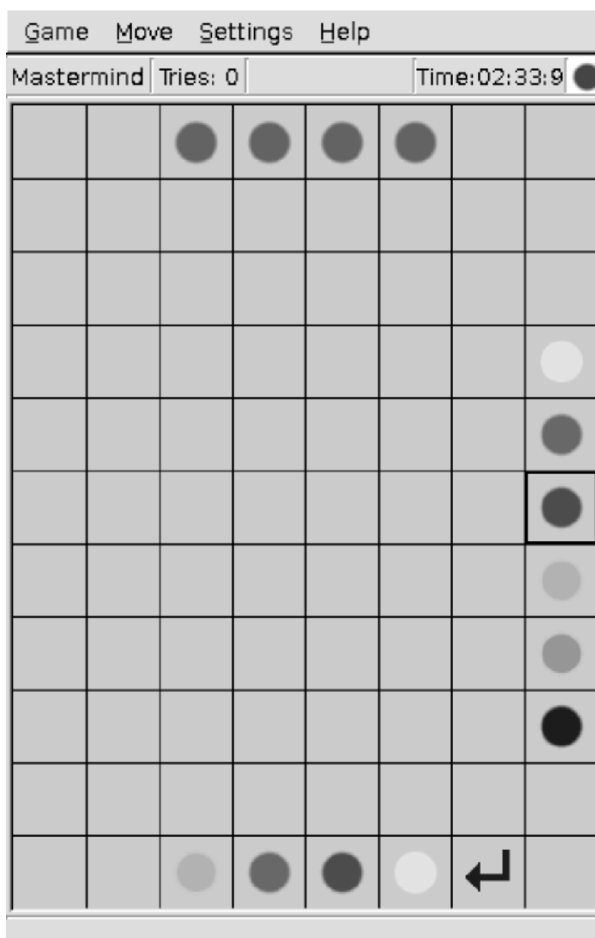


Abbildung 13.3. Screenshot eines Mastermind-Spiels aus dem Debian-Paket *gtkboard*

Es kann auf einem lokalen Server gespielt werden; im Internet stehen aber auch eine Handvoll Metaserver bereit, auf denen jedermann mitspielen kann.

Das Spiel setzt sich aus einem Server (*civserver*) und einer beliebigen Anzahl von Clients (Spielern, *civclient*) zusammen. Die Clients benötigen X11, der Server nicht. Das Debian-Paket für den Server heißt *freeciv* (*woody*) bzw. *freeciv-server* (*sarge*). Für die Klienten stehen je nach verwendeter Grafikbibliothek zwei Debian-Pakete zur Verfügung: *freeciv-gtk* oder *freeciv-xaw3d* für *woody* bzw. *freeciv-client-gtk* oder *freeciv-client-xaw3d* für *sarge*.

Beim Spielen im Netz durch eine Firewall hindurch ist unter Umständen darauf zu achten, dass der verwendete Port freigeschaltet ist (port forwarding). Im übrigen

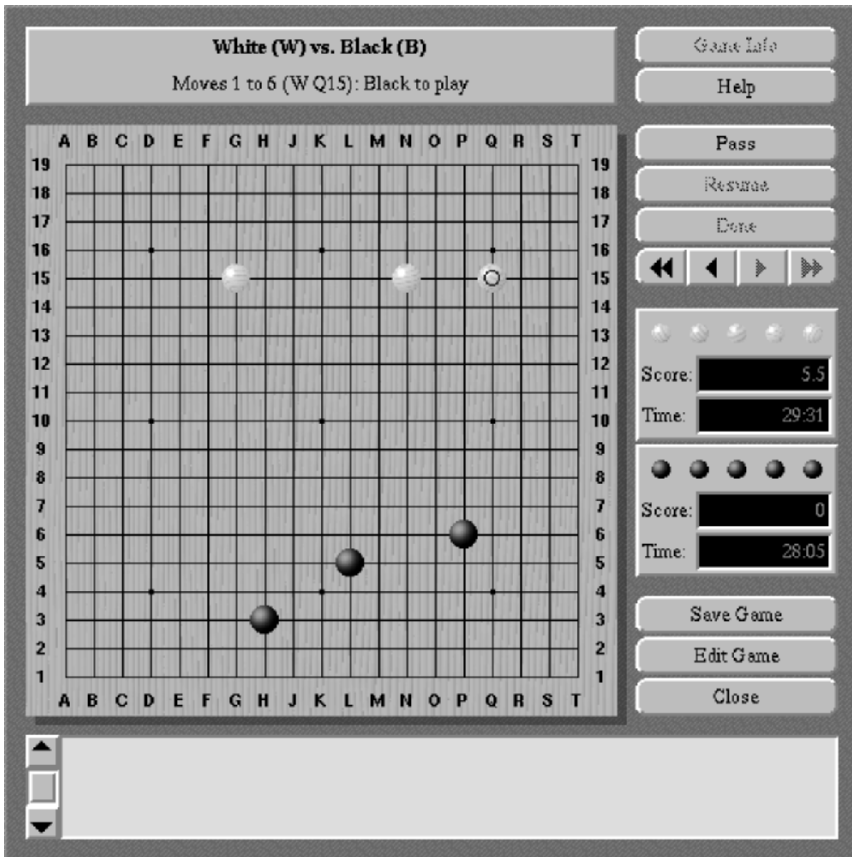


Abbildung 13.4. Screenshot von gnugo (GO-Spiel) mit cgoban

ist das Spiel in vielerlei Hinsicht konfigurierbar. Das Ziel ist, mit friedlichen und kriegesischen Mitteln zur mächtigsten Zivilisation im Spiel zu werden.

13.4 TuxRacer – für Wintersportler

Das Spiel TuxRacer belegt 15 MB auf der Platte. Es geht darum, den Pinguin Tux auf einer Abfahrtspiste in einer dreidimensionalen Winterlandschaft möglichst schnell ins Ziel zu bringen, Hindernissen (Slalomstangen, Bäume) auszuweichen und nebenbei noch herumliegende Heringe einzusammeln. Das Programm kennt keine Kommandozeilen-Argumente. Ein Blick in die Datei `$HOME/.tuxracer/options` und vorsichtiges Editieren kann weiterhelfen. Per Default wird TuxRacer mittels der Cursor-Tasten gesteuert. Die Escape-Taste bricht das Spiel ab.



Abbildung 13.5. Screenshot des Spieles FreeCiv. ALEXANDER DER GROSSE am Beginn seiner Laufbahn. Nicht ganz der historischen Wahrheit entsprechend, da er ein bereits großes Reich von seinem Vater erbt.

TuxRacer bringt Sound mit, allerdings berichten viele Benutzer von Problemen. Bei einem Start aus dem KDE-Menü (*Games -> Debian -> Arcade -> TuxRacer*) kam der Sound sofort, bei einem Start aus der Kommandozeile nicht. Abhilfe schaffte die Änderung der Option `set audio_frequ_mode` von 22 auf 44 kHz.

Bis zur Version 0.61 war TuxRacer frei im Sinne der GPL. Eine aktuelle Information von HOLGER THIELE findet man unter <http://www.hthiele.de/> – auch zu anderen Spielen unter Linux – die freie Version bei <http://tuxracer.sourceforge.net/>, Kommentare bei <http://www.linuxgames.com/>, <http://www.pro-linux.de/spiele/tuxracer.html> oder <http://happypenguin.org/>, die freie Weiterentwicklung OpenRacer bei <http://openracer.worldforge.org/>. Eine zweite freie Abwandlung namens PlanetPenguin Racer ist auf <http://projects.planetpenguin.de/racer/> zu Hause. Einen Strecken- oder Kurseditor und weitere Abfahrtsstrecken liegen auf <http://www.erinacom.de/tux/> oder <http://tuxracer.fubaby.com/>.

Die kommerzielle Fortsetzung, gegenwärtig in Version 1.1, von den Sunspire Studios wird von Amazon oder Ixsoft angeboten, der Hersteller hat sich aus dem Web verabschiedet.



Abbildung 13.6. Screenshot des Spieles TuxRacer, an heißen Sommertagen sehr erfrischend. Mit der Farbe von Tux stimmt etwas nicht.

13.5 Pingus – ein Lemmings-Klon

Pingus ist ein freier Lemmings-Klon, eine Art von Puzzle mit Hindernissen. Es belegt 20 MB auf der Platte. Die Manual-Seite ist kurz, die Info-Seite gibt mehr her, ausführliche Informationen findet man bei <http://pingus.seul.org/>. Zu dem Spiel gehört ein Level-Editor, mit dem man eigene Level gestalten kann. Die Entwicklung geht langsam voran, aber sie lebt.

13.6 Flight Gear – ein anspruchsvoller Flugsimulator

Das Debian-Paket *flightgear* stellt unter dem Aufruf `fgfs` einen Flugsimulator zur Verfügung:

```
joe@debian:~$ man fgfs
```

```
joe@debian:~$ fgfs -help
```

```
joe@debian:~$ fgfs
```

Die Dateien belegen im Grundausbau 200 MB auf der Platte; das *World Scenery Set* umfasst drei DVDs mit zusammen 12 GB. Einzelne Gegenden können aus dem



Abbildung 13.7. Screenshot des Startbildes von Pingus, einem Lemmings-Klone

Netz heruntergeladen werden. CPU, Arbeitsspeicher und Grafikkarte sollten nicht zu schwach ausgelegt sein. Ferner empfiehlt es sich, ergänzend zur Tastatur ein etwas realistischeres Eingabegerät zu gebrauchen. Vielleicht findet sich bei Ebay ein alter Link-Trainer, den man mit einer USB-Schnittstelle aufrüsten kann ... Das Spielen zu mehreren in einem lokalen Netz wird unterstützt (Multiplayer Protocol). Alles Weitere bei <http://www.flightgear.org/>.

Bei dem Simulator geht es nicht darum, feindliche Flugobjekte abzuschießen. Er will vielmehr das Verständnis für die Physik und Technik des Fliegens fördern und kommt daher auch mit verschiedenen Modellen für die Flugdynamik.

13.7 Das Planetarium auf dem Desktop (kstars)

Die Astronomen mögen verzeihen, dass wir ein ernsthaftes wissenschaftliches Gerät zu den Spielzeugen rechnen. Das Programm *kstars* wird entweder in KDE unter dem Menüpunkt *Applications* oder in einer Kommandozeile aufgerufen:

```
joe@debian:~$ kstars -help-all
```

```
joe@debian:~$ kstars
```

Als erstes vergewissere man sich über Zeit (LT, Local Time) und Ort der Beobachtung. Beides lässt sich einstellen. Für die Abbildung 13.9 auf Seite 422 haben wir

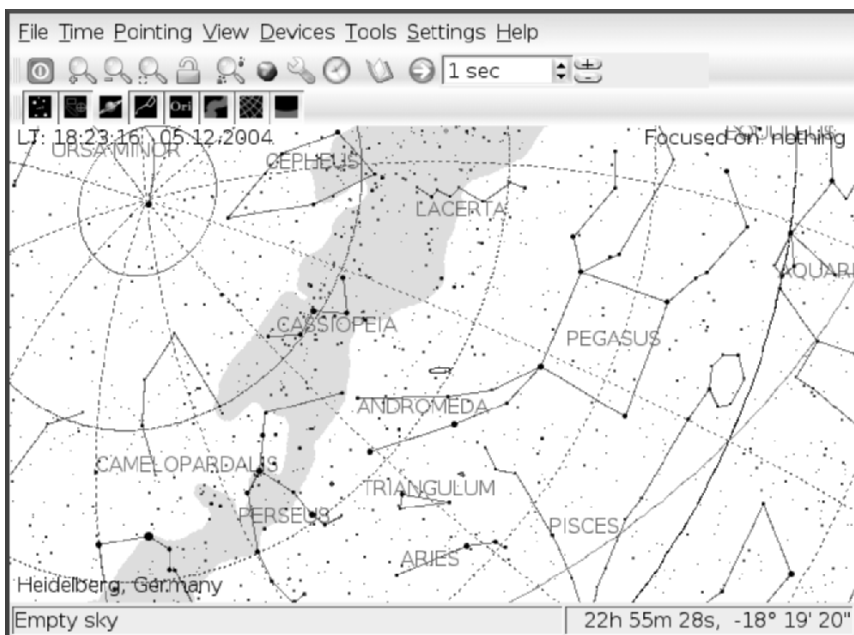


Abbildung 13.9. Screenshot des Programmes Kstars, eines Planetariums auf dem Desktop. Hier der Himmel über Heidelberg am 5. Dez. 2004 gegen 18.20, in Bildmitte der Andromeda-Nebel. Als Darstellungsart wurde *Sternenkarte* gewählt.

```
joe@debian:~$ megahal-personal
```

Anfängliche Klagen über fehlendes Hirn (`megahal.brn`) übergeht man, das wächst von allein. MegaHAL startet mit einer Begrüßung; dann kann man antworten. Die Antwort darf Zeilenwechsel enthalten. Erst ein zweifacher Zeilenwechsel beendet die Eingabe. Von Geburt aus redet MegaHAL englisch, aber wenn man stur bleibt und immer nur deutsch mit ihm redet, stellt es sich langsam darauf um. Während des Zwiegesprächs können Kommandos eingegeben werden, die mit einem Doppelkreuz (command prefix) beginnen:

- `#QUIT` MegaHAL beenden und Gelerntes speichern,
- `#EXIT` MegaHAL beenden, ohne Gelerntes zu speichern,
- `#HELP` die verfügbaren Kommandos anzeigen und kurz erläutern.

Es ist interessant zu verfolgen, wie MegaHAL lernt, vor allem, wenn man deutsch mit ihm redet, weil es in dieser Sprache keine Vorkenntnisse mitbringt. Tippfehler lernt es genau so eifrig wie schwierige Fremdwörter. Wird man grob zu MegaHAL, ändert es seine Ausdrucksweise ebenfalls rasch. Nach einigen Stunden Unterricht beginnt MegaHAL, beinahe intelligente Sätze zu äußern. Hängt vom Lehrer ab.

Eine Verwandte von MegaHAL namens Nicole (Nearly Intelligent Computer Operated Language Examiner) liegt unter <http://nicole.sourceforge.net>.

net. Sie ist nicht bei Debian zu haben. Die im Netz eingerichtete Nicole mit Web-Interface zeigte sich nicht Gesprächsbereit, weder unter Galeon noch unter Opera.

13.9 Fraktale (kfract, xaos, xfractint)

Die eigenartigen Bilder, die Fraktal-Generatoren liefern, kennt jeder. Die Mathematik dahinter zu beschreiben, würde den Rahmen des Buches sprengen. Am besten liest man bei **BENOIT MANDELBROT** nach, einem französischen Mathematiker, der die Fraktale populär gemacht hat.

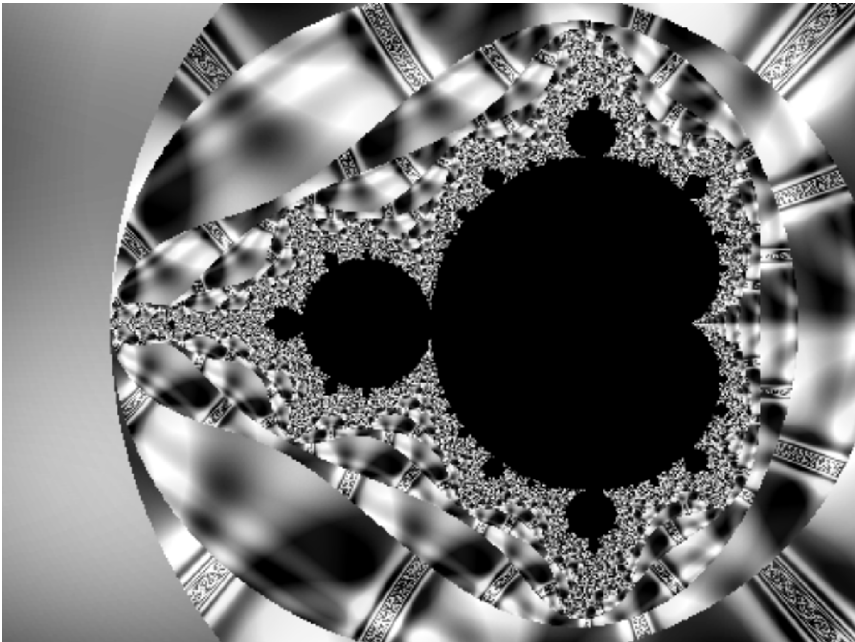


Abbildung 13.10. Screenshot eines Apfelmännchens (Fraktals), erzeugt mit Xaos, auf dem Bildschirm farbig

Der KDE Fraktal-Generator `kfract` wird aus der Kommandozeile aufgerufen oder als Menüpunkt *Graphics -> Fractal Generator*. Er zeichnet sofort ein Apfelmännchen in ein eigenes Fenster. Über das Menü *Options* lassen sich Zahlenwerte eingeben, jedoch nicht der Typ (die zu Grunde liegende Gleichung) oder die Farben ändern.

Xaos zeichnet ein Apfelmännchen auf den Bildschirm, Klickt man mit der linken Maustaste in das Bild, kommt es näher. Die rechte Maustaste verschiebt das Bild in die Ferne. Mit der Taste `<h>` holt man sich eine Hilfe ins Fenster, darunter eine gut

gemachte, animierte Einführung in Fraktale in mehreren Sprachen. Mit der Tastenfolge <h> <h> <1> <2> <h> <h> <2> <2> holt man sich eine deutschsprachige Einführung in Fraktale auf den Schirm. Abbrechen mit <q> oder <esc>.

Der Fraktal-Generator `xfractint` – ursprünglich von IBM für den PC entwickelt – ist für X11 umgeschrieben worden. Nach unserem Eindruck stimmen die in der Online-Hilfe (<h>) angegebenen Tastenzuordnungen nicht ganz zu dem Verhalten des Programms. Beenden mit <esc>. Näheres bei <http://www.fractint.org/>.

13.10 Türme von Hanoi

Die *Türme von Hanoi* sind ein Spiel für eine einzelne Person, bei dem ein Stapel unterschiedlich großer Scheiben von einem Turm auf einen zweiten Turm gebracht werden soll, ein dritter Turm als Zwischenlager dient, mit einem Zug immer nur eine Scheibe bewegt werden und niemals eine größere Scheibe über einer kleineren liegen darf. Das Spiel wurde 1883 von dem französischen Mathematiker FRANÇOIS EDUOUARD ANATOLE LUCAS erdacht. Es ist ein beliebtes Beispiel für rekursive Lösungswege; suchen Sie einmal im Web nach *Towers of Hanoi* und *recurs* sowie ihren deutschen Entsprechungen.

Das Spiel gibt es nicht als Debian-Paket, aber wir können es uns selbst schreiben. Tippen Sie mit einem Editor das Programmbeispiel 13.1 in eine Datei namens `hanoi`, setzen Sie deren Zugriffsrechte mittels `chmod 750 hanoi`, und rufen Sie es mittels `./hanoi` aus einem Textfenster auf. Diese Art von Programmen nennt man Shellskript. Im obigen Shellskript ist die Anzahl der Scheiben auf 16 begrenzt, weil mit steigender Scheibenzahl die Zeiten lang werden (Anzahl der Züge minimal $2^n - 1$, bei 10 Scheiben also 1023). Bei 16 Scheiben braucht selbst ein schneller Rechner etwas Zeit. Sie dürfen die Grenze im Skript aber beliebig erhöhen, bis Ihr Rechner streikt.

```
# Shellskript hanoi (Tuerme von Hanoi)
# Aufruf hanoi n mit n = Anzahl der Scheiben
# max. 16 Scheiben, wegen Zeitbedarf

# Funktion, rekursiv (selbstaufrufend)

function fhanoi
{
    typeset -i x=$1-1
    ((x>0)) && fhanoi $x $2 $4 $3
    echo "von Turm $2 nach Turm $3"
    ((x>0)) && fhanoi $x $4 $3 $2
}
```

```
# Hauptskript

case $1 in
[1-9] | [1][0-6])
    echo "Tuerme von Hanoi"
    echo "Start Turm 1, Ziel Turm 2, mit $1 Scheiben"
    echo "Bewege die oberste Scheibe"
    fanoi $1 1 2 3;;
*) echo "Argument zwischen 1 und 16 erforderlich"
    exit;;
esac
```

Quelle 13.1 . Shellskript Türme von Hanoi

Das Hauptskript ruft die Funktion `fanoi` mit vier Argumenten auf. Das erste ist die Anzahl der Scheiben, die weiteren Argumente sind Start-, Ziel- und Zwischenturm. Die Funktion `fanoi` setzt die ganzzahlige Variable `x` auf den um 1 verminderten Wert der Anzahl, im Beispiel also zunächst auf 2. Diese Variable begrenzt die Rekursionstiefe. Ist der Wert des ersten Argumentes im Aufruf bei 1 angekommen, ruft sich die Funktion nicht mehr auf, sondern gibt nur noch aus. Die Zeile:

```
((x>0)) && fanoi $x $2 $4 $3
```

ist in der Bourne-again-Shell so zu verstehen:

- berechne den Wert des booleschen Ausdrucks `x > 0`,
- falls TRUE herauskommt, rufe die Funktion `fanoi` mit den jeweiligen Argumenten auf, wobei `$2` das zweite Argument ist usw.

Schreiben wir uns die Folge der Funktionsaufrufe untereinander, erhalten wir:

```
fanoi 3 1 2 3
    fanoi 2 1 3 2
        fanoi 1 1 2 3 -> print 1 2
    print 1 3
        fanoi 1 2 3 1 -> print 2 3
print 1 2
    fanoi 2 3 2 1
        fanoi 1 3 1 2 -> print 3 1
    print 3 2
        fanoi 1 1 2 3 -> print 1 2
```

Die Ausgabe des Skripts für $n = 3$ sieht folgendermaßen aus:

Tuerme von Hanoi
Start Turm 1, Ziel Turm 2, mit 3 Scheiben

Bewege die oberste Scheibe
von Turm 1 nach Turm 2
von Turm 1 nach Turm 3
von Turm 2 nach Turm 3
von Turm 1 nach Turm 2
von Turm 3 nach Turm 1
von Turm 3 nach Turm 2
von Turm 1 nach Turm 2

Für $n = 1$ ist die Lösung trivial, für $n = 2$ offensichtlich, für $n = 3$ überschaubar. Bei größeren Werten muss man systematisch vorgehen. Ein entscheidender Moment ist erreicht, wenn nur noch die unterste (größte) Scheibe im Start liegt und sich alle übrigen Scheiben im Zwischenlager befinden, geordnet natürlich. Dann bewegen wir die größte Scheibe ins Ziel. Der Rest ist nur noch, den Stapel vom Zwischenlager ins Ziel zu bewegen, eine Aufgabe, die wir bereits beim Transport der $n - 1$ Scheiben vom Start ins Zwischenlager bewältigt haben. Damit haben wir die Aufgabe von n auf $n - 1$ Scheiben reduziert. Das Rezept wiederholen wir, bis wir bei $n = 2$ angelangt sind. Wir ersetzen also eine vom Umfang her zu schwierige Aufgabe durch eine gleichartige mit geringerem Umfang so lange, bis die Aufgabe einfach genug geworden ist. Das Problem liegt darin, sich alle angefangenen, aber noch nicht zu Ende gebrachten Teilaufgaben zu merken, aber dafür gibt es Rechner. Mit der Entdeckung eines Lösungsweges (Algorithmus), der mit Sicherheit in kürzester möglicher Zeit zum Ziel führt, ist der Charakter des Spiels verloren gegangen, es ist nur noch ein Konzentrations- und Gedächtnistest.

Was noch zu tun bliebe, wäre, dem Programm eine hübsche grafische Oberfläche zu schneiden, aber das ist mehr Arbeit, als das Programm selbst zu entwerfen, und verdeutlicht den Preis, den wir für bequeme Benutzeroberflächen zahlen.

13.11 Game of Life

Das Game of Life (GOL) wurde 1970 von dem britischen Mathematiker JOHN HORTON CONWAY erdacht. Eine gute Übersicht bieten die deutsche oder die englische Wikipedia. Den zugehörigen Artikel von MARTIN GARDNER im *Scientific American* und anderes findet man unter <http://www.ibiblio.org/lifepatterns/> und <http://www.ibiblio.org/pub/Linux/science/ai/life/>, eine aufbereitete Sammlung von Links bei <http://www.radicaleye.com/lifepage/>, eine FAQ bei <http://cafaq.com/lifefaq/>. Ein fertiges Programm namens *xlife* von JON BENNETT enthält das gleichnamige Debian-Paket aus der Abteilung *Games*, zwei weitere Programme gibt es als Tarball im Netz:

- GtkLife von SUZANNE BUTTON <http://ironphoenix.org/tril/gtklife/> oder <http://freshmeat.net/projects/gtklife/>,
- Life von DAVID BELL <ftp://sunsite.unc.edu/pub/Linux/games/amusements/life/dblife-6.0.tgz>

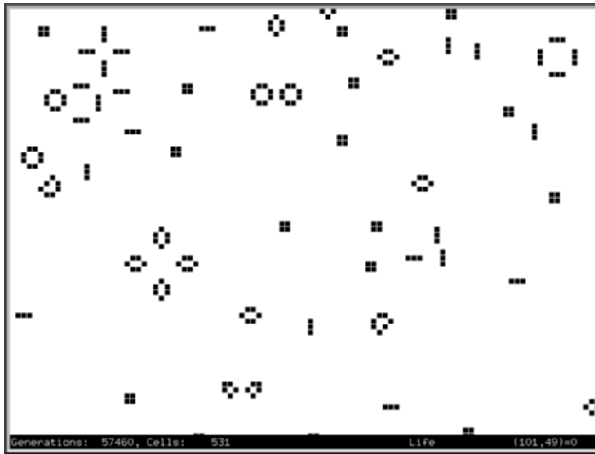


Abbildung 13.11. Screenshot von *xlife*, einem *Game of Life*

Das Game of Life ist ein zweidimensionaler zellulärer Automat, eine Art von Schachbrett, auf dem nach folgenden Regeln gespielt wird:

- Jedes Feld (Zelle) ist entweder tot (in Abbildung 13.11 weiß) oder lebendig (schwarz).
- Jede neue Generation von Feldern wird entsprechend den beiden nachfolgenden Regeln aus der vorangehenden Generation gebildet.
- Ein lebendes Feld mit 2 oder 3 lebenden Nachbarn (punkt- oder linienförmig benachbart, jedes Feld hat 8 Nachbarn) lebt weiter, andere Felder sterben.
- Ein totes Feld mit genau 3 lebenden Nachbarn beginnt zu leben.

Der Reiz des Spiels liegt in der Entwicklung der Muster. Man kann auch die Regeln variieren oder mit sechseckigen Feldern oder in höheren Dimensionen spielen. Das Spiel hat außer seinem Unterhaltungswert eine tiefere Bedeutung in der Informatik (Entscheidungsproblem).

13.12 Wortspielereien

Ein Anagramm ist ein Wort oder eine Wortgruppe (Phrase), die aus einem anderen Wort oder einer anderen Phrase durch Vertauschen (Permutieren) der Zeichen entsteht. Das ist für den Rechner eine Kleinigkeit. Die Anzahl der möglichen Permutationen von n Zeichen ist $n!$. Da in Anagrammen erleichterte Regeln für das Einfügen von Leer- oder Satzzeichen bestehen, ist die Anzahl der Anagramme größer. Beispiele: *Geburt* – *Erbgut*, *Klaus* – *Lukas*, *Strolche* – *rechtlos* oder *Briefmarke* – *Fabrikmeer* – *Imkerfarbe* – *Krambriefe* – *Markfieber*. Zum Ermitteln von Anagrammen braucht ein entsprechendes Programm ein Wörterbuch (dictionary), mit dessen Hilfe es aus der Menge aller möglichen Vertauschungen die sprachlich zulässigen herausfiltert. Die geistreichen muss der Benutzer selbst herausuchen.

Das Programm `an` sucht per Default in einem Wörterbuch `/usr/dict/words`, das vermutlich nicht eingerichtet ist. Eine einfach formatierte deutsche Wörterliste findet sich unter `ftp://ftp.cs.tu-berlin.de/pub/doc/wordlists/germanl.Z`. Wenn man die Datei mit dem alten Linux/UNIX-Werkzeug `uncompress` oder mit `gunzip` verdünnt hat, bleibt noch die Frage der Umlaute, die man eventuell mit Hilfe des Streaming-Editors `sed` angehen muss. Immerhin lieferte `an` nach dem Kopieren von `germanl` nach `/usr/bin/words` als Anagramm zu *alex* das Wort *axel*.

Ein zweiter Anagramm-Generator ist das Programm `wordplay` aus dem gleichnamigen Debian-Paket. Es bringt seine Default-Wörterliste `/usr/share/games/wordplay/words721.txt` mit, die entsprechend ihrer Herkunft nur amerikanische Wörter enthält. Die Liste ist lesbar, einfach aufgebaut und wird auch vom Programm `an` verstanden. Der Aufruf

```
joe@debian:~$ wordplay copenhagen
```

liefert unter anderem *CHANGE OPEN* zurück. Besteht das Eingangsargument aus mehreren Wörtern, sind sie mit Gänsefüßchen oder Single Quotes einzurahmen, damit die Shell sie als Ganzes übergibt.

Eine besondere Form des Anagramms sind Palindrome oder Kehrwörter. Das sind Wörter, die – gegebenenfalls mit anderer Bedeutung – von rechts und links gelesen werden können: Anna, Annasusanna, Otto, Rentner, Lagerregal. Weiteres zu Wortspielen bei <http://anagramme.de/>, <http://www.gnudung.de/>, <http://www.palindromes.org/> und wie immer in der Wikipedia.

Scrabble ist ein bekanntes Gesellschaftsspiel (<http://www.scrabble.de/>, <http://www.scrabble.com/>), das mit Kreuzworträtseln verwandt ist. Die *sarge*-Distribution enthält ein Debian-Paket dieses Namens. Das Spiel wird von der Kommandozeile ohne X11 gespielt. Durch Eingabe von `help` erfährt man Weiteres, `quit` beendet das Spiel. Die gegenwärtige Version bringt ein amerikanisches Wörterbuch mit; an Optionen kennt sie nur die Spielstärke. Das Programm verträgt noch Entwicklung.

Geht es um Kreuzworträtsel, so sind zwei Aufgaben für den Rechner zu unterscheiden:

- Erzeugen (Generieren, Konstruieren) von Kreuzworträtseln,
- Lösen oder Hilfe beim Lösen.

Bei Debian findet sich noch nichts Geeignetes, aber bei <http://sourceforge.net/> wird man mit dem Suchbegriff *crossword* fündig. Im deutschen Sprachraum kann man sein Glück bei <http://www.tea.ch/de/> versuchen, zwar ein kommerzielles Angebot, aber auch mit einigen kostenlosen (nicht *freien* im Sinne von Debian) Päckchen.

Wir sind am Ende. Während des Schreibens hat sich die Anzahl der Debian-Pakete um ein paar tausend vermehrt. Debian GNU/Linux lebt. Nicht alle Zweige grünen, aber doch so viele, dass aus der zarten Pflanze von 1993 ein stattlicher Baum geworden ist.

