Pearson Education

ADDISON-WESLEY

Linux-Server im kommerziellen Netzwerk

Pearson Education

ADDISON-WESLEY

Linux Specials

Pearson Education

ADDISON-WESLEY

Peter Samulat

Linux-Server im kommerziellen Netzwerk

Planung, Integration, Betrieb

Bitte beachten Sie: Der originalen Printversion liegt eine CD-ROM bei.
In der vorliegenden elektronischen Version ist die Lieferung einer CD-ROM nicht enthalten.
Alle Hinweise und alle Verweise auf die CD-ROM sind ungültig.

ADDISON-WESLEY

An imprint of Pearson Education

München • Boston • San Francisco • Harlow, England
Don Mills, Ontario • Sydney • Mexico City
Madrid • Amsterdam

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Ein Titeldatensatz für diese Publikation ist bei
der Deutschen Bibliothek erhältlich.

Umwelthinweis:
Dieses Buch wurde auf chlorfrei gebleichtem Papier gedruckt.
Die Einschrumpffolie – zum Schutz vor Verschmutzung – ist aus umweltfreundlichem und recyclingfähigem PE-Material.

10 9 8 7 6 5 4 3 2 1

03 02 01 00

ISBN 3-8273-1631-6

ein Imprint der Pearson Education Company Deutschland GmbH
Martin-Kollar-Straße 10–12, D-81829 München/Germany

Einbandgestaltung: Hommer Design Production, Haar bei München
Lektorat: Susanne Spitzer, sspitzer@pearson.de
Korrektorat: Angelika Obermayr
Herstellung: TYPisch Müller, Gräfelfing
Satz: reemers publishing services gmbh, Krefeld
Druck und Verarbeitung: Druckerei Schoder, Geisthofen
Printed in Germany

Pearson Education

ADDISON-WESLEY

Inhaltsverzeichnis

Vorwort **VII**

1 Netzwerkplanung **1**

1.1 Linux im Aufwind 2
1.2 Netzwerkressourcen 8
1.3 Serverhardware 23

2 Basiskonfiguration Linux-Server **41**

2.1 Grundinstallation 41
2.2 Systemsicherheit 66
2.3 Hochverfügbarkeitssysteme 67
2.4 Grundlagen der Systemverwaltung 69

3 Konfiguration der Netzwerkdienste **127**

3.1 Standard-Netzwerkfunktionen 127
3.2 Druckdienste 164
3.3 Datei- und Verzeichnisdienste 185
3.4 Zeitserver 235
3.5 Programmdienste 246
3.6 Internetdienste 284
3.7 E-Mail – elektronische Post 308
3.8 Router zur Netzwerkverbindung 317

A Anhang **321**

A.1 Die GPL im Einzelnen 321
A.2 Wie wird die GPL auf eigene neue Programme angewendet? 325
A.3 Die beiliegende CD-ROM 327

Quellenverzeichnis **329**

Stichwortverzeichnis **331**

Pearson Education

ADDISON-WESLEY

Pearson Education

ADDISON-WESLEY

Vorwort

Ziel dieses Buches ist es, Personen mit Erfahrungen in PC-Netzwerken, insbesondere auch den Systemverwaltern von Novell- und Windows-NT-Netzwerken, den einfachen Aufbau und Einsatz linuxbasierter Systeme nahe zu bringen. Hier geht es darum, einen kompetenten und vollständigen Schritt-für-Schritt-Leitfaden zum kommerziellen Einsatz von Linux-Systemen zu geben, im Gegensatz zu vielen anderen Büchern, in denen die Einführung in das Betriebssystem Linux, die einfache Installation und die Grundkonfiguration kleiner Systeme zentrales Thema ist.

Dargestellt wird die Planung, Realisierung und der Betrieb eines typischen Büronetzwerkes, in dem Linux-Server die zentralen Komponenten darstellen.

In kurzer Zeit ist deutlich geworden, dass Linux eine ernstzunehmende Konkurrenz für die »etablierten« PC-Netzwerkbetriebssysteme darstellt. Stabilität, geringe Kosten und die Tatsache, dass immer mehr Hard- und Softwareanbieter Linux-Lösungen präsentieren, zeigen die Eignung von Linux für einen kommerziellen Einsatz.

Damit bietet sich an, bestehende Novell- oder Windows-NT-Systeme um Linux-Komponenten zu erweitern oder sogar eine Migration hin zu Linux-basierten Systemen durchzuführen.

Weiterhin ermöglicht Linux dem erfahrenen Systemverwalter von Novell- oder Windows-basierten Netzwerken den Einblick und Einstieg in ein seit Jahrzehnten bewährtes und weltweit verfügbares Netzwerkbetriebssystem. UNIX-basierte Systeme dominieren die Internet-Strukturen; Novell- und Windows-Systeme werden mit Hochdruck weiterentwickelt, um zumindest einem Teil dieser Anforderungen gerecht zu werden. Die Auseinandersetzung mit den Leistungsmöglichkeiten von Linux hilft dem Systemverwalter, die Leistungsdaten, die Stabilität und den Verwaltungsaufwand seines Netzwerkes kritisch zu beurteilen.

Dieses Buch beschreibt schrittweise Aufbau, Test und Betrieb von typischen Netzwerk-Funktionen: Der Linux-Server wird vom einfachen Datei- und Druckserver zum multifunktionalen Kommunikationsserver mit Internetzugang, E-Mail- und Faxdiensten, Datenbankanbindung und weiteren Leistungsmerkmalen erweitert. Die Darstellung basiert auf der SuSE-Distribution in der Version 6.3 (Kernel 2.2.13).

Peter Samulat

Hemme, Februar 2000

Pearson Education

ADDISON-WESLEY

1 Netzwerkplanung

PC-Netzwerke sollen heute wesentlich mehr bereitstellen als nur Druck- und Dateidienste. Immer mehr Funktionen werden zentralisiert; Intranet- und Internet-Dienste, wie z.B. E-Mail und der Zugriff auf Informationssysteme, werden Standard.

Schon im Kleinbetrieb mit nur wenigen PC-Arbeitsplätzen entstehen so enorme Kosten: Die Hardware für den Server, die PC-Arbeitsplätze und für die Netzwerk-Infrastruktur muss gekauft werden; die Betriebssysteme werden nach der Anzahl der Lizenzen berechnet. An den Arbeitsplätzen haben sich die Microsoftsysteme Win9x/Windows NT so weit durchgesetzt, dass nur mit einer enormen Anstrengung noch über Alternativen nachgedacht werden kann. Jeder, der mit einem PC umgehen muss, hat den Umgang mit Microsoft- Produkten gelernt. Der Einsatz anderer Programme erfordert zusätzlichen Schulungsaufwand, denn andere Office-Pakete unterscheiden sich zwar kaum in ihrer Leistungsfähigkeit von Microsoft-Produkten, aber in ihrer Benutzeroberfläche. Damit ist es heute (noch) die Ausnahme, über den Einsatz von Linux auch am PC-Arbeitsplatz nachzudenken, denn es ist weder Zeit noch Geld da, sich mit konkurrierenden Office-Paketen zu beschäftigen. Wer aber heute nur die Kosten für einen PC-Arbeitsplatz mit Office-Paket zu der Microsoftlösung in Relation setzt, kann doch ins Wanken kommen. Vielleicht rechnet sich der zusätzliche Aufwand ja doch ...

In diesem Buch soll es also um den Einsatz von Linux als Server-Betriebssystem gehen. Der PC-Arbeitsplatz spielt dabei nur eine untergeordnete Rolle, so wird z.B. auf die typischen Arbeitsplatz-Anwendungen unter Linux nicht eingegangen.

Was ist jetzt grundsätzlich anderes, wenn für das Netzwerk ein Server unter Linux geplant wird? Sind die Hardwareanforderungen wirklich so gering, wie es die »Einstiegsliteratur« behauptet? Soll tatsächlich der uralte 486er aus dem Keller geholt und zum Hochleistungsserver aufgebaut werden? Kann da nicht unheimlich viel Geld gespart werden?

Im folgenden Abschnitt werden die Vorausetzungen für den Aufbau eines kommerziellen Netzwerkes unter Linux dargestellt. Im Vordergrund stehen Planung sowie hard- und softwareseitige Realisierung eines Büronetzwerkes in einem Klein- und Mittelbetrieb, in dem ein oder mehrere Server unter Linux, vielleicht in Kombination mit Novell- oder Windows NT-Servern, eine zentrale Rolle spielen. Die PC-Arbeitsplätze werden aber weiterhin unter Windosw 9x/NT betrieben.

1.1 Linux im Aufwind

Linux ist ein frei verfügbares Unix-ähnliches 32-/64-Bit-Betriebssystem für IBM-PC kompatible Rechner und für eine Reihe weiterer Hardwareplattformen. Echtes Multitasking, Multiuserbetrieb und Mehrprozessorfähigkeit, verbunden mit der auch für kommerzielle Anforderungen ausreichenden Stabilität haben Linux in den letzten Jahren immer weiter auf den Markt vordringen lassen.

Die rechtliche Grundlage für den Einsatz der »Open Source«-Software Linux liefert die im Anhang abgedruckte GNU General Public License GPL. Der Einsatz von Linux wird damit nicht von kommerziellen Interessen bestimmt. Wichtig ist, dass die lauffähigen Betriebssystemteile und die dazugehörigen Sourcecodes frei verfügbar sind und damit die Basis eigener und gemeinsamer Weiterentwicklung dieses Systems darstellen. Die einzige Auflage ist, die Arbeit, inklusive des geänderten Sourcecodes, wieder der Allgemeinheit zur Verfügung zu stellen.

Linux ist tatsächlich auf PCs ab dem Intel-386er-Prozessor einsetzbar. Dieses Argument wird immer noch als ein Grund für den Einsatz von Linux angeführt; es soll die geringen Hardwareanforderungen verdeutlichen. In der Praxis ist diese Hardware aber schon lange nicht mehr die Basis kommerzieller Anwendungen. Wer sich heute vornimmt, ein Linux-System auf einem 386er aufzubauen, wird nur wenig Freude daran haben. Zu umfangreich sind die Einschränkungen im Vergleich zu Systemen auf aktuellen Hardwareplattformen.

Es ist aber sicherlich reizvoll, z. B. auf einem 486er ein leistungsfähiges »Diskless X-Terminal« unter Linux zu realisieren [Such99], das insbesondere für den harten Einsatz in Schulen oder anderen Bildungseinrichtungen als vollwertiger Netzwerk-arbeitsplatz eingesetzt werden kann. Der Trend zurück zum »dummen« Terminal am Arbeitsplatz ist heute in vielen Bereichen spürbar, hier kann tatsächlich ältere Hardware unter Linux noch sinnvoll eingesetzt werden.

Im Einsatz für den Schul- und Ausbildungsbetrieb kann Linux von erheblichem Vorteil sein. Von den Softwarekosten einmal ganz abgesehen, können robuste und praxisgerechte Systeme realisiert werden, die alle benötigten Funktionen zur Verfügung stellen. Die Programme am PC-Arbeitsplatz müssen dazu nicht aufwändig vor unbefugtem Zugriff abgesichert werden; der Rechner wird entweder als einfaches Terminal betrieben oder die auf dem Arbeitsplatz laufende Sofware wird bei einem Neustart des Systems über eine »Image« vom Linux-Server geladen. Der PC-Arbeitsplatz ist wieder im Grundzustand. Über grafisch orientierte »Fernsteuersoftware« kann der Ausbilder direkt auf einem Schüler-PC eingreifen oder er überträgt das Bild seines Arbeitsplatzes auf alle PCs im Ausbildungsnetzwerk. Dass ein solches System auch über einen Internetzugang und E-Mail verfügt, ist schon fast selbstverständlich.

Neben dem kommerziellen Bereich fasst Linux auch immer weiter Fuß in »privaten« Anwendungen. Grafische Bedieneroberflächen, leistungsfähige Office-Pakete, kaufmännische Anwendungen und Kommunikationsprogramme für das Internet lassen Linux auch als Arbeitsplatzbetriebssystem interessanter werden. Auch immer mehr Spiele sind in einer Linux-Version verfügbar; damit steigt auch für den Einsteiger der Reiz, die Benutzung dieses Betriebsystems zu lernen. Die Rolle der Spiele sollte nicht unterschätzt werden, denn für viele, die später am PC arbeiten, sind sie der Anfang der Lernens am Computer.