A

Zahlensysteme

Außer dem *Dezimalsystem* sind das *Dual*-, das *Oktal*- und das *Hexadezimalsystem* gebräuchlich. Ferner spielt das *Binär codierte Dezimalsystem (BCD)* bei manchen Anwendungen eine Rolle. Bei diesem sind die einzelnen Dezimalstellen für sich dual dargestellt. Die folgende Tabelle enthält die Werte von 0 bis dezimal 255. Bequemlichkeitshalber sind auch die zugeordneten ASCII-Zeichen aufgeführt.

dezimal	dual	oktal	hex	BCD	ASCII
0	0	0	0	0	nul
1	1	1	1	1	soh
2	10	2	2	10	stx
3	11	3	3	11	etx
4	100	4	4	100	eot
5	101	5	5	101	enq
6	110	6	6	110	ack
7	111	7	7	111	bel
8	1000	10	8	1000	bs
9	1001	11	9	1001	ht
10	1010	12	a	1.0	lf
11	101	13	b	1.1	vt
12	1100	14	c	1.10	ff
13	1101	15	d	1.11	cr
14	1110	16	e	1.100	so
15	1111	17	f	1.101	si
16	10000	20	10	1.110	dle
17	10001	21	11	1.111	dc1
18	10010	22	12	1.1000	dc2
19	10011	23	13	1.1001	dc3
20	10100	24	14	10.0	dc4
21	10101	25	15	10.1	nak
22	10110	26	16	10.10	syn

23	10111	27	17	10.11	etb
24	11000	30	18	10.100	can
25	11001	31	19	10.101	em
26	11010	32	1a	10.110	sub
27	11011	33	1b	10.111	esc
28	11100	34	1c	10.1000	fs
29	11101	35	1d	10.1001	gs
30	11110	36	1e	11.0	rs
31	11111	37	1f	11.1	us
32	100000	40	20	11.10	space
33	100001	41	21	11.11	!
34	100010	42	22	11.100	”
35	100011	43	23	11.101	#
36	100100	44	24	11.110	\$
37	100101	45	25	11.111	%
38	100110	46	26	11.1000	&
39	100111	47	27	11.1001	,
40	101000	50	28	100.0	(
41	101001	51	29	100.1)
42	101010	52	2a	100.10	*
43	101011	53	2b	100.11	+
44	101100	54	2c	100.100	,
45	101101	55	2d	100.101	-
46	101110	56	2e	100.110	.
47	101111	57	2f	100.111	/
48	110000	60	30	100.1000	0
49	110001	61	31	100.1001	1
50	110010	62	32	101.0	2
51	110011	63	33	101.1	3
52	110100	64	34	101.10	4
53	110101	65	35	101.11	5
54	110110	66	36	101.100	6
55	110111	67	37	101.101	7
56	111000	70	38	101.110	8
57	111001	71	39	101.111	9
58	111010	72	3a	101.1000	:
59	111011	73	3b	101.1001	;
60	111100	74	3c	110.0	<
61	111101	75	3d	110.1	=
62	111110	76	3e	110.10	>
63	111111	77	3f	110.11	?
64	1000000	100	40	110.100	@
65	1000001	101	41	110.101	A
66	1000010	102	42	110.110	B
67	1000011	103	43	110.111	C

68	1000100	104	44	110.1000	D
69	1000101	105	45	110.1001	E
70	1000110	106	46	111.0	F
71	1000111	107	47	111.1	G
72	1001000	110	48	111.10	H
73	1001001	111	49	111.11	I
74	1001010	112	4a	111.100	J
75	1001011	113	4b	111.101	K
76	1001100	114	4c	111.110	L
77	1001101	115	4d	111.111	M
78	1001110	116	4e	111.1000	N
79	1001111	117	4f	111.1001	O
80	1010000	120	50	1000.0	P
81	1010001	121	51	1000.1	Q
82	1010010	122	52	1000.10	R
83	1010011	123	53	1000.11	S
84	1010100	124	54	1000.100	T
85	1010101	125	55	1000.101	U
86	1010110	126	56	1000.110	V
87	1010111	127	57	1000.111	W
88	1011000	130	58	1000.1000	X
89	1011001	131	59	1000.1001	Y
90	1011010	132	5a	1001.0	Z
91	1011011	133	5b	1001.1	[
92	1011100	134	5c	1001.10	\
93	1011101	135	5d	1001.11]
94	1011110	136	5e	1001.100	^
95	1011111	137	5f	1001.101	_
96	1100000	140	60	1001.110	,
97	1100001	141	61	1001.111	a
98	1100010	142	62	1001.1000	b
99	1100011	143	63	1001.1001	c
100	1100100	144	64	1.0.0	d
101	1100101	145	65	1.0.1	e
102	1100110	146	66	1.0.10	f
103	1100111	147	67	1.0.11	g
104	1101000	150	68	1.0.100	h
105	1101001	151	69	1.0.101	i
106	1101010	152	6a	1.0.110	j
107	1101011	153	6b	1.0.111	k
108	1101100	154	6c	1.0.1000	l
109	1101101	155	6d	1.0.1001	m
110	1101110	156	6e	1.1.0	n
111	1101111	157	6f	1.1.1	o
112	1110000	160	70	1.1.10	p

113	1110001	161	71	1.1.11	q
114	1110010	162	72	1.1.100	r
115	1110011	163	73	1.1.101	s
116	1110100	164	74	1.1.110	t
117	1110101	165	75	1.1.111	u
118	1110110	166	76	1.1.1000	v
119	1110111	167	77	1.1.1001	w
120	1111000	170	78	1.10.0	x
121	1111001	171	79	1.10.1	y
122	1111010	172	7a	1.10.10	z
123	1111011	173	7b	1.10.11	{
124	1111100	174	7c	1.10.100	
125	1111101	175	7d	1.10.101	}
126	1111110	176	7e	1.10.110	~
127	1111111	177	7f	1.10.111	del
128	10000000	200	80	1.10.1000	
129	10000001	201	81	1.10.1001	
130	10000010	202	82	1.11.0	
131	10000011	203	83	1.11.1	
132	10000100	204	84	1.11.10	
133	10000101	205	85	1.11.11	
134	10000110	206	86	1.11.100	
135	10000111	207	87	1.11.101	
136	10001000	210	88	1.11.110	
137	10001001	211	89	1.11.111	
138	10001010	212	8a	1.11.1000	
139	10001011	213	8b	1.11.1001	
140	10001100	214	8c	1.100.0	
141	10001101	215	8d	1.100.1	
142	10001110	216	8e	1.100.10	
143	10001111	217	8f	1.100.11	
144	10010000	220	90	1.100.100	
145	10010001	221	91	1.100.101	
146	10010010	222	92	1.100.110	
147	10010011	223	93	1.100.111	
148	10010100	224	94	1.100.1000	
149	10010101	225	95	1.100.1001	
150	10010110	226	96	1.101.0	
151	10010111	227	97	1.101.1	
152	10011000	230	98	1.101.10	
153	10011001	231	99	1.101.11	
154	10011010	232	9a	1.101.100	
155	10011011	233	9b	1.101.101	
156	10011100	234	9c	1.101.110	
157	10011101	235	9d	1.101.111	

158	10011110	236	9e	1.101.1000
159	10011111	237	9f	1.101.1001
160	10100000	240	a0	1.110.0
161	10100001	241	a1	1.110.1
162	10100010	242	a2	1.110.10
163	10100011	243	a3	1.110.11
164	10100100	244	a4	1.110.100
165	10100101	245	a5	1.110.101
166	10100110	246	a6	1.110.110
167	10100111	247	a7	1.110.111
168	10101000	250	a8	1.110.1000
169	10101001	251	a9	1.110.1001
170	10101010	252	aa	1.111.0
171	10101011	253	ab	1.111.1
172	10101100	254	ac	1.111.10
173	10101101	255	ad	1.111.11
174	10101110	256	ae	1.111.100
175	10101111	257	af	1.111.101
176	10110000	260	b0	1.111.110
177	10110001	261	b1	1.111.111
178	10110010	262	b2	1.111.1000
179	10110011	263	b3	1.111.1001
180	10110100	264	b4	1.1000.0
181	10110101	265	b5	1.1000.1
182	10110110	266	b6	1.1000.10
183	10110111	267	b7	1.1000.11
184	10111000	270	b8	1.1000.100
185	10111001	271	b9	1.1000.101
186	10111010	272	ba	1.1000.110
187	10111011	273	bb	1.1000.111
188	10111100	274	bc	1.1000.1000
189	10111101	275	bd	1.1000.1001
190	10111110	276	be	1.1001.0
191	10111111	277	bf	1.1001.1
192	11000000	300	c0	1.1001.10
193	11000001	301	c1	1.1001.11
194	11000010	302	c2	1.1001.100
195	11000011	303	c3	1.1001.101
196	11000100	304	c4	1.1001.110
197	11000101	305	c5	1.1001.111
198	11000110	306	c6	1.1001.1000
199	11000111	307	c7	1.1001.1001
200	11001000	310	c8	10.0.0
201	11001001	311	c9	10.0.1
202	11001010	312	ca	10.0.10

203	11001011	313	cb	10.0.11
204	11001100	314	cc	10.0.100
205	11001101	315	cd	10.0.101
206	11001110	316	ce	10.0.110
207	11001111	317	cf	10.0.111
208	11010000	320	d0	10.0.1000
209	11010001	321	d1	10.0.1001
210	11010010	322	d2	10.1.0
211	11010011	323	d3	10.1.1
212	11010100	324	d4	10.1.10
213	11010101	325	d5	10.1.11
214	11010110	326	d6	10.1.100
215	11010111	327	d7	10.1.101
216	11011000	330	d8	10.1.110
217	11011001	331	d9	10.1.111
218	11011010	332	da	10.1.1000
219	11011011	333	db	10.1.1001
220	11011100	334	dc	10.10.0
221	11011101	335	dd	10.10.1
222	11011110	336	de	10.10.10
223	11011111	337	df	10.10.11
224	11100000	340	e0	10.10.100
225	11100001	341	e1	10.10.101
226	11100010	342	e2	10.10.110
227	11100011	343	e3	10.10.111
228	11100100	344	e4	10.10.1000
229	11100101	345	e5	10.10.1001
230	11100110	346	e6	10.11.0
231	11100111	347	e7	10.11.1
232	11101000	350	e8	10.11.10
233	11101001	351	e9	10.11.11
234	11101010	352	ea	10.11.100
235	11101011	353	eb	10.11.101
236	11101100	354	ec	10.11.110
237	11101101	355	ed	10.11.111
238	11101110	356	ee	10.11.1000
239	11101111	357	ef	10.11.1001
240	11110000	360	f0	10.100.0
241	11110001	361	f1	10.100.1
242	11110010	362	f2	10.100.10
243	11110011	363	f3	10.100.11
244	11110100	364	f4	10.100.100
245	11110101	365	f5	10.100.101
246	11110110	366	f6	10.100.110
247	11110111	367	f7	10.100.111

248	11111000	370	f8	10.100.1000
249	11111001	371	f9	10.100.1001
250	11111010	372	fa	10.101.0
251	11111011	373	fb	10.101.1
252	11111100	374	fc	10.101.10
253	11111101	375	fd	10.101.11
254	11111110	376	fe	10.101.100
255	11111111	377	ff	10.101.101

B

Zeichensätze und Sondertasten

B.1 EBCDIC, ASCII, Roman8, IBM-PC

Die Zeichensätze sind in den Ein- und Ausgabegeräten (Terminal, Drucker) gespeicherte Tabellen, die die Zeichen in Zahlen und zurück umsetzen.

dezimal	oktal	EBCDIC	ASCII-7	ROMAN8	IBM-PC
0	0	nul	nul	nul	nul
1	1	soh	soh	soh	Grafik
2	2	stx	stx	stx	Grafik
3	3	etx	etx	etx	Grafik
4	4	pf	eot	eot	Grafik
5	5	ht	enq	enq	Grafik
6	6	lc	ack	ack	Grafik
7	7	del	bel	bel	bel
8	10		bs	bs	Grafik
9	11	rlf	ht	ht	ht
10	12	smm	lf	lf	lf
11	13	vt	vt	vt	home
12	14	ff	ff	ff	ff
13	15	cr	cr	cr	cr
14	16	so	so	so	Grafik
15	17	si	si	si	Grafik
16	20	dle	dle	dle	Grafik
17	21	dc1	dc1	dc1	Grafik
18	22	dc2	dc2	dc2	Grafik
19	23	dc3	dc3	dc3	Grafik
20	24	res	dc4	dc4	Grafik
21	25	nl	nak	nak	Grafik
22	26	bs	syn	syn	Grafik
23	27	il	etb	etb	Grafik
24	30	can	can	can	Grafik
25	31	em	em	em	Grafik
26	32	cc	sub	sub	Grafik

27	33		esc	esc	Grafik
28	34	ifs	fs	fs	cur right
29	35	igs	gs	gs	cur left
30	36	irs	rs	rs	cur up
31	37	ius	us	us	cur down
32	40	ds	space	space	space
33	41	sos	!	!	!
34	42	fs	”	”	”
35	43		#	#	#
36	44	byp	\$	\$	\$
37	45	lf	%	%	%
38	46	etb	&	&	&
39	47	esc	,	,	,
40	50		(((
41	51)))
42	52	sm	*	*	*
43	53		+	+	+
44	54		,	,	,
45	55	enq	-	-	-
46	56	ack	.	.	.
47	57	bel	/	/	/
48	60		0	0	0
49	61		1	1	1
50	62	syn	2	2	2
51	63		3	3	3
52	64	pn	4	4	4
53	65	rs	5	5	5
54	66	uc	6	6	6
55	67	eot	7	7	7
56	70		8	8	8
57	71		9	9	9
58	72		:	:	:
59	73		;	;	;
60	74	dc4	<	<	<
61	75	nak	=	=	=
62	76		>	>	>
63	77	sub	?	?	?
64	100	space	@	@	@
65	101		A	A	A
66	102	â	B	B	B
67	103	ä	C	C	C
68	104	à	D	D	D
69	105	á	E	E	E
70	106	ã	F	F	F
71	107	å	G	G	G
72	110	ç	H	H	H
73	111	ñ	I	I	I
74	112	[J	J	J
75	113	.	K	K	K

76	114	<	L	L	L
77	115	(M	M	M
78	116	+	N	N	N
79	117	!	O	O	O
80	120	&	P	P	P
81	121	é	Q	Q	Q
82	122	ê	R	R	R
83	123	ë	S	S	S
84	124	è	T	T	T
85	125	í	U	U	U
86	126	î	V	V	V
87	127	ï	W	W	W
88	130	ì	X	X	X
89	131	ß	Y	Y	Y
90	132]	Z	Z	Z
91	133	\$	[[[
92	134	*	\	\	\
93	135)]]]
94	136	;	^	^	^
95	137	^	‘	‘	‘
96	140	—			
97	141	/	a	a	a
98	142	Â	b	b	b
99	143	Ã	c	c	c
100	144	À	d	d	d
101	145	Á	e	e	e
102	146	Ã	f	f	f
103	147	Å	g	g	g
104	150	Ç	h	h	h
105	151	Ñ	i	i	i
106	152		j	j	j
107	153	,	k	k	k
108	154	%	l	l	l
109	155	—	m	m	m
110	156	>	n	n	n
111	157	?	o	o	o
112	160	ø	p	p	p
113	161	É	q	q	q
114	162	Ê	r	r	r
115	163	Ë	s	s	s
116	164	È	t	t	t
117	165	Í	u	u	u
118	166	Î	v	v	v
119	167	Ï	w	w	w
120	170	Ì	x	x	x
121	171	‘	y	y	y
122	172	:	z	z	z
123	173	#	{	{	{

124	174	@			
125	175	'	}	}	}
126	176	=	~	~	~
127	177	”	del	del	Grafik
128	200	Ø			Ç
129	201	a			ü
130	202	b			é
131	203	c			â
132	204	d			ä
133	205	e			à
134	206	f			å
135	207	g			ç
136	210	h			ê
137	211	i			ë
138	212	«			è
139	213	»			ı
140	214				î
141	215	ý			ì
142	216				Ä
143	217	±			Å
144	220				É
145	221	j			œ
146	222	k			Æ
147	223	l			ô
148	224	m			ö
149	225	n			ò
150	226	o			û
151	227	p			ù
152	230	q			y
153	231	r			Ö
154	232	<u>a</u>			Ü
155	233	<u>o</u>			
156	234	æ			£
157	235	—			Yen
158	236	Æ			Pt
159	237				f
160	240	μ			á
161	241	~		À	í
162	242	s		Â	ó
163	243	t		È	ú
164	244	u		Ê	ñ
165	245	v		Ë	Ñ
166	246	w		Î	<u>a</u>
167	247	x		Ï	<u>o</u>
168	250	y		,	ı
169	251	z		‘	Grafik
170	252	i		^	Grafik
171	253	ı			1/2

172	254		~	1/4
173	255	Ý	Û	ı
174	256		Ũ	«
175	257			»
176	260			Grafik
177	261	£		Grafik
178	262	Yen		Grafik
179	263		◦	Grafik
180	264	f	Ç	Grafik
181	265	§	ç	Grafik
182	266	¶	Ñ	Grafik
183	267		ñ	Grafik
184	270		ı	Grafik
185	271		ı	Grafik
186	272			Grafik
187	273		£	Grafik
188	274	–	Yen	Grafik
189	275		§	Grafik
190	276		f	Grafik
191	277	=		Grafik
192	300	{	â	Grafik
193	301	A	ê	Grafik
194	302	B	ô	Grafik
195	303	C	û	Grafik
196	304	D	á	Grafik
197	305	E	é	Grafik
198	306	F	ó	Grafik
199	307	G	ú	Grafik
200	310	H	à	Grafik
201	311	I	è	Grafik
202	312		ò	Grafik
203	313	ô	ù	Grafik
204	314	ö	ä	Grafik
205	315	ò	ë	Grafik
206	316	ó	ö	Grafik
207	317	õ	ü	Grafik
208	320	}	Å	Grafik
209	321	J	î	Grafik
210	322	K	Ø	Grafik
211	323	L	Æ	Grafik
212	324	M	å	Grafik
213	325	N	í	Grafik
214	326	O	ø	Grafik
215	327	P	æ	Grafik
216	330	Q	Ä	Grafik
217	331	R	ì	Grafik
218	332		Ö	Grafik
219	333	û	Ü	Grafik
220	334	ü	É	Grafik

221	335	ù	ı	Grafik
222	336	ú	ß	Grafik
223	337	y	Ô	Grafik
224	340	\	Á	α
225	341		Ã	β
226	342	S	ã	Γ
227	343	T		π
228	344	U		Σ
229	345	V	Í	σ
230	346	W	Ì	μ
231	347	X	Ó	τ
232	350	Y	Ò	Φ
233	351	Z	Õ	θ
234	352		õ	Ω
235	353	Ô	Š	δ
236	354	Ö	š	∞
237	355	Ò	Ú	Ø
238	356	Ó	Y	€
239	357	Õ	y	∩
240	360	0	thorn	≡
241	361	1	Thorn	±
242	362	2		≥
243	363	3		≤
244	364	4		Haken
245	365	5		Haken
246	366	6	–	÷
247	367	7	1/4	≈
248	370	8	1/2	◦
249	371	9	ª	•
250	372		º	·
251	373	Û	≪	√
252	374	Ü	⊐	n
253	375	Ù	≫	2
254	376	Ú	±	⊐
255	377			(FF)

B.2 German-ASCII

Falls das Ein- oder Ausgabegerät einen deutschen 7-Bit-ASCII-Zeichensatz enthält, sind folgende Ersetzungen der amerikanischen Zeichen durch deutsche Sonderzeichen üblich:

Nr.	US-Zeichen	US-ASCII	German ASCII
91	linke eckige Klammer	[Ä
92	Backslash	\	Ö
93	rechte eckige Klammer]	Ü

123	linke geschweifte Klammer	{	ä
124	senkrechter Strich		ö
125	rechte geschweifte Klammer	}	ü
126	Tilde	~	ß

Achtung: Der IBM-PC und Ausgabegeräte von Hewlett-Packard verwenden keinen 7-Bit-ASCII-Zeichensatz, sondern eigene 8-Bit-Zeichensätze, die die Sonderzeichen unter Nummern höher 127 enthalten, siehe vorhergehende Tabelle.

B.3 ASCII-Steuerzeichen

Die Steuerzeichen der Zeichensätze dienen der Übermittlung von Befehlen und Informationen an das empfangende Gerät und nicht der Ausgabe eines sicht- oder druckbaren Zeichens. Die Ausgabegeräte kennen in der Regel jedoch einen Modus (transparent, Monitor, Display Functions), in der die Steuerzeichen nicht ausgeführt, sondern angezeigt werden. Die meisten Steuerzeichen belegen keine eigene Taste auf der Tastatur, sondern werden als Kombination aus der <ctrl>-Taste und einer Zeichentaste eingegeben. In C/C++ läßt sich jedes Zeichen durch seine oktale Nummer in der Form \123 oder durch seine hexadezimale Nummer in der Form \x53 eingeben (hier das S).

dezimal	C-Konst.	ASCII	Bedeutung	Tasten
0	\x00	nul	ASCII-Null	control @
1		soh	Start of heading	control a
2		stx	Start of text	control b
3		etx	End of text	control c
4		eot	End of transmission	control d
5		enq	Enquiry	control e
6		ack	Acknowledge	control f
7	\a	bel	Bell	control g
8	\b	bs	Backspace	control h, BS
9	\t	ht	Horizontal tab	control i, TAB
10	\n	lf	Line feed	control j, LF
11	\v	vt	Vertical tab	control k
12	\f	ff	Form feed	control l
13	\r	cr	Carriage return	control m, RETURN
14		so	Shift out	control n
15		si	Shift in	control o
16		dle	Data link escape	control p
17		dc1	Device control 1, xon	control q
18		dc2	Device control 2, tape	control r
19		dc3	Device control 3, xoff	control s
20		dc4	Device control 4, tape	control t
21		nak	Negative acknowledge	control u
22		syn	Synchronous idle	control v
23		etb	End transmission block	control w
24		can	Cancel	control x

25		em	End of medium	control y
26		sub	Substitute	control z
27	\x1b	esc	Escape	control [, ESC
28		fs	File separator	control \
29		gs	Group separator	control]
30		rs	Record separator	control ^
31		us	Unit separator	control _
127		del	Delete	DEL, RUBOUT

B.4 Latin-1 (ISO 8859-1), Unicode

Die Norm ISO 8859 beschreibt zehn Zeichensätze, die jedes Zeichen durch jeweils ein Byte darstellen. Jeder Zeichensatz umfaßt also maximal 256 druckbare Zeichen und Steuerzeichen. Der erste – Latin-1 genannt – ist für west- und mitteleuropäische Sprachen – darunter Deutsch – vorgesehen. Latin-2 deckt Mittel- und Osteuropa ab, soweit das lateinische Alphabet verwendet wird. Wer einen polnisch-deutschen Text schreiben will, braucht Latin 2. Die deutschen Sonderzeichen liegen in Latin 1 bis 6 an denselben Stellen. Weiteres siehe in der ISO-Norm und im RFC 1345 *Character Mnemonics and Character Sets* vom Juni 1992. Auch <http://wwwws.cs.tu-berlin.de/~czyborra/charsets/> hilft weiter.

Die erste Hälfte (0 – 127) aller Latin-Zeichensätze stimmt mit US-ASCII überein. Unicode, UCS oder ISO/IEC 10646 stimmt in den untersten 256 Positionen mit Latin-1 überein. Zu jedem Zeichen gehört eine standardisierte verbale Bezeichnung. Einige Zeichen wie das isländische Thorn oder das Cent-Zeichen konnten hier nicht dargestellt werden.

dezimal	oktal	hex	Zeichen	Bezeichnung
000	000	00	nu	Null (nul)
001	001	01	sh	Start of heading (soh)
002	002	02	sx	Start of text (stx)
003	003	03	ex	End of text (etx)
004	004	04	et	End of transmission (eot)
005	005	05	eq	Enquiry (enq)
006	006	06	ak	Acknowledge (ack)
007	007	07	bl	Bell (bel)
008	010	08	bs	Backspace (bs)
009	011	09	ht	Character tabulation (ht)
010	012	0a	lf	Line feed (lf)
011	013	0b	vt	Line tabulation (vt)
012	014	0c	ff	Form feed (ff)
013	015	0d	cr	Carriage return (cr)
014	016	0e	so	Shift out (so)
015	017	0f	si	Shift in (si)
016	020	10	dl	Datalink escape (dle)

017	021	11	d1	Device control one (dc1)
018	022	12	d2	Device control two (dc2)
019	023	13	d3	Device control three (dc3)
020	024	14	d4	Device control four (dc4)
021	025	15	nk	Negative acknowledge (nak)
022	026	16	sy	Synchronous idle (syn)
023	027	17	eb	End of transmission block (etb)
024	030	18	cn	Cancel (can)
025	031	19	em	End of medium (em)
026	032	1a	sb	Substitute (sub)
027	033	1b	ec	Escape (esc)
028	034	1c	fs	File separator (is4)
029	035	1d	gs	Group separator (is3)
030	036	1e	rs	Record separator (is2)
031	037	1f	us	Unit separator (is1)
032	040	20	sp	Space
033	041	21	!	Exclamation mark
034	042	22	”	Quotation mark
035	043	23	#	Number sign
036	044	24	\$	Dollar sign
037	045	25	%	Percent sign
038	046	26	&	Ampersand
039	047	27	,	Apostrophe
040	050	28	(Left parenthesis
041	051	29)	Right parenthesis
042	052	2a	*	Asterisk
043	053	2b	+	Plus sign
044	054	2c	,	Comma
045	055	2d	-	Hyphen-Minus
046	056	2e	.	Full stop
047	057	2f	/	Solidus
048	060	30	0	Digit zero
049	061	31	1	Digit one
050	062	32	2	Digit two
051	063	33	3	Digit three
052	064	34	4	Digit four
053	065	35	5	Digit five
054	066	36	6	Digit six
055	067	37	7	Digit seven
056	070	38	8	Digit eight
057	071	39	9	Digit nine
058	072	3a	:	Colon
059	073	3b	;	Semicolon
060	074	3c	<	Less-than sign
061	075	3d	=	Equals sign
062	076	3e	>	Greater-than sign
063	077	3f	?	Question mark
064	100	40	@	Commercial at
065	101	41	A	Latin capital letter a

066	102	42	B	Latin capital letter b
067	103	43	C	Latin capital letter c
068	104	44	D	Latin capital letter d
069	105	45	E	Latin capital letter e
070	106	46	F	Latin capital letter f
071	107	47	G	Latin capital letter g
072	110	48	H	Latin capital letter h
073	111	49	I	Latin capital letter i
074	112	4a	J	Latin capital letter j
075	113	4b	K	Latin capital letter k
076	114	4c	L	Latin capital letter l
077	115	4d	M	Latin capital letter m
078	116	4e	N	Latin capital letter n
079	117	4f	O	Latin capital letter o
080	120	50	P	Latin capital letter p
081	121	51	Q	Latin capital letter q
082	122	52	R	Latin capital letter r
083	123	53	S	Latin capital letter s
084	124	54	T	Latin capital letter t
085	125	55	U	Latin capital letter u
086	126	56	V	Latin capital letter v
087	127	57	W	Latin capital letter w
088	130	58	X	Latin capital letter x
089	131	59	Y	Latin capital letter y
090	132	5a	Z	Latin capital letter z
091	133	5b	[Left square bracket
092	134	5c	\	Reverse solidus
093	135	5d]	Right square bracket
094	136	5e	^	Circumflex accent
095	137	5f	_	Low line
096	140	60	`	Grave accent
097	141	61	a	Latin small letter a
098	142	62	b	Latin small letter b
099	143	63	c	Latin small letter c
100	144	64	d	Latin small letter d
101	145	65	e	Latin small letter e
102	146	66	f	Latin small letter f
103	147	67	g	Latin small letter g
104	150	68	h	Latin small letter h
105	151	69	i	Latin small letter i
106	152	6a	j	Latin small letter j
107	153	6b	k	Latin small letter k
108	154	6c	l	Latin small letter l
109	155	6d	m	Latin small letter m
110	156	6e	n	Latin small letter n
111	157	6f	o	Latin small letter o
112	160	70	p	Latin small letter p
113	161	71	q	Latin small letter q
114	162	72	r	Latin small letter r

115	163	73	s	Latin small letter s
116	164	74	t	Latin small letter t
117	165	75	u	Latin small letter u
118	166	76	v	Latin small letter v
119	167	77	w	Latin small letter w
120	170	78	x	Latin small letter x
121	171	79	y	Latin small letter y
122	172	7a	z	Latin small letter z
123	173	7b	{	Left curly bracket
124	174	7c		Vertical line
125	175	7d	}	Right curly bracket
126	176	7e	~	Tilde
127	177	7f	dt	Delete (del)
128	200	80	pa	Padding character (pad)
129	201	81	ho	High octet preset (hop)
130	202	82	bh	Break permitted here (bph)
131	203	83	nh	No break here (nbh)
132	204	84	in	Index (ind)
133	205	85	nl	Next line (nel)
134	206	86	sa	Start of selected area (ssa)
135	207	87	es	End of selected area (esa)
136	210	88	hs	Character tabulation set (hts)
137	211	89	hj	Character tabulation with justification (htj)
138	212	8a	vs	Line tabulation set (vts)
139	213	8b	pd	Partial line forward (pld)
140	214	8c	pu	Partial line backward (plu)
141	215	8d	ri	Reverse line feed (ri)
142	216	8e	s2	Single-shift two (ss2)
143	217	8f	s3	Single-shift three (ss3)
144	220	90	dc	Device control string (dcs)
145	221	91	p1	Private use one (pu1)
146	222	92	p2	Private use two (pu2)
147	223	93	ts	Set transmit state (sts)
148	224	94	cc	Cancel character (cch)
149	225	95	mw	Message waiting (mw)
150	226	96	sg	Start of guarded area (spa)
151	227	97	eg	End of guarded area (epa)
152	230	98	ss	Start of string (sos)
153	231	99	gc	Single graphic character introducer (sgci)
154	232	9a	sc	Single character introducer (sci)
155	233	9b	ci	Control sequence introducer (csi)
156	234	9c	st	String terminator (st)
157	235	9d	oc	Operating system command (osc)
158	236	9e	pm	Privacy message (pm)
159	237	9f	ac	Application program command (apc)
160	240	a0	ns	No-break space
161	241	a1	¡	Inverted exclamation mark
162	242	a2	¢	Cent sign
163	243	a3	£	Pound sign

164	244	a4		Currency sign (künftig Euro?)
165	245	a5		Yen sign
166	246	a6		Broken bar
167	247	a7	§	Section sign
168	250	a8		Diaresis
169	251	a9	©	Copyright sign
170	252	aa	^a	Feminine ordinal indicator
171	253	ab	◀◀	Left-pointing double angle quotation mark
172	254	ac	¬	Not sign
173	255	ad	-	Soft hyphen
174	256	ae		Registered sign
175	257	af	—	Overline
176	260	b0	°	Degree sign
177	261	b1	±	Plus-minus sign
178	262	b2	²	Superscript two
179	263	b3	³	Superscript three
180	264	b4	’	Acute accent
181	265	b5	μ	Micro sign
182	266	b6	¶	Pilcrow sign
183	267	b7	·	Middle dot
184	270	b8		Cedilla
185	271	b9	¹	Superscript one
186	272	ba	◊	Masculine ordinal indicator
187	273	bb	≫	Right-pointing double angle quotation mark
188	274	bc	1/4	Vulgar fraction one quarter
189	275	bd	1/2	Vulgar fraction one half
190	276	be	3/4	Vulgar fraction three quarters
191	277	bf	¿	Inverted question mark
192	300	c0	À	Latin capital letter a with grave
193	301	c1	Á	Latin capital letter a with acute
194	302	c2	Â	Latin capital letter a with circumflex
195	303	c3	Ã	Latin capital letter a with tilde
196	304	c4	Ä	Latin capital letter a with diaresis
197	305	c5	Å	Latin capital letter a with ring above
198	306	c6	Æ	Latin capital letter ae
199	307	c7	Ç	Latin capital letter c with cedilla
200	310	c8	È	Latin capital letter e with grave
201	311	c9	É	Latin capital letter e with acute
202	312	ca	Ê	Latin capital letter e with circumflex
203	313	cb	Ë	Latin capital letter e with diaresis
204	314	cc	Ì	Latin capital letter i with grave
205	315	cd	Í	Latin capital letter i with acute
206	316	ce	Î	Latin capital letter i with circumflex
207	317	cf	Ï	Latin capital letter i with diaresis
208	320	d0		Latin capital letter eth (Icelandic)
209	321	d1	Ñ	Latin capital letter n with tilde
210	322	d2	Ò	Latin capital letter o with grave
211	323	d3	Ó	Latin capital letter o with acute

212	324	d4	Ô	Latin capital letter o with circumflex
213	325	d5	Ö	Latin capital letter o with tilde
214	326	d6	Ö	Latin capital letter o with diaeresis
215	327	d7	×	Multiplication sign
216	330	d8	Ø	Latin capital letter o with stroke
217	331	d9	Û	Latin capital letter u with grave
218	332	da	Ú	Latin capital letter u with acute
219	333	db	Û	Latin capital letter u with circumflex
220	334	dc	Ü	Latin capital letter u with diaeresis
221	335	dd	Ý	Latin capital letter y with acute
222	336	de		Latin capital letter thorn (Icelandic)
223	337	df	ß	Latin small letter sharp s (German)
224	340	e0	à	Latin small letter a with grave
225	341	e1	á	Latin small letter a with acute
226	342	e2	â	Latin small letter a with circumflex
227	343	e3	ã	Latin small letter a with tilde
228	344	e4	ä	Latin small letter a with diaeresis
229	345	e5	å	Latin small letter a with ring above
230	346	e6	æ	Latin small letter æ
231	347	e7	ç	Latin small letter c with cedilla
232	350	e8	è	Latin small letter e with grave
233	351	e9	é	Latin small letter e with acute
234	352	ea	ê	Latin small letter e with circumflex
235	353	eb	ë	Latin small letter e with diaeresis
236	354	ec	ì	Latin small letter i with grave
237	355	ed	í	Latin small letter i with acute
238	356	ee	î	Latin small letter i with circumflex
239	357	ef	ï	Latin small letter i with diaeresis
240	360	f0		Latin small letter eth (Icelandic)
241	361	f1	ñ	Latin small letter n with tilde
242	362	f2	ò	Latin small letter o with grave
243	363	f3	ó	Latin small letter o with acute
244	364	f4	ô	Latin small letter o with circumflex
245	365	f5	õ	Latin small letter o with tilde
246	366	f6	ö	Latin small letter o with diaeresis
247	367	f7	÷	Division sign
248	370	f8	ø	Latin small letter o with stroke
249	371	f9	ù	Latin small letter u with grave
250	372	fa	ú	Latin small letter u with acute
251	373	fb	û	Latin small letter u with circumflex
252	374	fc	ü	Latin small letter u with diaeresis
253	375	fd	ý	Latin small letter y with acute
254	376	fe		Latin small letter thorn (Icelandic)
255	377	ff	ÿ	Latin small letter y with diaeresis

B.5 Latin-2 (ISO 8859-2)

Der Zeichensatz Latin-2 deckt folgende Sprachen ab: Albanisch, Bosnisch, Deutsch, Englisch, Finnisch, Irisch, Kroatisch, Polnisch, Rumänisch, Serbisch (in lateinischer Transskription), Serbokroatisch, Slowakisch, Slowenisch, Sorbisch, Tschechisch und Ungarisch. Samisch wird in Latin-9 berücksichtigt. Auf: <http://sizif.mf.uni-lj.si/linux/cee/iso8859-2.html> finden sich Einzelheiten und weitere URLs. Hier nur die Zeichen, die von Latin-1 abweichen:

dezimal	oktal	hex	Zeichen	Bezeichnung
161	241	a1		Latin capital letter a with ogonek
162	242	a2		Breve
163	243	a3	Ł	Latin capital letter l with stroke
165	245	a5	Ł̇	Latin capital letter l with caron
166	246	a6	Ś	Latin capital letter s with acute
169	251	a9	Ṥ	Latin capital letter s with caron
170	252	aa	Ş	Latin capital letter s with cedilla
171	253	ab	Ť	Latin capital letter t with caron
172	254	ac	Ž	Latin capital letter z with acute
174	256	ae	Ž̇	Latin capital letter z with caron
175	257	af	Ž̈	Latin capital letter z with dot above
177	261	b1		Latin small letter a with ogonek
178	262	b2		Ogonek (Schwänzchen)
179	263	b3	ł	Latin small letter l with stroke
181	265	b5		Latin small letter l with caron
182	266	b6		Latin small letter s with acute
183	267	b7		Caron
185	271	b9		Latin small letter s with caron
186	272	ba		Latin small letter s with cedilla
187	273	bb		Latin small letter t with caron
188	274	bc		Latin small letter z with acute
189	275	bd		Double acute accent
190	276	be		Latin small letter z with caron
191	277	bf		Latin small letter z with dot above
192	300	c0		Latin capital letter r with acute
195	303	c3		Latin capital letter a with breve
197	305	c5		Latin capital letter l with acute
198	306	c6		Latin capital letter c with acute
200	310	c8		Latin capital letter c with caron
202	312	ca		Latin capital letter e with ogonek
204	314	cc		Latin capital letter e with caron
207	317	cf		Latin capital letter d with caron
208	320	d0		Latin capital letter d with stroke
209	321	d1		Latin capital letter n with acute

210	322	d2	Latin capital letter n with caron
213	325	d5	Latin capital letter o with double acute
216	330	d8	Latin capital letter r with caron
217	331	d9	Latin capital letter u with ring above
219	333	db	Latin capital letter u with double acute
222	336	de	Latin capital letter t with cedilla
224	340	e0	Latin small letter r with acute
227	343	e3	Latin small letter a with breve
229	345	e5	Latin small letter l with acute
230	346	e6	Latin small letter c with acute
232	350	e8	Latin small letter c with caron
234	352	ea	Latin small letter e with ogonek
236	354	ec	Latin small letter e with caron
239	357	ef	Latin small letter d with caron
240	360	f0	Latin small letter d with stroke
241	361	f1	Latin small letter n with acute
242	362	f2	Latin small letter n with caron
245	365	f5	Latin small letter o with double acute
248	370	f8	Latin small letter r with caron
249	371	f9	Latin small letter u with ring above
251	373	fb	Latin small letter u with double acute
254	376	fe	Latin small letter t with cedilla
255	377	ff	Dot above

B.6 HTML-Entities

HTML-Entities sind eine Ersatzschreibweise für Zeichen, die nicht direkt in HTML-Text eingegeben werden können. Zu diesen Zeichen gehören:

- Sonderzeichen außerhalb des US-ASCII-Zeichensatzes (Umlaute),
- Zeichen, die eine besondere Bedeutung in HTML haben (&, <),
- mathematische und andere Symbole (\pm , \copyright).