Welche Anwendungsbandbreite Linux-Systeme heute bereits erreicht haben, zeigen Ansätze, auch technische Anwendungen, z.B. aus dem Meß und Regelungstechnik, unter Linux auszuführen. Das Betriebssystem ist dazu so erweitert worden, dass auch zeitkritische Aufgaben mit vorhersagbaren Antwortzeiten erfüllt werden können. Ein entsprechend modifiziertes »RT-Linux« (Real Time Linux) ist verfügbar oder der interessante Versuch, ein Linux-System als Emulation oder sogar »Boot«-Plattform für beliebige andere PC-Betriebssysteme einzurichten. Die Windows-Emulation *wine* gehört heute schon zum Standard-Lieferumfang vieler Linux-Distributionen; die Software der Firma *Vmware* ermöglicht das Booten von DOS-, Win9x- und Windows-NT-Systemen unter Linux [Kirs99].

Linux ist »Jahr-2000-fest«. Bei Linux (und bei allen anderen UNIX-Systemen) wird die Zeit relativ zum Jahreswechsel am 01. 01. 1970 berechnet. Hierfür wird ein Sekundenzähler verwendet, dessen Länge derzeit 32 Bit beträgt und der damit bis zum Jahr 2038 ausreicht (Die Umstellung auf 64 Bit wird bestimmt nicht so lange dauern!).

1.1.1 Netzwerke im kommerziellen Einsatz

Bei der Neuanschaffung PC-basierter Netzwerke anschaffen oder der Aktualisierung älterer Netzwerksysteme fällt die Entscheidung in vielen Fällen zu Gunsten der Microsoft-Systeme unter Windows NT. Sowohl die Netzwerkarbeitsplätze, die Clients, wie auch die Server werden auf dieses Betriebssystem umgestellt. Enorme Investitionen im Bereich der Hard- und Software werden getätigt.

Warum hat sich dieser »Trend« auf dem Markt so stark durchsetzen können? Wieso wird eine Vielzahl bewährter und stabiler Netzwerke insbesondere unter Novell Netware teilweise mit erheblichem Kostenaufwand zu Windows NT migriert bzw. vollständig durch solche Systeme abgelöst?

Microsoft hat es in den vergangenen Jahren geschafft, den PC-Anwendern eines deutlich zu machen: Windows-NT-Systeme fast beliebiger Größe sind einfach zu installieren, stabil und sicher. Gerade die »Entscheider« in Industrie und Wirtschaft kennen seit Jahren Betriebssysteme wie Windows 95/98 oder Windows NT vom eigenen Arbeitsplatz. Die Meinung herrscht vor, dass Netzwerke auf dieser Basis ebenso einfach und unkompliziert zu bedienen seien. Schnell fällt so der Entschluß,

das gesamte Netzwerk auf eine »durchgängige Basis« zu stellen, nur Software eines Herstellers zu verwenden und dann den Systemverwalter, der ja kaum noch Aufgaben zu haben scheint, nur noch in Zweit- oder Drittfunktion zu besetzen.

Sehr schnell stellt sich jedoch heraus, was seit Jahren eigentlich schon selbstverständlich ist: PC-Netzwerken ab einer bestimmten Größe benötigen einen gut ausgebildeten und erfahrenen Systemverwalter, der Sicherheitslücken schließt und einen stabilen Systembetrieb gewährleistet. Für Microsoftnetzwerke ist hier die Qualifikation *MCP* (Microsoft Certified Professional) ode besser sogar *MCSE* (Microsoft Certified Systems Engeneer) unbedingte Voraussetzung.

Ist die Beschaffung anderer Netzwerkbetriebssysteme wie z.B. Novell Netware, Unix oder auch Linux geplant, ist es selbstverständlich, bereits im Vorfeld gut ausgebildetes Fachpersonal vor Ort hinzuzuziehen. Kann bei Novell Netware auch auf entsprechend qualifiziertes Personal zurückgegriffen werden (Certified Novell Adminstator, *CNA*, oder Certified Novell Engeneer, *CNE*), so ist das bei Unix- oder Linux-basierten Netzwerken schwierig. Es stellt sich die Frage, woher das unbedingt benötigte Fachwissen kommen soll, ob es vergleichbare Zertifikate gibt oder welche Systemhäuser professionellen Support geben können.

1.1.2 Linux – ein System ohne Schulungsangebot?

Linux kommt aus dem Ausbildungsbereich; es entstand im Rahmen der Informatikausbildung und wird auch heute zu Lehr- und Schulungszwecken in der universitären Ausbildung eingesetzt.

Aber was ist mit der Aus- und Weiterbildung der Netzwerksystemverwalter, wer führt entsprechende Qualifizierungen durch und belegt das erworbene Fachwissen durch eine anerkannte Prüfung oder Zertifizierung?

Hier hat sich in wenigen Monaten die Weiterbildungslandschaft erheblich entwickelt. Immer mehr namhafte System- und Schulungshäuser (DITEC, HP, ...) bieten Kurse an, in denen Grundlagen und Spezialwissen im Bereich der Linux-Systeme vermittelt und geprüft werden.

Es fehlt aber immer noch ein anerkannter, firmenunabhängiger Qualifizierungsnachweis. Der zunehmende kommerzielle Erfolg und die Verbreitung von Linux erzeugen einen Zertifizierungsbedarf, gefordert werden ähnliche Nachweise, wie sie mit dem *MCP/MCSE* (Microsoft) oder *CNA/CNE* (Novell) seit Jahren Standard sind. Verfügen einige Distributoren wie z.B. *Caldera* und *Red Hat* bereits heute über eigene Trainings- und Zertifizierungsprogramme, so wird es schwierig werden, ein *herstellerneutrales* Zertifikat in einem Open-Source-System wie Linux zu realisieren. Anerkannte Organisationen wie Linux International (*LI*, in Deutschland der Linux Verband *LIVE*) oder die German Unix Users Group (*GUUG*) haben noch keine eigenen Zertifizierungen angestrebt. Man kann jedoch davon ausgehen, dass ab 2000 die Anzahl der angebotenen Linux-Zertifizierungen sprunghaft ansteigen wird.

1.1.3 Linux – ein System für Universitäten, aber nicht für den kommerziellen Einsatz?

Linux eignet sich hervorragend als preiswertes »Lehrmaterial« für die Aus- und Weiterbildung. Zu diesem Zweck existiert eine Vielzahl von Programmen , die in den unterschiedlichen Linux- Distributionen enthalten sind, z. B. Lernsoftware, Dokumentationen oder auch eine beindruckende Menge von Programmiersprachen.

Von dieser riesigen Fülle an Programmen wird nur ein vergleichweise geringer Teil, der zudem seit Jahren im Netzwerkbereich der Unix-Welt Standard ist, kommerziell eingesetzt Die hier verfügbaren Funktionen sind in der Praxis bewährt, gut programmiert und mehr als ausreichend stabil. Im kommerziellen Bereich sollte man sich zunächst auf diesen Umfang beschränken, den ein Linux-System kann »totinstalliert« werden.

Ein weiteres wichtiges Detail liegt in der guten und umfangreichen Systemdokumentation. Weiterhin ist das gesamte Betriebssystem im Sourcecode verfügbar. Damit ergeben sich Möglichkeiten zur eigenen Fehlersuche und -analyse, die mit Systemen unter Novell oder Windows nicht möglich sind oder teuer erkauft werden müssen. Wird Unterstützung benötigt, so ist eine Internetrecherche in der Regel schnell erfolgreich. Die Vielzahl an Informationen, die z. B. in spezialisierten Newsgroups verfügbar ist, kann gerade auch dem Neuling Hilfestellung leisten.

1.1.4 Linux – ein instabiles System?

Linux ist ein echtes 32-/64-Bit-Betriebssystem mit preemptiven Multitasking. Damit müßte es ausreichend stabil sein, um auch kommerzielle Anwendungen zuverlässig ausführen zu können.

Jedes preemptive Multitaskingsystem hat aber eine Schwachstelle. Die Kommunikation der Anwendungen mit der Hardware erfolgt nicht direkt über I/O-Zugriffe, sondern immer nur über das Betriebssystem, über im Kernel eingebettete Gerätetreiber. Das Betriebssystem wertet die von den laufenden Anwendungen gestellten I/O-Anfordungen aus, koordiniert diese und überwacht die Hardwarefunktion. Damit soll erreicht werden, dass langsame oder vielleicht sogar nicht mehr reagierende I/O-Schnittstellen auf keinen Fall zum Stehenbleiben des gesamten Systems führen, sondern vom Betriebssystem »isoliert« werden.

Unter Linux wird dieses über *Special Files* realisiert, die in einem speziellen Verzeichnis zusammengefasst sind und wie eine Datei angesprochen werden. Wird in eine solche Datei hineingeschrieben oder werden Daten gelesen, führt das Betriebssystem den entsprechenden Hardwarezugriff aus.

Ist der im Kernel enthaltene Teil eines Gerätetreibers falsch oder unsauber programmiert, so kann auch ein preemptives Multitaskingsystem »stehenbleiben«. Die Stabilität ist also abhängig von der Qualität der eingesetzten Gerätetreiber.

In diesem Bereich hat Linux in den vergangenen Jahren enorme Fortschritte gemacht. Es gibt heute für fast jede auf dem Markt verfügbare Hardware kommerziell einsetzbare und stabile Linux-Treiber.

Kommt neue Hardware auf den Markt, sind in der Regel sehr schnell Gerätetreiber für Linux verfügbar und werden dann auch konsequent weiterentwickelt. Immer mehr gehen die Hardwarehersteller sogar dazu über, selbst entsprechende Software mit anzubieten.

Linux entwickelt sich schnell, weil bei der Masse an Entwicklern, die weltweit am Linux-Projekt beteiligt sind, die Intervalle bei Neuerscheinungen relativ kurz sind. Aber man ist keinesfalls gezwungen, jeden Schritt mitzumachen. Innerhalb der Kernelentwicklung gibt es zwei verschiedene Hauptlinien, den stabilen und den Entwicklerkernel.

Beim stabilen Systemkern bestehen die eventuellen Änderungen hauptsächlich aus dem Eliminieren von Sicherheitslöchern oder anderen Fehlern. Neue Funktionalitäten, Treiber und grundlegende Veränderungen werden ausschließlich im Entwicklerkernel eingearbeitet. Dieser durchläuft eine langwierige und intensive Testphase (*kernel freeze*), bis er zum nächsten stabilen Kernel wird. Dem Benutzer steht also frei, ob er ein stabiles System verwenden oder die neuesten Funktionen und Treiber ausreizen will.

Linux entwickelt hohe Stabilität und Verfügbarkeit über Hard- und Softwarelösungen, die mit redundanten Strukturen dazu führen, dass der Ausfall einzelner Komponenten nicht sofort zum Systemausfall führt. Im Festplattenbereich sind RAID-Systeme (soft- oder hardwaremäßig aufgebaut) verfügbar, die den Ausfall einer Festplatte tolerieren. Bei Bedarf können aus mehreren gekoppelten Linux-Servern gebildete Hochverfügbarkeitssysteme realsiert werden.

1.1.5 Linux – ein System ohne Systemhaus?

Bei geplanten Änderungen im eigenen EDV-System hat sich es bewährt, ein Systemhaus zu beauftragen. Damit liegt, ein entsprechend gut formuliertes und oft gemeinsam erarbeitetes Pflichtenheft vorausgesetzt, die Verantwortung für die Qualität der Lösung beim Auftragnehmer. Für Unix-, Novell- oder Windows-Netzwerke ist es in der Regel einfach, ein geeignetes Systemhaus zu finden, das die anfallenden Arbeiten mit qualifiziertem Personal erfüllen kann.

Neben diesen größeren Netzwerkarbeiten, z.B. auch Migrationen auf ein neues Betriebsystem, werden immer mehr die laufende Betreuung des Netzwerkes, der Benutzersupport oder alle im DV-Bereich anfallenden Arbeiten an externe Dienstleister gegeben. Dieses *Outsourching* der Netzwerkverwaltung ist mittlerweile weit verbreitet.

Während dies für Unix-, Novell oder Windows-Systeme in der Regel schnell realiserbar ist, fehlen für Linux Systemhäuser ausreichender Größe. Viele kleine Firmen bieten zwar ihre Dienstleistungen für Linux-Systeme an; Qualifikationsnachweise, insbesondere herstellerneutralen Zertifikate fehlen.

Schnell verfügbare, kompetente Ansprechpartner sind selten. Es gibt wenige auf Linux spezialisierte Freiberufler oder Firmen. Telefon-Hotlines für Linux-Systeme sind im Aubau; Tests haben bisher noch keine befriedigenden Ergebnisse erbracht. Getestet wurden z. B. Telefon-Hotlines mit festen Taktzyklen (0190x-Hotlines); wobei die gestellten Probleme nur bedingt gelöst werden konnten. Auffällig war, dass die Anbieter standardmäßig immer nur ein System unterstützen [DuPR99].