Für den Ersatz gibt es zwei Möglichkeiten:

- die dezimale oder hexadezimale Nummer des Zeichens im Zeichensatz,
- eine Umschreibung mit ASCII-Zeichen.

Soweit die Zeichen im Latin-1-Zeichensatz enthalten sind, können die dort angegebenen Nummern verwendet werden. Die vollständige Tabelle entnimmt man am einfachsten der HTML-Spezifikation. Hier nur die häufigsten Zeichen:

dezimal	hex	char ent	Zeichen	Bezeichnung
&	&	&	&	ampersand
<	L	<	<	less-than sign

>	N	>	>	greater-than sign
 	 	 		non-breaking space
¡	¡	¡	¡	inverted exclamation mark
£	£	£	£	pound sign
©	©	©	©	copyright sign
«	«	«	«	left pointing guillemet
­	­	­		soft or dicretionary hyphen
®	®	®		registered sign
°	°	°	°	degree sign
±	±	±	±	plus-minus sign
²	²	²	²	superscript two
³	³	³	³	superscript three
µ	µ	µ	μ	micro sign
»	»	»	»	right pointing guillemet
½	½	½	1/2	fraction one half
À	À	À	À	latin capital letter A with grave
Á	Á	Á	Á	latin capital letter A with acute
Â	Â	Â	Â	latin capital letter A with circumflex
Ã	Ã	Ã	Ã	latin capital letter A with tilde
Ä	Ä	Ä	Ä	latin capital letter A with diaeresis
Å	Å	Å	Å	latin capital letter A with ring above
Æ	Æ	Æ	Æ	latin capital ligature AE
Ç	Ç	Ç	Ç	latin capital letter C with cedilla
Ñ	Ñ	Ñ	Ñ	latin capital letter N with tilde
Ø	Ø	Ø	Ø	latin capital letter O with stroke
ß	ß	ß	ß	latin small letter sharp s
ë	ë	ë	ë	latin small letter e with diaeresis
ñ	ð	ñ	ñ	latin small letter n with tilde
€	&#x ;	€	€	euro sign

B.7 Sondertasten

Computertastaturen weisen im Vergleich zu den Tastaturen herkömmlicher Schreibmaschinen zusätzliche Tasten (E: key, F: touche) auf. Obwohl die PC-Tastatur (MF-2) weit verbreitet ist, gibt es eine Vielzahl abweichender Tastaturen. Die Wirkung, die von einer Taste ausgelöst wird, hängt von der Software ab und lässt sich daher nicht allgemein angeben. Typische Beispiele sind die Funktionstasten, aber auch fast alle anderen Tasten lassen sich umprogrammieren. In der folgenden Tabelle werden die wichtigsten Sondertasten aufgeführt:

	DÜZ	Datenübertragung zeilenweise
ACK		Acknowledge
AlphaLock		wie Caps Lock

Alt	Alt	Alternate, Meta
AltGr	AltGr	Alternate Graphic(s)
BackTab		
BS, Backspace	Rücktaste	einen Schritt zurück
BEL		Glocke, ASCII-Nr. 7
Break	Untbr	Signal SIGINT, Signal Nr. 2
CAN		Cancel
Caps Lock	Feststelltaste	Umschaltung feststellen
CBT		Cursor back tab
CE, ClearEntry		
CharDel		Lösche Zeichen
CharInsert		Füge Zeichen vor Cursor ein
ClearSpace		
CR		Carriage Return, Wagenrücklauf, ASCII-Nr. 13
Ctrl, Ctl	Strg	Control, Steuerung
CUB		Cursor back
CUD		Cursor down
CUF		Cursor forward
CUU		Cursor up
DCH		Delete character
DC1		Device control 1, ASCII-Nr. 17
DC2		Device control 2, ASCII-Nr. 18
DC3		Device control 3, ASCII-Nr. 19
DC4		Device control 4, ASCII-Nr. 20
Dead Keys	Tod-Tasten, Akzent-Tasten	T. für diakritische Zeichen, Akzente <i>ohne</i> Zeichenvorschub
DEL, Delete	Entf	Zeichen entfernen, löschen
DL		Delete line
DLE		Data link enable
Down	Pfeil abwärts	Cursor abwärts
Edit		siehe Alt
EM		End of media
End	Ende	Cursor zum Seitenende
ENQ		Enquiry
Enter	Zeilenwechsel	Eingabe vollziehen
EOT		End of transmission
Esc	Escape	Escape, Fluchtsymbol, ASCII-Nr. 27
ETB		End of transaction block
ETX		End of text
ExtChar		
F1	F1	Funktionstaste 1 usw.
FF		Form Feed, Blattvorschub, ASCII-Nr. 12
Funct		ähnlich wie Ctrl
Home	Pos1	Cursor zum Seitenanfang

HT		Horizontal tab
INS, Insert	Einfüg	Zeichen vor Cursor einfügen
IS1		Information separator 1
IS2		Information separator 2
IS3		Information separator 3
IS4		Information separator 4
Left	Pfeil links	Cursor nach links
LF		Line feed, ASCII-Nr. 10
LineDel		Lösche Zeile
LineErase		Lösche Zeile ab Cursor
LineFeed		wie Down
LineIns		Füge Zeile oberhalb Cursor ein
LocEsc		Local Escape
Meta		siehe Alt
NAK		Not acknowledge
NoScroll		Hält Bildschirm an
NumLock	Num	Number lock key
Page		Nächste Seite des Speichers
Page Down	Bild abwärts	Nächste Seite des Speichers
PageErase		Seite löschen
Page Up	Bild aufwärts	Vorige Seite des Speichers
Pause	Pause	Pause
PFK		Program Function Key
PrevWin		Previous window
PrintScreen	Druck	Hardcopy des Bildschirms
Reset		Setzt Terminal zurück
Return	Zeilenwechsel	Zeilenwechsel, Eingabe vollziehen
Right	Pfeil rechts	Cursor nach rechts
Rubout		Zeichen löschen, wie DEL
ScrollLock	Rollen	Hält Bildschirm an
Send		Sende Seite zum Computer
Setup		Zeigt Konfiguration an
Shift	Umschaltung	Umschaltung groß-klein
Space	Leertaste	ASCII-Nr. 32, Leerschritt
SOH		Start of header
SUB		Substitute character, ASCII-Nr. 26
SYN		Synch idle
SysRq	S-Abf	System request, System-Anfrage
Tab	Tab	Horizontal-Tabulator
Up	Pfeil aufwärts	Cursor nach oben

Daneben finden sich auf den Tastaturen einzelner Hersteller noch Tasten mit abgebissenen Äpfeln, fliegenden Fenstern, einem fröhlichen Pinguin und dergleichen. Multimediatastaturen verfügen über Tasten zum Einstellen der Lautstärke usw.

C

Metazeichen

C.1 Shell

Metazeichen (E: special character) sind Zeichen, die in bestimmten Zusammenhängen eine besondere Bedeutung erlangen. Sollen sie in einem solchen Zusammenhang als gewöhnliches Zeichen – buchstäblich – aufgefasst werden, sind sie zu quoten (maskieren, sperren, entwerten, zitieren). Zum Quoten – siehe Abschnitt 2.1.1 *Gesprächspartenr* auf Seite 53 – dienen der Gegenschrägstrich, Gänsefüßchen und Hochkommas (Apostrophe).

In Datei- oder Verzeichnisnamen ersetzt (expandiert) die Bash folgende Metazeichen (Jokerzeichen):

- Fragezeichen: genau ein beliebiges Zeichen,
- Stern: eine beliebige Anzahl beliebiger Zeichen,
- Tilde: zu Beginn eines Pfades dasselbe wie `$HOME`, das Home-Verzeichnis.

Als Trennzeichen (die nicht innerhalb eines Wortes verwendet werden dürfen) kennt die Bash:

- Leerzeichen, Tabulator,
- senkrechter Strich,
- `&`, et-Zeichen,
- Semikolon,
- runde Klammern,
- spitze Klammern (kleiner/größer).

Als Steuerzeichen (control operators), die eine Tätigkeit der Bash auslösen, werden verwendet:

- Zeilenwechsel (newline),
- senkrechter Strich.
- ein Paar senkrechter Striche,
- `&`, et-Zeichen,
- ein Paar von et-Zeichen,

- Semikolon,
- ein Paar von Semikolons,
- runde Klammern.

In Kommandos oder Shellscripts kennt die Bash noch folgende Metazeichen:

- Doppelkreuz: Kommentar bis Zeilenende,
- Dollarzeichen: Ersetzen von Variablen durch ihren Wert, z. B. \$TERM.

Darüber hinaus kennen Shells reservierte Wörter wie `if`, `fi`, `select`, `time`, `while` usw. Zu diesen rechnen auch das Befehlszeichen (Negation), geschweifte Klammern sowie doppelte eckige Klammern.

C.2 LaTeX

In \LaTeX -Quelltexten haben folgende Zeichen eine tiefere Bedeutung:

- Doppelkreuz `#`: Ersetzungszeichen bei der Parameterübergabe, buchstäbliche Eingabe durch `\#` oder `\symbol{35}`,
- Dollar `$`: Umschalten vom Textmodus in den mathematischen Modus und zurück, buchstäbliche Eingabe durch `\$` oder `\symbol{36}`,
- Ampersand `&`: Spaltentrenner in Tabellen, buchstäbliche Eingabe durch `\&` oder `\symbol{38}`,
- Tilde `~`: Zwischenraum (non-breaking space), buchstäbliche Eingabe nur durch `\symbol{126}`,
- Unterstrich `_`: Tiefstellung in Formeln, buchstäbliche Eingabe durch `_` oder `\symbol{95}`,
- Circumflex `^`: Hochstellung in Formeln, buchstäbliche Eingabe nur durch `\symbol{94}`,
- Prozentzeichen `%`: Zeilenkommentar, buchstäbliche Eingabe durch `\%` oder `\symbol{37}`,
- geschweifte Klammern `{}`: rahmen Parameter von \LaTeX -Befehlen ein, buchstäbliche Eingabe durch `\}` oder `\symbol{123}` und entsprechend,
- Gegenschrägstrich (backslash): quotet die meisten Metazeichen, buchstäbliche Eingabe nur durch `\symbol{92}`.

Grundsätzlich kann jedes überhaupt verfügbare Zeichen in der Form `\symbol{n}` eingegeben werden, wobei `n` die dezimale Nummer des Zeichens im Zeichensatz ist. Für häufig buchstäblich verwendete Metazeichen legt man sich zweckmäßig eigene, kurze \LaTeX -Befehle an, beispielsweise:

```
\newcommand{\bsl}{\symbol{92}}
\newcommand{\cfl}{\symbol{94}}
\newcommand{\til}{\symbol{126}}
```

Natürlich darf man dabei keine bereits vergebenen Kommandonamen verwenden.

C.3 Reguläre Ausdrücke

Innerhalb regulärer Ausdrücke sind folgende Metazeichen gebräuchlich:

- Stern: eine beliebige Anzahl (auch null) des vorangehenden Zeichens (Atoms),
- Pluszeichen: mindestens einmaliges Vorkommen des vorangehenden Zeichens (Atoms),
- Fragezeichen: null- oder einmaliges Vorkommen des vorangehenden Zeichens (Atoms),
- Punkt: genau ein beliebiges Zeichen.

Dazu kommen runde, eckige und geschweifte Klammern, Zirkumflex, Dollar, Backslash und senkrechter Strich.

C.4 make

In Makefiles werden folgende Metazeichen verwendet:

- Doppelkreuz: Zeilenkommentar,
- Tabulator: leitet eine Kommandozeile ein,
- Doppelpunkt: kennzeichnet eine Zeile mit Abhängigkeiten, ein Ziel,
- Gleichheitszeichen: kennzeichnet eine Zeile mit einer Makrodefinition,
- Dollarzeichen: ersetzt (expandiert) ein Makro

sowie runde und geschweifte Klammern. Da die Kommandozeilen von einer Shell ausgeführt werden, können dort Metazeichen der Shell vorkommen, insbesondere die Jokerzeichen in Dateinamen.

C.5 HTML

In HTML-Dokumenten (Webseiten) haben folgende Zeichen eine besondere Bedeutung:

- kleiner als `<`, leitet HTML-Anweisung ein, buchstäbliche Eingabe mittels HTML-Entity `<` oder `<`;
- größer als `>`, beendet HTML-Anweisung, buchstäbliche Eingabe mittels HTML-Entity `>` oder `>`;
- et-Zeichen (ampersand) `&`, leitet HTML-Entity ein, Eingabe mittels HTML-Entity `&` oder `&`;

Auch alle anderen Zeichen – soweit überhaupt verfügbar – können in obiger Form mit Hilfe ihrer dezimalen Nummer im Zeichensatz dargestellt werden.

D

Papier- und Schriftgrößen

D.1 Papierformate

In der EDV sind folgende Papierformate gebräuchlich (1 Zoll (inch) = 25,4 mm oder 72 Punkte):

- **DIN/ISO A2** Blattgröße 420 mm x 594 mm entsprechend 1190 p x 1684 p.
- **DIN/ISO A3** Blattgröße 297 mm x 420 mm entsprechend 842 p x 1190 p.
- **DIN/ISO A4** Blattgröße 210 mm x 297 mm entsprechend 595 p x 842 p, näherungsweise 8 Zoll x 12 Zoll.
- **DIN/ISO A5** Blattgröße 148 mm x 210 mm entsprechend 421 p x 595 p,
- **DIN/ISO A6** Blattgröße 105 mm x 148 mm entsprechend 298 p x 421 p,
- **PA4** Blattgröße 210 mm x 280 mm, Breite wie DIN A4, Länge wie Letter.
- **Briefumschlag DIN lang** 110 mm x 220 mm
- **Tabellierpapier schmal** Blattgröße einschließlich Lochrand 240 mm x 305 mm, ohne Lochrand 210 mm x 305 mm, also annähernd DIN A4. 305 mm sind 12 Zoll.
- **Tabellierpapier breit** Blattgröße einschließlich Lochrand 375 mm x 305 mm. Bei 10 Zeichen/Zoll fasst die Zeile 132 Zeichen, bei 12 Zeichen/Zoll 158 Zeichen, bei 15 Zeichen/Zoll 198 Zeichen.
- **Executive** amerikanisch, Blattgröße 7.25 Zoll x 10.5 Zoll (184,2 mm x 266,7 mm) entsprechend 540 p x 720 p. Man findet aber auch 7,5 Zoll x 10.0 Zoll (194 mm x 254 mm).
- **Legal** amerikanisch, Blattgröße 8.5 Zoll x 14 Zoll (215,9 mm x 355,6 mm) entsprechend 612 p x 1008 p.
- **Letter** amerikanisch, Blattgröße 8.5 Zoll x 11 Zoll (215,9 mm x 279,4 mm) entsprechend 612 p x 792 p, Schreibmaschinenpapier.
- **Ledger/Tabloid** amerikanisch, Blattgröße 11 Zoll x 17 Zoll (279,4 mm x 431,8 mm) entsprechend 792 p x 1224 p.
- **Toilet** amerikanisch, Blattgröße 4,5 Zoll x 4,5 Zoll (114 mm x 114 mm) entsprechend 324 p x 324 p.

Weiteres zu den Papierformaten bei <http://www.cl.cam.ac.uk/~mgk25/iso-paper.html>. Es ist zwischen der Blattgröße, der Größe des adressierbaren Bereichs (logische Seite) und der Größe des beschreibbaren Bereichs zu unterscheiden. Näheres beispielsweise im *HP PCL 5 Printer Language Technical Reference Manual*, das nicht zum Lieferumfang der HP Laserdrucker gehört.

Die Papierstärke wird in Gramm pro Quadratmeter gemessen. Im Geschäftsverkehr üblich sind 80 g/m². Etwas fester und und ansprechender ist eine Stärke von 90 g/m².

D.2 Schriftgrößen

Bei den Schriftgrößen oder -graden finden sich mehrere Maßsysteme. In Mitteleuropa ist traditionell das aus dem Bleisatz stammende Punktsystem nach FRANÇOIS AMBROISE DIDOT gebräuchlich, das die Höhe der Schrifttype (nicht des Buchstaben) angibt. Es geht auf den französischen Fuß zurück; ein Punkt war früher 0,376 mm, heute ist er in Angleichung an das metrische System 0,375 mm. Die wichtigsten Größen werden auch mit eigenen Namen bezeichnet (9 pt = Borgis, 12 pt = Cicero). Die untere Grenze der Lesbarkeit bei längerem Lesen beträgt 9 pt, optimal sind 11 oder 12 pt. Großschrift beginnt mit 14 pt.

Auf englische Füße geht das Pica-System zurück, in dem 1 Pica gleich 12 Points oder 4,216 mm (1/6 Zoll) ist. Hiervon abgeleitet ist das IBM-Pica-System, in dem Pica eine Schrift mit 6 Zeilen pro Zoll und 10 Zeichen pro Zoll ist. Elite bedeutet 12 Zeichen pro Zoll bei gleichem Zeilenabstand.

Auf deutschen Schreibmaschinen war Pica eine Schriftart mit 2,6 mm Schrittweite (etwa 10 Zeichen pro Zoll) und Perl eine Schriftart mit 2,3 mm Schrittweite.

E

Farben

E.1 X11-Farben

Eine Auswahl der von X11 verwendeten Farben. Die vollständige Liste ist in `/etc/X11/rgb.txt`, `/usr/X11R6/lib/X11/rgb.txt` oder `/usr/lib/x11/rgb.txt` zu finden. Mittels `showrgb` werden die Farben mit ihren Zahlenwerten (RGB-Triplets) aufgelistet. Das Kommando `xcolors` stellt in einem großen Fenster alle Farben gleichzeitig samt ihren Bezeichnungen dar, das Kommando `xcolorsel` in einem kleinen Fenster einen Ausschnitt samt Hexwerten und Bezeichnungen. Die tatsächliche Farbdarstellung hängt vom Bildschirm bzw. Drucker ab, deshalb sind die Farbmuster nur als Orientierungshilfe zu verstehen.

X11 stellt auch Farben dar, die nicht in dieser Liste genannt sind. Darüber hinaus kann sich ein Benutzer mit einem Farbeditor eigene Farben für X11 mischen. Nach dem RGB-Modell arbeiten `xcolmix` (Debian) und `xcoloredit`, zu holen bei <http://freebsd.home4u.ch/de/> oder <ftp://export.lcs.mit.edu/R5contrib/>. Ein durchschnittlicher Benutzer kommt jedoch mit den vorgefertigten Farben aus.

Farbe	Colour	RGB-Vektor	Hex, 8bittig
weiß	white	255,255,255	#ffffff
gelb	yellow	255,255,0	#ffff00
fuch sienrot	magenta1	255,0,255	#ff00ff
cyan	cyan1	0,255,255	#00ffff
rot	red	255,0,0	#ff0000
maigrün	green	0,255,0	#00ff00
blau	blue1	0,0,255	#0000ff
grau	grey52	133,133,133	#868686
olivgrün	olive drab	107,142,35	#698e20
purpur	purple	160,32,240	#9e20ef
dunkelgrün	green4	0,139,0	#008a00
kastanienbraun	maroon	176,48,96	#ae3061

grün	green	0,255,0	#00ff00
marineblau	navy blue	0,0,128	#000086
lachsrot	salmon	250,128,114	#f78271
dunkelrosa	hot pink	255,105,180	#ff69b6
dunkelorange	dark orange	255,140,0	#ff8e00
königsblau	royal blue	65,105,225	#4169df
altweiß	antique white	250,235,215	#f7ebd7
ocker	navajo white	255,222,173	#ffdfae
hellgrau	light grey	211,211,211	#cfd3cf
schwarz	black	0,0,0	#000000

E.2 HTML-Farben

Eine Auswahl von Farben nach dem RGB-Farbmodell, ihren Bezeichnungen und ihren Hex-Codes, wie sie in HTML-Seiten verwendet werden. Die ersten 16 Farben können auf Webseiten über ihre englische Bezeichnung angesprochen werden; die Hex-Codes sollten immer richtig verstanden werden. Weiteres unter <http://www.htmlhelp.com/cgi-bin/color.cgi> oder <http://www.seifert-web.de/info/bgclick.htm>.

Farbe	Colour	Hex Code
weiß	white	#ffffff
schwarz	black	#000000
gelb	yellow	#ffff00
fuch sienrot	magenta, fuchsia	#ff00ff
cyan	cyan, aqua	#00ffff
rot	red	#ff0000
maigrün	lime	#00ff00
blau	blue	#0000ff
grau	gray	#808080
olivgrün	olive	#808000
purpur	purple	#800080
dunkelgrün	teal	#008080
kastanienbraun	maroon	#800000
grün	green	#008000
marineblau	navy	#000080
silbern	silver	#c0c0c0
lachsrot	salmon	#fa8072
dunkelrosa	hot pink	#ff69b4
dunkel orange	dark orange	#ff8c00
königsblau	royal blue	#4169e1
ocker	navajo white	#ffdead

F

Datei-Merkmale

F.1 Zugriffsrechte

Bezeichnungen, wie sie von `ls` und `chmod` verwendet werden:

- `d` Verzeichnis (directory),
- `l` weicher (symbolischer) Link,
- `c` Character Device File, zeichenweise arbeitende Gerätedatei (z. B. Terminal, Drucker),
- `b` Block Device File, blockweise arbeitende Gerätedatei (z. B. Platte, Floppy),
- `p` Named Pipe (FIFO),
- `n` Network Device File,
- `s` an vorderster Stelle: Socket,
- `-` an vorderster Stelle: reguläre Datei, sonst gleichbedeutend mit nicht gesetztem Recht, Zahlenwert 0,
- `r` lesen (read), kopieren, bei Verzeichnissen: mit `ls` auflisten, Zahlenwert 4,
- `w` schreiben (write), ändern, bei Verzeichnissen Dateien anlegen oder löschen, Zahlenwert 2,
- `x` ausführen (execute, was nur bei Programmen sinnvoll ist), bei Verzeichnissen mit `cd` hinein- oder hindurchgehen (search), Zahlenwert 1,
- `S` bei den Besitzerrechten: Set-User-ID-Bit gesetzt, Zahlenwert 4, `x` nicht gesetzt,
- `s` bei den Besitzerrechten: Set-User-ID-Bit gesetzt, Zahlenwert 4, `x` gesetzt,
- `S` bei den Gruppenrechten: Set-Group-ID-Bit gesetzt, Zahlenwert 2, `x` nicht gesetzt,
- `s` bei den Gruppenrechten: Set-Group-ID-Bit gesetzt, Zahlenwert 2, `x` gesetzt,
- `T` Sticky Bit gesetzt, Zahlenwert 1, `x` nicht gesetzt,
- `t` Sticky Bit, Zahlenwert 1, und `x` gesetzt,

Das Sticky Bit bewirkte früher, dass ein Programm im Arbeitsspeicher verblieb, auch wenn es vorübergehend nicht benutzt wurde. Unter Linux führt es dazu, dass in Verzeichnissen wie `/tmp/`, in denen jeder Dateien anlegen kann, nur der Besitzer eine Datei auch wieder löschen darf.

Tabelle F.1. Zugriffsrechte von Dateien und Verzeichnissen, Beispiele

Rechte	Rechte	Besitzer	Gruppe	Rest der Welt
000	- - - - -	Rechte ändern	nichts	nichts
700	r w x - - - - -	alles	nichts	nichts
600	r w - - - - -	lesen + schreiben	nichts	nichts
500	r - x - - - - -	lesen + ausführen	nichts	nichts
400	r - - - - -	lesen	nichts	nichts
300	- w x - - - - -	schreiben + ausführen	nichts	nichts
200	- w - - - - -	schreiben	nichts	nichts
100	- - x - - - - -	ausführen	nichts	nichts
750	r w x r - x - - -	alles	lesen + ausführen	nichts
640	r w - - - - -	lesen + schreiben	lesen	nichts
755	r w x r - x r - x	alles	lesen + ausführen	lesen + ausführen
644	r w - r - - r - -	lesen + schreiben	lesen	lesen
664	r w - r w - r - -	lesen + schreiben	lesen + schreiben	lesen
775	r w x r w x r - x	alles	alles	lesen + ausführen
777	r w x r w x r w x	alles	alles	alles
1000	- - - - - T	nichts	nichts	nichts, Sticky Bit
2000	- - - - - S - - -	nichts	nichts, sgid-Bit	nichts
4000	- - S - - - - -	nichts, suid-Bit	nichts	nichts
4600	r w S - - - - -	lesen + schreiben, suid-Bit	nichts	nichts
4700	r w s - - - - -	alles, suid-Bit	nichts	nichts
6750	r w s r - x - - -	alles, suid-Bit	lesen + ausführen, sgid-Bit	nichts
4555	r - s r - x r - x	lesen + ausführen, suid-Bit	lesen + ausführen	lesen + ausführen
7555	r - s r - s r - t	lesen + ausführen, suid-Bit	lesen + ausführen, sgid-Bit	lesen + ausführen, Sticky Bit

F.2 Kennungen

Unter Linux/UNIX ist es nicht gebräuchlich, den Typ einer Datei durch eine Kennung (Endung, Erweiterung, Extension) an seinen Namen zu kennzeichnen, aber es ist erlaubt. Die Länge der Kennung ist beliebig, es dürfen auch mehrere Kennungen angefügt werden, zum Beispiel `.ps.gz`. Unter <http://www.endungen.de/> und <http://www.wotsit.org/> liegen umfangreiche Listen. Folgende Kennungen (a = ASCII-Text, b = binäre Datei, ? = wechselnd oder unbekannt) kommen im Buch oder seinem Umfeld vor:

a	b	UNIX-Archiv, mit <code>ar (1)</code> erzeugt
aac	b	Advanced Audio Coding, MP 4
adi	b	AutoCAD DXF
aiff	b	Audio Interchange File Format
arc	?	Archiv, mit <code>arc</code> oder <code>pkpak</code> komprimiert
arj	b	Archiv, mit <code>arj</code> komprimiert
asc	a	ASCII-Textfile
au	b	Sound (Sun, NeXT)
aux	?	L ^A T _E X Hilfsfile, mit Bezügen
avi	b	Microsoft RIFF
bak	a	Backup
bat	a	Batchfile unter DOS, entsprechend Shellskript
bin	b	binäre Datei
bit	b	L ^A T _E X-Ausgabe des Druckertreibers
blend	b	Blender-Datei
bm	b	X11 Bitmap
bmp	b	Microsoft Bitmap Graphics
bz2	b	bzip2-komprimierte Datei
c	a	C-Quelle
C	a	C++-Quelle
cat	a	Catalogue, Verzeichnis
cc	a	C++-Quelle
cdf	?	Comma delimited format
cfg	?	Konfigurationsdatei
cgm	b	Computer Graphics Metafile
cmd	a	Command, Skript, Batchfile unter OS/2
cnf	?	Konfigurationsdatei
com	b	Kommando (DOS), ausführbares Programm
cpp	a	C++-Quelle
csv	a	Comma Separated Values
ctx	b	Signaturfile (PGP)
dat	a	Daten
dct	?	Dictionary
deb	?	Debian-Paket
def	?	Definitionen, Defaultwerte

des	?	Description, Beschreibung
dic	?	Dictionary
diff	?	Differenz, Datei mit Änderungen
dir	a	Directory, Verzeichnis
dll	b	Dynamically Linked Library
doc	a	Dokument, Text (MS Word)
drv	b	Device Driver
drw	b	Drawing
dsc	?	Description, Beschreibung
dvi	b	Zwischendatei von \LaTeX , geräteunabhängig
dxf	b	Data Exchange Format (AutoCAD)
emf	b	Enhanced Metafile Graphic
eml	a	Electronic Mail
enc	b	encoded
eps	b	Adobe Encapsulated PostScript
err	a	Error, Fehlerprotokoll
exe	b	executable, ausführbares Programm (DOS)
f	a	FORTRAN-Quelle
fax	b	G3 Fax
fd	?	Font Description
fig	b	xfig -Datei
fli	b	Em \TeX Fontlibrary
fnt	b	Font
g3	b	G3 Fax
gif	b	CompuServe Graphics Interchange Format
gz	b	mit GNU gzip komprimiert
h	a	Header-Datei , include-Datei
hex	b	Hexdump
hpp	a	Header-Datei C++
hqx	b	Macintosh BinHex encoded
htm	a	Hypertext-Markup-Language (DOS, MS Windows)
html	a	Hypertext-Markup-Language (Linux/UNIX)
idx	a	Index
img	b	Image, GEM Paint, Rastergrafik/Bitmap
inf	a	Information, Installationsskript
ini	?	Konfigurationsdatei (Initialisierung)
iso	?	CD-Image, ISO-9660-Dateisystem
jar	?	Java Archiv
jfi	b	JPEG File Interchange Format
jif	b	JPEG Interchange Format
jpeg	b	Joint Photographic Experts Group Compressed
jpg	b	Joint Photographic Experts Group Compressed
js	a	Javascript
jse	b	Javascript encoded
lib	b	Library, Bibliothek

lof	a	L ^A T _E X Bildverzeichnis (list of figures)
log	a	Protokolldatei, L ^A T _E X-Meldungen während Übersetzung
lot	a	L ^A T _E X Tabellenverzeichnis (list of tables)
lzh	b	Archiv, mit lha komprimiert
mak	a	Makefile
mdf	?	Menu Definition
mid	b	MIDI File
mif	b	Maker Interchange Format (FrameMaker)
miff	b	Magick Image File Format
mod	b	MODULA-Quelle, Amiga-Modul (Sound)
mp3	b	MPEG 1 Audio Layer 3
mpeg	b	Moving Pictures Expert Group (Filme)
ogg	b	Ogg Vorbis Audio
ovl	b	Overlay
ovr	b	Overlay, Overview
pak	b	Archiv, mit pkpak komprimiert
pbm	b	Portable Bitmap
pcd	b	Kodak Photo Compact Disc
pcl	a	HP Printer Control Language
pcx	b	Paintbrush Graphic
pdb	b	Palm Database
pdf	b	Adobe Portable Document Format
pdi	b	Personal Data Interchange
pgm	b	Portable Graymap
pic	b	Lotus/Pixar/Radiance Picture, Softimage Image
pif	b	Program Information
pit	b	PackIt (Kompressor Macintosh)
pk	b	Packed Format Font
png	b	Portable Network Graphics
pnm	b	Portable Anymap, Bitmap Graphics
pov	b	Persistance of Vision
ppm	b	Portable Pixel Map
pps	?	Power Point Slide Show
ppt	?	Power Point
prn	a	Druckerfile
ps	b	Adobe PostScript
psf	b	Permanent Swap File
qtm	b	QuickTime
ra	b	Real Audio
raw	b	Raw PCM Audio Data
rib	b	RenderMan Interface Bytestream
riff	b	Resource Interchange File Format
rm	b	Real Media
rpm	?	Red Hat Package Manager
rtf	?	Rich Text Format

sdf	?	Space Delimited File
sea	?	Self Extracting Archive (Macintosh)
sh	a	Bourne-Shell-Skript
shar	a	Bourne-Shell-Archiv
shr	a	Shell-Archiv
shtml	a	Server Parsed HTML Document
sit	b	Macintosh StuffIt Archive
sl	a	Shader Language Source
slx	b	Shader Language Executable
src	a	Program Source Code
svg	b	Scalable Vector Graphics
swf	g	Shockwave Flash
swp	b	Swap
swx	b	OpenOffice.org Dokument
sxc	b	Star/OpenOffice Calc
sxd	b	Star/OpenOffice Draw
sxi	b	Star/OpenOffice Impress
sxw	b	Star/OpenOffice Writer
tar	b	tar-Archiv
tex	a	T _E X- oder L ^A T _E X-Quelltext
tfm	b	T _E X-Font-Metrics
tga	b	Truevision Targa Graphic
tgz	b	.tar.gz, tar-Archiv, GNUzip komprimiert
tif	b	Tag Image File Format
tiff	b	Tag Image File Format
tmp	?	temporäre Datei
toc	a	Hilfsdatei von L ^A T _E X (table of contents)
ttf	b	TrueType Font
txt	a	Textdatei, lesbar
tz	b	tar.Z, tar-Archiv, compress-komprimiert
uu	a	uuencoded file, ASCII, aber nicht lesbar
vcf	b	Visiting Card File
voc	b	Creative Labs Sound
wav	b	Waveform Audio File (RIFF)
xbm	b	X11 Bitmap
xim	b	X11 Xim Toolkit
xls	?	MS Excel Datei
xpm	b	X11 Pixmap
xwd	b	X11 Dump Image
Z	b	mit compress komprimiert
z	b	mit GNU gzip komprimiert (veraltet, besser gz)
zip	b	mit pkzip komprimiert
zoo	b	mit zoo komprimiert

G

Prozess-Signale

Die Default-Reaktion eines Prozesses auf die meisten Signale ist seine Beendigung; sie können aber abgefangen und umdefiniert werden. Die Signale 09, 24 und 26 können nicht abgefangen werden. Die Bezeichnungen sind nicht ganz einheitlich. Weiteres unter `signal(2)`, `signal(5)` oder `signal(7)`.

Name	Nr.	Bedeutung
SIGHUP	01	hangup
SIGINT	02	interrupt (meist Break-Taste)
SIGQUIT	03	quit
SIGILL	04	illegal instruction
SIGTRAP	05	trace trap
SIGABRT	06	software generated abort
SIGIOT	06	software generated signal
SIGEMT	07	software generated signal
SIGFPE	08	floating point exception
SIGKILL	09	kill (sofortiger Selbstmord)
SIGBUS	10	bus error
SIGSEGV	11	segmentation violation
SIGSYS	12	bad argument to system call
SIGPIPE	13	write on a pipe with no one to read it
SIGALRM	14	alarm clock
SIGTERM	15	software termination (bitte Schluß machen)
SIGUSR1	16	user defined signal 1
SIGSTKFLT	16	stack fault on coprocessor
SIGUSR2	17	user defined signal 2
SIGCHLD	18	death of a child
SIGCLD	18	= SIGCHLD
SIGPWR	19	power fail
SIGINFO	19	= SIGPWR
SIGVTALRM	20	virtual timer alarm
SIGPROF	21	profiling timer alarm

SIGIO	22	asynchronous I/O signal
SIGPOLL	22	= SIGIO
SIGWINDOW	23	window change or mouse signal
SIGSTOP	24	stop
SIGTSTP	25	stop signal generated from keyboard
SIGCONT	26	continue after stop
SIGTTIN	27	background read attempt from control terminal
SIGTTOU	28	background write attempt from control terminal
SIGURG	29	urgent data arrived on an I/O channel
SIGLOST	30	NFS file lock lost
SIGXCPU	30	CPU time limit exceeded
SIGXFSZ	31	file size limit exceeded

Die Nr. 0 wird nicht für ein Signal verwendet. Wird sie dem Shell-Kommando `trap` übergeben, so ist damit das Ende der Shell (`exit`) gemeint.