Aber auch hier ist Besserung in Sicht: Namhafte Hard- und Softwarehäuser und Linux-Distributoren bieten nach dem schon längere Zeit verfügbaren Installationssupport kostenpflichtigen kommerziellen Support an (z. B. *SuSE: Business Support*). Die Kosten orientieren sich an den üblichen Sätzen, wofür professionelle Leistungen erbracht werden.. Vertraglich können definierte Support-Leistungen vereinbart werden, z. B. auch auf bestimmte Programme oder Netzwerkfunktionen begrenzt.

Es gibt mittlerweile eine ganze Reihe von Firmen, die weitere Linux-bezogene Dienstleistungen wie Installation, Schulungen und die komplette Systemadministration abdecken. Firmen wie *SuSE*, *ID Pro*, *Innominate* und viele mehr bieten neben bundesweiten Schulungsreihen Distributionen mit Business-Support-Lösungen.

1.1.6 Linux – ein System ohne Gewährleistung?

Als Argument gegen den kommerziellen Einsatz von Linux wird oft angeführt, dass in diesem »Open Source«-System jeder Änderungen vornehmen kann, für die letztendlich niemand eine Gewährleistungspflicht erfüllen will und kann. Es ist auch kaum davon auszugehen, dass sich hier in der Zukunft etwas ändern wird. Wer also übernimmt in dem »freien« System Linux die für den kommerziellen Einsatz erforderliche Gewährleistungspflicht? Für die Software an sich ist dies offensichtlich nicht möglich.

Damit bleibt die Gewährleistungspflicht dort, wo sie auch bei Planung, Realisierung und Betrieb von komplexen Unix, Novell- oder Windows-Systemen ist: beim ausführenden Systemhaus.

Linux stellt also hier keine Ausnahme dar. Natürlich besteht bei einem etablierten PC-Netzwerkbetriebssystem bei offensichtlichen Fehlern eine Gewährleistungspflicht des Herstellers. Aber wo wurden mit Erfolg Schadenersatzforderungen gestellt?

Jeder PC-Netzwerk-Spezialist wurde sicherlich schon einmal auf die neue Programmversion, ein fehlerhaftes Update oder auf ein noch nicht verfügbares Ser-

vice-Pack vertröstet. Hier ist schneller und zuverlässiger Support wichtiger als kostspielige und zeitaufwändige Rechtsmechanismen.

Hier also wieder die Forderung nach qualifiziertem Personal, die auch umfangreichere Netzwerkprojekte unter Linux durchführen können. Unbestritten ist, dass Dienstleistungen ihren Preis haben, dies gilt auch für den Linux Support. Hoffentlich kann diese Gratwanderung zwischen offenen System und kommerziellen Einsatz weiter erfolgreich gegangen werden.

1.2 Netzwerkressourcen

Moderne PC-Netzwerke werden mit dem Netzwerkprotokoll *TCP/IP* betrieben, um Internet- oder Intranet-Dienste in vollem Umfang nutzen zu können. Dieses gilt inzwischen auch für die aktuellen PC-Netzwerke unter Novell und Windows: Insbesondere Novell hat sich nach Versuchen, das eigene Protokoll *IPX/SPX* zum »weltweiten« Standard zu erheben, auch dieser mit hoher Dynamik vom Internet geprägten Entwicklung gebeugt.

Neben TCP/IP noch weitere Protokolle im Netzwerk zu betreiben, vermindert in jedem Fall den Datendurchsatz und die Reaktionszeiten erheblich. Grundsätzlich sollte dabei eine Beschränkung auf ein Protokoll erfolgen; wenn notwendig können andere Protokoll-Welten über Gateways integriert werden.

Welche Planungsgrundsätze sind für eine typische Netzwerkumgebung zu berücksichtigen? Wie sieht eine entsprechende Server-Grundkonfiguration aus?

Wichtig ist natürlich die geforderte Leistung des Netzwerks und wie es nach der Fertigstellung betrieben werden soll:

- Welche Ressourcen werden wo und in welcher Menge benötigt? Wie lang sind die maximalen Reaktionszeiten; sind (langsame) Weitverkehransbindungen erforderlich?
- Wie soll das Netzwerk technisch realisiert werden? Welche aktiven und passiven Netzwerkkomponenten können im *LAN/WAN* verwendet werden?
- Welche Absicherungsmaßnahmen sind erforderlich zum Schutz von Datenbeständen? Welche gesetzlichen Auflagen sind zu erfüllen, z.B. im Bereich Datenschutz oder bei Aufbewahrungsfristen?
- Ist ein besonderer Zugangschutz durch technische Absicherungen zu realisieren? Müssen Codekarten, andere Personenidentifikationssysteme oder Schlüsselverfahren eingesetzt werden?
- Was sind die Verfügbarkeitsanforderungen im Bezug auf alle Netzwerkkomponenten? Was »kostet« ein Systemausfall, wie lange darf die Wiederherstellung der Systemfunktionen nach typischen Fehlern oder nach einem Datenverlust dauern?

- Welche Maßnahmen zur Sicherung und Wiederherstellung der Datenbestände müssen grundsätzlich getroffen werden? Ist eine Archivierung der Datenbestände notwendig?
- Welche Betriebssysteme sollen am Arbeitsplatz eingesetzt werden? Welche Anwendungen müssen installiert und gepflegt werden? Wenn eine Revisonskontrolle aufgebaut werden soll, wie soll diese technisch realisiert werden?
- Welche Ressourcen stehen für die Administration im laufenden Netzwerkbetrieb zur Verfügung? Wie sehen die erforderlichen und tatsächlichen Qualifizierungen der hier eingesetzten Personen aus? Muss die Netzwerkplanung entsprechend angepasst werden und welche Weiterbildungsmaßnahmen sind mindestens erforderlich?

Dazu sollen zunächst die typischen Netzwerkdienste im direkten Vergleich zu den PC-Netzwerkbetriebssystemen Novell Netware und Windows NT (Windows 2000) vorgestellt werden. Überlegungen zur Absicherung von Netzwerkressourcen, zur Planung und Realisierung der Multiuserumgebung spielen dabei ebenso eine Rolle wie die Darstellung der auf dem Markt verfügbaren Lösungen im Bereich von Hard- und Software.

1.2.1 Basis-Netzwerkfunktionen

Der Server soll seine Ressourcen im Netzwerk zur Verfügung stellen. Die Herstellung der Standard-Netzwerkfunktionen ist damit Voraussetzung für alle weiteren Konfigurationsschritte. Aus den schon dargestellten Gründen soll hier nur auf dem Netzwerkbetrieb unter TCP/IP eingegangen werden, obwohl gerade unter Linux auch andere Protokolle möglich wären.