H

Slang im Netz

Diese Sammlung von im Netz vorkommenden Slang-Abkürzungen ist ein Auszug aus dem Abkürzungs-Lexikon <http://www.abklex.de/abklex/> mit rund 10000 Abkürzungen aus Informatik und Telekommunikation.

2L8	Too Late
AAMOF	As A Matter Of Fact
ACK	Acknowledge
ADN	Any Day Now
AFAIAA	As Far As I Am Aware
AFAIC	As Far As I am Concerned
AFAIK	As Far As I Know
AFAIKT	As Far As I Know Today
AFAIR	As Far As I Remember
AFJ	April Fool's Joke
AFK	Away From Keyboard
AIMB	As I Mentioned Before
AISI	As I See It
AIUI	As I Understand It
AKA	Also Known As
AMBW	All My Best Wishes
ANFSCD	And Now For Something Completely Different
APOL	Alternate Person On Line
ASAP	As Soon As Possible
ATFSM	Ask The Friendly System Manager
ATM	At The Moment
ATST	At The Same Time
ATT	At This Time
AWA	A While Ago
AWB	A While Back
AWGHTGTTA	Are We Going To Have To Go Through This Again?
AWHIFY	Are We Having Fun Yet?

AYOR	At Your Own Risc
AYST	Are You Still There?
B4	Before
B4N	Bye For Now
BAK	Back At Keyboard
BBL	Be Back Later
BBR	Burnt Beyond Recognition
BCNU	Be Seeing You
BFB	Brute Force and Bloody Ignorance
BFI	Brute Force and Ignorance
BFMI	Brute Force and Massive Ignorance
BFN	Bye For Now
BG	Big Grin
BHOF	Bald Headed Old Fart
BION	Believe It Or Not
BITD	Back In The Day
BM	Byte Me
BNF	Big Name Fan
BNFSCD	But Now For Something Completely Different
BOF	Birds Of a Feather
BOFS	Birds Of a Feather Session
BOT	Back On Topic
BRB	Be Right Back
BSF	But Seriously Folks
BTA	But Then Again
BTAIM	Be That As It May
BTC	Biting The Carpet
BTDT	Been There Done That
BTHOM	Beats The Hell Outta Me
BTIC	But Then, I'am Crazy
BTK	Back To Keyboard
BTSOOM	Beats The Shit Out Of Me
BTW	By The Way
BYKT	But You Knew That
CMH	Cross My Heart
CMIIW	Correct Me If I'm Wrong
CU	See You
CU2	See You, Too
CUL	See You Later
CYA	Cover Your Ass
CYL	See You Later
D+C	Duck + Cover
DAU	Dümmster Anzunehmender User
DHRVVF	Ducking, Hiding and Running Very Very Fast
DILLIGAS	Do I Look Like I Give A Shit

DIY	Do It Yourself
DLTM	Don't Lie To Me
DSH	Desperately Seeking Help
DUCWIC	Do You See Waht I See
DWIM	Do What I Mean
DWISNWID	Do What I Say Not What I Do
DYHWIH	Do You Hear What I Hear
E2EG	Ear To Ear Grin
EG	Evil Grin
EMFBI	Excuse Me For Butting In
EOD	End Of Discussion
ESOSL	Endless Snorts Of Stupid Laughter
ETOL	Evil Twin On Line
F2F	Face to Face
FA	For Auction
FAFWOA	For A Friend Without Access
FB	Fine Business
FCFS	First Come First Served
FHS	For Heaven's Sake
FIAWOL	Fandom Is A Way Of Life
FISH	First In, Still Here
FITB	Fill In The Blank
FITNR	Fixed In The Next Release
FOAF	Friend Of A Friend
FOAD	Fuck Off And Die
FTASB	Faster Than A Speeding Bullet
FTL	Faster Than Light
FUBAR	Fouled Up Beyond All Repair/Recognition
FUBB	Fouled Up Beyond Belief
FUD	(spreading) Fear, Uncertainty, and Disinformation
FWIW	For What It's Worth
FYA	For Your Amusement
GA	Go Ahead
GAFIA	Get Away From It All
GAL	Get A Life
GDR	Grinning, Ducking and Running
GDRVVF	Grinning, Ducking and Running Very, Very Fast
GFC	Going for Coffee
GIGO	Garbage In, Garbage/Gospel Out
GIWIST	Gee I Wish I'd Said That
GOK	God Only Knows
GR+D	Grinning, Running + Ducking
HAK	Hugs And Kisses
HAND	Have A Nice Day
HGZH	Hoffe Geholfen Zu Haben

HHOK	Ha Ha Only Kidding
HHOS	Ha Ha Only Serious
HSIK	How Should I Know
HTH	Hope This/That Helps
IAC	In Any Case
IAE	In Any Event
IANAL	I Am Not A Lawyer
IBC	Inadequate But Cute
IBTD	I Beg To Differ
IC	I See
IC	In Characters
ICOCBW	I Could, Of Course, Be Wrong
ICQ	I Seek You
IDC	I Don't Care
IDST	I Didn't Say That
IDTS	I Don't Think So
IIRC	If I Remember Correctly
IITYWYBMAD	If I Tell You Will You Buy Me Another Drink?
IIMW	If It Were Me/Mine
ILLAB	Ich Liege Lachend Am Boden, vgl. ROTFL
IMAO	In My Arrogant Opinion
IMCO	In My Considered Opinion
IME	In My Experience
IMHO	In My Humble Opinion
IMI	I Mean It
IMNSCO	In My Not So Considered Opinion
IMNSHO	In My Not So Humble Opinion
IMO	In My Opinion
IMOBO	In My Own Biased Opinion
INPO	In No Particular Order
IOW	In Other Words
IRAS	I Really Am Sorry
IRL	In Real Life
ISBT	It's Strange But True
ISTM	It Seems To Me
ISTR	I Seem To Remember
IUTY	It's Up To You
IWBNI	It Would Be Nice If
IYSWIM	If You See What I Mean
ISWYM	I See What You Mean
JAM	Just A Minute
JAUA	Just Another Useless Answer
JFTR	Just For The Record
JIC	Just In Case
JJ	Just Joking

JSNM	Just Stark Naked Magic
KIBO	Knowledge In, Bullshit Out
KWIM	Know What I Mean?
LLP	Live Long and Prosper
LLTA	Lots and Lots of Thundering Applause
LMAO	Laughing My Ass Off
LOL	Laughing Out Loud
LOL	Lots Of Luck
LTNS	Long Time No See
MDA	Merci d'avance
MHOTY	My Hat's Off To You
MMN	Meiner Meinung Nach
MNRE	Manual Not Read Error
MOTAS	Member Of The Appropriate Sex
MTFBWY	May The Force Be With You
MUMN	Meiner Unmaßgeblichen Meinung Nache
MUSEN	Meinem Unmaßgeblichen Sprachempfinden Nach
MYOB	Mind Your Own Business
NAK	Negative Acknowledge
NBD	No Big Deal
NCNCNC	No Coffee, No Chocolate, No Computer
NFI	No Frigging Idea
NFW	No Fucking Way
NIMBY	Not In My Backyard
NLA	Not Long Ago
NLMB	Nolifemuddinbastard
NN	Nightmare Networker
NRN	No Reply Necessary
NSO	Nonvirtual Significant Other
NTTAWWT	Not That There's Anything Wrong With That
OAQ	Over And Out
OATUS	On A Totally Unrelated Subject
OAUS	On An Unrelated Subject
OBTW	Oh, By The Way
OBO	Or Best Offer
OFAB	Old Fart At Play
OIC	Oh, I See
ONNA	Oh No, Not Again
ONNTA	Oh No, Not This Again
OOC	Out Of Characters
OOP	Out Of Print
OOTB	Out Of The Box
OOTC	Obligatory On-Topic Comment
OT	Off Topic
OTL	Out To Lunch

OTOH	On The Other Hand
OTTH	On The Third Hand
PAL	Problem Anderer Leute
PDQ	Pretty Darned Quick
PEBKAC	Problem Exists Between Keyboard And Chair
PFM	Pure Fantastic Magic
PITA	Pine In The Ass
PMETC	Pardon Me, etc.
PMFIJB	Pardon Me For Jumping In But
POV	Point Of View
PTO	Please Turn Over
RAEBNC	Read And Enjoyed, But No Comment
RL	Real Life
ROFBTC	Rolling On the Floor Biting The Carpet
ROFL	Rolling On the Floor Laughing
ROFLASTC	Rolling On the Floor Laughing And Scaring The Cat
ROFLMAO	Rolling On The Floor Laughing My Ass Off
ROTFL	Rolling On The Floor Laughing
RSN	Real Soon Now
RTFAQ	Read The FAQ
RTFB	Read The Funny Binary
RTFF	Read The Fantastic FAQ
RTFM	Read The Fine/Fantastic/Funny ... Manual
RTFS	Read The Funny Source
RTM	Read The Manual
RTWFQ	Read The Whole Fantastic Question
RUOK	Are You Ok
RYS	Read Your Screen
SCNR	Sorry, Could Not Resist
SEP	Somebody Else's Problem
SFMJI	Sorry For My Jumping In
SIASL	Stranger In A Strange Land
SIMCA	Sitting In My Chair Amused
SITD	Still In The Dark
SNAFU	Situation Normal All Fed/Fucked Up
TABYAS	Thinking All But You Are Stupid
TAFN	That's All For Now
TANJ	There Ain't No Justice
TANSTAAFL	There Ain't No Such Things As A Free Lunch
TARFU	Things Are Really Fouled Up
TBH	To Be Honest
TFTT	Thanks For The Thought
TGAL	Think Globally, Act Locally
TGIF	Thank God It's Friday
THWLAIAS	The Hour Was Late, And I Am Senile

TIA	Thanks In Advance
TIC	Tongue In Cheek
TINAR	This Is Not A Recommendation/Review
TINWIS	That Is Not What I Said
TMTOWTDI	There's More Than One Way To Do It
TNX	Thanks
TOFU	Text Oben, Fullquote Unten
TOJ	Tears Of Joy
TPTB	The Powers That Be
TRDMC	Tears Running Down My Cheeks
TSOHF	Total Sense Of Humour Failure
TTBOMK	To The Best Of My Knowledge
TTFN	Ta Ta For Now
TTT	That's The Truth
TTYL	Type/Talk To You Later
TIUI	That's What I Understand/Understood
TYCLO	Turn Your Caps Lock Off
TYVM	Thank You Very Much
UDP	Usenet Death Penalty
UL	Urban Legend
UTSL	Use The Source, Luke
VBG	Very Big Grin
VL	Virtual Life
WAB	What Another Bill
WAMKSAM	Why Are My Kids Staring At Me?
WDYMBT	What Do You Mean By That
WDYS	What Did You Say
WHAK	With Hugs And Kisses
WIBNI	Would It Be Nice If
WIMRE	Wenn Ich Mich Recht Erinnere
WMRS	Write Me Real Soon
WOMBAT	Waste Of Money, Brains, And Time
WRT	With Regard/Respect To
WT	Without Thinking
WTB	Wanted To Buy
WTH	What The Heck/Hell
WTT	Wanted To Trade
WTTM	Without Thinking Too Much
WTTF	Want To Test First
WWG	Wieder Was Gelernt
WYS	Whatever You Say
YKWIS	You Know What I'm Saying
YMMV	Your Mileage May Vary

I

Beispiele LaTeX

I.1 Gelatexte Formeln

$$c = \sqrt{a^2 + b^2} \tag{I.1}$$

$$\sqrt[3]{1+x} \approx 1 + \frac{x}{3} \quad \text{für } x \ll 1 \tag{I.2}$$

$$r = \sqrt[3]{\frac{3}{4\pi}V} \tag{I.3}$$

$$\lim_{x \rightarrow 0} \frac{\sin x}{x} = 1 \tag{I.4}$$

$$a = \frac{F_0}{k} \frac{1}{\sqrt{(1 - \frac{\Omega^2}{\omega_0^2})^2 + (\frac{\Omega_c}{k})^2}} \tag{I.5}$$

$$m\ddot{x} + c\dot{x} + kx = \sum_k F_k \cos \Omega_k t \tag{I.6}$$

$$\Theta \frac{d^2 \varphi}{dt^2} + k^* \frac{d\varphi}{dt} + D^* \varphi = |\mathbf{D}| \tag{I.7}$$

$$\bar{Y} \approx f(\bar{x}) + \frac{1}{2} \frac{N-1}{N} f''(\bar{x}) s_x^2 \tag{I.8}$$

$$\mathbf{F}_\mathbf{r} = -\frac{\Gamma m M}{r^2} \mathbf{e}_\mathbf{r} = -\frac{\Gamma m M}{r^3} \mathbf{r} \tag{I.9}$$

$$\begin{aligned}\text{Arbeit} &= \lim_{\Delta r_i \rightarrow 0} \sum \mathbf{F}_i \Delta \mathbf{r}_i \\ &= \int_{\mathbf{r}_0}^{\mathbf{r}(t)} \mathbf{F}(\mathbf{r}) \, d\mathbf{r}\end{aligned}\tag{I.10}$$

$$\prod_{j \geq 0} \left(\sum_{k \geq 0} a_{jk} z^k \right) = \sum_{n \geq 0} z^n \left(\sum_{\substack{h_0, k_1, \dots \geq 0 \\ k_0 + k_1 + \dots = 0}} a_{0k_0} a_{1k_1} \dots \right)\tag{I.11}$$

$$\mathbf{x} = \begin{pmatrix} x - x_s \\ y - y_s \\ z - z_s \end{pmatrix}\tag{I.12}$$

$$\Psi = \begin{pmatrix} \begin{pmatrix} ab \\ cd \end{pmatrix} & \frac{e+f}{g-h} \\ \Re z & \begin{vmatrix} ij \\ kl \end{vmatrix} \end{pmatrix}\tag{I.13}^1$$

$$dE_\omega = V \frac{\hbar}{\pi^2 c^3} \omega^3 \cdot e^{-\frac{\hbar \omega}{T}} \cdot d\omega\tag{I.14}$$

$$\oint \mathbf{E} \, ds = - \frac{\partial}{\partial t} \int \mathbf{B} \, d\mathbf{A}\tag{I.15}$$

$$\frac{1}{2\pi j} \int_{x-j\infty}^{x+j\infty} e^{ts} f(s) \, ds = \begin{cases} 0 & \text{für } t < 0 \\ F(t) & \text{für } t > 0 \end{cases}\tag{I.16}$$

$$\binom{n+1}{k} = \binom{n}{k} + \binom{n}{k-1}\tag{I.17}$$

$$\boxed{\forall x \in \mathbb{R} : \quad x^2 \geq 0}\tag{I.18}$$

$$\forall x, y, z \in \mathbb{M} : \quad (xRy \wedge xRz) \Rightarrow y = z\tag{I.19}$$

$$A \cdot B = \overline{\overline{A} + \overline{B}}\tag{I.20}$$

¹Fette Griechen gibt es nur als Großbuchstaben.

$$\rho \cdot \bar{v}_k \cdot \frac{\partial \bar{v}_j}{\partial x_k} = -\frac{\partial \bar{p}}{\partial x_j} + \frac{\partial}{\partial x_k} \left(\mu \frac{\partial \bar{v}_j}{\partial x_k} - \rho \overline{v'_k v'_j} \right) \quad (\text{I.21})$$

$$r' = \frac{\overline{v'_1 v'_2}}{\sqrt{\bar{v}_1^2} \sqrt{\bar{v}_2^2}} \quad (\text{I.22})$$

$$\tau_{tur} = \rho l^2 \left| \frac{\partial \bar{v}_1}{\partial x_2} \right| \frac{\partial \bar{v}_1}{\partial x_2} \quad (\text{I.23})$$

$$\begin{aligned} \ddot{R} &= \frac{1}{\bar{\Psi}} (V_r - \dot{R}) + R \dot{\Phi}^2 \\ \ddot{\Phi} &= \left[\frac{1}{\bar{\Psi}} (V_\varphi - R \dot{\Phi}) - 2 \dot{R} \dot{\Phi} \right] \frac{1}{R} \\ \ddot{Z} &= \frac{1}{\bar{\Psi}} (V_Z - \dot{Z}) \end{aligned} \quad (\text{I.24})$$

$$\hat{\chi}^2 = \frac{n(n-1)}{B(n-B)} \sum_{i=1}^k \frac{(B_i - E_i)^2}{n_i} > \chi^2_{k-1;\alpha} \quad (\text{I.25})$$

$$V(r,\vartheta,\varphi)=\sum_{l=0}^{\infty}\frac{4\pi}{2l+1}\sum_{m=-l}^l q_{l,m}\frac{Y_{l,m}\vartheta,\varphi}{r^{l+1}} \quad (\text{I.26})$$

$$\mathbf{q} = \begin{pmatrix} q_{0,0} \\ q_{1,1} \\ q_{1,0} \\ q_{1,-1} \\ q_{2,2} \\ q_{2,1} \\ q_{2,0} \\ q_{2,-1} \\ q_{2,-2} \end{pmatrix} \quad (\text{I.27})$$

$$q'_{l',m'} = \sum_{o=0}^{\infty} \sum_{p=-o}^o n_{o,p} \, q_{o,p} \, (\nabla)_{l'+o,m'-p;o,p} \, \frac{Y_{l'+o,m'-p}(\vartheta_a,\varphi_a)}{a^{l'+o+1}} \quad (\text{I.28})$$

$$q_{l,m} = -q'_{l,m} R^{2l+1} \frac{l(\epsilon_i - \epsilon_a)}{l(\epsilon_a + \epsilon_i) + \epsilon_a} \quad (\text{I.29})$$

$$\mathbf{q}'_{1,1} = \mathbf{q}'_{1,0} + I_{2,1} \, \mathbf{q}'_{2,0} \quad (\text{I.30})$$

$$\mathbf{q}'_{2,1} = \mathbf{q}'_{2,0} + I_{1,2} \, \mathbf{q}'_{1,0} \quad (\text{I.31})$$

$$\begin{aligned} \nabla_{\pm 1} \sum_{l=0}^{\infty} \frac{4\pi}{2l+1} \sum_{m=-l}^l q_{l,m} \frac{Y_{l,m}(\vartheta, \varphi)}{r^{l+1}} \\ = \sum_{l=0}^{\infty} \frac{4\pi}{2l+1} \sum_{m=-l}^l (\tilde{\nabla}_{\pm 1})_{l,m} q_{l,m} \frac{Y_{l+1,m\pm 1}(\vartheta, \varphi)}{r^{l+2}} \end{aligned} \quad (I.32)$$

$$\underbrace{a + \overbrace{b + \cdots + y + z}^{123}}_{\alpha\beta\gamma} \quad (I.33)$$

Lange Formeln muß man selbst in Zeilen umbrechen:

$$\begin{aligned} w + x + y + z = \\ a + b + c + d + e + f + \\ g + h + i + j + k + l \end{aligned} \quad (I.34)$$

$$N_{+1} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ (\tilde{\nabla}_+)_{0,0} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & (\tilde{\nabla}_+)_{1,1} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & (\tilde{\nabla}_+)_{1,0} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & (\tilde{\nabla}_+)_{1,-1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & (\tilde{\nabla}_+)_{2,2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & (\tilde{\nabla}_+)_{2,1} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & (\tilde{\nabla}_+)_{2,0} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Wie man an Gleichung (I.24) auf Seite 478 sieht, muss man erforderlichenfalls bei einem Gleichungssystem (eqnarray) mittels des Kommandos nonumber dafür sorgen, dass es nur eine einzige, gemeinsame Nummer erhält. Zugleich ist dies hier ein Beispiel für einen Bezug auf eine Gleichung. Hat man oft schwierige Formeln zu schreiben, legt man sich zweckmäßig eine Sammlung an und erzeugt neue Formeln durch Kopieren und Editieren ähnlicher Formeln.

Es gibt auch L^AT_EX-Pakete zum Schreiben von chemischen Strukturformeln, Musiknoten, Runen, Keilschrift, karolingischen Minuskeln, phonetischen Alphabeten, Braille und Kreuzworträtseln. Beim Suchen hilft *The T_EX Catalogue Online* auf <http://www.dante.de/>.

I.2 Formeln im Quelltext

(1)

```
\begin{equation}
c = \sqrt{a^{2} + b^{2}}
\end{equation}
```

(2)

```
\begin{equation}
\sqrt[3]{1 + x} \approx 1 + \frac{x}{3}
\qquad \mbox{ur} \quad x \ll 1
\end{equation}
```

(3)

```
\begin{equation}
r = \root 3 \of {\frac{3}{4\pi} V}
\end{equation}
```

(4)

```
\begin{equation}
\lim_{x \rightarrow 0} \frac{\sin x}{x} = 1
\end{equation}
```

(5)

```
\begin{equation}
a = \frac{F_0}{k} \cdot \frac{1}{\sqrt{(1 - \frac{\Omega^2}{\omega_0^2})^2 + (\frac{\Omega_c}{k})^2}}
\end{equation}
```

(6)

```
\begin{equation}
m\ddot{x} + c\dot{x} + kx = \sum_k F_k \cos \Omega_k t
\end{equation}
```

(7)

```
\begin{equation}
\Theta \frac{d^2 \varphi}{dt^2} + k^{\ast}
\frac{d \varphi}{dt} + D^{\ast} \varphi = | \vec{D} |
\end{equation}
```

(8)

```
\begin{equation}
\bar{Y} \approx f(\bar{x}) + \frac{1}{2}\,, \frac{N-1}{N}
\,, f''(\bar{x})\,, s_x^2
\end{equation}
```

```
(9)
\begin{equation}
\vec{F_{\Gamma}} = - \frac{\Gamma m M}{r^2} \vec{e_r} =
- \frac{\Gamma m M}{r^3} \vec{r}
\end{equation}
```

```
(10)
\begin{eqnarray}
\mbox{Arbeit} & = & \lim_{\Delta r_i \rightarrow 0} \sum
\{\vec{F}_i \Delta \vec{r}_i\} \nonumber \\
& = & \int \limits_{\vec{r}_0}^{\vec{r}(t)}
\{\vec{F}(\vec{r},) \} : d\vec{r}
\end{eqnarray}
```

```
(11)
\begin{equation}
\prod_{j \geq 0} \left( \sum_{k \geq 0} a_{jk} z^k \right) =
\sum_{n \geq 0} z^n \left( \sum_{h_0, k_1 \ldots}
\atop k_0 + k_1 + \cdots = 0}
a_{0k_0} a_{1k_1} \ldots \right)
\end{equation}
```

```
(12)
\begin{equation}
\vec{x} = \left( \begin{array}{c}
x - x_s \quad y - y_s \quad z - z_s
\end{array} \right)
\end{equation}
```

```
(13)
\begin{minipage}{120mm}
\begin{displaymath}
\{\bf\Psi\} = \left( \begin{array}{cc}
\displaystyle{ab \choose cd}
& \displaystyle{\frac{e+f}{g-h}}
\end{array} \right)
\Re z \ \& \ \displaystyle{\left| \atop ij \atop kl \right|}
\end{array} \right)
\end{displaymath}
\end{minipage}
\stepcounter{equation}
\hspace*{\fill}
(\theequation) \makebox[0pt][l]{\footnotemark}
\footnotetext{Fette Griechen gibt es nur als
Gro\3buchstaben.}
```

`\vspace{1mm}`

(14)

`\begin{equation}`

$$dE_{\omega} = V \frac{\hbar}{\pi^2 c^3} \omega^3 \cdot e^{-\frac{\hbar \omega}{T}} \cdot d\omega$$

`\end{equation}`

(15)

`\begin{equation}`

$$\oint \vec{E} \cdot d\vec{s} = - \frac{\partial}{\partial t} \int \vec{B} \cdot d\vec{A}$$

`\end{equation}`

(16)

`\begin{equation}`

$$\frac{1}{2} \pi j \int_{-\infty}^{\infty} e^{ts} f(s) ds =$$

$$\left\{ \begin{array}{l} r @{\quad \text{f"ur}} \quad 1 \\ 0 \text{ \& } t < 0 \\ F(t) \text{ \& } t > 0 \end{array} \right.$$

`\end{equation}`

(17)

`\begin{equation}`

$$\{n+1 \text{ choose } k\} = \{n \text{ choose } k\} + \{n \text{ choose } k-1\}$$

`\end{equation}`

(18)

`\begin{equation}`

$$\text{\mbox{\fbox{\parbox{60mm}{\begin{displaymath}$$

$$\text{\forall } x \in \mathbb{R}: \quad x^2 \geq 0$$

`\end{equation}`

(19)

`\begin{equation}`

$$\text{\forall } x,y,z \in \mathbb{M}: \quad (xRy \wedge xRz) \Rightarrow y = z$$

`\end{equation}`

(20)

`\begin{equation}`

$$A \cdot B = \overline{\bar{A} + \bar{B}}$$

\end{equation}

(21)

```
\begin{equation}
\rho \cdot \bar{v}_k \cdot \frac{\partial \bar{v}_j}{\partial x_k} = - \frac{\partial p}{\partial x_j} + \frac{\partial}{\partial x_k}
\left( \mu \frac{\partial \bar{v}_j}{\partial x_k} \right)
- \rho \cdot \overline{v'_k v'_j} \right)
\end{equation}
```

(22)

```
\begin{equation}
r' = \frac{\overline{v'_1 v'_2}}{\sqrt{\bar{v'^2_1}}}
\backslash: \sqrt{\bar{v'^2_2}}
\end{equation}
```

(23)

```
\begin{equation}
\tau_{tur} = \rho l^2 \backslash; \mid \frac{\partial \bar{v}_1}{\partial x_2}
\mid \backslash; \frac{\partial \bar{v}_1}{\partial x_2}
\end{equation}
```

(24)

```
\begin{eqnarray}
\label{formel}
\ddot{R} &= & \frac{1}{\Psi} (V_r - \dot{R})
+ R \{\dot{\Phi}\}^2 \nonumber\backslash\backslash
\ddot{\Phi} &= & \bigl\lbrack \frac{1}{\Psi}
(V_{\varphi} - R \dot{\Phi})
- 2 \dot{R} \dot{\Phi} \bigr\rbrack \frac{1}{R} \backslash\backslash
\ddot{Z} &= & \frac{1}{\Psi} (V_Z - \dot{Z}) \nonumber\backslash\backslash
\end{eqnarray}
```

(25)

```
\begin{equation}
\hat{\chi}^2 = \frac{n(n-1)}{B(n-B)} \sum_{i=1}^k
\frac{(B_i - E_i)^2}{n_i} > \chi_{k-1;\alpha}^2
\end{equation}
```

(26)

```
\begin{equation}
V(r,\vartheta,\varphi)=\sum\limits_{l=0}^{\infty}
\frac{4\pi}{2l+1}
```



```
\sum\limits_{m=-1}^1 \, \, q_{1,m} \, \frac{Y_{1,m}}{\vartheta,\varphi)}\{r^{1+1}\}
\end{equation}
```

(27)

```
\begin{eqnarray}
\vec{q}= \left(\begin{array}{c}
q_{0,0}\\q_{1,1}\\q_{1,0}\\q_{1,-1}\\q_{2,2}\\q_{2,1}\\q_{2,0}\\q_{2,-1}\\q_{2,-2}
\end{array}\right)
\end{eqnarray}
```

(28)

```
\begin{equation}
q'_{1',m'}=\sum\limits_{o=0}^{\infty}\sum\limits_{p=-o}^on_{o,p} \, \, ; \, \, q_{o,p} \, \, ; \, \, (\nabla)_{1'+o,m'-p;o,p} \, \, ,
\frac{Y_{1'+o,m'-p}}{\vartheta_a,\varphi_a)}\{a^{1'+o+1}\}
\end{equation}
```

(29)

```
\begin{equation}
q_{1,m}=-q'_{1,m} R^{2l+1} \, \, ,
\frac{l(\epsilon_i-\epsilon_a)}{l(\epsilon_a+\epsilon_i)+\epsilon_a}
\end{equation}
```

(30)

```
\begin{eqnarray}
\vec{q'}_{1,1}=\vec{q'}_{1,0}+I_{2,1} \, \, ; \, \, \vec{q'}_{2,0} \\
\vec{q'}_{2,1}=\vec{q'}_{2,0}+I_{1,2} \, \, ; \, \, \vec{q'}_{1,0}
\end{eqnarray}
```

(31)

```
\begin{eqnarray}
\lefteqn{\nabla_{\pm 1} \sum\limits_{l=0}^{\infty} \frac{4\pi}{2l+1} \, \, ,} \\
\sum\limits_{m=-1}^1 \, \, : \, \, q_{1,m} \, \, : \, \, \frac{Y_{1,m}}{\vartheta,\varphi)}\{r^{1+1}\} \\
\nonumber \\
&=&\sum\limits_{l=0}^{\infty} \frac{4\pi}{2l+1} \, \, , \\
\sum\limits_{m=-1}^1 \, \, : \, \, (\tilde{\nabla}_{\pm 1})_{1,m} \\
\, \, , \, \, q_{1,m} \, \, : \\
\frac{Y_{1+1,m \pm 1}}{\vartheta,\varphi)}\{r^{1+2}\} \\
\end{eqnarray}
```

```
(32)
\begin{equation}
\underbrace{a + \overbrace{b + \cdots + y}^{\{123\}}
+ z}_{\{\alpha\beta\gamma\}}
\end{equation}
```

Lange Formeln mu"s man selbst in Zeilen umbrechen (33):

```
\begin{eqnarray}
\lefteqn{w + x + y + z = } \hspace{1cm} \nonumber\\
& \& a + b + c + d + e + f + \\
& \& g + h + i + j + k + l \nonumber
\end{eqnarray}
```

```
(34)
\begin{eqnarray*}
N_{+1}=\left(\begin{array}{*{7}{c@{\;}c}}
0 & 0 & 0 & 0 & 0 & 0 & 0\\
(\tilde{\nabla}_{+})_{0,0} & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & (\tilde{\nabla}_{+})_{1,1} & 0 & 0 & 0 & 0 & 0\\
0 & 0 & (\tilde{\nabla}_{+})_{1,0} & 0 & 0 & 0 & 0\\
0 & 0 & 0 & (\tilde{\nabla}_{+})_{1,-1} & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & (\tilde{\nabla}_{+})_{2,2} & 0 & 0\\
0 & 0 & 0 & 0 & 0 & (\tilde{\nabla}_{+})_{2,1} & 0\\
0 & 0 & 0 & 0 & 0 & 0 & (\tilde{\nabla}_{+})_{2,0}\\
0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}\right)
\end{eqnarray*}
```

J

Modem-Kommandos (Hayes)

Der nordamerikanische Modem-Hersteller *Hayes* hat einen Satz von Modem-Kommandos geschaffen, der sich weltweit durchgesetzt hat. Die meisten Modems verstehen eine Übermenge des Standard-Kommandosatzes von Hayes. Da diese Kommandos mit einer Ausnahme mit `at` (= attention please) beginnen, werden sie auch als `at`-Kommandos bezeichnet. Es dürfen Groß- oder Kleinbuchstaben verwendet werden, jedoch nicht gemischt. Zu dem Thema siehe auch <http://www.linuxhaven.de/dlhp/HOWTO/DE-Modem-HOWTO.html>.

<code>ata</code>	Answer Command, antworten
<code>atb0</code>	Select CCITT Communication Standard
<code>atb1</code>	Select USA/Canada Communication Standard
<code>atc</code>	Carrier Control Selection
<code>atd</code>	Dial Command, wählen
<code>ate</code>	Command State Character Echo Selection
<code>atf</code>	On-line State Character Echo Selection
<code>ath0</code>	Hook Command Options, auflegen
<code>ath1</code>	Hook Command Options, Anruf annehmen
<code>ati</code>	Internal Memory Tests
<code>atl</code>	Speaker Volume Level Selection, Lautstärke, von 0 bis 3
<code>atm</code>	Speaker On/Off Selection
<code>atn</code>	Negotiation of Handshake Options
<code>ato</code>	On-line Command
<code>atp</code>	Select Pulse Dialing Method, Pulswahl
<code>atq</code>	Result Code Display Options
<code>atsn=r</code>	Write to S-Register n
<code>atsn?</code>	Read from S-Register n
<code>att</code>	Select Tone Dialing Method, Tonwahl
<code>atv</code>	Result Code Format Options
<code>atw</code>	Negotiation Progress Message Selection
<code>atx</code>	Call Progress Options
<code>aty</code>	Long Space Disconnect Options

atz	Soft Reset, zurücksetzen
at&b	V.32 Auto Retrain Options
at&c	Data Carrier Detect Options
at&d	Data Terminal Ready Options
at&f	Recall Factory Profile
at&g	Guard Tone Selection
at&j	Jack Type Selection
at&k	Local Flow Control Options
at&l	Line Type Selection
at&o	PAD Channel Selection
at&q	Communications Mode Options
at&r	RTS/CTS Options
at&s	Data Set Ready Options
at&t	Test Options
at&u	Trellis Coding Options
at&v	View Configuration Profiles
at&w	Write Active Profile to Memory
at&x	Synchronous Transmit Clock Source
at&y	Select Stored Profile for Hard Reset
at&z	Store Telephone Number

Mehrere Kommandos dürfen zu einer Zeile vereinigt werden, wobei der `at`-Vorspann nur einmal erforderlich ist. Mit `<cr>` wird das Kommando abgeschlossen und die Ausführung veranlasst. Um im transparenten Modus (das heißt während des Übertragens) in den Befehlsmodus zu schalten, ohne die Verbindung abzubrechen, ist das Kommando `+++` gefolgt von einer Pause von mindestens 1 sec zu geben. Eine in Deutschland verwendbare Startsequenz lautet:

```
ATZ
AT S7=45 S0=0 V1 X3
ATDP 0721,38340
```

Die erste Zeile sorgt für ein Reset, dann wird das Register S7 auf 45 Sekunden gesetzt. Falls das Modem innerhalb dieser Zeit nach dem Wählvorgang keine Antwort bekommt, legt es wieder auf. Der Wert 0 im Register S0 bewirkt, dass das Modem ankommende Anrufe nicht beantwortet, also nur für abgehende Verbindungen eingerichtet ist. Das Kommando V1 teilt dem Modem mit, Statusmeldungen (result codes) in Wortform, nicht als Zahlen auszugeben. X3 schließlich legt ein bestimmtes Antwortverhalten des Modems fest. Die letzte Zeile veranlasst die Wahl der Nummer 0721-Pause-38340 im Pulswählverfahren. Geht alles gut, pfeift ein Modem des Rechenzentrums der Universität Karlsruhe zurück.