Netzwerkprotokoll: TCP/IP

Die Anfänge von TCP/IP gehen auf ein 1969 von der amerikanischen *Defense Advanced Research Projects Agency* (ARPA) finanziertes Forschungsprojekt zurück. Was damals unter dem Namen *ARPANET* als experimentelles Netzwerk begann, wurde 1975 in den normalen Betrieb übernommen. 1983 wurde das Protokoll als Standard TCP/IP zur Netzwerkkommunikation definiert.

Während das *ARPANET* langsam zum Internet heranwuchs, hatte sich TCP/IP schon sehr weit auf Netzwerken außerhalb des Internets verbreitet, insbesondere auf lokalen UNIX-Netzwerken.

TCP/IP ist ein vollständig transparentes Netzwerkprotokoll. Anwendungen können über das Netzwerk »getrennt« werden, d.h. im Rahmen von Client-Server-Strukturen können die Standard-Netzwerkfunktionen eines Servers von einem Client über das Netzwerk ferngesteuert werden.

TCP/IP verwendet eine starre, für jeden Host im Netzwerk eindeutige, Adresszuweisung (fast) ohne jegliche Dynamik. Die typische Schreibweise solcher IP-Adressen ist

```
a.b.c.d
```

wobei die dezimalen Angaben für *a, b, c* und *d* jeweils 8-Bit-Zahlenwerte darstellen, also einen zugelassenen Wertebereich von 0 bis 255 haben. Die IP-Adresse hat eine Gesamtlänge von 32 Bit. Für das Internet erfolgen die Zuweisungen der IP-Adressräume über das *Network Information Center* NIC, im Bereich der *privaten* Firmennetzwerke, die zunächst noch keine direkte Anbindung an das Internet haben, können Adressen aus einem dafür reservierten Adressbereich verwendet werden, die typischerweise mit *a = 192* beginnen.

Die weiteren, tatsächlich sehr umfangreichen Grundsätze für die Planung und Realisierung von IP-Adressräumen würden den Umfang dieses Buches sprengen. In der Literatur ist dazu eine Vielzahl von Grundlagenwerken verfügbar, z. B. [Kirh96], [WaEv97].

Ein Hinweis aus der Praxis: Die Planung von größeren Netzwerken unter TCP/IP sollte nur durch qualifiziertes Personal erfolgen, da die möglichen Fehlerquellen sehr vielseitig und komplex sind. Bereits kleine Planungsfehler können dazu führen, dass Netzwerke unter TCP/IP nicht richtig funktionieren, extrem langsam sind, Daten verloren gehen und die gesamte Planung dann neu durchgeführt werden muss.

Namensauflösung

Jeder Host verfügt über seine eindeutige 32-Bit umfassende IP-Adresse und einen Hostnamen. Die Auflösung von Hostnamen (*Hostname Resolution*) ordnet einem Hostnamen die IP-Adresse zu, sie gehört zu den Standarddiensten von TCP/IP. Ist diese Namensauflösung nicht möglich oder nicht richtig konfiguriert, kommt es auch im lokalen Netzwerk zu sehr langen Antwortzeiten durch *Time-out*. Die Namensauflösung ist also ein sehr wichtiges Konfigurationsdetail jedes TCP/IP-Netzwerkes.

In kleinen Netzwerken kann die einfache Zuordnung *Hostname-IP-Adresse* ausreichend sein. Die dafür auf jedem Rechner erforderliche Tabelle, überlicherweise die Datei *hosts* (bei Linux-Systemen im Verzeichnis */etc/*), enthält die notwendigen Zuordnungen. Wird ein Host hinzugefügt oder die IP-Adresse geändert, müssen die Tabellen aller zum Netzwerk gehörenden Rechner entsprechend korrigiert werden. Mit zunehmender Neztwerkgröße wird dies zunehmend mühsam und fehleranfällig.

Auch im Internet wurden zu Beginn alle Adressinformationen in einer einzigen Datenbank *HOSTS.TXT* gespeichert. Diese Datei wurde von NIC verwaltet und musste von allen angeschlossenen Rechnern geladen werden. 1984 wurde ein

neues Schema für die Auflösung von Adressnamen eingeführt, das *Domain Name System*. DNS organisiert Hostnamen in einer Hierarchie von Domänen (*Domain*: diese Domäne haben nichts zu tun mit den Windows-NT-Domänen). Eine Domain ist eine Sammlung von *Sites* oder *Hosts*, die organisatorisch zusammenhängen, z. B. alle Hosts in einem Netzwerk oder alle Netzwerke einer Firma.

Jeder Host erhält einen eindeutigen, voll qualifizierten Domänennamen (*full qualified domanin name*, FQDN), der ihn eindeutig identifiziert. Mit

```
linux01.samulat.de
```

wird der Host *linux01* in der *Top-Level-Domain* Deutschland (*de:* zweistelliger Ländercode nach ISO 3166) und dem Domain-Namen *samulat* eindeutig identifiziert. Diese Namen beginnen, von links nach rechts gelesen, immer mit einer *Top-Level-Domain*, danach folgt (zumindest in Deutschland) in der Regel ein etwas längerer Name, der dann direkt auf die tatsächliche Firma oder Organisation hinweist.

Die Organisation von Namen in der Hierarchie von Domänen löst das Problem der Eindeutigkeit von Hostnames sehr elegant. Bei DNS muss der Hostname nur innerhalb der Domäne eindeutig sein, die voll qualifizierten Namen sind einfach zu merken.

Domainname:	samulat.de
Domaininhaber:	IngBuero Samulat Dorfstr. 67 D-25774 Hemme Germany
Administrativer Ansprechpartner:	PS5942-RIPE
Technischer Ansprechpartner:	MH375-RIPE
Zonenverwalter:	MH375-RIPE
Nameserver:	ns5.hetzner.de
Nameserver:	ns.hetzner.net
Status:	konnektiert
Letzte Aktualisierung:	Mittwoch, 6. Oktober 1999
Stand Datenbank:	Samstag, 13. November 1999

Personendaten

Name:	Peter Samulat
Kontakttyp:	Person
Adresse:	IngBuero Samulat Dorfstr. 67 D-25774 Hemme Germany
Telefax:	+49 4837 9121
E-Mail:	samulat@samulat.de
NIC-Handle:	PS5942-RIPE

Abbildung 1-1 Informationen zur Domäne samulat.de

Die Zuweisung eines Internet-Namens erfolgt über das nationale NIC. Um zu überprüfen, ob der gewünschte Name noch verfügbar ist, stellen viele Provider

und auch das deutsche *DE-NIC* (www.de-nic.de) dafür Suchmaschinen zur Verfügung. Das nachfolgende Beispiel (Abbildung 1-1) zeigt das Ergebnis einer über *DE-NIC* durchgeführten Suche für die Domäne *samulat.de* (Auszug).

In den in der Regel überschaubaren Strukturen eines lokalen Netzwerks kann die Namensauflösung über die bereits vorgestellten Dateien *hosts* erfolgen, auch wenn eine Internetanbindung vorhanden oder geplant ist. In großen Netzwerken muss ein serverbasiertes System zur Namensauflösung eingerichtet werden; aus Gründen der Fehlertoleranz sind hier mindestens zwei DNS-Server erforderlich. Die Konfiguration dieser Server, die Wartung der Informationen während des Betriebes und vor allem die Anbindung des eigenen Netzwerkes an das Internet sind aufwändig und fehlerträchtig. Diese Verfahren sind in der grundlegenden TCP/IP-Literatur ausreichend beschrieben; Details sollen daher an dieser Stelle nicht wiederholt werden. In den hier vorgestellten Lösungen wird die Namensauflösung über die Dateien *hosts* realisiert.

DHCP-Server

Die IP-Adresszuweisung ist in der Regel statisch; jedem Host wird eine eindeutige Adresse zugewiesen. Das bedeutet, dass die Programmierung von IP-Adresse, Subnet-Mask und weitere optionalen Angaben manuell an jedem Host erfolgen muss. Auch diese Arbeit wird mit zunehmender Größe des Netzwerkes fehleranfällig, insbesondere dann, wenn Rechner öfter das Teilnetz wechseln oder neue Rechner integriert werden sollen.

Das *Dynamic Host Configuration Protocol* DHCP ermöglicht es, die vollständige Zuweisung der IP-Adressinformationen zu zentralisieren und aus einer Datenbank heraus durchzuführen. Dazu muss ein DHCP-Server eingerichtet werden, in dem für jeden Netzwerkstrang Adressräume und weitere Konfigurationsdetails definiert werden.

Wird jetzt ein als DHCP Client konfigurierter Rechner gestartet, erhält er Adressdaten vom DHCP-Server; als gemeinsame Referenz dient die weltweit eindeutige, 12-stellige Seriennummer (*MAC Number*) der Client-PC-Netzwerkkarte. Nahezu alle PC- Betriebssysteme können unter TCP/IP als DHCP-Client konfiguriert werden; die am Client durchzuführenden Arbeiten im Rahmen der Einrichtung des Protokolls werden damit erheblich verinfacht.

Dieses optimale Verfahren zur zentralen Verwaltung und Zuweisung von IP-Adressdaten sollte nicht dazu führen, dass demselben Rechner immer wieder unterschiedliche freie IP-Adressen zugewiesen werden. Bei der Konfiguration sollte die Bindung *MAC-Adresse-IP-Adresse* auch vom DHCP Server so weit wie möglich berücksichtigt werden, um auch die Namensauflösung weiterhin ohne Einschränkungen zu ermöglichen.

Ein DHCP-Server sollte mindestens Daten für

- IP-Adresse,
- Subnet-Mask,
- Gateway(s),
- Hostname,
- Domänen-Name,
- WINS-Server (bei Windows-Clients) und
- DNS Server (bei Namensauflösung über DNS)

zentral verwalten und zuweisen.

Werden Windows Clients eingesetzt, sollten in der Windows IP Konfiguration neben der Auswahl *Adresszuweisung über DHCP* keine weitere Einträge vorgenommen werden, da lokale Definitionen die über DHCP zugewiesenen Werte überschreiben.

Zeitsynchronisierung

Alle Rechner im Netzwerk, Server oder Client, arbeiten mit der gleichen Uhrzeit. Ist dies nicht der Fall, können Bearbeitungsstände von kopierten Dateien nicht mehr nachvollzogen werden; im Netzwerk gemeinsam genutzte Dienste arbeiten nicht oder nur noch fehlerhaft.

Clients in Novell-Netzwerken (z.B. die *DOS-Requester*) synchronisieren die Systemuhr der Arbeitsstation bei jeder Netzwerkanmeldung mit dem Server. In Windows- NT-Netzwerken scheint dies nicht so wichtig gewesen zu sein, hier muss der Systemverwalter ein eigenes Startskript schreiben, das bei jeder Anmeldung abgearbeitet wird und dann die notwendige Zeitsynchronisierung durchführt.

Jedes Verfahren zur Zeitsynchronisierung setzt voraus, dass im Netzwerk eine Zeitreferenz zur Verfügung steht, mit der alle Rechner ihre Systemuhren synchronisieren können. Zweckmäßigerweise sollte dies nicht nur Zusatzaufgabe für einen ständig im Netzwerk verfügbaren Server sein, dieser Server sollte auch in der Lage sein, sich selbst mit einer externen Referenz zu synchronisieren.

Auf dem Markt sind eine Vielzahl von Funkuhren (*DCF-Systeme, GPS*) verfügbar, die direkt am Server angeschlossen werden und die Systemuhr auf die amtliche Uhrzeit stellen. Möglich ist auch die Nutzung ähnlich guter Zeitquellen, wie z.B. der ISDN-Zeitstempel oder auch der Zugriff auf externe Zeitserver über das Internet. Die dazu notwendigen Verfahren werden an späterer Stelle ausführlich beschreiben.

Server-Fernwartung

Bereits durch die Grundfunktionen von TCP/IP stehen eine Reihe von Werkzeugen zur Verfügung, mit denen die Server-Fernwartung z.B. über das Terminalpro-

gramm *telnet* über das Netzwerk durchgeführt werden kann. Novell-Systemverwalter kennen dieses Prinzip, wenn auch mit Einschränkungen, durch die *Remote Console*.

Grafisch-orientierte Bedieneroberflächen, wie z.B. bei Windows-NT-Servern, ermöglichen standardmäßig keine Fernbedienung. Man ist darauf angewiesen, am Server selbst zu arbeiten, oder mit Fremdprodukten wie *PC-Anywhere* oder *Carbon Copy* mit einem Programm, das die Grafikdarstellung des Servers an einen Client-Arbeitsplatz transportiert und bedienbar macht. Unter Windows NT kann man dazu auch das Backoffice-Programm *SMS-Server* einsetzen.