K

ISO 3166 Ländercodes

Bei den Namen von Computern im Internet ist es außerhalb der USA üblich, als letzten Teil den Ländercode nach ISO 3166 anzugeben. Dies ist nicht zwingend, es gibt Bezeichnungen, die eine Organisation angeben und daher nicht von ISO 3166 erfasst werden. Die vollständige Tabelle findet sich unter <ftp://ftp.ripe.net/iso3166-countrycodes>. Die Internet-TLD EU (Europa) erscheint nicht in der Länderliste.

Land	A 2	A 3	Nummer
ALBANIA	AL	ALB	008
ALGERIA	DZ	DZA	012
ANDORRA	AD	AND	020
ARGENTINA	AR	ARG	032
AUSTRALIA	AU	AUS	036
AUSTRIA	AT	AUT	040
BAHAMAS	BS	BHS	044
BELARUS	BY	BLR	112
BELGIUM	BE	BEL	056
BOLIVIA	BO	BOL	068
BOSNIA AND HERZEGOWINA	BA	BIH	070
BRAZIL	BR	BRA	076
BULGARIA	BG	BGR	100
CANADA	CA	CAN	124
CHILE	CL	CHL	152
CHINA	CN	CHN	156
CHRISTMAS ISLAND	CX	CXR	162
COCOS ISLANDS	CC	CCK	166
COLOMBIA	CO	COL	170
CROATIA	HR	HRV	191
CYPRUS	CY	CYP	196
CZECH REPUBLIC	CZ	CZE	203

DENMARK	DK	DNK	208
ECUADOR	EC	ECU	218
EGYPT	EG	EGY	818
ESTONIA	EE	EST	233
FAROE ISLANDS	FO	FRO	234
FINLAND	FI	FIN	246
FRANCE	FR	FRA	250
GEORGIA	GE	GEO	268
GERMANY	DE	DEU	276
GIBRALTAR	GI	GIB	292
GREECE	GR	GRC	300
GREENLAND	GL	GRL	304
HONG KONG	HK	HKG	344
HUNGARY	HU	HUN	348
ICELAND	IS	ISL	352
INDIA	IN	IND	356
INDONESIA	ID	IDN	360
IRELAND	IE	IRL	372
ISRAEL	IL	ISR	376
ITALY	IT	ITA	380
JAPAN	JP	JPN	392
KOREA, REPUBLIC OF	KR	KOR	410
LATVIA	LV	LVA	428
LEBANON	LB	LBN	422
LIECHTENSTEIN	LI	LIE	438
LITHUANIA	LT	LTU	440
LUXEMBOURG	LU	LUX	442
MACEDONIA, REPUBLIC OF	MK	MKD	807
MALAYSIA	MY	MYS	458
MALTA	MT	MLT	470
MEXICO	MX	MEX	484
MONACO	MC	MCO	492
MOROCCO	MA	MAR	504
NETHERLANDS	NL	NLD	528
NEW ZEALAND	NZ	NZL	554
NIUE	NU	NIU	570
NORWAY	NO	NOR	578
PAKISTAN	PK	PAK	586
PERU	PE	PER	604
POLAND	PL	POL	616
PORTUGAL	PT	PRT	620
QATAR	QA	QAT	634
ROMANIA	RO	ROM	642
RUSSIAN FEDERATION	RU	RUS	643
SAN MARINO	SM	SMR	674

SAUDI ARABIA	SA	SAU	682
SINGAPORE	SG	SGP	702
SLOVAKIA	SK	SVK	703
SLOVENIA	SI	SVN	705
SOUTH AFRICA	ZA	ZAF	710
SPAIN	ES	ESP	724
SRI LANKA	LK	LKA	144
SWEDEN	SE	SWE	752
SWITZERLAND	CH	CHE	756
TAIWAN	TW	TWN	158
TONGA	TO	TON	776
TRINIDAD + TOBAGO	TT	TTO	780
TUNISIA	TN	TUN	788
TURKEY	TR	TUR	792
UKRAINE	UA	UKR	804
UNITED KINGDOM	GB	GBR	826
UNITED STATES	US	USA	840
URUGUAY	UY	URY	858
VATICAN	VA	VAT	336
VENEZUELA	VE	VEN	862
YUGOSLAVIA	YU	YUG	891

L

Requests For Comments (RFCs)

Das Internet wird durch RFCs (Request For Comments) beschrieben, Mitte 2005 etwa 4100 an der Zahl. Wird ein RFC durch einen neueren abgelöst, bekommt dieser eine neue, höhere Nummer. Die BCPs und FYIs behalten ihre Nummer, auch wenn dahinter ein neuer RFC steht. Sucht man eine Information, besorgt man sich einen Index der RFCs und startet in den Titeln, bei der höchsten Nummer beginnend, eine Stichwortsuche. Einige RFCs sind zugleich FYIs (For Your Information) mit eigener Zählung. Diese enthalten einführende Informationen. Andere RFCs sind zugleich BCPs (Best Current Practice), siehe RFC 1818, oder RTRs (RARE Technical Report) der Réseaux Associés pour la Recherche Européenne. Aktuelle, vollständige Sammlungen finden sich auf <ftp://ftp.nic.de/pub/rfc/>, <http://www.faqs.org/rfcs/>, <http://rfc.net/> oder <http://www.zvon.org/>, ähnlich auch bei <http://www.rfc-editor.org/>. Die Dateien mit den Übersichten sind:

- Request For Comments: `rfc-index.txt`,
- For Your Information: `fyi-index.txt`,
- Internet Standard: `std-index.txt`,
- Best Current Practice: `bcp-index.txt`.

Zusammenfassungen der Entwürfe (Internet Drafts) zu kommenden RFCs finden sich bei <http://www.ietf.org/ietf/lid-abstracts.txt>. Genießen Sie RFCs, die an einem ersten April erschienen sind, mit Vorsicht.

L.1 Alle FYIs

- 0001 FYI on FYI: Introduction to the FYI notes (1990) (RFC1150)
- 0002 FYI on a Network Management Tool Catalog: Tools for monitoring and debugging TCP/IP internets and interconnected devices (1993) (RFC1470)
- 0003 FYI on Where to Start: A bibliography of internetworking information (1991) (RFC1175)

- 0004 FYI on Questions and Answers - Answers to Commonly asked New
Internet User Questions (1999) (RFC2664)
- 0005 Choosing a name for your computer (1991) (RFC1178)
- 0006 FYI on the X Window System (1991) (RFC1198)
- 0007 FYI on Questions and Answers: Answers to commonly asked
experienced Internet user questions (1991) (RFC1207)
- 0008 Site Security Handbook (1997) (RFC2196)
- 0009 Who's Who in the Internet: Biographies of IAB, IESG and
IRSG Members (1992) (RFC1336)
- 0010 There is Gold in them thar Networks! or Searching for Treasure
in all the Wrong Places (1993) (RFC1402)
- 0011 X.500 Implementations Catalog-96 (1997) (RFC2116)
- 0012 Building a Network Information Services Infrastructure (1992) (RFC1302)
- 0013 Executive Introduction to Directory Services Using the
X.500 Protocol (1992) (RFC1308)
- 0014 Technical Overview of Directory Services Using the
X.500 Protocol (1992) (RFC1309)
- 0015 Privacy and Accuracy Issues in Network Information Center
Databases (1992) (RFC1355)
- 0016 Connecting to the Internet – What Connecting Institutions
Should Anticipate (1992) (RFC1359)
- 0017 The Tao of IETF - A Guide for New Attendees of the Internet
Engineering Task Force (2001) (RFC3160)
- 0018 Internet Users' Glossary (1996) (RFC1983)
- 0019 FYI on Introducing the Internet – A Short Bibliography of
Introductory Internetworking Readings (1993) (RFC1463)
- 0020 FYI on What is the Internet (1993) (RFC1462)
- 0021 A Survey of Advanced Usages of X.500 (1993) (RFC1491)
- 0022 Frequently Asked Questions for Schools (1996) (RFC1941)
- 0023 Guide to Network Resource Tool (1994) (RFC1580)
- 0024 How to Use Anonymous FTP (1994) (RFC1635)
- 0025 A Status Report on Networked Information Retrieval:
Tools and Groups (1994) (RFC1689)
- 0026 K-12 Internetworking Guidelines (1994) (RFC1709)
- 0027 Tools for DNS debugging (1994) (RFC1713)
- 0028 Netiquette Guidelines (1995) (RFC1855)
- 0029 Catalogue of Network Training Materials (1996) (RFC2007)
- 0030 A Primer On Internet and TCP/IP Tools and Utilities (1997) (RFC2151)
- 0031 Humanities and Arts: Sharing Center Stage on the Internet
(1997) (RFC2150)
- 0032 Hobbes' Internet Timeline (1997) (RFC2235)
- 0033 Some Testing Tools for TCP Implementors (1998) (RFC2398)
- 0034 Users' Security Handbook (1999) (RFC2504)
- 0035 DON'T SPEW. A Set of Guidelines for Mass Unsolicited Mailings and
Postings (spam*) (1999) (RFC2635)

- 0036 Internet Security Glossary (2000) (RFC2828)
- 0037 Guide to Administrative Procedures of the Internet Infrastructure
(2000) (RFC2901)
- 0038 How to Advertise Responsibly Using E-Mail and Newsgroups
(2001) (RFC3098)

L.2 Alle BCPs

- 0001 Best Current Practices (1995) (RFC1818)
- 0002 Addendum to RFC 1602 – Variance Procedure (1995) (RFC1871)
- 0003 Variance for The PPP Connection Control Protocol and The PPP
Encryption Control Protocol (1996) (RFC1915)
- 0004 An Appeal to the Internet Community to Return Unused
IP Networks (Prefixes) to the IANA (1996) (RFC1917)
- 0005 Address Allocation for Private Internets (1996) (RFC1918)
- 0006 Guidelines for creation, selection, and registration of an
Autonomous System (AS) (1996) (RFC1930)
- 0007 Implications of Various Address Allocation Policies for
Internet Routing (1996) (RFC2008)
- 0008 IRTF Research Group Guidelines and Procedures (1996) (RFC2014)
- 0009 The Internet Standards Process – Revision 3 (1996) (RFC2026)
- 0010 IAB and IESG Selection, Confirmation, and Recall Process Operation
of the Nominating and Recall Committees (2000) (RFC2727)
- 0011 The Organizations Involved in the IETF Standards Process
(1996) (RFC2028)
- 0012 Internet Registry IP Allocation Guidelines (1996) (RFC2050)
- 0013 Multipurpose Internet Mail Extension (MIME) Part Four:
Registration Procedures (1996) (RFC2048)
- 0014 Key words for use in RFCs to Indicate Requirement Levels
(1997) (RFC2119)
- 0015 Deployment of the Internet White Pages Service (1997) (RFC2148)
- 0016 Selection and Operation of Secondary DNS Servers (1997) (RFC2182)
- 0017 Use of DNS Aliases for Network Services (1997) (RFC2219)
- 0018 IETF Policy on Character Sets and Languages (1998) (RFC2277)
- 0019 IANA Charset Registration Procedures (2000) (RFC2978)
- 0020 Classless IN-ADDR.ARPA delegation (1998) (RFC2317)
- 0021 Expectations for Computer Security Incident Response (1998)
(RFC2350)
- 0022 Guide for Internet Standards Writers (1998) (RFC2360)
- 0023 Administratively Scoped IP Multicast (1998) (RFC2365)
- 0024 RSVP over ATM Implementation Guidelines (1998) (RFC2379)
- 0025 IETF Working Group Guidelines and Procedures (1998) (RFC2418)
- 0026 Guidelines for Writing an IANA Considerations Section in RFCs
(1998) (RFC2434)

494	L Requests For Comments (RFCs)
0027	Advancement of MIB specifications on the IETF Standards Track (1998) (RFC2438)
0028	Enhancing TCP Over Satellite Channels using Standard Mechanisms (1999) (RFC2488)
0029	Procedure for Defining New DHCP Options (1999) (RFC2489)
0030	Anti-Spam Recommendations for SMTP MTAs (1999) (RFC2505)
0031	Media Feature Tag Registration Procedure (1999) (RFC2506)
0032	Reserved Top Level DNS Names (1999) (RFC2606)
0033	URN Namespace Definition Mechanisms (1999) (RFC2611)
0034	Changing the Default for Directed Broadcasts in Routers (1999) (RFC2644)
0035	Registration Procedures for URL Scheme Names (1999) (RFC2717)
0036	Guidelines for Writers of RTP Payload Format Specifications (1999) (RFC2736)
0037	IANA Allocation Guidelines For Values In the Internet Protocol and Related Headers (2000) (RFC2780)
0038	Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing (2000) (RFC2827)
0039	Charter of the Internet Architecture Board (IAB) (2000) (RFC2850)
0040	Root Name Server Operational Requirements (2000) (RFC2870)
0041	Congestion Control Principles (2000) (RFC2914)
0042	Domain Name System (DNS) IANA Considerations (2000) (RFC2929)
0043	Procedures and IANA Guidelines for Definition of New DHCP Options and Message Types (2000) (RFC2939)
0044	Use of HTTP State Management (2000) (RFC2964)
0045	IETF Discussion List Charter (2000) (RFC3005)
0046	Recommended Internet Service Provider Security Services and Procedures (2000) (RFC3013)
0047	Tags for the Identification of Languages (2001) (RFC3066)
0048	End-to-end Performance Implications of Slow Links (2001) (RFC3150)
0049	Delegation of IP6.ARPA (2001) (RFC3152)
0050	End-to-end Performance Implications of Links with Errors (2001) (RFC3155)
0051	IANA Guidelines for IPv4 Multicast Address Assignments (2001) (RFC3171)
0052	Management Guidelines & Operational Requirements for the Address and Routing Parameter Area Domain (arpa) (2001) (RFC3172)
0053	GLOP Addressing in 233/8 (2001) (RFC3180)
0054	IETF Guidelines for Conduct () (RFC3184)
0055	Guidelines for Evidence Collection and Archiving () (RFC3227)
0056	On the use of HTTP as a Substrate () (RFC3205)
0057	IANA Considerations for IPv4 Internet Group Management Protocol (IGMP) () (RFC3228)
0058	Defining the IETF () (RFC3233)

- 0059 A Transient Prefix for Identifying Profiles under Development () (RFC3349)
- 0060 Inappropriate TCP Resets Considered Harmful () (RFC3360)
- 0061 Strong Security Requirements for Internet Engineering Task Force Standard Protocols () (RFC3365)
- 0062 Advice to link designers on link Automatic Repeat reQuest (ARQ) () (RFC3366)
- 0063 Session Initiation Protocol (SIP) for Telephones (SIP-T): Context and Architectures () (RFC3372)
- 0064 Internet Assigned Numbers Authority (IANA) Considerations for the Lightweight Directory Access Protocol (LDAP) (????) (RFC3383)
- 0065 Dynamic Delegation Discovery System (DDS) Part Five: URLARPA Assignment Procedures (2002) (RFC3405)
- 0066 Uniform Resource Names (URN) Namespace Definition Mechanisms (2002) (RFC3406)
- 0067 Change Process for the Session Initiation Protocol (SIP) (2002) (RFC3427)
- 0068 Layer Two Tunneling Protocol (L2TP) IANA Considerations Update (2002) (RFC3438)
- 0069 TCP Performance Implications of Network Path Asymmetrie (2002) (RFC3449)
- 0070 Guidelines for the Use of XML within IETF Protocols (2003) (RFC3470)
- 0071 TCP over Second (2.5G) and Third (3G) Generation Wireless Networks (2003) (RFC3481)
- 0072 Guidelines for Writing RFC Text on Security Considerations (2003) (RFC3552)
- 0073 An IETF URN Sub-namespace for Registered Protocol Parameters (2003) (RFC3553)
- 0074 Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework (2003) (RFC3584)
- 0075 Session Initiation Protocol (SIP). Basic Call Flow Examples (2003) (RFC3665)
- 0076 Session Initiation Protocol (SIP). Public Switched Telephone Network (PSTN) Call Flows. (2003) (RFC3666)
- 0077 IETF ISOC Board of Trustee Appointment Procedures (2003) (RFC3677)
- 0078 IETF Rights in Contributions (2004) (RFC3667)
- 0079 Intellectual Property Rights in IETF Technology (2004) (RFC3668)
- 0080 Delegation of E.F.F.3.IP6.ARPA. (2004) (RFC3681)
- 0081 The IETF XML Registry. (2004) (RFC3688)
- 0082 Assigning Experimental and Testing Numbers Considered Useful (2004) (RFC3692)
- 0083 A Practice for Revoking Posting Rights to IETF Mailing Lists (2004) (RFC3683)
- 0084 Ingress Filtering for Multihomed Networks (2004) (RFC3704)
- 0085 Best Current Practices for Third Party Call Control (3pcc) in the Session Initiation Protocol (SIP) (2004) (RFC3725)

- 496 L Requests For Comments (RFCs)
- 0086 Determining Strength for Public Keys Used for Exchanging
Symmetric Keys. (2004) (RFC3766)
- 0087 Use of Interior Gateway Protocol (IGP) Metric as a second MPLS
Traffic Engineering (TE) Metric (2004) (RFC3785)
- 0088 IANA Considerations for the Point-to-Point Protocol (PPP)
(2004) (RFC3818)
- 0089 Advice for Internet Subnetwork Designers (2004) (RFC3819)
- 0090 Registration Procedures for Message Header Fields, (2004) (RFC3864)
- 0091 DNS IPv6 Transport Operational Guidelines, (2204) (RFC3901)
- 0092 The IESG and RFC Editor Documents: Procedures, (2004) (RFC3932)

L.3 Einige 1.-April-RFCs

- 0748 Telnet randomly-lose option (1978)
- 0852 ARPANET short blocking feature (1983)
- 1097 Telnet subliminal-message option (1989)
- 1147 FYI on a Network Management Tool Catalog: Tools for Monitoring and
Debugging TCP/IP Internets and Interconnected Devices (1990)
- 1149 Standard for the transmission of IP datagrams on avian carriers (1990)
- 1151 Version 2 of the Reliable Data Protocol (RDP) (1990)
- 1217 Memo from the Consortium for Slow Commotion Research (1991)
- 1438 IETF Statements of Boredom (SOBs) (1993)
- 1605 SONET to Sonnet Translation (1994)
- 1606 A Historical Perspective On The Usage Of IP Version 9 (1994)
- 1607 A View from the 21st Century (1994)
- 1776 The Address is the Message (1995)
- 1925 The Twelve Networking Truths (1996)
- 1926 An Experimental Encapsulation of IP Datagrams on Top of ATM (1996)
- 2321 RITA – The Reliable Internetwork Troubleshooting Agent (1998)
- 2324 Hypertext Coffee Pot Control Protocol HTCPCP/1.0 (1998)
- 2325 Definitions of Managed Objects for Drip-Type Heated Beverage
Hardware Devices Using SMIV2 (1998)
- 2549 IP Over Avian Carriers with Quality of Service (1999)
- 2550 Y10K and Beyond (1999)
- 2551 The Roman Standards Process – Revision III (1999)
- 2795 The Infinite Monkey Protocol Suite (2000)
- 3091 Pi Digit Generation Protocol (2001)
- 3092 Etymology of *Foo* (2001)
- 3093 Firewall Enhancement Protocol (FEP) (2001)
- 3251 Electricity over IP (2002)
- 3252 Binary Lexical Octet Ad-hoc Transport (2002)
- 3514 The Security Flag in the IPv4 Header (2003)
- 3751 Omniscience Protocol Requirements (2004)
- 4041 Requirements for Morality Sections in Routing Area Drafts (2005)
- 4042 UTF-9 and UTF-18, Efficient Transformation Formats of Unicode (2005)

M

Zeittafel

Ausführlichere Angaben sind den im Anhang N *Zum Weiterlesen* in Abschnitt *Geschichte* auf Seite 516 aufgeführten Werken zu entnehmen. Die meisten Errungenschaften entwickelten sich über manchmal lange Zeitspannen, so dass vor einige Jahreszahlen *um etwa* zu setzen ist.

-10E8	Der beliebte Tyrannosaurus hatte zwei Finger an jeder Hand und rechnete vermutlich im Dualsystem, wenn überhaupt.
-2000	Die Babylonier verwenden für besondere Aufgaben ein gemischtes Stellenwertsystem zur Basis 60.
-400	In China werden Zählstäbchen zum Rechnen verwendet.
20	In der Bergpredigt wird das Binärsystem erwähnt (Matth. 5, 37). Die Römer schieben Rechensteinchen (calculi).
600	Die Inder entwickeln das heute übliche reine Stellenwertsystem, die Null ist jedoch älter. Etwa gleichzeitig entwickeln die Mayas in Mittelamerika ein Stellenwertsystem zur Basis 20.
1200	LEONARDO VON PISA, genannt FIBONACCI, setzt sich für die Einführung des indisch-arabischen Systems im Abendland ein.
1550	Die europäischen Rechenmeister verwenden sowohl die römische als auch die indisch-arabische Schreibweise.
1617	JOHN NAPIER erfindet die Rechenknochen (Napier's Bones).
1623	Erste mechanische Rechenmaschine mit Zehnerübertragung und Multiplikation, von WILHELM SCHICKARD, Tübingen.
1642	Rechenmaschine von BLAISE PASCAL, Paris für kaufmännische Rechnungen seines Vaters.
1674	GOTTFRIED WILHELM LEIBNIZ baut eine mechanische Rechenmaschine für die vier Grundrechenarten und befasst sich mit der dualen Darstellung von Zahlen. In der Folgezeit technische Verbesserungen an vielen Stellen in Europa.
1714	HENRY MILL erhält ein Patent auf eine Schreibmaschine.
1801	JOSEPH MARIE JACQUARD erfindet die Lochkarte und steuert Webstühle damit.

- 1821 CHARLES BABBAGE stellt der Royal Astronomical Society eine programmierbare mechanische Rechenmaschine vor, die jedoch keinen wirtschaftlichen Erfolg hat. Er denkt auch an das Spielen von Schach oder Tic-tac-toe auf Maschinen.
- 1840 SAMUEL FINLEY BREEZE MORSE entwickelt einen aus zwei Zeichen plus Pausen bestehenden Telegrafencode, der die Buchstaben entsprechend ihrer Häufigkeit codiert.
- 1847 GEORGE BOOLE entwickelt die symbolische Logik.
- 1861 JOHANN PHILIPP REIS erfindet das Telephon.
- 1873 ELIPHALET REMINGTON and Sons, NY, stellen außer Gewehren und Nähmaschinen auch Schreibmaschinen her. 1886 trennen sie sich vom Schreibmaschinenbereich, der später den Namen Remington Rand und noch später den Namen Sperry Rand trägt.
- 1876 ALEXANDER GRAHAM BELL erhält ein Patent auf sein Telefon.
- 1877 Gründung der Bell Telephone Company.
- 1885 Aus Bell Telephone Co. wird American Telephone + Telegraph Co.
- 1890 HERMAN HOLLERITH erfindet die Lochkartenmaschine und setzt sie bei einer Volkszählung in den USA ein. Das ist der Anfang von IBM.
- 1894 OTTO LUEGERS *Lexikon der gesamten Technik* führt unter dem Stichwort *Elektrizität* als Halbleiter Äther, Alkohol, Holz und Papier auf.
- 1895 Erste Übertragungen mittels Radio (ALEXANDER POPOW, GUGLIELMO MARCONI).
- 1896 Gründung der Tabulating Machine Company, der späteren IBM.
- 1898 VALDEMAR POULSEN erfindet die magnetische Aufzeichnung von Tönen (*Telegraphon*).
- 1900 01. Januar 1900 00:00:00 GMT Nullpunkt der gegenwärtigen NTP-Ära (eine NTP-Ära umfasst 136 Jahre).
- 1910 Gründung der Deutschen Hollerith Maschinen GmbH, Berlin, der Vorläuferin der IBM Deutschland.
- 1918 Das Enigma-Verschlüsselungsverfahren entwickelt.
- 1924 Aus der Tabulating Machine Company von HERMAN HOLLERITH, später in Computing-Tabulating-Recording Company umbenannt, wird die International Business Machines (IBM).
- 1930 EDWIN LINK baut – anstatt Pianos und Orgeln wie sein Vater – einen mechanischen Flugsimulator für Übungs- und Vergnügungszwecke und erhält ein Patent darauf. Der Link-Trainer erlangt Verbreitung.
- 1932 Die Firma AEG baut das erste Tonbandgerät mit dem Namen *Magnetophon*. Die Bänder dazu liefert die BASF.
- 1937 ALAN TURING veröffentlicht sein Computermodell.
- 1938 KONRAD ZUSE stellt den programmgesteuerten Rechner Z 1 fertig. Ein späterer Nachbau schafft 1 Rechenoperation pro Sekunde. Elektronische binäre Addiermaschine von JOHN VINCENT ATANASOFF und CLIFFORD BERRY, Iowa State University, zur Lösung linearer Gleichungssysteme.

- 1939 KONRAD ZUSE stellt die Z 2 fertig.
Gründung der Firma Hewlett-Packard, Palo Alto, Kalifornien durch WILLIAM HEWLETT und DAVID PACKARD. Ihr erstes Produkt ist ein Oszillator für Tonfrequenzen (Messtechnik).
- 1941 KONRAD ZUSE stellt die Z 3 fertig.
- 1942 Die Purdue University beginnt mit der Halbleiterforschung und untersucht Germaniumkristalle.
- 1943 Der Computer *Colossus*, Bletchley Park/Buckinghamshire UK, entschlüsselt deutsche Militärnachrichten (Enigma).
IBM-Chef THOMAS WATSON schätzt den weltweiten Bedarf an Computern auf 5 (fünf) Stück.
- 1944 Die Zuse Z 4 wird fertig (2200 Relais, mechanischer Speicher).
Sie arbeitet von 1950 bis 1960 in der Schweiz.
An der Harvard University bauen HOWARD AIKEN und GRACE HOPPER die Mark I in Relaistechnik. Die Maschine läuft bis 1959.
- 1945 KONRAD ZUSE entwickelt den Plankalkül, die erste höhere Programmiersprache.
WILLIAM BRADFORD SHOCKLEY startet ein Forschungsprojekt zur Halbleiterphysik in den Bell-Labs.
VANNEVAR BUSH entwickelt ein System zur Informationsspeicherung und -suche, das auf Mikrofilmen beruht.
- 1946 JOHN VON NEUMANN veröffentlicht sein Computerkonzept.
JOHN PRESPEER ECKERT und JOHN WILLIAM MAUCHLY bauen in den USA die ENIAC (Electronic Numerical Integrator and Calculator). Die ENIAC rechnet dezimal, enthält 18000 Vakuumröhren, wiegt 30 t, ist 5,5 m hoch und 24 m lang, braucht für eine Addition 0,2 ms, ist an der Entwicklung der Wasserstoffbombe beteiligt und arbeitet bis 1955. Sie ist der Urahne der UNIVAC.
- 1948 CLAUDE ELWOOD SHANNON begründet die Informationstheorie.
JOHN BARDEEN, WALTER HOUSER BRATTAIN und WILLIAM BRADFORD SHOCKLEY entwickeln in den Bell-Labs den Transistor, der 10 Jahre später die Vakuumröhre ablöst.
- 1949 Erster Schachcomputer: Manchester MADM. Das Wort *Bit* kreiert.
- 1950 An der ETH Zürich geht die Zuse Z 4 in Betrieb.
- 1952 IBM bringt ihre erste elektronische Datenverarbeitungsanlage, die IBM 701, heraus.
- 1953 IBM baut die erste Magnetbandmaschine zur Datenspeicherung (726).
- 1954 Remington-Rand bringt die erste UNIVAC heraus, IBM die 650.
Silizium beginnt, das Germanium zu verdrängen.
- 1955 IBM entwickelt die erste höhere Programmiersprache, die Verbreitung erlangt: FORTRAN (Formula Translator) und verwendet Transistoren in ihren Computern.
- 1956 KONRAD ZUSE baut die Z 22, die mit Röhren arbeitet.
Sie kommt 1958 auf den Markt. Bis 1961 werden 50 Stück verkauft.

- BARDEEN, BRATTAIN und SHOCKLEY erhalten den Nobelpreis für Physik.
- IBM stellt die erste Festplatte vor (IBM 350 Disk File für den Computer RAMAC 305), Kapazität 5 MByte, groß wie ein Schrank, Gewicht 1 to, bestehend aus 50 Scheiben zu 24 Zoll, 50.000 US-\$.
 1957 Die IBM 709 braucht für eine Multiplikation 0,12 ms.
 Weltweit arbeiten rund 1300 Computer.
 Seminar von Prof. JOHANNES WEISSINGER über *Programm-gesteuerte Rechenmaschinen* im SS 1957 der TH Karlsruhe.
 KARL STEINBUCH (Firma SEL, später TH Karlsruhe) prägt den Begriff *Informatik*.
- Erster Satellit (Sputnik, Sowjetunion) kreist um die Erde.
 1958 Als eine Reaktion auf den Sputnik gründet das us-amerikanische Verteidigungsministerium (DoD) die Denkfabrik Advanced Research Projects Agency (ARPA), die später das ARPANET aufbaut.
 MARVIN LEE MINSKY prägt den Begriff *Artificial Intelligence*.
 Die TH Karlsruhe erhält ihren ersten Computer, eine ZUSE Z 22, finanziert vom Land Baden-Württemberg.
 Die Maschine verwendet 400 Vakuumröhren und wiegt 1 t. Der Arbeitsspeicher fasst 16 Wörter zu 38 Bits, d. h. 76 Byte. Der Massenspeicher, eine Magnettrommel, fasst rund 40 KByte. Eine Gleitkommaoperation dauert 70 ms. Das System versteht nur Maschinensprache (Freiburger Code) und läuft bis 1972.
 Im SS 1958 hält Priv.-Doz. KARL NICKEL (Institut für Angew. Mathematik) eine Vorlesung *Programmieren mathematischer und technischer Probleme für die elektronische Rechenmaschine Z 22*.
 Die Programmiersprache ALGOL 58 kommt heraus.
 Bei Texas Instruments baut JACK ST. CLAIR KILBY den ersten IC; im Jahr 2000 erhält er dafür den Nobelpreis für Physik.
- 1959 Im SS 1959 hält Priv.-Doz. KARL NICKEL erstmals die Vorlesung *Programmieren I*, im WS 1959/60 die Vorlesung *Programmieren II*. Erstes Werk von Hewlett-Packard in Deutschland. Siemens baut die Siemens 2002.
- 1960 Programmieren steht noch in keinem Studienplan, sondern ist freiwillig. Die Karlsruher Z 22 arbeitet Tag und Nacht.
 Die Programmiersprache COBOL wird veröffentlicht.
 Ein Computerspiel namens *Spacewar* läuft auf einer Digital Equipment Corporation (DEC) PDP-1 im MIT.
 ALAN SHUGART entwickelt ein Verfahren zur Aufzeichnung von Daten auf einer magnetisch beschichteten Scheibe.
- 1961 Die TH Karlsruhe erhält im Zuge der Berufungsverhandlungen von Prof. Nickel eine Zuse Z 23, die mit 2400 Transistoren arbeitet.
 Ihr Hauptspeicher fasst 240 Wörter zu 40 Bits.
 Eine Gleitkommaoperation dauert 15 ms. Außer Maschinensprache versteht sie ALGOL.
 Weltweit arbeiten etwa 7300 Computer.

- 1962 Die TH Karlsruhe erhält einen SEL ER 56, der bis 1968 läuft.
An der Purdue University wird die erste Fakultät für Informatik (Department of Computer Science) gegründet.
Texas Instruments und Fairchild nehmen die Serienproduktion von ICs (Chips) auf.
JOSEPH CARL ROBNETT LICKLIDER hat zwei Visionen: den interaktiven Computer und das galaktische Netz (wenn schon, denn schon). Er wird Direktor in der ARPA und geht an die Verwirklichung seiner Visionen.
- 1963 Weltweit arbeiten etwa 16.500 Computer.
Erster geostationärer Satellit (Syncom).
IVAN E. SUTHERLAND entwickelt in seiner Doktorarbeit am MIT das Sketchpad, einen grafischen Bildschirm mit Lichtgriffel, und wird damit zum Vater der Computergrafik.
- 1964 Die Programmiersprache BASIC erscheint.
DOUGLAS CARL ENGELBART erfindet am Stanford Research Institute die Maus und die Fenstertechnik.
IBM legt das Byte zu 8 Bits fest (IBM 360).
Ein Chip enthält auf $0,5 \text{ cm}^2$ 10 Transistoren.
- 1965 Beginn des Betriebssystems MULTICS bei MIT, Bell und General Electric. Aus dessen Misserfolg erwächst später UNIX.
- 1966 Die TH Karlsruhe erhält eine Electrologica X 8, die bis 1973 betrieben wird. Gründung des Karlsruher Rechenzentrums.
Hewlett-Packard steigt in die Computerei ein (HP 2116 A).
- 1967 Erster elektronischer Taschenrechner (Texas Instruments).
IVAN E. SUTHERLAND entwickelt an der Harvard University einen Helm mit binokularem Display und bringt damit die Virtual Reality ein gutes Stück voran.
- 1968 Am 26. Februar entscheiden sich maßgebende Vertreter der Computerwissenschaft im deutschsprachigen Raum im Anschluss an ein internationales Kolloquium in Dresden für die Bezeichnung *Informatik* nach französischem Beispiel.
Die Programmiersprache PASCAL kommt heraus. Die Firma Intel gegründet. Hewlett-Packard baut den ersten wissenschaftlichen programmierbaren Tischrechner (HP 9100 A).
- 1969 In Karlsruhe wird am 1. Januar das Institut für Informatik gegründet, Direktor KARL NICKEL. Im WS 1969/70 beginnt in Karlsruhe die Informatik als Vollstudium mit 91 Erstsemestern.
Gründung der Gesellschaft für Informatik (GI) in Bonn.
In den Bell Labs UNIX in Assembler auf einer DEC PDP 7.
Beginn des ARPANET-Projektes, erste Teilnehmer U. of California at Los Angeles, Stanford Research Institute, U. of California at Santa Barbara und U. of Utah, allesamt mit DEC PDP-10 Maschinen.
RFC 0001: Host Software, von STEVE CROCKER.

- 1970 Die Universität Karlsruhe erhält eine UNIVAC 1108, die bis 1987 läuft und damit den hiesigen Rekord an Betriebsjahren hält. Preis 23 MDM, 3 Zentraleinheiten, 256 Kilo-Wörter zu je 36 Bits Arbeitsspeicher, 20 Bildschirme.
Die Karlsruher Fakultät für Informatik wird gegründet.
Am 01. Januar 1970 00:00:00 GMT beginnt die UNIX-Uhr zu laufen.
- 1971 UNIX auf C umgeschrieben, erster Mikroprozessor (Intel 4004).
ALAN SHUGART entwickelt bei IBM die Floppy Disk.
Die Internet-Protokolle FTP (RFC 114) und Telnet (RFC 137) werden vorgeschlagen und diskutiert.
- 1972 IBM entwickelt das Konzept des virtuellen Speichers und stellt die 8-Zoll-Floppy-Disk vor. Xerox (ROBERT METCALFE), DEC und Intel entwickeln den Ethernet-Standard.
Das ARPANET wird der Öffentlichkeit vorgestellt.
Ein Student namens STEPHAN G. WOZNIAK lötet sich einen Computer zusammen, der den Smoke-Test nicht übersteht.
In der Bundesrepublik arbeiten rund 8.200 Computer.
Erster wissenschaftlicher Taschenrechner (Hewlett-Packard 35).
- 1973 Erste internationale Teilnehmer am ARPANET: NORSAR (Norwegian Seismic Array), Norwegen und U. College of London.
- 1974 Der erste programmierbare Taschenrechner kommt auf den Markt (Hewlett-Packard 65), Preis 2500 DM.
- 1975 UNIX wird veröffentlicht (Version 6), Beginn der BSD-Entwicklung.
Die Zeitschrift *Byte* wird gegründet.
Erste, mäßig erfolgreiche Personal Computer (Xerox, Altair).
Die Firma Microsoft Corporation von WILLIAM HENRY GATES III. und PAUL ALLEN gegründet.
- 1976 STEVEN P. JOBS und STEPHAN G. WOZNIAK gründen die Firma Apple und bauen den Apple I. Er kostet 666,66 Dollar.
ALAN SHUGART stellt die 5,25-Zoll-Diskette vor.
Die nichtprozedurale Datenbanksprache SQL – entwickelt von EDGAR F. CODD bei IBM – wird veröffentlicht.
- 1977 ROBERT KAHN und VINTON G. CERF veröffentlichen das Konzept von TCP/IP, anfangs Kahn-Cerf-Protokolle genannt.
- 1978 In der Bundesrepublik arbeiten rund 170.000 Computer.
Der Commodore PET 2001 – ein Vorläufer des C64 – kommt heraus, 4 bis 32 kbyte Arbeitsspeicher, Bildschirm 25 Zeilen zu 40 Zeichen.
Erste Tabellenkalkulation *Visicalc*, für den Apple II, von DAN BRICKLIN und BOB FRANKSTON, Harvard University.
Erste Fassung von \TeX (DONALD ERVIN KNUTH) veröffentlicht.
Das Network Time Protocol (NTP) wird in Gebrauch genommen.
- 1979 Faxdienst in Deutschland eingeführt.
Beginn des Usenet in der Duke University und der University of North Carolina auf der Basis von uucp-Verbindungen.

- Die Zusammenarbeit von Apple mit Rank Xerox führt zur Apple Lisa, ein Mißerfolg, aber der Wegbereiter für den Macintosh. Plattenherstellerfirma *Seagate* gegründet.
- Gründung der Satelliten-Kommunikations-Firma Inmarsat.
- BJARNE STROUSTRUP beginnt mit der Entwicklung von C++.
- Programmiersprache Ada veröffentlicht.
- Betriebssystem DOS für Intel 8086/8088 von Fa. Seattle Computer Products entwickelt, später von Microsoft erworben.
- 1980 Erster Jugendprogrammier-Wettbewerb der GI.
- Erster Home-Computer: Sinclair ZX-80, für rund 500 DM.
- Sony führt die 3,5-Zoll-Diskette ein. In den Folgejahren entwickeln andere Firmen auch Disketten mit Durchmessern von 3 bis 4 Zoll.
- Microsoft bringt Xenix, ein UNIX für PCs, heraus.
- 1981 Die Universität Karlsruhe erhält eine Siemens 7881 als zentralen Rechner.
- IBM bringt in den USA den IBM-PC heraus mit PC-DOS 1.0 (MS DOS) als wichtigstem Betriebssystem.
- In Berlin wird der *Chaos Computer Club* gegründet.
- Xanadu-Projekt von TED NELSON, ein Vorläufer des Web.
- 1982 Die Firma Sun Microsystems wird gegründet, entscheidet sich für UNIX und baut die ersten Workstations.
- JIM CLARK gründet Silicon Graphics, Inc. (SGI)
- Beginn des EuNETs, einer der ersten deutschen Internet-Provider, an der Universität Dortmund.
- WILLIAM GIBSON prägt das Wort *Cyberspace*.
- MORTON HEILIG präsentiert einen Spielautomaten für Motorrad- und Auto-Simulationen mit Stereotonfilm, Gebläse, Gerüchen und vibrierenden Sitzen, echt multimedial, aber erfolglos, da zu teuer.
- 1983 Die Universität Karlsruhe erhält einen Vektorrechner Cyber 205 und eine Siemens 7865. Die Cyber leistet 400 Mio. Flops.
- Beginn des *Lokalen Informatiknetzes Karlsruhe* (LINK), ab 1984 Xlink, in der Fakultät für Informatik der Universität Karlsruhe.
- IBM bringt den PC auf den deutschen Markt.
- UNIX kommt als System V von AT&T in den Handel, die erste Ausgabe der Zeitschrift *Computertechnik* (c't) erscheint (Nr. 12/83 vom Oktober 1983).
- Gründung der X/Open-Gruppe.
- MS-DOS 2.0 (PC-DOS 2.0) und Novell Netware kommen heraus.
- Microsoft Windows wird angekündigt.
- Das ARPAnet wechselt von NCP auf TCP/IP.
- 1984 Der erste Apple Macintosh (128K) und der Hewlett-Packard Thinkjet, der erste Tintenstrahldrucker, kommen auf den Markt.
- GNU-Projekt von RICHARD MATTHEW STALLMAN gegründet.
- Der IBM PC/AT mit Prozessor Intel 80 286 und MS-DOS 3.0 kommen heraus.

- Siemens steigt in UNIX (Sinix) ein.
Die Universität Karlsruhe wird Email-Relay für Deutschland zum Computer Science Net (CSNet) in den USA. Als erste Mail erhält Prof. ZORN, U. Karlsruhe, einen Gruß vom CSNet.
Entwicklung des X Window Systems am MIT.
- 1985 MS-Windows 1.0, IBM 3090 und IBM Token Ring Netz.
XLink an der Universität Karlsruhe stellt als erstes deutsches Netz eine Verbindung zum nordamerikanischen ARPANET her.
Hewlett-Packard bringt den ersten Laserjet-Drucker heraus.
- 1986 Weltweit etwa eine halbe Million UNIX-Systeme und 3000 öffentliche Datenbanken.
Mit dem Computer-Investitionsprogramm des Bundes und der Länder (CIP) kommen mehrere HP 9000/550 unter UNIX an die Universität Karlsruhe.
- 1987 Microsoft XENIX (ein UNIX) für den IBM PC/AT
IBM bringt die PS/2-Reihe unter MS-OS/2 heraus.
Weltweit mehr als 5 Millionen Apple Computer und etwa 100 Millionen PCs nach Vorbild von IBM.
Das MIT veröffentlicht das X Window System Version 11 (X11).
In Berkeley wird die RAID-Technologie entwickelt.
Beginn des baden-württembergischen BelWue-Netzes.
- 1988 JARKKO OIKARINEN, Finnland, entwickelt den IRC.
Das Karlsruher Campusnetz KARLA wird durch das Glasfasernetz KLINK ersetzt. VBN-Strecke Karlsruhe - Stuttgart im BelWue-Netz.
Frankreich geht ans Internet (INRIA, Rocquencourt bei Paris).
Gründung der Open Software Foundation (OSF) und der UNIX International Inc. MS-DOS 4.0 für PCs.
Ein Internet-Wurm namens Morris geht auf die Reise, darauf hin Gründung des Computer Emergency Response Teams (CERT).
Erster Hoax (2400-baud-Modem-Hoax) im Internet, siehe CIAC.
Erstes landmobiles Satellitensystem für Datenfunk (Inmarsat-C).
- 1989 Das NFSNET löst das ARPANET als Backbone des Internet ab.
UNIX System V Release 4 vereinheitlicht System V, BSD und Xenix.
Im Rechenzentrum Karlsruhe löst die IBM 3090 die Siemens 7881 ab. ISDN in Deutschland eingeführt.
Erster deutscher Internet-Direktanschluss via Xlink, Karlsruhe.
- 1990 Zunehmende Vernetzung, Anschluss an weltweite Netze.
Die Internet Society (ISOC) schätzt das Internet auf 500.000 Knoten.
Computer-Kommunikation mittels E-Mail, Btx und Fax vom Arbeitsplatz aus. Optische Speichermedien (CD-ROM, WORM).
Das Web (URL, HTTP, HTML) von TIMOTHY BERNERS-LEE und ROBERT CAILLIAU am CERN in Genf entwickelt.
UNIX System V Version 4.
Die mittlere Computerdichte in technisch orientierten Instituten und Familien erreicht 1 pro Mitglied.

- 1991 Das UNIX-System OSF/1 mit dem Mach-Kernel der Carnegie-Mellon-Universität kommt heraus.
17. Sep.: Anfang von LINUX (LINUS BENEDICT TORVALDS).
Erster Web-Server in den USA: Stanford Linear Accelerator Center.
MS-DOS 5.0 für PCs. Anfänge von Microsoft Windows NT.
Das DE-NIC an der Universität Dortmund gegründet.
IBM, Apple und Motorola kooperieren mit dem Ziel, einen Power PC zu entwickeln.
- 1992 Die Universität Karlsruhe nimmt den massiv parallelen Computer MasPar 1216A mit 16000 Prozessoren in Betrieb.
Novell übernimmt von AT&T die UNIX-Aktivitäten (USL).
FORTRAN 90 verabschiedet.
Eine Million Knoten im Internet. Weltweit etwa 50 Web-Server.
Erster deutscher Web-Server, am DESY in Hamburg.
- 1993 MS-DOS Version 6.0. Microsoft kündigt Windows-NT an.
DEC stellt PC mit Alpha-Prozessor vor, 150 MHz, 14.000 DM.
Novell tritt das Warenzeichen UNIX an die X/Open-Gruppe ab.
MARC ANDREESSEN, NCSA, schreibt einen Web-Browser für das X Window System mit der Möglichkeit, farbige Grafiken darzustellen.
IAN MURDOCK, Student an der Purdue University, gründet das Debian-Projekt. Gegen Jahresende Debian GNU/Linux Version 0.90.
Weltweit etwa 250 Web-Server.
Das DE-NIC zieht ans Rechenzentrum der Universität Karlsruhe.
- 1994 Weltweit 10 Mio. installierte UNIX-Systeme prognostiziert.
Linux 1.0 veröffentlicht.
Das Internet umfasst etwa 4 Mio. Knoten und 20 Mio. Benutzer.
Erste Spam-Mail (Canter + Siegel). Erste Banner-Werbung (Wired).
MARC ANDREESSEN und JIM CLARK gründen die Firma Netscape.
- 1995 Kommerzielle Netze lösen in den USA das NFSNET als Backbone ab.
Die X/Open-Gruppe führt die Bezeichnung *UNIX 95* für Systeme ein, die der *Single UNIX Specification* genügen.
Die Universität Karlsruhe ermöglicht in Zusammenarbeit mit dem Oberschulamt nordbadischen Schulen den Zugang zum Internet. Ähnliche Projekte werden auch an einigen anderen Hoch- und Fachhochschulen durchgeführt.
Die Programmiersprache JAVA wird von Sun veröffentlicht.
Online-Auktionshaus *Ebay* als Sammlerbörse in den USA gegründet.
Weltweit etwa 50000 Web-Server.
- 1996 Die Massen und Medien entdecken das Internet.
Debian GNU/Linux Version 1.1 (*buzz*) wird veröffentlicht, zum Jahresende folgt Version 1.2 (*rex*).
FORTRAN 95, eine revidierte Fassung von FORTRAN 90, fertig.
Die Open Software Foundation (OSF) und X/Open schließen sich zur Open Group zusammen.

- 1997 100-Ethernet ist erschwinglich geworden, über das Gigabit-Ethernet wird geredet. In Deutschland gibt es rund 20 Mio. PCs und 1 Mio. Internetanschlüsse (Quelle: Fachverband Informationstechnik). Debian GNU/Linux Version 1.3 (*bo*) freigegeben, rund 1000 Pakete. Single UNIX Specification Version 2 im Web veröffentlicht. HTML 4.0 freigegeben.
Der Buchversender Amazon meldet ein Patent an dergestalt, dass man mit einem Mausklick im Internet eine Ware bestellt.
- 1998 Compaq übernimmt die Digital Equipment Corporation (DEC). IBM bringt DOS 2000 heraus, Microsoft kündigt Windows 2000 an. Debian GNU/Linux Version 2.0 (*hamm*) freigegeben, 1500 Pakete. KDE 1.0 veröffentlicht. 9-GB-Festplatten kosten 500 DM. Gigabit-Ethernet-Standard IEEE 802.3z verabschiedet. JONATHAN B. POSTEL, einer der Apostel des Internet und Autor vieler RFCs, gestorben. Siehe RFC 2441: *Working with Jon* und RFC 2468: *I Remember IANA*.
- 1999 Das Y2K-Problem – die Jahrtausendwende – beschäftigt die Gemüter, weil die Programmierer früherer Jahrzehnte mit den Bits knauserten. Der RFC 2550 löst auch gleich das Y10K-Problem. Debian GNU/Linux Version 2.1 (*slink*) kommt heraus. Betreiber großer Suchmaschinen schätzen die Anzahl der Web-Seiten weltweit auf 1 Milliarde.
LINUS B. TORVALDS wird Ehrendoktor der Universität Stockholm.
- 2000 Das Y2K-Problem hat sich praktisch nicht ausgewirkt. Den 29. Februar 2000 haben wir auch gut überstanden, einen Schalttag nach einer Regel, die nur alle 400 Jahre angewendet wird. Debian GNU/Linux Version 2.2 (*potato*) kommt heraus, 6500 Pakete. Microsoft Windows 2000 ist erhältlich. Ein Macro-Virus namens *Love Letter* sorgt für Aufregung – außerhalb der Linux/UNIX-Welt. Der Intel Pentium kommt bei einer Taktfrequenz von 1,5 GHz an. Zum Jahresende 2 Mio. Internet-Hosts in Deutschland (Quelle: RIPE).
- 2001 CLAUDE ELWOOD SHANNON gestorben, gilt als Erfinder des Bits und Begründer der Informationstheorie.
In den USA wird die Wikipedia gegründet, eine Online-Enzyklopädie.
- 2002 Die Einführung der Euro-Währung führt zu einem neuen Zeichen in internationalen Zeichensätzen. Debian GNU/Linux Version 3.0 (*woody*) wird als stabil freigegeben. Die Distribution umfasst 8700 Pakete.
- 2004 PCs werden zunehmend ohne Floppy-Laufwerk ausgeliefert. Politische Überlegungen, die Patentierung von Software zuzulassen, bedrohen die Open Source Welt.
- 2005 Anfang Juni wird Debian GNU/Linux Version 3.1 (*sarge*) nach reiflicher Überlegung als stabil freigegeben und löst 3.0r6 (*woody*) ab; die Distribution umfasst rund 15.000 Pakete.

Die deutsche Wikipedia nähert sich der Marke von 300.000 Einträgen.
Internet-Telefonie (Voice over IP) ist stark im Kommen.
Der weltweit leistungsfähigste Rechner – ein IBM Blue Gene – läuft
unter einem Linux (Quelle: <http://www.top500.org/>).

N

Zum Weiterlesen

Die Auswahl ist subjektiv und enthält Werke, die wir noch lesen wollen, schon gelesen haben oder oft benutzen. Einige Bücher können vergriffen oder inzwischen in neuerer Auflage erschienen sein.

1. Lexika, Glossare, Wörterbücher

- Fachausdrücke der Informationsverarbeitung Englisch - Deutsch - Englisch
IBM Deutschland, IBM-Form-Nr. GQ12-1044-1, 1985, 1700 S.

W. Alex Abkürzungs-Liste ABKLEX (Informatik, Telekommunikation)

<http://www.abklex.de/abklex/>

V. Claus, A. Schwill Duden Informatik

Dudenverlag, Mannheim, 2003, 768 S.

E. Kajan Information Technology Encyclopedia and Acronyms

Springer, Berlin + Heidelberg, 2002, 720 S.

B. Nguyen Linux Dictionary

The Linux Documentation Project, <http://www.tldp.org/>, 2004, 1757 S.

E. S. Raymond The New Hacker's Dictionary

The MIT Press, Cambridge/USA, 1996, 547 S., als Jargon-File bekannt.

2. Informatik

E. Glatz Betriebssysteme

dpunkt.verlag, Heidelberg, 2005, 700 S.

T. Flik Mikroprozessortechnik und Rechnerstrukturen

Springer, Berlin + Heidelberg, 2005, 649 S.

W. K. Giloi Rechnerarchitektur

Springer, Berlin + Heidelberg, 1993, 482 S.

G. Goos Vorlesungen über Informatik

Band 1: Grundlagen und funktionales Programmieren,

Springer, Berlin + Heidelberg, 2000, 396 S.

Band 2: Objektorientiertes Programmieren und Algorithmen,

Springer, Berlin + Heidelberg, 2001, 368 S.

- Band 3: Berechenbarkeit, formale Sprachen, Spezifikationen,
Springer, Berlin + Heidelberg, 1997, 284 S.
- Band 4: Paralleles Rechnen und nicht-analytische Lösungsverfahren,
Springer, Berlin + Heidelberg, 1998, 292 S.
- E. Kinber, C. Smith Theory of Computing: A Gentle Introduction
Prentice Hall, London, 2001, 207 S.
- D. E. Knuth The Art of Computer Programming, 4 Bände
Addison-Wesley, Boston
Klassiker, stellenweise mathematisch, 7 Bände geplant,
Band 5 soll im Jahr 2009 fertig sein, Homepage
des Meisters: <http://www-cs-staff.stanford.edu/~uno/>
- W. Schiffmann et al. Technische Informatik
Springer, Berlin + Heidelberg, 2004/05, 1. Teil Grundlagen der
digitalen Elektronik, 337 S.; 2. Teil Grundlagen der
Computertechnik, 399 S.: Übungsbuch, 279 S.
- R. B. Thompson, B. F. Thompson PC Hardware in a Nutshell
O'Reilly, Sebastopol, 2003, 880 S.
3. Algorithmen, Numerische Mathematik, Statistik
- J.M. Chambers Programming with Data
Springer, Berlin + Heidelberg, 2000, 470 S.
- T. H. Cormen et al. Introduction to Algorithms
The MIT Press, Cambridge/USA, 2001, 1200 S.
- P. Dalgaard Introductory Statistics with R
Springer, Berlin + Heidelberg, 2004, 270 S.
- G. Engeln-Müllges et al. Numerik-Algorithmen
Springer, Berlin + Heidelberg, 2005, 677 S.
- M. Kaplan Computeralgebra
Springer, Berlin + Heidelberg, 2005, 400 S.
- D. E. Knuth Algorithmen
(deutsche Übersetzung von *Fundamental Algorithms*)
Springer, Berlin + Heidelberg, 2004, 700 S.
- K. Loudon Mastering Algorithms in C
O'Reilly, Sebastopol, 1999, 560 S.
- W. H. Press u. a. Numerical Recipes in C
Cambridge University Press, 1993, 994 S.
- L. Sachs Angewandte Statistik
Springer, Berlin + Heidelberg, 2004, 900 S.
- H. R. Schwarz Numerische Mathematik
Teubner, Stuttgart, 1993, 575 S.
- R. Sedgewick Algorithmen in C
Addison-Wesley, Bonn, 1992, 742 S.
- R. Sedgewick Algorithmen in C++
Addison-Wesley, Bonn, 1992, 742 S.

- J. Stoer, R. Bulirsch Numerische Mathematik
 Springer, Berlin + Heidelberg, 1. Teil 2005, 383 S.,
 2. Teil 2005, 394 S.

4. Linux/UNIX allgemein

- SuSE Linux Benutzerhandbuch
 Novell, Waltham, 2004, 340 S.
- M. J. Bach Design of the UNIX Operating System
 Prentice-Hall, London, 1986, 486 S.
- S. R. Bourne Das UNIX System V (The UNIX V Environment)
 Addison-Wesley, Bonn, 1988, 464 S.
- P. H. Ganten, W. Alex Debian GNU/Linux
 Springer, Berlin + Heidelberg, 2004, 970 S.
- J. Gulbins, K. Obermayr, Snoopy Linux
 Springer, Berlin + Heidelberg, 2003, 900 S.
- R. Hattenhauer Linux-Livesysteme
 Galileo Press, Bonn, 2005, 447 S.
- H. Herold Linux-Unix-Shells
 Addison-Wesley, München, 1999, 832 S.
- R. Hertzog, C. Le Bars Debian GNU/Linux
 Eyrolles, Paris, 2004, 246 S.
- B. W. Kernighan, R. Pike Der UNIX-Werkzeugkasten
 Hanser, München, 1986, 402 S.
- M. Kofler Linux – Installation, Konfiguration, Anwendung
 Addison-Wesley, München, 7. Auflage 2005, 1318 S.
- M. Kofler Ubuntu Linux
 Addison-Wesley, München, 2005, 336 S.
- D. G. Korn, M. I. Borsky The Kornshell, Command and
 Programming Language
 deutsch: Die KornShell, Hanser, München, 1991
- M. F. Krafft The Debian System – Concepts and Techniques
 Open Source Press, München, 2005, 608 S.
- N. Petreley, J. Bacon Linux Desktop Hacks
 O'Reilly, Sebastopol, 2005, 253 S.
- A. Robbins UNIX in a Nutshell
 O'Reilly, Sebastopol, 1999, 624 S.
- F. Ronneburg Debian GNU/Linux Anwenderhandbuch
 Addison-Wesley, München, 2005, 744 S.
- K. Rosen u. a. UNIX: The Complete Reference
 Osborne/McGraw-Hill, Berkeley, 1999, 1302 S.
- K. Sarnow Linux in der Schule
 Millin, Poing, 2004, 395 S.
- E. Siever et al. Linux in a Nutshell
 O'Reilly, Sebastopol, 2003, 944 S.

R.W.Smith *FreeBSD The Complete Reference*
 Osborne McGraw-Hill, Berkeley, 2003, 900 S.

5. Einige Linux/UNIX Einzelthemen

- A. V. Aho, B. W. Kernighan, P. J. Weinberger *The AWK Programming Language*
 Addison-Wesley, Boston, 1988, 210 S.
- D. J. Barnett, R. Silverman, R. G. Byrnes *SSH, The Secure Shell*
 O'Reilly, Sebastopol, 2005, 672 S.
- D. Cameron et al. *Learning GNU Emacs*
 O'Reilly, Sebastopol, 2004, 544 S.
- B. Collins-Sussman et al. *Version Control with Subversion*
 O'Reilly, Sebastopol, 2004, 320 S.
- D. Dougherty, A. Robbins *sed & awk*
 O'Reilly, Sebastopol, 1997, 432 S.
- J. E. F. Friedl *Mastering Regular Expressions*
 O'Reilly, Sebastopol, 2002, 484 S.
- H. Herold *awk & sed*
 Addison-Wesley, München, 2003, 290 S.
- H. Herold *lex & yacc*
 Addison-Wesley, München, 2003, 390 S.
- T. Krumbein *OpenOffice.org*
 Galileo Press, Bonn, 2003, 525 S.
- L. Lamb, A. Robbins *Textbearbeitung mit dem vi-Editor*
 O'Reilly, Köln, 1999, 333 S. (vergriffen?)
- K. Lunde *CJKV Information Processing*
 O'Reilly, Sebastopol, 1999, 1128 S.
- N. Matthew, R. Stones *Linux Programmierung*
 MITP, Bonn, 2005, 890 S.
- R. Mecklenburg *GNU make*
 O'Reilly, Köln, 2005, 328 S.
- A. Robbins, N. H. F. Beebe *Classic Shell Scripting*
 O'Reilly, Sebastopol, 2005, 560 S.
- R. M. Stallman *GNU Emacs Manual*
 GNU Press, Boston, 2002, 644 S.
- J. Vesperman *Essential CVS*
 O'Reilly, Sebastopol, 2003, 336 S.
- R. Wobst *vim GE-PACKT*
 MITP, Bonn, 2004, 350 S.

6. Datenbanken etc.

- G. Carter *LDAP System Administration*
 O'Reilly, Sebastopol, 2003, 312 S.
- J. Banning *LDAP unter Linux*
 Addison-Wesley, München, 2001, 250 S.

- R. Dyer MySQL in a Nutshell
O'Reilly, Sebastopol, 2005, 352 S.
 - A. Ebersbach et al. Wiki-Tools
Springer, Berlin + Heidelberg, 2005, 401 S.
 - P. Eisentraut PostgreSQL GE-PACKT
MITP, Bonn, 2004, 350 S.
 - T. A. Howes et al. Understanding and Deploying LDAP Directory Services
Addison-Wesley, Reading, 2003, 936 S.
 - A. Kemper, A. Eickler Datenbanksysteme
Oldenbourg, München, 2004, 640 S.
 - M. Kofler MySQL 5
Addison-Wesley, München, 2005, 1032 S.
 - A. Konar Computational Intelligence
Springer, Berlin + Heidelberg, 2005, 708 S.
 - B. Leuf, W. Cunningham The Wiki Way
Addison-Wesley, Reading, 2001, 440 S.
 - F. Puppe Systematic Introduction to Expert Systems
Springer, Berlin + Heidelberg, 1993,
 - F. Puppe et al. Wissensbasierte Diagnosesysteme im Service-Support
Springer, Berlin + Heidelberg, 2001, 229 S.
 - G. Reese MySQL kurz & gut
O'Reilly, Sebastopol, 2002, 619 S.
 - V. Schwaberow OpenLDAP
Addison-Wesley, München, 1998, 890 S.
 - J. C. Worsley, J. D. Drake Practical PostgreSQL
O'Reilly, Sebastopol, 2002, 640 S.
 - R. J. Yarger et al. MySQL & mSQL
O'Reilly, Sebastopol, 2000, 503 S.
7. X Window System (X11), Motif, GNOME, KDE
- F. Culwin An X/Motif Programmer's Primer
Prentice-Hall, New York, 1994, 344 S. (vergriffen?)
 - T. + M. K. Dalheimer KDE Anwendung und Programmierung
O'Reilly, Sebastopol, 1999, 321 S. (vergriffen?)
 - K. Gottheil u. a. X und Motif
Springer, Berlin + Heidelberg, 1992, 694 S.
 - N. Mansfield The Joy of X
Addison-Wesley, Boston, 1993, 368 S.
 - A. Nye XLib Programming Manual, Rel. 5
O'Reilly, Sebastopol, 1994, 821 S.
 - V. Quercia, T. O'Reilly X Users Guide Motif R5
O'Reilly, Sebastopol, 1994, 955 S.
 - R. J. Rost X and Motif Quick Reference Guide
Digital Press, Bedford, 1993, 400 S.

8. Textverarbeitung mit \LaTeX
 - K. Braune, J. Lammarsch, M. Lammarsch \LaTeX
Springer, Berlin + Heidelberg, 2006, 700 S.
 - J. Gulbins, C. Kahrmann Mut zur Typographie
Springer, Berlin + Heidelberg, 2000, 430 S.
 - H. Kopka \LaTeX , 3 Bände
 - Band 1: Einführung
Addison-Wesley, München, 2000, 520 S.
 - Band 2: Ergänzungen
Addison-Wesley, München, 1997, 456 S.
 - Band 3: Erweiterungen
Addison-Wesley, München, 1996, 512 S.
9. Multimedia (Grafik, Sound, Video)
 - G. Born The File Formats Handbook
International Thomson Computer Press, Boston, 1995, 1274 S.
 - C. Braut The Musician's Guide to MIDI
Sybex (Wiley), 1994, 1136 S. (vergriffen)
 - D. Fellner Computer-Grafik
Springer, Berlin + Heidelberg, 2006, 350 S.
 - J. D. Foley et al. Introduction to Computer Graphics
Addison-Wesley, Boston, 1994, 632 S.
 - J. D. Foley et al. Computer Graphics – Principles and Practice in C
Addison-Wesley, Boston, 1996, 1200 S.
 - B. Fries, M. Fries Digital Audio Essentials
O'Reilly, Sebastopol, 2005, 384 S.
 - K. Kylander, O. S. Kylander GIMP
MITP-Verlag, Bonn, 2001, 700 S.
 - T. Lama 3D-Welten
Hanser, München, 2004, 336 S.
 - N. Magnus, T. Spindler Video – Wiedergabe, Bearbeitung und Streaming unter Linux
Open Source Press, München, 2005, 320 S.
 - T. Maschke Digitale Bildbearbeitung
Springer, Berlin + Heidelberg, 2004, 212 S.
 - D. M. McIntyre Rosegarden Companion
Bomots, Forbach/F, 2005, 180 S.
 - D. Orlamünder, W. Mascolus Computergrafik und OpenGL
Hanser, München, 2004, 320 S.
 - J. Osterberg GIMP 2
dpunkt.verlag, Heidelberg, 2005, 500 S.
 - D. Phillips Linux Music & Sound
No Starch Press, San Francisco, 2000, 408 S.
 - C. Roads The Computer Music Tutorial
MIT Press, Cambridge/USA, 1996, 1234 S.

- T. Roosendaal et al. The Official Blender 2.3 Guide
No Starch Press, San Francisco, 2004, 784 S., oder
<http://www.blender3d.org/e-shop/>, Nachdruck 2005
 - R. Steinmetz, K. Nahrstedt Multimedia Systems
Springer, Berlin + Heidelberg, 2004, 466 S.
 - J. Tranter Linux Multimedia Guide
O'Reilly, Sebastopol, 1996 (vergriffen?)
 - J. Vince Introduction to Virtual Reality
Springer, Berlin + Heidelberg, 2004, 163 S.
10. Netze allgemein (Internet, OSI)
- WikiReader Internet
<http://www.wikireader.de/>, 2004, 194 S.
 - D. E. Comer Internetworking with TCP/IP (3 Bände)
Prentice-Hall, Englewood Cliffs, I. Band 2006, 688 S.
II. Band 1991, 530 S., 88 DM; III. Band 2001, 601 S.
 - D. E. Comer The Internet Book
Prentice-Hall, Englewood Cliffs, 2000, 351 S.
 - K. Conner-Sax, E. Krol Die Welt des Internet
O'Reilly, Sebastopol, 2000, 400 S., vergriffen
 - H. Hahn, R. Stout The Internet Complete Reference
Osborne McGraw-Hill, Berkeley, 1994, 818 S.
 - C. Hunt TCP/IP Netzwerk-Administration
O'Reilly, Köln, 2003, 792 S.
 - O. Kirch, T. Dawson Linux Network Administrator's Guide
O'Reilly, Sebastopol, 2000, 500 S.
 - J. F. Kurose, K. W. Ross Computer Networking
Addison-Wesley, Boston, 2003, 784 S.
 - A. S. Tanenbaum Computer Networks
Prentice-Hall, London, 2003, 912 S.
11. Netzdienste Einzelthemen
- P. Albitz, C. Liu DNS und BIND
O'Reilly, Köln, 2001, 668 S.
 - A. Badach Voice over IP
Hanser, München, 2005, 394 S.
 - R. Deutz Mambo
Springer, Berlin + Heidelberg, 2005, 331 S.
 - A. Ebersbach et al. WikiTools
Springer, Berlin + Heidelberg, 2005, 400 S.
 - J. E. Hellbusch Barrierefreies Webdesign
KnowWare, <http://www.knowware.de/>, 2001, 86 S.
 - P. J. Lynch, S. Horton Web Style Guide
Yale University Press, New Haven, 1999, 165 S.

- C. Meinel, H. Sack WWW
Springer, Berlin + Heidelberg, 2004, 1200 S.
- S. Münz, W. Nefzger HTML 4.0 Handbuch
Franzis, München, 1999, 992 S.
- J. Niederst Web Design in a Nutshell
O'Reilly, Köln, 2002, 668 S.
- A. Schwartz Managing Mailing Lists
O'Reilly, Sebastopol, 1998, 320 S. (vergriffen?)
- P. von Weitershausen Web Component Development with Zope 3
Springer, Berlin + Heidelberg, 2005, 467 S.

12. Sicherheit

- Department of Defense Trusted Computer Systems
Evaluation Criteria (Orange Book)
1985, 120 S., ASCII. Abgelöst durch:
Federal Criteria for Information Technology Security
1992, 2 Bände mit zusammen 500 S., Postscript
- D. J. Barrett et al. Linux-Sicherheits-Kochbuch
O'Reilly, Köln, 2003, 364 S.
- F. L. Bauer Entzifferte Geheimnisse
Springer, Berlin + Heidelberg, 2000, 503 S.
- S. Garfinkel et al. Practical Unix + Internet Security
O'Reilly, Sebastopol, 2003, 992 S.
- L. Packschies Praktische Kryptographie unter Linux
Open Source Press, München, 2005, 368 S.
- B. Schneier Angewandte Kryptographie
Addison-Wesley, Bonn, 1996, 844 S.
- M. Schumacher, U. Roedig, M.-L. Moschgath Hacker Contest
Springer, Berlin + Heidelberg, 2003, 300 S.

13. Computerrecht

- Juristisches Internetprojekt Saarbrücken
<http://www.jura.uni-sb.de/>
- Netlaw Library (Universität Münster)
<http://www.jura.uni-muenster.de/netlaw/>
- Online-Recht <http://www.online-recht.de/>
- Computerrecht (Beck-Texte)
Beck, München, 1994
- U. Dammann, S. Simitis Bundesdatenschutzgesetz
Nomos Verlag, Baden-Baden, 1993, 606 S.
- A. Junker Computerrecht. Gewerblicher Rechtsschutz,
Mängelhaftung, Arbeitsrecht. Reihe Recht und Praxis
Nomos Verlag, Baden-Baden, 1988, 267 S.
- F. Koch Handbuch Software- und Datenbank-Recht
Springer, Berlin + Heidelberg, 2003, 1000 S.

D. Kröger, M. A. Gimmy Handbuch zum Internetrecht
Springer, Berlin + Heidelberg, 2. Auflage 2002, 1000 S.

14. Geschichte der Informatik

- Kleine Chronik der IBM Deutschland
1910 – 1979, Form-Nr. D12-0017, 138 S.
1980 – 1991, Form-Nr. D12-0046, 82 S.
Reihe: Über das Unternehmen, IBM Deutschland
- Die Geschichte der maschinellen Datenverarbeitung Band 1
Reihe: Enzyklopädie der Informationsverarbeitung
IBM Deutschland, 228 S., Form-Nr. D12-0028
- 100 Jahre Datenverarbeitung Band 2
Reihe: Über die Informationsverarbeitung
IBM Deutschland, 262 S., Form-Nr. D12-0040
- Open Source
O'Reilly, Köln, 1999, 70 S.
- P. E. Ceruzzi A History of Modern Computing
The MIT Press, Cambridge/USA, 1998, 400 S.
- O. A. W. Dilke Mathematik, Maße und Gewichte in
der Antike (Universalbibliothek Nr. 8687 [2])
Reclam, Stuttgart, 1991, 135 S.
- M. Hauben, R. Hauben Netizens – On the History and
Impact of Usenet and the Internet
IEEE Computer Society Press, Los Alamitos, 1997, 345 S.
- A. Hodges Alan Turing, Enigma
Kammerer & Unverzagt, Berlin, 1989, 680 S.
- S. Levy Hackers – Heroes of the Computer Revolution
Penguin Books, London, 1994, 455 S.
- D. Shasha, C. Lazere Out of Their Minds
Springer, Berlin + Heidelberg, 1995, 295 S.
- D. Siefkes u. a. Pioniere der Informatik
Springer, Berlin + Heidelberg, 1998, 160 S.
- L. Torvalds, D. Diamond Just for Fun
dtv, 2004, 275 S.
- K. Zuse Der Computer - Mein Lebenswerk
Springer, Berlin + Heidelberg, 3. Aufl. 1993, 220 S.

15. Philosophie

- E. Dyson Release 2.1 – A Design for Living in the Digital Age
Petersen, Hamburg, 2000, 370 S.
- D. R. Hofstadter Gödel, Escher, Bach - ein Endloses
Geflochtenes Band
dtv/Klett-Cotta, München, 1992, 844 S.
- P. Schefe u. a. Informatik und Philosophie
BI Wissenschaftsverlag, Mannheim, 1993, 326 S.

- R. M. Stallman Free Software – Free Society
GNU Press, Boston, 2002, 224 S.
- K. Steinbuch Die desinformierte Gesellschaft
Busse + Seewald, Herford, 1989, 269 S. (vergriffen)
- J. Weizenbaum Die Macht der Computer und die Ohnmacht
der Vernunft (Computer Power and Human Reason.
From Judgement to Calculation)
Suhrkamp Taschenbuch Wissenschaft 274, Frankfurt (Main),
1990, 369 S.
- H. Zemanek Das geistige Umfeld der Informationstechnik
Springer, Berlin + Heidelberg, 1992, 303 S.

16. Zeitschriften

- c't magazin für computertechnik
für alle Fragen der Computerei, technisch.
<http://www.heise.de/ct/>
- iX Magazin für professionelle Informationstechnik
für Anwender von Multi-User-Systemen, technisch.
<http://www.heise.de/ix/>
- Linux Enterprise
<http://www.linuxenterprise.de/>
- LinuxFocus
<http://linuxfocus.org/>
- Linux Gazette
<http://www.linuxgazette.com/>
- Linux Magazin
<http://www.linux-magazin.de/>
- linuxUSER
<http://www.linux-user.de/>
- Linux Journal
<http://www.linuxjournal.com/>

Und über allem, mein Sohn, laß dich warnen;
denn des vielen Büchermachens ist kein Ende,
und viel Studieren macht den Leib müde.

Prediger 12, 12

Personenverzeichnis

- Adleman, L. 235
Aho, A. V. 206
Aiken, H. 497
Allen, J. H. 195
Almquist, K. 69
Andreessen, M. 497
Arens, R. 186
Atanasoff, J. V. 497
Atterer, R. 142
- Babbage, C. 497
Bardeen, J. 497
Bauer, F. L. 236
Berners-Lee, T. 144, 224, 497
Berry, C. 497
Boole, G. 497
Bostic, K. 175
Bourne, S. R. 33
Bradley, J. 302
Brattain, W. H. 497
Brehm, A. 6
Bresenham, J. E. 293
Bricklin, D. 262, 497
- Caesar, C. J. 109, 234
Cailliau, R. 144, 224, 497
Campbell, C. E. 186
Cerf, V. G. 497
Chambers, J. 268
Clark, J. 497
Codd, E. F. 497
- Conway, J. H. 426
Crocker, S. 497
Cunningham, W. 284
Czyborra, R. 161
- Dersch, H. 310
Didot, F. A. 458
- Eckert, J. P. 497
Eisen, L. S. 372
Elgamal, T. 237
Engelbart, D. C. 497
Esser, T. 214
Euler, L. 267
Ewing, L. 6
- Fibonacci 497
Fox, B. 33
Frankston, B. 497
Friedman-Hill, E. J. 289
- Gantt, H. L. 390
Gibson, W. 497
Goethe, J. W. von 403
Goldfarb, C. 227
Gregor XIII. 109
Grothmann, R. 296
Gutenberg, J. 162
- Haible, B. 161
Hänel, M. 25
Hawking, S. W. 25

- Heilig, M. 497
 Hermitte, L. 186
 Herrmann, M. 303
 Hewlett, W. 497
 Hollerith, H. 497
 Hopper, G. 497

 Jacquard, J. M. 497
 Jeeves, R. 147
 Jobs, S. P. 497
 Joy, B. 33, 69, 175

 Kahn, R. 497
 Karzauninkat, S. 133
 Kernighan, B. W. 206
 Kilby, J. St. C. 497
 Knuth, D. E. 122, 213, 223, 260, 497
 Kopka, H. 223
 Kuhn, M. 161

 Lagotzki, S. 171, 218
 Lamport, L. 214
 Leibniz, G. W. 497
 Licklider, J. C. R. 497
 Link, E. 497
 Lucas, E. A. L. 424

 Mandelbrot, B. 423
 Marillat, C. 355
 Mauchly, J. W. 497
 Metcalfe, R. 497
 Mill, H. 497
 Minsky, M. L. 497
 Moolenaar, B. 175
 Morse, S. F. B. 497
 Münz, S. 227, 291
 Murdock, I. 7, 497
 Murphy, E. A., Jr. 379

 Napier, J. 497
 Nelson, T. 223, 497
 Neumann, J. von 497
 Newton, Sir I. 293
 Nickel, K. 497

 Oikarinen, J. 497

 Packard, D. 497
 Partl, H. 214, 227
 Pascal, B. 497
 Piaf, E. 24
 Pike, R. 159
 Pisa, L. von 497
 Poskanzer, J. 303
 Postel, J. B. 497
 Poulsen, W. 497
 Proust, M. 391

 Raymond, E. S. 124, 374
 Reis, J. P. 497
 Remington, E. 497
 Ritchie, D. 6
 Rivest, R. 235

 Sander-Beuermann, W. 133
 Schickard, W. 497
 Schliemann, H. 403
 Schneier, B. 46, 236
 Schultz, R. 313
 Shakespeare, W. 403
 Shamir, A. 235
 Shannon, C. E. 327, 497
 Shockley, W. B. 497
 Shugart, A. 497
 Singer, C. 204
 Socher, K. 310, 320
 Stallman, R. M. 6, 188, 497
 Stayton, B. 213
 Steinbuch, K. 497
 Stroustrup, B. 497
 Sutherland, I. E. 497
 Suvasa, E. 6

 Thiele, H. 418
 Thompson, K. 6, 159
 Tichy, W. F. 244
 Torvalds, L. B. 6, 497
 Trieloff, L. 213
 Tschichold, J. 208
 Turing, A. 497
 Tux 6

 Wall, L. 69

Waller, F. 24
Walsh, N. 163, 213
Weinberger, P. J. 206
Weissinger, J. 497
Weizenbaum, J. 421
Wodehouse, P. G. 147
Wolfram, S. 269
Wood, A. 161
Wozniak, S. G. 497
Wrage, J. 52

Ylonen, T. 110

Zuse, K. 497

Sachverzeichnis

- .LRN 402
- .bashrc 32
- .emacs 190
- .exrc 177
- /dev/null 76
- /etc/alternatives/ 175
- /etc/fstab 90
- /etc/shells 66
- /etc/ssh/ 110
- 3Delight 318
- 8-bit-clean 158

- a2ps 105, 170
- AAC → Advanced Audio Coding
- abcde 340
- AbiWord 199
- Abklex 138
- Abkürzung 138, 469
- Abkürzung (vi) 179
- abmelden 8, 45
- abook 389
- abschalten → herunterfahren
- Abtasttheorem 327
- AC3D 319
- Access Control List 75
- accessibilité → Zugänglichkeit
- accessibility → Zugänglichkeit
- acheck 232
- ACL → Access Control List
- acroread 166
- Adabas 283

- Administrator → Verwalter
- Adobe Reader 166, **171**, 214
- Advanced Audio Coding 328
- Advanced Linux Sound Architecture 330
- Advanced Revision Control System 244
- Agnula 328
- agrep 123, 124
- AIFF → Audio Interchange File Format
- AIM → AOL Instant Messenger
- Akronym 138
- Aktion (awk) 206
- aktuelles Verzeichnis → Arbeitsverzeichnis
- alevtd 361
- alias 54
- alien 50, 302
- Allesklar 148
- ALSA → Advanced Linux Sound Architecture
- alsamixer 331
- AlsaPlayer 337
- Alt Gr (Taste) 451
- Altavista 146
- amavis 47
- Amaya **143**, 224, 225
- American Standard Code for Information Interchange → ASCII

- amixer 330, 331
- amorce → booten
- ampersand → et-Zeichen
- an 427
- anacron 109
- Anagramm 427
- Analog Real-Time Synthesizer 356
- Analog-Digital-Wandler 327
- and 120
- Angel (Renderer) 318
- animate 308
- animieren 313
- Anmeldemanager → Display-Manager
- anmelden 8, 43
- Anonymous FTP 140
- Ant 392
- ant  moire → Cache
- antiword 172
- Anweisung (LaTeX) 215
- Anwendung 33
- AOL Instant Messenger 377
- aoss 330
- Apache 131, 392
- appel syst  me → Systemaufruf
- AppleTalk 86
- application → Anwendung
- appres 20
- apropos 37
- apt-file 129
- apt-get 49, 51
- Aqsis 313
- ar 84
- ara 128
- Arachnophilia 225
- Arbeitsspeicher 3
- Arbeitsverzeichnis 44, 71
- arch 250
- Archiv (Dateien) 50, 84
- Archiv (News) 374
- ardour 334
- Argument 9
- arobace → Klammeraffe
- arping 134
- Array
 - A. mit Inhaltindizierung 208
 - assoziatives A. 208
 - awk-Array 208
- arri  re-plan → Hintergrund
- arrow key → Pfeiltaste
- Art of Illusion 319
- aRts 356
- ASCII
 - German-ASCII 157
 - Steuerzeichen 156, 442
 - Zeichensatz **156**, 429, 436
- Ask Jeeves 146
- asmixer 331
- aspell 231
- Asterisk 384
- asynchron 98
- at 108
- at-Kommando (Modem) 486
- Athena-Projekt 11
- atime 127
- AUCTeX 190
- Audacity 334
- Audio Interchange File Format 328
- Audioeditor 334
- Aufbau eines Rechners 1
- Aufl  sung (Audio) 327
- Aufl  sung (Grafik) 292
- Aufmachung 208
- Auftrag (Drucken) 102
- Ausdruck, regul  rer **116**, 182, 205, 456
- ausf  hren 74
- ausloggen → abmelden
- Aussage 120
- ausschalten → herunterfahren
- Authentizit  t 112
- Automat 1, 427
- autopano 310
- autotrace 303
- avant-plan → Vordergrund
- awk 116, **206**, 278
- awk-Skript 207
- axe 198
- Axiom 271
- Ayam 313
- Azureus 142

- Babel 138
- background → Hintergrund
- Backup 45, 85, 276
- balsa 366
- Band 89
- Barrierefreiheit → Zugänglichkeit
- base de données → Datenbank
- bash 33
- bashburn 94
- bashrc 32
- batch 97
- bc 254
- BCD → Dezimalsystem, binär codiertes
- BCP → Best Current Practice
- bed 202
- Befehl → Kommando
- Befehlsfenster 9
- Befehlszeilschalter → Option
- Behinderter 25
- Beleuchtung 295
- Benutzer 2, 130, 280
- Benutzer-ID 73
- Benutzername 9, 43
- Benutzeroberfläche 11
- Beowulf 97
- Berkeley Database 279
- Berkeley Services 110
- Berkeley Software Distribution 5
- Besitzer 74
- Best Current Practice 491
- Betriebssystem 4, 5
- Bewegungsmelder 362
- bfs 232
- bg 99
- bidirektional 169
- bidiv 169
- Big Eight 373
- Big Red Switch 8
- Bildbetrachter 302
- Bildschirm 2, 18, 45
- Bildschirmlupe 27
- Bildschirm Sperre 45
- Binär-Editor 202
- BitTornado 142
- BitTorrent 141
- bitweise Verknüpfung 121
- blackbook 389
- Blender 319, 320
- Blinde 26
- Blinux 26
- Blowfish 235
- Blue Moon Rendering Tools 313
- bluefish 225
- Bluetooth 113
- BMRT → Blue Moon Rendering Tools
- Bobby 226
- Bodoni 162
- bookmark → Lesezeichen
- booten 8
- boule de commande → Rollkugel
- Bourne-again-Shell 33, 53
- Bourne-Shell 33
- bouton → Knopf
- Brahms (Musikeditor) 345
- Braillezeile 349
- Break-Taste 451, 467
- brennen (CD/DVD) 91
- Brenner 92
- brlTTY 349
- Brooktree Video Decoder 358
- browsen (MS Windows) 88
- Browser 142
- Brute Force Attack 237
- BSD → Berkeley Software Distribution
- BSD-Drucken 103
- BSI → Bundesamt für Sicherheit in der Informationstechnik
- Builder 241
- Bundesamt für Sicherheit in der Informationstechnik 48
- bunzip2 83
- burner → Brenner
- button → Knopf
- bvi 202
- Byte 155
- bzgrep 123
- bzip2 83

- C-Shell 33
- Cache 3
- CAD → Computer Aided Design
- cal 108
- calc (OpenOffice.org) 265
- calcoo 258
- calendar 109, 388
- Carbon Copy 370
- CAS → Computeralgebrasystem
- Cascading Style Sheet 226
- CAST5 235
- cat 77, 168
- catdoc 172
- CBW → Crypt Breaker's Workbench
- CC → Carbon Copy
- ccal 108
- CCC → Chaos Computer Club
- ccdecoder 29
- CCITT V.3 156
- cd 54, 71
- CD → Compact Disc
- cda 338
- CDDb → Internet CD Database
- CDE → Common Desktop Environment
- cdparanoia 339
- cdrdao 95
- cdrecord 91
- central processing unit → Zentralprozessor
- central unit → Zentraleinheit
- CERT → Computer Emergency Response Team
- cgoban 415
- chaîne de caractères → Zeichenkette
- Chandler 387
- Chaos Computer Club 497
- character → Zeichen
- Chatbot 421
- chfn 130
- chgrp 73
- chiffrement → Verschlüsselung
- Chinesisch, Japanisch, Koreanisch, Vietnamesisch 30, 158
- Chipmunk 300
- chktex 221
- chmod 74
- Cholesterol Free Desktop Environment 13
- chown 73
- chroot 66
- chsh 66
- CIFS → Common Internet File System
- ciphertext → Geheimtext
- CJKV → Chinesisch, Japanisch, Koreanisch, Vietnamesisch
- clamav 47
- clara 173
- clavier → Tastatur
- clicher → dumpen
- Client 11, 12
- Client-Server-Modell 11
- CLIPS 289
- cliquer → klicken
- cmp 229
- CMS → Content Management System
- CMYK-Farbmodell 294
- Codec 328
- Codetafel 156
- col 210, 232
- collecticiel → Groupware
- comm 232
- command line → Kommandozeile
- commercial at → Klammeraffe
- Common Desktop Environment 13
- Common Internet File System 86
- Common UNIX Printing System 104
- Compact Disc 3, 89
- compare 308
- Composer 225
- composite 308
- Comprehensive TeX Archive Network 214
- compress 83

compte → Konto
 computer → Rechner
 Computer Aided Design 297
 Computer Emergency Response Team 48
 Computer Incident Advisory Capability 48
 Computeralgebrasystem 214, 269
 Concurrent Versions System 243, 246
 Configurable Unified Search Engine 132
 configure 50
 conglomerate 227
 conjure 308
 connexion → anmelden
 Console → Konsole
 Content Based Image Retrieval System 126
 Content Management System 400
 Content Scrambling System 352
 convert 308
 convmv 174
 Cookie (X11) 16
 Courier 162
 courriel → Email
 courrier électronique → Email
 cp 75
 Cpmtools 91
 Cracking 237
 crafty 413
 CRMS → Customer Relationship Management System
 cron 108
 crontab 108
 CrossOver Office 412
 crypt 235
 Crypt Breaker's Workbench 235
 cryptogramme → Geheimtext
 csh 33
 CSS → Cascading Style Sheet, Content Scrambling System
 CTAN → Comprehensive TeX Archive Network
 ctrl-alt-del 8

CUPS → Common UNIX Printing System
 curses 11, 158
 Cursor 27, 43
 Cursortaste → Pfeiltaste
 CUSI → Configurable Unified Search Engine
 Customer Relationship Management System 388
 cut 232
 CVS → Concurrent Versions System
 CYCAS 319

 DAN → Domestic Area Network
 DANTE 214
 Data Base Management System 274
 Data Encryption Standard 235
 Data Exchange Format 292
 date → Datum
 date 107
 Datei
 Allgemeines 71
 Audiodatei 328, 333
 Besitzer 74
 Besitzer ändern 73
 Bezeichner 60, 61
 D. ausführen 74
 D. kopieren 75
 D. lesen 74
 D. löschen 75, 76
 D. packen 82
 D. schreiben 74
 D. umbenennen 76
 D. verlinken 78
 D. verschieben 76
 D.-Manager 34, 80
 Ende 164
 Format 328
 Gruppe 74
 Gruppe ändern 73
 Kennung 76, 463
 Name 56, 76
 Pointer 60

- Rest der Welt 74
- Typ einer D. 73
- verborgene D. 72
- Zeitstempel 79
- Zugriffsrecht 74
- Datenbank
 - Aufgabe 273
 - Datenbasis 274
 - Grundbegriffe 273
 - relationale D. 274
 - Tabelle 274
 - View 282
- Datenaustauschdienst 142
- Datum 107, 252, 347
- DAV → Distributed Authoring and Versioning
- db2latex 213
- DBMS → Data Base Management System
- dc 257
- dd 75
- Debian
 - Accessibility 26
 - D. Free Software Guidelines 33
 - D. Weekly News 42
 - Distribution 7
 - Forum 42
 - HOWTO 39
 - Liste 371
 - Paket 49, 128
 - Wiki 42
- Decoder 327
- déconnexion → abmelden
- Deep Blue 413
- Defaultwert 10
- delatex 232
- DeMuDi 328
- DENIC 135
- Denoiser 333
- dérouleur → Laufwerk
- Desktop Environment 13, 22
- Desktop File Manager 81
- detex 232
- Deutsche Kommission Elektrotechnik 158
- Deutsches Institut für Normung 158
- Dezimalsystem 429
- Dezimalsystem, binär codiertes D. 429
- Dezimalzahl 253
- dfm 81
- DFSG → Debian Free Software Guidelines
- Dialer 48
- dict 124
- Dienstprogramm 34
- diff 229
- diff3 229
- digikam 302
- Digital Versatile Disc 3, 89
- Digital Video Broadcasting 352
- DIN → Deutsches Institut für Normung
- DIN 66003 157
- DIN-Brief 216
- ding 124
- diplom 233
- directory → Verzeichnis
- Diskette 3, 89
- display 308
- DISPLAY 14
- Display-Manager 13, 43
- dispositif de pointage → Zeigegerät
- disque dur → Festplatte
- disque optique compact → Compact Disc
- disquette → Diskette
- distcc 97
- Distiller 166
- Distinguished Name 287
- Distributed Authoring and Versioning 243
- Distribution 7
- Dithering 294
- DKE → Deutsche Kommission Elektrotechnik
- dlocate 129
- dlume 389
- DN → Distinguished Name
- DNS → Domain Name System

DOC → disque optique compact
 doc2xml 172
 DocBook 213
 docbook2html 213
 docbook2pdf 213
 Dock 13
 Dockapp 13, 332
 Document Type Definition 227
 Dokumentklasse (LaTeX) 215
 Domäne (DNS) 135
 domain → Domäne
 Domain Name System 135
 domaine → Domäne
 Domestic Area Network 3
 dpkg 49–51
 dpkg-iaquery 129
 dpkg-make-index 129
 dpkg-query 129
 Dresden, Arbeitsgruppe Studium für
 Blinde und Sehbehinderte 26
 drive → Laufwerk
 drucken 102
 Drucker 3
 DTD → Document Type Definition
 Dualsystem 429
 Dualzahl 253
 dumpen 21, **75**, 169
 durchsuchen 74
 DVB → Digital Video Broadcas-
 ting
 DVD → Digital Versatile Disc
 DVD Video 352
 dvi2ps 170
 dvips 170, 215

 e2ps 170
 eac 341
 Eagle 299
 Easy MetaGer 152
 eboard 413
 ecawave 334
 echo 56, 59
 ECMA → European Computer Ma-
 nufacturers Association
 ECMA-94 157

ECMS → Enterprise Content Ma-
 nagement System
 écran → Bildschirm
 ed 175, 229
 EDA → Electronic Design Auto-
 mation
 éditeur → Editor
 Editor
 AbiWord 199
 axe 198
 Binär-E. 202
 Emacs 116, 188
 gedit 197
 geresh 202
 Grundbegriffe 167
 Hex-E. 202
 joe 195
 kate 196
 katoob 203
 KWord 200
 Microemacs 189
 Musikeditor 345
 nano 195
 nedit 196
 oowriter 201
 pico 195
 sed 205
 Streaming E. 205
 vi 116
 XEmacs 189
 yudit 203
 efax 381
 eflite 350
 egrep 123
 eGroupWare 393
 EGW → eGroupWare
 EIFF → Exchangeable Image File
 Format
 Einfügemarke → Cursor
 Eingabeaufforderung → Prompt
 einloggen → anmelden
 Electronic Design Automation 298
 Electronic Mail → Email
 elinks 143
 Eliza 421

- elm 366
- EMA-XPS 290
- Emacs
 - Allgemeines 188
 - aufrufen 191
 - beenden 192
 - Emacspeak 350
 - Konfiguration 189
 - navigieren 192
 - Psychiatrist 195
 - regulärer Ausdruck 116
 - Schnellstart 188
 - suchen 122
 - Texinfo 211
 - zurückschreiben 191
- Emacspeak 195, 350
- emacspeakconfig 350
- Email 45, **365**
- Emulator 52
- en-tête → Kopf
- enblend 310
- Encapsulated PostScript 166, 292
- Encoder 327, 339
- Enlightened Sound Daemon 330
- Enlightenment 13
- enscript 170
- Enterprise Content Management System 400
- ENUM 383
- environment → Umgebung
- environnement → Umgebung
- Enzyklopädie
 - Brockhaus 137
 - E. Britannica 137
 - Larousse 137
 - Meyers 137
 - Wikipedia 137
- Epiphany 143
- Epos 348
- EPS → Encapsulated PostScript
- eps2eps 170
- ERP5 400
- Erweiterung (Dateiname) → Kennung
- Erweiterung (Shell) 54
- Escape-Taste 451
- ESD → Enlightened Sound Daemon
- ESR → Raymond, E. S.
- esvn 250
- Eszett 186
- et-Zeichen 98
- euler 267
- European Computer Manufacturers Association 157
- Eurosymbol 157
- Evince 171
- Evolution 393, 394
- ex 180
- Exact Audio Copy 341
- Excel 262
- Exchangeable Image File Format 303
- EXIF → Exchangeable Image File Format
- EXINIT 176
- exit 45
- expand 232
- Expansion 54
- Expertensystem 273, 278, **289**
- exports 87
- Extensible Hypertext Markup Language 228
- Extensible Markup Language 227
- factor 253
- faktorisieren 253
- Family Area Network 3
- FAN → Family Area Network
- FAQ → Frequently Asked Questions
- Farbe
 - Grundbegriffe 293
 - HTML 460
 - X11 459
- Farbeditor 459
- Farbmischung 293
- Farbmodell 293
- Farbschwäche 26
- Farbtemperatur 293

- Farbtiefe 294
- Farbvektor 292
- Fax 381
- fdformat 89
- feh 302
- Fehler
 - Fehlermeldung 10
 - LaTeX 221
- Feld
 - awk 206
 - Datenbank 274
- FELt 300
- FEM → Finite-Elemente-Methode
- fenêtre → Fenster
- Fenster
 - F. aktivieren 19
 - F. selektieren 19
 - F-Manager 13
 - Grundbegriffe 11
- Fernsehen 352, 358
- Festival 348
- Festplatte 3
- fetchmail 368
- FFEP → Free Finite Element Package
- fg 99
- fgrep 123
- fichier → Datei
- file → Datei
- file 73
- File Transfer Protocol 140
- Filter (Programm) 60, 206
- find 126
- finger 130
- fingerd 130
- Finite-Elemente-Methode 300
- Fireball 146
- Firefox 143, 224
- Firewall 141
- fixfmps 105
- fixmacps 105
- fixwfwps 105
- Flight Gear 419
- flite 347
- Floppy → Diskette
- floppy disc → Diskette
- Flugsimulator 419
- fluidsynth 343
- Fluxbox 13
- Foire Aux Questions → Frequently Asked Questions
- Fokus 19
- fold 232
- Font
 - Bitmap-F. 163
 - Grundbegriffe 162
 - Open Type 163
 - PostScript Type 1 163
 - True Type 163
 - Vektor-F. 163
- For Your Information 491
- foreground → Vordergrund
- Format
 - Audiodateiformat 328
 - Font-F. 163
 - Hochformat 163
 - Querformat 163
 - Rastergrafik 292
 - Textdateiformat 164
 - Vektorgrafik 292
 - Videodateiformat 353
- formatieren (Datenträger) 89
- formatieren (Text) 209
- forme → Muster
- Formel (LaTeX) 476
- fpm 46
- Fraktal 423
- FrameMaker 209
- Free Finite Element Package 301
- Free Software Directory 50
- Free Software Foundation 50
- freeamp 336
- FreeBSD 5
- FreeCiv 415
- freedb 339
- FreeFem 301
- FreeMind 398
- Freepats 344
- FreeTTS 348

Frequently Asked Questions 40, 375

Freudenknüppel 24

FSF → Free Software Foundation

fstab 87

ftp 140

FTP → File Transfer Protocol

fuser 90

fvwm 13

FYI → For Your Information

gaby 389

Gaim 376, 378

galculator 258

Galeon 143, 224

Game of Life 426

gamix 331

GanttProject 392

Ganzzahl 251

gattrib 298

gawk 206

gbase 159, 253

gcalctool 259

gcombust 94

gdbm 279

gdm 13, 43

Gebrauch eines Kommandos 10

gEDA 298

Gedächtnisstöpsel → Memory
Stick

gedit 197

Gegensprechen 363

Geheimtext 234

gentoo 81

gerbv 298

geresh 202

gestionnaire → Manager

gfloppy 89

GForge 392

ggv 170

Ghostscript 105, 163, 166, **170**, 218

Ghostview 105, 170

GID → Gruppen-ID

Giessen, Zentrum für blinde und seh-
behinderte Studierende 26

GIF → Graphics Interchange For-
mat

gift 126

gimp 21

GIMP 166, **304**, 309, 310, 320, 354

gjots2 399

GLAME 334

Gleitkommazahl 251

Glimpse 123

glimpseindex 124

Globbering 54

Glyph 156

gManager 298

gnetlist 298

GNOME

AbiWord 199

Desktop 13, 22

Gnumeric 263

Office-Paket 410

gnome-system-monitor 96

gnome-terminal 20

gnome-think 399

gnomemeeting 383

Gnopernicus 27, 349

gnotime 391

GNU Chess 413

GNU Network Object Model Envi-
ronment → GNOME

GNU Privacy Guard 237

GNU Privacy Handbook 237

GNU-Manual 38

GNU-Projekt 6, 188

gnucap 298

gnugo 415

gnui ft 126

Gnumeric 263

gnupg 237, 371

gnuplot 253

gnus 372

gnusound 334

GNUstep 13

GNUTls 113

GO (Spiel) 415

gocr 173

Google 145, 374

- Googlehupf 145
- gpdf 171
- gphoto2 301
- gq 287
- graphical user interface → Benut-
zeroberfläche
- Graphics Interchange Format 292
- graveur → Brenner
- grep 116, 119, **122**, 278
- grip 340
- groff 209
- group → Gruppe
- groupe de nouvelles → Newsgrup-
pe
- Groupware 385, 393
- grpn 258
- Gruppe 74, 280
- Gruppen-ID 73
- gschem 298
- GSview 170
- gsymcheck 298
- gtkam 301
- gtkboard 414
- GtkLife 426
- gtkmorph 311
- GTKWave 298
- GUI → graphical user interface
- Guide 40
- gunzip 83
- Gutenberg (Projekt) 173
- gv 105, **170**, 171, 209
- gvim 175
- gwave 298
- gwc 333
- gwget 140
- gxine 354
- gzip 83

- H.323 383
- Hackbrett → Tastatur
- Hacker's Dictionary 124
- Hadifix 349
- Hanoi, Türme von H. 243, 256, **424**
- hard disc → Festplatte
- Hardlink → Link, harter

- Hardware 1
- Hauptkatalog → Home-Verzeichnis
- Hayes-Kommando 486
- head 75
- Heimatverzeichnis → Home-
Verzeichnis
- Heise Security 48
- Helix Player 357
- Help Browser 38
- Helvetica 162
- herunterfahren 8
- Hewlett-Packard Graphic Language
292
- Hex-Editor 202
- Hexadezimalsystem 429
- Hexadezimalzahl 253, 429
- hexcat 169
- Hilfe 36
- Hilfezentrum 38
- Hintergrund 98
- hnb 386
- Home-Verzeichnis 44, 71
- Homepage → Startseite
- Hörvermögen 29
- host 134
- hostname 106
- HOWTO
 - Accessibility 26
 - Allgemeines 39
 - CVS-RCS 246
 - International keyboard and Xmod-
map H. 17
 - Keyboard and Console H. 17
 - Modem 486
 - vim 175
 - Wine 52
 - X11 12
 - XFree86 12
- HP-GL → Hewlett-Packard Gra-
phic Language
- htm 226
- html 226
- HTML → Hypertext Markup Lan-
guage
- HTML-Browser 224

- HTML-Entity 450
- HTML-Farbe 460
- HTML-Prüfer 226
- html2ps 170
- HTTP → Hypertext Transfer Protocol
- Hugin 310
- HylaFAX 381
- Hyperlink 223
- Hypermedia 223
- Hypertext 223
- Hypertext Markup Language 142, 174, 209, **224**
- Hypertext Transfer Protocol 142
- hyphen 232
- i18n 30
- iCalendar 388
- Icarus 298
- ICMP-Paket 133
- Icon 19
- iconv 174
- id 106
- IDEA 235
- identify 308
- Identitätsklau 48
- IEC → International Electrotechnical Commission
- IIM → Information Interchange Model
- IM → Instant Messaging
- ImageMagick 218, **308**, 320
- IMAP → Internet Message Access Protocol
- IMC → Internet Mail Consortium
- IMcom 377
- imgseek 126
- import 308
- Impress 379
- imprimante → Drucker
- Index 121
- info 38, 211
- Information Interchange Model 303
- inkscape 296
- Inode 77
- INRIA 267, 497
- Instant Messaging 377
- Integrität 112
- interface → Schnittstelle
- interface utilisateur → Benutzeroberfläche
- interface utilisateur graphique → Benutzeroberfläche
- Interleaving 352
- International Electrotechnical Commission 158
- International Press Telecommunications Council 303
- International Standard Book Number 139
- International Standardization Organization 158
- International Telecommunication Union 352, 383
- internationalisiert 30
- Internet CD Database 338
- Internet Explorer 224
- Internet Mail Consortium 388
- Internet Message Access Protocol 368
- Internet Printing Protocol 104
- Internet Relay Chat 375
- Internet Service Provider 3
- Internet-Telefonie 382
- Internetradio 336, 357
- invite → Prompt
- IPP → Internet Printing Protocol
- IPTC → International Press Telecommunications Council
- IRC → Internet Relay Chat
- IRIT 319
- ISBN → International Standard Book Number
- ISO → International Standardization Organization
- ISO 10646 158
- ISO 3166 488
- ISO 646 156
- ISO 8859 157, 443
- ISO 8879 227

ISO/IEC 10646 443

ISP → Internet Service Provider

ispell 186, 221, **231**

Ispell.vim 186

ITU → International Telecommuni-
cation Union

iwconfig 114

iwlist 114

Jabber 377

Jabber ID 377

jack 340

JACK Audio Connection Kit 330

Jargon 124

Java 398

Java Data Base Connectivity 282

JDBC → Java Data Base Connecti-
vity

jed 198

jedit 198

Jess 289

jeu de caractères → Zeichensatz

Jigdo 142

Jigsaw Download 142

Job 98

jobs 98

jocr 173

joe 195

Joint Photographic Experts Group
292Jokerzeichen **54**, 121, 454

Jotter 398

jove 189

joystick → Freudenknüppel

JPEG → Joint Photographic Ex-
perts Group

K Desktop Environment → KDE

k3b 94

Kaboodle 356

KAddressBook 389

Kaffeine 354

Kalender 108

Kaltstart 8

Kanal (Audio) 327

kandy 397

Karlsruhe

Beginn der Informatik in K. 497

Informatikstudium in K. 497

Rechenzentrum der Universität K.
497Studienzentrum für Sehgeschädig-
te 26

ZUSE Z 22 497

Karlsruher Virtueller Katalog 139

Katalog (Dateien) → Verzeichnis

Katalog (Web) 148

kate 196

katoob 203

kcalc 258

KDE

Desktop 13, 22

KSpread 265

KWord 200

Office-Paket 410

KDE Integrated L^AT_EX Environment
223*kdepim* (Metapaket) 388

kdesktop_lock 45

kdissert 400

kdm 13, 43

Kehrwort 428

Kennung 76, 463

Kern 5

kernel → Kern

key → Taste, Schlüssel

keyboard → Tastatur

Keymap 16

Keyserver 239

kfocus 391

kfract 423

kghostview 170

khelpcenter 38

khexedit 202

kile 223

Kindersuchmaschine 147

kitchensync 397

kjots 399

Klammeraffe 155

Klartext 234

- klatin 404
- klicken 19
- Klient → Client
- kmag 28
- kmail 366
- kmenuedit 23
- KMix 331
- knews 372
- knode 372
- Knopf 19
- Knoppix 7
- Knoten 133
- Kodak Photo Compact Disc 302
- Kolab 393, 395
- Kommando
 - externes K. 54
 - Grundform 9
 - internes K. 54
- Kommando-Interpreter 33, 53
- Kommando-Substitution 56
- Kommandozeile 9, 11
- Kommentar
 - make 241
 - awk 207
 - HTML 224
 - LaTeX 216
 - Shell 57
- Kommunikation 363
- Konfiguration
 - Emacs 189
 - vi 176
- Konqueror 22, 34, **81**, 143, 211, 224
- konsole 20
- Konsole **2**, 9, 22, 44
- kontakt 387
- Konto 43
- Kontrollterminal 2, 95
- Kopete 378
- Kopf (Mail) 370
- KOrganizer 388
- kpdf 171
- kphone 383
- KPresenter 380
- Kryptanalyse 234, 237
- Kryptologie 233
- KsCD 339
- kseg 296
- ksnapshot 21
- KSpread 265
- Kstars 420
- ksync 397
- ktalkd 365
- ktouch 229
- KVocabulary 404
- kvoctrain 404
- kwalletmanager 46
- kwavecontrol 114
- kwlan 114
- kwm 13
- KWord 200, 209
- kxmleditor 227
- lacheck 221
- LADSPA → Linux Audio Developer's Simple Plugin API
- Ländercode 488
- LAME 333
- LAMP 400
- LAN → Local Area Network
- LANG 31
- LanManager 86
- Last File Manager 80
- LaTeX
 - Abbildung 217
 - Anweisung 215
 - Compiler 215
 - DIN-Brief 216
 - Dokumentklasse 215
 - Editor 215
 - Fehlersuche 221
 - Formel 218
 - Grundbegriffe 209, 214
 - LyX 201
 - Register 218
 - Syntaxprüfer 221
 - Tabelle 218
 - teTeX 214
 - Umwandlung 174
- latex2html 174
- latex2rtf 166

- Laufwerk 3
- Layout 208
- LC_ALL 31
- LDAP → Lightweight Directory Access Protocol
- LDAP Data Interchange Format 287
- ldapadd 287
- ldapmodify 287
- ldapsearch 287
- lde 77
- LDIF → LDAP Data Interchange Format
- LDP → The Linux Documentation Project
- leave 107
- Lernkartei 404
- Lesbarkeit 232
- lesen 74
- Lesezeichen 143
- less 169
- lfhex 202
- lfm 80
- lgrep 169
- Licht 293
- lien → Link
- Life 426
- liferea 372
- Ligatur 162
- Lightweight Directory Access Protocol 286
- ligne de commande → Kommandozeile
- LinCVS 246, 249
- Link
 - harter L. 78
 - Link (Hypertext) 223
 - symbolischer L. → weicher L.
 - weicher L. 78
- Linkfäule 149
- linklint 226
- links2 143
- Linux 5
- Linux Audio Developer's Simple Plugin API 335
- Linux Standard Base 6
- Linux TV 352
- Linux User Group 42
- LinuxWiki 42
- LinVDR 352
- Lisa 290
- LISP 188, 190
- Liste (awk) 206
- Liste (Email) 371
- liste d'écoute → Wiedergabeliste
- liste de diffusion → Liste (Email)
- Listengenerator 206
- ln 78
- Load Balancing 97
- loadkeys 16
- Local Area Network 3
- locale 30
- Locale 30
- locale-gen 31
- locale.gen 31
- locate 51, 121, **128**, 313
- Log 300
- logiciel → Software
- Login-Manager → Display-Manager
- Login-Shell 66
- logische Verknüpfung 120
- logout 45
- Lokalisierung 30
- look + feel 9
- lookat 326
- lookup 123
- ls 72, 461
- LUG → Linux User Group
- lv 169
- lynx 143
- LyX 201
- MAC-Adresse 134
- MacOS-Emulator 52
- magnifier 27
- mail 366
- Mail 365, 370
- Mail User Agent 366
- mailcap 368

- mailman 371
- mailx 366
- Majordomo 371
- make 50, 211, **241**, 392
- Makefile 50, 212, 217, **241**
- Makro (Emacs) 194
- Makro (vi) 178
- Malware 47
- Mambo 400, 402
- man 32, 36
- Manager
 - Allgemein 81
 - Datei-M. 80
 - Passwort-M. 46
- manche à balais → Freudenknüppel
- mandb 32
- Mantra 238
- Manual 36
- MaPiVi 303
- Maple 269
- Maske (Datenbank) 276
- Masquerading → Network Address Translation
- Massachusetts Institute of Technology 11
- Massenspeicher 3
- Master List 372
- matériel → Hardware
- Mathematica 269
- MatLab 266
- mattrib 91
- Maus 2
- Maven 392
- MaxDB 283
- Maximize-Button 19
- MBROLA 349
- mc 80
- mcd 91
- mcoppy 91
- mcrypt 235
- mdel 91
- mdir 91
- MegaHAL 421
- Mehrbenutzersystem 8
- mémoire → Speicher
- mémoire de masse → Massenspeicher
- mémoire vive → Arbeitsspeicher
- memory → Arbeitsspeicher
- Memory Stick 3, 89
- mencoder 356
- Menü 10, 23
- Menüdatei 23
- Menü-Button 19
- mesg 365
- messaging électronique → Email
- Meta Email Search Agent 136
- Metacrawler 151
- Metafinder 149
- Metafont 223
- MetaGer 151
- metamail 368
- Metaphone 121
- Metaspinner 151
- Metasprache 227
- Metasuchmaschine 151
- MetaXplorer 151
- Metazeichen
 - HTML 456
 - LaTeX 455
 - make 456
 - regulärer Ausdruck 117, 456
 - Shell 54, 454
 - Texinfo 211
- mformat 91
- mgetty 381
- micq 377
- micro-httpd 42, 131
- Microemacs 189
- MIDI → Musical Instrument Digital Interface
- Midnight Commander 80
- mikmod 329
- MIME → Multipurpose Internet Mail Extensions
- Mindmap 398
- minimalist 371
- Minimize-Button 19
- Mischer 330, 331
- Mixer.app 332

- MiXViews 334
- mkdir 77
- mkfifo 102
- mkfs 90
- mkisofs 91
- mlchat 365
- MLDonkey 142
- mlterm 20
- modellieren 313
- Modem 486
- Modul (Musik) 329
- mogrify 308
- mol 52
- montage 308
- more 168
- morph 311
- morphen 311
- Mosaic 224
- Mosix 97
- mot de passe → Passwort
- moteur de recherche → Suchmaschine
- Motif 13
- motion 362
- motv 360
- mounten 87, 90
- Mounting Point 90
- mouse → Maus
- Moving Picture Experts Group 328, 352
- Mozilla 143, 224
- MP3 328
- mp3blaster 337
- mpack 368
- MPEG → Moving Picture Experts Group
- mpg321 337
- MPlayer 355
- Mr.Project 391
- mren 91
- mserv 338
- Mtools 91
- mtr 135
- mtt 360
- mtype 91
- MUA → Mail User Agent
- Multimedia 351
- Multipurpose Internet Mail Extensions 368
- multisync 397
- MuPAD 270
- MusE 345
- Musical Instrument Digital Interface 329, 342
- Musik-Server 338
- Musikeditor 345
- Muster (awk) 206
- Muster (reg. Ausdruck) 117
- mutt 366, 371
- mv 76
- mwm 13
- mypasswordsafe 46
- MySQL 283
- Nachbarschaft 121
- Nachrüsten von Paketen 49
- Name
 - Benutzer 43
 - Datei 49, 76
 - Paket 49
 - Programm 49
- nano 195
- NAT → Network Address Translation
- NAT-Server 134
- Nationale Teilnehmerrufnummer 383
- Native Language Support 158
- Nautilus 22, 34, 81
- navigateur → Browser
- Navigator 143
- nedit 196
- Negation 120
- neqn 210
- Netatalk 88
- NetBIOS 86
- NetBSD 5
- Netiquette 374
- Netnews 372
- netpbm 303

- Netscape 143, 224
- network → Netz
- Network Address Translation 134
- Network File System 86, 87
- Network Information Center 135
- Network Information Service 87
- Network News Transport Protocol 372
- Network Time Protocol 87
- Network Virtual Terminal 164
- Netz 3
- New Century Schoolbook 162
- newgrp 106
- Newline 115
- Newsgruppe 372
- Newsreader 372
- NF (awk) 207
- NFS → Network File System
- ngspice 298
- NIC → Network Information Center
- nice 97
- nice-Faktor 97
- nicole 422
- NIS → Network Information Service
- Niue 136
- nl 232
- NLS → Native Language Support
- nn 372
- NNTP → Network News Transport Protocol
- no-break space 158
- Noatun 356
- node → Knoten
- nohup 97, 99
- nom de utilisateur → Benutzername
- non-us.debian.org 135
- normalisieren 334
- notation suffixiée → Umgekehrte Polnische Notation
- NoteEdit 345
- Notizbuch 386
- noyau → Kern
- NR (awk) 208
- nroff 209
- nslookup 134
- NTP → Network Time Protocol
- ntpdate 107
- NTR → Nationale Teilnehmerrufnummer
- Nullbyte 115
- Nuxeo CPS 400
- nvi 175
- NVT → Network Virtual Terminal
- NVT ASCII 164
- nvu 225
- OACS → OpenACS
- Oberfläche
 - Benutzer-O. 9
 - multimediale O. 24
- OCR → optical character recognition
- octave 266
- octet → Byte
- od 169
- ODBC → Open Data Base Connectivity
- ODP → Open Directory Project
- Office
 - CrossOver Office 412
 - GNOME-O. 410
 - KOffice 410
 - O.-Paket 276, 405
 - OpenOffice.org 407
- Ogg Vorbis 328
- ogg123 337
- Ogle 356
- Oktalsystem 429
- Oktalzahl 253, 429
- Oktett → Byte
- Online-Handbuch → Manual
- oocalc 265
- oointpress 379
- oowriter 201
- Open Data Base Connectivity 282
- Open Directory Project 149
- Open Movie 319

- Open Sound System 330
- open-EIS 393, 396
- OpenACS 400, 402
- OpenBSD 5
- openCRX 388
- Openexchange 393
- OpenFacts 42
- OpenGroupware 393, 396
- OpenLDAP 287
- OpenOffice.org
 - Office-Paket 407
 - oocalc 265
 - oointpress 379
 - oowriter 201
- OpenRacer 418
- opensched 391
- OpenSG 326
- OpenSSH 110
- openssl 113
- OpenSSL 112
- OpenTracker 326
- Opera 143, 224
- operating system → Betriebssystem
- Opie 354
- optical character recognition → Zeichenerkennung
- Option 9
- or 120
- ordinateur → Rechner
- Ordner → Verzeichnis
- Organizer 385
- Orientierung 163
- ORS (awk) 207
- OSS → Open Sound System
- otl2docbook 187
- otl2html 187
- otl2pdb 187
- outil → Werkzeug
- Outline Prozessor 187
- owner → Besitzer

- package → Paket
- packagesearch 129
- packen 82

- page d'accueil → Startseite
- page de base → Startseite
- Paket
 - Debian-P. 49
 - Office-P. 405
 - P. nachrüsten 49
 - RPM-Paket 50
- Palindrom 428
- pan 372
- Panorama 310
- Panorama Tools 310
- Pantone Matching System 294
- Parallel Virtual Machine 97
- Passphrase 238
- passwordsafe 46
- Passwort 9, 43, 45
- Passwort-Manager 46
- paste 232
- Patch 344
- path → Pfad
- PATH 62
- pattern → Muster
- Pauker 404
- PCB → Printed Circuit Board
- PCL → Printer Control Language
- PCM → Puls-Code-Modulation
- PDF → Portable Document Format
- pdflatex 214
- pdftk 171
- PDI → Personal Data Interchange
- pearpc 52
- Peripherie 4
- Perl 69, 116, 208
- perluète → et-Zeichen
- permission → Zugriffsrecht
- Persistence Of Vision Raytracer 319
- Personal Data Interchange 388
- Personal Information Manager 385
- Perspektive 294
- Pfad
 - absoluter P. 71
 - PATH 62
 - relativer P. 71
- Pfeiltaste 451

- pg 168
- pgaccess 281
- pgadmin 281
- PGP → Pretty Good Privacy
- Pharming 48
- Phase5 225
- Phishing 48
- photopc 302
- PhotoRealistic RenderMan 318
- Photoshop 304
- PHP 69
- phpGroupWare 393, 395
- Physikalisch-Technische Bundesanstalt 107, 293
- pi 252
- pico 195
- Picture Transfer Protocol 301
- PID → Prozess-ID
- pilote → Treiber
- PIM → Personal Information Manager
- pinfo 211
- ping 133
- PingoS 229
- Pingus 419
- Pipe 101
- Pixie 318
- PL/pgSQL 275
- plain text → Klartext
- Planetarium 420
- PlanetPenguin Racer 418
- Planner 391
- play 333
- playlist → Wiedergabeliste
- playmidi 344
- Plone 400, 403
- pmidi 344
- PNG → Portable Network Graphics
- poignée → Freudenknüppel
- pointing device → Zeigegerät
- police → Font
- pollupostage → Spam
- POP → Post Office Protocol
- Portable Document Format 166, 170, 214
- Portable Network Graphics 292
- Pos1-Taste 451
- Positionsparameter 65
- POSIX 6, 30
- Post Office Protocol 368
- PostgreSQL 279
- PostNuke 403
- PostScript 105, 165, 170, 292
- PostScript Printer Description 105
- POV-Ray 319
- PPD → PostScript Printer Description
- Präsentation 379
- Pretty Good Privacy 366
- Primfaktor 253
- primitive → Systemaufruf
- Primzahl 236, 253
- Printed Circuit Board 298
- printer → Drucker
- Printer Control Language 105
- Priorität 97
- Private-Key-Verschlüsselung 234
- Privileg (Datenbank) 280
- PRMan → PhotoRealistic RenderMan
- processeur central → Zentralprozessor
- procmail 367
- Programm 95
- Project Revision Control System 244
- Prompt 9, 64, 106
- proteusworksheet 270
- protocol → Protokoll
- Protokoll 364
- Protokolldatei → Logdatei
- Protux 334
- proximité → Nachbarschaft
- proximity → Nachbarschaft
- Proxy 144
- Prozess
 - Client-P. 11
 - Grundbegriffe 95
 - Kontrollterminal 95
 - nice-Faktor 97

- Priorität eines P. 97
- Prozess-ID 95
- Server-P. 11
- Status 95
- ps 95
- PS → PostScript
- ps2ascii 170
- ps2eps 170
- ps2ps 170
- psbook 105
- Pseudo-Zufallszahl 260
- psi 377
- psmerge 105
- psnup 105
- psql 281
- psresize 105
- pselect 105
- pstops 105
- pstotext 170
- PSUtils 105, 170
- PTP → Picture Transfer Protocol
- ptx 232
- Public-Key-Verschlüsselung 234
- Puffer (Emacs) 191
- Puffer (vi) 184
- Pull-down-Menü 19
- Puls-Code-Modulation 327, 331
- PVM → Parallel Virtual Machine
- pwd 71
- Python 69
- Pyxine 354
- qcad 297
- qiv 302
- QTorrent 142
- quanta 228
- queue 97
- quiteinsane 309
- quoten 56, 117
- qwerty 4
- qwertz 4
- R (Software) 268
- r-Kommando 110
- RAL-Farbe 294
- Ranking 144
- RARE → Réseaux Associés pour la
Recherche Européenne
- RARE Technical Report 491
- Rastergrafik 292
- rastern 293
- Raytracing 295, 313
- rbash 66
- RCS → Revision Control System
- rdesktop 20
- RDN → Relative Distinguished Na-
me
- RE → regulärer Ausdruck
- Real Player 357
- Real Time Streaming Protocol 336
- RealAudio 328
- reboot 8
- rec 333
- Rechner 1
- recode 174
- reconnaissance optique de caractères
→ Zeichenerkennung
- reconnaissance vocale → Spracher-
kennung
- recover 77
- recover (vi) 180
- Red Hat Package Manager 50
- Redaktionssystem 400
- REDUCE 269
- Regional Playback Control 352
- Regionalcode → Regional Play-
back Control
- Register 218
- regulärer Ausdruck **116**, 182, 205, 456
- Rekursion 424
- Relative Distinguished Name 287
- RenderMan 315
- RenderMan Interface Bytestream
313
- rendern 294, 313
- renice 97
- répertoire → Verzeichnis (Dateien)
- Répertoire (Zeichenvorrat) 155
- Report (Datenbank) 276
- request → Auftrag

Request For Comments

allgemein 41
 ausgewählte RFCs 491
 RFC 1282 110
 RFC 1459 375
 RFC 1642 161
 RFC 1939 368
 RFC 2045 368
 RFC 2060 368
 RFC 2229 124
 RFC 2246 112
 RFC 2251 286
 RFC 2307 289
 RFC 2550 281
 RFC 2793 378
 RFC 2849 287
 RFC 2910 104
 RFC 2911 104
 RFC 2916 383
 RFC 3261 383
 RFC 3283 388
 RFC 3380 104
 RFC 3629 161
 RFC 3912 286
 RFC 821 365
 RFC 854 110, 164
 RFC 959 140
 RFC 977 372
 Son-of-1036 370
 requête → Auftrag
 réseau → Netz
 réseau étendu → Wide Area Network
 réseau local → Local Area Network
 Réseaux Associés pour la Recherche Européenne 491
 Reset-Knopf 8
 Resource Interchange File Format 328
 Rest der Welt 74
 rev 232
 Reverse Polish Notation → Umgekehrte Polnische Notation
 Revision Control System 243, 244
 rexima 331

ReZound 334
 RFC → Request For Comments
 RGB-Farbmodell 293, 459
 Rhythmbox 336
 RIB → RenderMan Interface Byte-stream
 Rich Text Format 166, 198
 RIFF → Resource Interchange File Format
 Ripper 339
 rlogin 110
 rm 52, **75**, 76
 rmdir 77
 rmligs-german 221
 rmnl 232
 RMS → Stallman, R. M.
 Rollkugel 2
 Rootverzeichnis → Wurzelverzeichnis
 Rosegarden 345
 Roster 377
 ROT13 205, 234
 RPM → Red Hat Package Manager
 RPM-Paket 50
 RPN → Reverse Polish Notation
 RSA-Verfahren 235
 RTF → Rich Text Format
 rtf2latex 166
 rtime 107
 RTSP → Real Time Streaming Protocol
 rubrica 389
 Runlevel 106
 rwho 106, 110

 S (Software) 268
 Sachregister 207
 Samba 88
 SANE 309
 SAP DB 283
 SAS (Software) 268
 SASL → Simple Authentication and Security Layer
 Satz
 awk 206

Datenbank 274
 Satzprogramm 209
 sauvegarde → Sicherungskopie
 say 348
 saydate 347
 saytime 347
 sc 263
 Scalable Vector Graphics 292
 scanimage 309
 Scanner 309
 scanneur → Scanner
 SCCS → Source Code Control Sys-
 tem
 Sced 319
 Scene Description Language 319
 Schall 327
 Schlappscheibe → Diskette
 Schlüssel (Kryptografie) 234
 Schlüssel (Sortieren) 121
 Schlüsselbund 239
 Schnittstelle 4
 schreiben 74
 Schrift
 Art 162
 Grad 162
 Proportionalschrift 162
 Schnitt 162
 Weite 162
 Scilab 267
 scp 110, 111
 Scrabble 428
 screen → Bildschirm
 Screen Dump → Screenshot
 Screenreader 349
 Screenshot 21, 320
 Scribus 213
 script 85
 script de commandes → Shellskript
 ScrollKeeper 38
 ScrollServer 38
 SDL → Scene Description Language
 ge
 Secure Internet Live Conferencing
 376
 Secure Shell 110

Secure Sockets Layer 112
 sed 116, 164, 174, **205**, 278
 sed-Script 205
 Sehgeschädigte 26
 Sehvermögen 26
 SelfHTML 227
 SelfLinux 40
 seq 252
 Server 11, 12
 Server Message Block 86
 Server Parsed Document 226
 Server Side Include 226
 serveur → Server
 serveur mandaté → Proxy
 service d'annuaire → Verzeichnis-
 dienst
 session → Sitzung
 Session Initiation Protocol 383
 setxkbmap 17
 sftp 111
 SGML → Standard Generalized
 Markup Language
 sgrep 123
 sh 33
 Shading Language 315
 Shebang 68
 shed 202
 Shell 33
 C-Shell 69
 Distributed S. 69
 Funktion 424
 rc (Plan 9) 69
 restricted S. 66
 Secure S. 16
 TENEX-C-Shell 69
 Z-Shell 69
 Shellskript 67, 424
 Shortcut 10
 showrgb 459
 shtml 226
 shutdown → herunterfahren
 Sicherheit 45
 Sicherungskopie 45
 Signatur (Mail) 370
 Signatur (Unterschrift) 371

- signet → Lesezeichen
- SILC → Secure Internet Live Conferencing
- Silky 376
- Simple Authentication and Security Layer 287
- Sinek 354
- Single UNIX 6
- SINGULAR 272
- SIP → Session Initiation Protocol
- Sitzung 8, 43
- Skript
 - awk 207
 - GIMP 307
 - ImageMagick 308
 - sed 205
 - Shell 67
 - SQL 275
- SL → Shading Language
- slapd 287
- slogin 16, **110**, 111
- SLOX → SuSE Linux Openexchange Server
- smartlist 367, 371
- SMB → Server Message Block
- smbc 88
- smbclient 88
- Smoke Test 24
- snd 334
- sodipodi 296
- Softlink → Link, weicher
- Softphone 382
- Software
 - S. entfernen 51
 - S. vs. Hardware 1
- son → Schall
- sort 230, 278
- sound → Schall
- Soundex 121
- Soundfont 344
- Source Code Control System 243, 244
- sources.list 355
- souris → Maus
- SoX 333
- Spam 364, 367
- Speakup 349
- speech recognition → Spracherkennung
- Speicher 3
- spell 231
- sphinx2 347
- split 77
- Spoofing 48
- Spracherkennung 346
- SPSS (Software) 268
- SQL → Structured Query Language
- ssh 110
- SSH → Secure Shell
- ssh-keygen 111
- sshd 110
- SSI → Server Side Include
- SSL → Secure Sockets Layer
- ssp 232
- Standard Generalized Markup Language 227
- StarOffice 408
- Startseite 144
- startx 13
- Statistik 268
- stderr 60
- stdin 60
- stdout 60
- Steuerzeichen (ASCII) 442
- Stitcher 310
- storage → Massenspeicher
- Storage Area Network 3
- Streaming Editor 205
- strings 123
- Structured Query Language 275, 289
- Struktur (Text) 208
- style 233
- Subversion 243, 250
- Suchfibel 133
- Suchlexikon 133, 149
- Suchmaschine 144
- Suchmaschinenlabor 133

suite bureaucratique → Office-Paket
 superformat 89
 Supervisor → Verwalter
 SuSE Linux Openexchange Server 393, 396
 suspendieren 98
 SVG → Scalable Vector Graphics
 SVK 250
 Swap-Datei (vi) 180
 sweep 334, 335
 Swirl 7
 sylpheed 366
 Symlink → Link, symbolischer
 sync 75
 synchron 98
 Syntax-Hervorhebung (Emacs) 190
 Syntax-Hervorhebung (vim) 178
 Syntaxprüfer 221
 system call → Systemaufruf
 System-V-Drucken 103
 Systemanfrage → Prompt
 Systemaufruf 35
 système d'exploitation → Betriebssystem
 Systemmanager → Verwalter
 Systemprompt 9

 Tabellenkalkulation 262, 273
 Tablespace 274
 tableur → Tabellenkalkulation
 Tagged Image File Format 292
 tail 75
 talk 365
 talkd 365
 tar 50, **84**, 89
 Tarball 50
 Taschenrechner 254
 Tastatur
 deutsche T. 16
 DVORAK-T. 16
 internationale T. 16
 T. umschalten 16
 Terminal 2
 Taste 16, 451

Tastenanordnung 16
 Taubblinde 26
 tbl 210
 Tcl/Tk 69
 tclspice 298
 ted 198
 tee 85
 Teilnehmerrufnummer, Nationale 383
 tel écran – tel écrit → WYSIWYG
 Teletext 360
 telnet 110
 TERM 11, **62**, 177
 termcap 177
 Terminal 2, 9
 terminfo 177
 Terminkalender 388
 teTeX 214
 TeX 213
 texi2dvi 211
 Texinfo 38, 210
 Text over IP 378
 Text-To-Speech Control Protocol 348
 texte en clair → Klartext
 Textterminal 44
 The Linux Documentation Project 39
 think 399
 Thunderbird 366, 372
 tidy 226
 TIFF → Tagged Image File Format
 Time stretching 344
 Times Roman 162
 TiMidity 343
 tin 372
 tina 386
 Tipptrainer 229
 tkhtml 225
 tksmb 88
 TLDP → The Linux Documentati-on Project
 TOCHNOG 301
 Tod-Taste 451
 ToIP → Text over IP

- tool → Werkzeug
- top 96
- Totem 354
- touch 80
- touche → Taste
- tr 164, 174, **232**
- traceroute 135
- trackball → Rollkugel
- Tracker 329
- Transaktion 274
- transfig 295
- Transport Layer Security 112
- Treiber 35
- Trennzeichen (awk) 207
- Trennzeichen (Bash) 62, 454
- Triple-DES 235
- trn 372
- troff 209
- tset 177
- TTSCP → Text-To-Speech Control Protocol
- Türme von Hanoi 243, 256, **424**
- TUTOS 393, 395
- TuxCards 387
- TuxRacer 417
- tuxtype 229
- tvtime 360
- TWAIN 309
- twm 13
- txt2pho 349
- TYPO3 400, 403
- u2ps 170
- UBE → Unsolicited Bulk Electronic Mail
- Ubuntu 7
- UCS → Universal Character Set
- Uhrzeit 107, 347
- UID → Benutzer-ID
- umask 75
- Umgebung 62
- Umgekehrte Polnische Notation 257
- Umlaut 157, 186
- Umlenkung 60
- uncompress 83
- unexpand 232
- unicode 159
- Unicode 158, 443
- Unicode Consortium 158
- Unicode Transformation Format 159
- unicnv 174
- Uniform Resource Identifier 142
- Uniform Resource Locator 142
- uniq 207, 232
- unison 397
- Universal Character Set 158, 443
- Universität
 - Berlin 145
 - Bochum 145
 - Bonn 347
 - Cambridge, UK 361
 - Carnegie Mellon 347
 - Edinburgh 347, 348
 - Eichstätt 150, 296
 - Erlangen 137, 148
 - Essen 301
 - Hannover 133, 136, 151
 - Kaiserslautern 145, 272
 - Karlsruhe 145, 148
 - Königsberg 374
 - Konstanz 133, 136, 145
 - KTH Stockholm 347
 - Leipzig 137
 - Mainz 145
 - Mons 349
 - München 137
 - Pierre et Marie Curie, Paris 301
 - Saarbrücken 291
 - Siegen 132
 - Stuttgart 346
 - Trier 137
- UNIX 6
- unpaper 309
- unrtf 166, 173
- Unsolicited Bulk Email 364, 367
- unzip 84
- update-menus 23
- updatedb 51, 121, **128**, 313

UPN → Umgekehrte Polnische Notation
 URI → Uniform Resource Identifier
 URL → Uniform Resource Locator
 usage → Gebrauch
 Usenet 372
 user → Benutzer
 user interface → Benutzeroberfläche
 UserLinux 7
 UTF → Unicode Transformation Format
 utilisateur → Benutzer
 utilitaire → Dienstprogramm
 utility → Dienstprogramm
 uudevview 368

 V.E.R.A 138
 V4L → Video4Linux
 Variable
 awk-Variable 207
 Positionsvariable 65
 Shellvariable 62
 Umgebungsvariable 62
 vbs 298
 vCalendar 388
 vCard 388
 vche 202
 VDR → Video Disc Recording
 vec2web 297
 Vektorgrafik 292
 vektorisieren 293
 Verilog 298
 Verknüpfung
 bitweise V. 121
 logische V. 120
 V. einer Datei → Link
 verlinken 78
 Verschlüsselung
 Asymmetrische V. 235
 Content Scrambling 352
 Email 371
 Hybride V. 237
 Kryptologie 233

ROT13 234
 RSA-V. 235
 Secure Shell 111
 Secure Sockets Layer 112
 Symmetrische V. 234
 Version 229
 Versionsverwaltung 243
 Verwalter 2, 8, 43, 45, 49, 102, 128, 280
 Verwalter (Datenbank) 276, 279
 Verweis → Link
 Verzeichnis
 aktuelles V. → Arbeitsverzeichnis
 Allgemeines 71, 277
 Arbeitsv. anzeigen 71
 Rootverzeichnis → Wurzelv.
 V. anlegen 77
 V. auflisten 72
 V. durchsuchen 74
 V. löschen 77
 V. umbenennen 77
 V. verschieben 77
 V. wechseln 71
 Wurzelv. 71
 Verzeichnisdienst 273, 277
 vi 116, 122, 158, 164, **174**, 278
 aufrufen 179
 beenden 180
 Konfiguration 176
 navigieren 181
 recover 180
 Schnellstart 174
 zurückschreiben 180
 Video Disc Recording 352
 Video4Linux 352
 Videotext 361
 view 169, 179
 View (Datenbank) 282
 vim 175
 VimOutliner 187
 vimspell.vim 186
 Viper 126
 Virens Scanner 47
 Virtual Library 148
 Virtual Private Network 3

Virtual Reality Markup Language 326
 Virtuelle Realität 325
 Virus 47
 vis 232
 VLC Media Player 353
 VO → VimOutliner
 vo_maketags 187
 Voice over IP 382
 voice recognition → Spracherkennung
 VoIP → Voice over IP
 Vordergrund 98
 VPN → Virtual Private Network
 VR → Virtuelle Realität
 VR Juggler 326
 VRML → Virtual Reality Markup Language

 W3C → World Wide Web Consortium
 wall 365
 WAN → Wide Area Network
 Warmstart 8
 warpen 311
 WAV → Waveform Audio Data
 Waveform Audio Data 328
 wavemon 114
 wc 232
 WCMS → Web Content Management System
 wdm 13
 Web → World Wide Web
 Web Content Management System 400
 Web-Browser 142
 WebCalendar 388
 webcam 361
 WebGlimpse 123
 Webkamera 361
 Webmail 369
 Webmin 130
 Webserver 130
 websvn 250
 Wechselsprechen 363

Weltzeituhr 107
 Werkzeug 9
 wget 140, 142
 What You See Is What You Get 167, **199**, 209
 whereis 128
 which 128
 whichman 37
 Whitespace 115
 who 106
 whoami 106
 whois 135
 Whois 286
 Wide Area Network 3
 Widerrufszeug 239
 Wiedergabeliste 356
 Wiki 273, 277, **284**
 Wiki (Debian) 42
 Wiki (Linux) 42
 Wikipedia 41, 137, **284**
 WikiReader 41
 Wildcard → Jokerzeichen
 window → Fenster
 Window-Manager 43
 WindowMaker 13
 Windows Internet Name Service 88
 Windows Terminal Server 20
 Windows-Emulator 52
 wine 52
 Wine Tools 52
 WINS → Windows Internet Name Service
 wipe 77
 Wireless Local Area Network 3, 113
 wireless-tools 114
 WLAN → Wireless Local Area Network
 wmaker 13
 wmmixer 332
 word2x 172
 wordplay 428
 wordview 172
 WorkBone 338
 World Wide Web 142, 223

World Wide Web Consortium 158
 Wörterbuch 124
 Wortschatz 403
 write 365
 Wurzelverzeichnis 71
 wv 172
 WWW → World Wide Web
 WYSIWYG → What You See Is
 What You Get

X → X Window System
 X Color Management System 294
 X File Explorer 81
 X MultiMedia System 337,356
 X Northern Captain 81
 X Window System 11
 X-Client 13
 X-Printer 12
 X-Server 12
 X.500 286
 X11 → X Window System
 X11-Farbe 459
 X11-Protokoll 12
 X11R6 → X Window System
 Xaos 423
 xargs 127
 xauth 16
 Xauthority 16
 xawtv 359
 xboard 413
 xcdroast 94
 xclock 107
 xcolmix 459
 xcoloredit 459
 xcolors 459
 xcolorsel 459
 xdm 13,43
 xdpinfo 19
 xdvi 209
 XEmacs 189
 Xerox 11
 xfe 81
 xfig 295
 xfractiont 424

XFree86 12
 xgrab 21
 xhost 14
 Xine 354
 xine-check 354
 xinit 13
 xjed 198
 xkbsel 17
 xkeysw 17
 xlhhtml 173
 xlife 426
 xls2csv 172
 XmAddressBook 389
 xmag 27
 xmcid 338
 XML → Extensible Markup Lan-
 guage
 xmlto 227
 xmms 337,356
 xmodmap 17
 xmorph 311
 xnc 81
 xor 120
 xpcd 302
 xpdf 171
 xprop 19
 xsane 309
 xscanimage 309
 xsmbrowser 88
 xterm 9,20,44,59
 XTerm Control Sequence 59
 xtermcontrol 20
 xtermset 20
 xv 21,302
 XVoice 347
 xwd 21
 xwininfo 19

 YACAS 270
 YafRay 319
 Yahoo 148
 yasr 349
 Yellow Pages → Network Informa-
 tion System
 Yelp 38

- ytalk 365
- yudit 203
- Zahl
 - binär codierte Dezimalz. 429
 - Dezimalzahl 253, 429
 - Dualzahl 253, 429
 - Ganzzahl 251
 - Gleitkommazahl 251
 - Hexadezimalzahl 253
 - Oktalzahl 253
 - Primzahl 236, 253
 - Pseudo-Zufallszahl 260
 - Zufallszahl 260
- Zahlensystem 429
- Zeichen
 - Steuerzeichen 156
 - Umlaut 157
 - Zeichenvorrat 155
- Zeichenerkennung, optische 173
- Zeichenkette 115
- Zeichensatz
 - ASCII 156, 436
 - ASCII-Steuerzeichen 442
 - Codetafel 156
 - EBCDIC 157, 436
 - German ASCII 441
 - Grundbegriffe 155
 - IBM-PC 157, 436
 - ISO 10646 158
 - ISO 8859-1 157
 - ISO 8859-15 157
 - Latin-1 157, 443
 - Latin-2 449
 - Latin-9 157
 - NVT ASCII 164
 - ROMAN8 157, 436
 - Tabelle Nr. 850 155
 - UCS-2 159
 - UCS-4 159
 - Unicode 158
 - Unicode Transformation Format 159
 - Universal Character Set 158
 - UTF-16 159
 - UTF-18 496
 - UTF-32 159
 - UTF-7 161
 - UTF-8 159
 - UTF-9 496
 - Zeichencodierung 156
 - Zeichenmenge 155
 - Zeichenposition 155
 - Zeichenvorrat 155
- Zeigegerät 2
- Zeilenabstand 163
- Zeilenwechsel 164
- Zeitserver 107
- Zeitstempel 79
- Zentraleinheit 3
- Zentralprozessor 3
- zgrep 123
- Ziffer 251
- zinf 336
- zip 84
- Zirkel und Lineal 296
- ZMS 400
- Zope 400, 402
- zsafe 46
- Zufallszahl 260
- Zugänglichkeit 25
- Zugriffsrecht 74, 280, 461
- ZUSE Z 22 497