Das im TCP/IP-Protokoll enthaltene Client-Server-Prinzip sollte es grundsätzlich ermöglichen, das Bild eines *Grafikservers* auf einen als *Grafikclient* eingerichteten Arbeitsplatz-PC zu transportieren. Dazu sollte im Prinzip nur eine geeignete Grafik-Client-Software notwendig sein, die unter dem Betriebssystem des Arbeitsplatzes läuft. Linux bietet solche Möglichkeiten: Die dazu notwendige Software ist im Standardumfang der meisten Distributionen enthalten. Zusätzliche Kosten für den Kauf von Fremdprodukten entfallen.

1.2.2 Datei- und Verzeichnisdienste

Eine Standardaufgabe fast jedes Netzwerkservers ist die Bereitstellung von Dateien und Datenbeständen zur gemeinsamen Bearbeitung. Die Aufgaben eines solchen *Fileservers* wurden in den letzten Jahren sehr häufig von Novell-Servern der Version 3.12 abgedeckt, die, sehr stabil und einfach in der Bedienung, weit verbreitet waren. Natürlich können auch Windows-NT- oder Linux-Server die Aufgaben eines Fileservers wahrnehmen und Dateien im Netzwerk bereitstellen. Mit Windows-Clients wird in der Regel dann über *logische Laufwerke* auf dieses Datenbestände zugegriffen.

In modernen Netzwerken haben die Dateigrößen und die Menge der über das Netzwerk zu transportierenden Datenmengen stark zugenommen. Festplattenkapazitäten im Bereich um 10 Gigabyte stellen keine Seltenheit dar, Druckaufträge haben schnell Größenordnungen von mehreren Megabytes erreicht. Damit werden die Anforderungen im Bereich der Geschwindigkeit des Festplattensubsystems und der Netzwerke immer höher. Der Server benötigt eine möglichst schnelle Festplatte und eine leistungsfähige Netzwerkanbindung, typischerweise mit 100 MBit/s.

Trotz der sehr großen Festplattenkapazitäten besteht für jeden Fileserver immer die Gefahr, dass ein Benutzer unbefugt zu große Datenmenge speichert und damit die zur Verfügung stehenden Kapaziät aufbraucht. Bietet Novell schon seit Jahren Verfahren an, um den Speicherplatz bezogen auf den Benutzer oder auf einzelne Verzeichnisse zu limitieren (*Quoting*), kann Windows NT das immer noch nicht. Erst Windows 2000 soll endlich über dieses unbedingt notwendige Feature verfügen! Auf Linux-Systemen kann ein *Quoting* eingerichtet werden, die Administration und Kontrolle ist vergleichsweise einfach.

Bei der Installation der Netzwerk-Clients können Fileserver eine wichtige Rolle spielen: Ein im Netzwerk verfügbarer Installationspunkt stellt die Setup-Software für Betriebssysteme und Anwendungen bereit oder ermöglicht das Laden von Festplattenimages, die dann vollständige Client-Installationen sind. Für den Netzwerk-Client müssen hierzu nur Bootdisketten vorbereitet werden, über die dann bei Bedarf eine Verbiundung zum Installationspunkt hergestellt wird.

Einen Sonderfall des Fileservers stellt der CD-ROM-Server dar. Die in einem oder mehreren CD-ROM- (oder DVD-) Laufwerken eingelegten Datenträger werden im Netzwerk bereitgestellt. Novell und Windows NT erfordern vom Client für jedes Laufwerk einen eigenen Laufwerkbuchstaben; der ohnehin knappe Buchstabenvorrat ist dann schnell aufgebraucht. Wesentlich zweckmäßiger ist es, alle CD-ROM-Laufwerke eines Servers in einem gemeinsamen Verzeichnis zu *mounten* und dann dieses Verzeichnis über einen Laufwerkbuchstaben zu erreichen. Erhalten die dort verfügbaren Unterverzeichnisse dann noch Namen, über die die Datenträger eindeutig identifizierbar sind, ist ein sehr einfacher Umgang mit diesen Ressourcen möglich. Voraussetzung dafür ist, dass das Server-Betriebssystem es erlaubt, CD-ROM-Laufwerke direkt in die Verzeichnisstruktur des Servers zu *mounten*. Dies ist unter UNIX, LINUX und Windows 2000 möglich – unter Novell und Windows NT nicht.

Werden auf dem Fileserver Datenbestände gespeichert, die von Windows-Clients bearbeitet werden, ist natürlich auch die Absicherung gegen *Viren* wichtig. Das Betriebssystem Linux ist zwar selber nicht durch Viren gefährdet, jedoch die dort gespeicherten Dateien.

1.2.3 Druckdienste

Eine weitere Netzwerk-Standardanforderung ist die gemeinsame Nutzung von Druckgeräten. Dabei geht es weniger darum, die im Netzwerk verfügbaren Drucker räumlich zu zentralisieren, vielmehr sollen heterogene Druckumgebungen, in denen die zur Verfügung stehenden Drucker teilweise direkt am (Linux-) Server, an Printserver-Boxen oder auch unmittelbar den Arbeitsplätzen angeschlossen sind (Abbildung 1-2) möglichst für alle Netzwerkbenutzer bereitsgestellt werden.

Direkt am PC-Arbeitsplatz angeschlossene Drucker können dabei erheblich zur Reduzierung der Netzwerklast durch Druckaufträge beitragen, da die teilweise sehr großen Datenmengen direkt ausgegeben werden können. Nach Abbildung 1-2 können dabei Drucker am Windows 9x/NT-Client ebenso lokal betrieben werden wie z. B. am Linux Client. Innerhalb des Protokolls TCP/IP ist es bereits beiden Clients auf sehr einfachem Wege möglich, den jeweils anderen Drucker zu verwenden.

Die gemeinsame Nutzung eines Druckers ist insbesondere dann sinnvoll, wenn es sich um teure Geräte handelt wie z. B. Hochleistungsdrucker oder A3-Farbplotter, die nicht in größerer Stückzahl im Netzwerk bereitgestellt werden können. Damit

diese Geräte zu jedem Zeitpunkt erreichbar sind, sollten sie nicht an einem PC-Arbeitsplatz angeschlossen werden, sondern über eine ständig erreichbare Netzwerkkomponente, wie einen Server oder eine Printerserver-Box.

Durch die Zentralisierung der Druckdienste können alle im Netzwerk verfügbaren Druckgeräte bei Bedarf über eine einzigen Server erreicht werden, z. B. über einen Linux-Server.

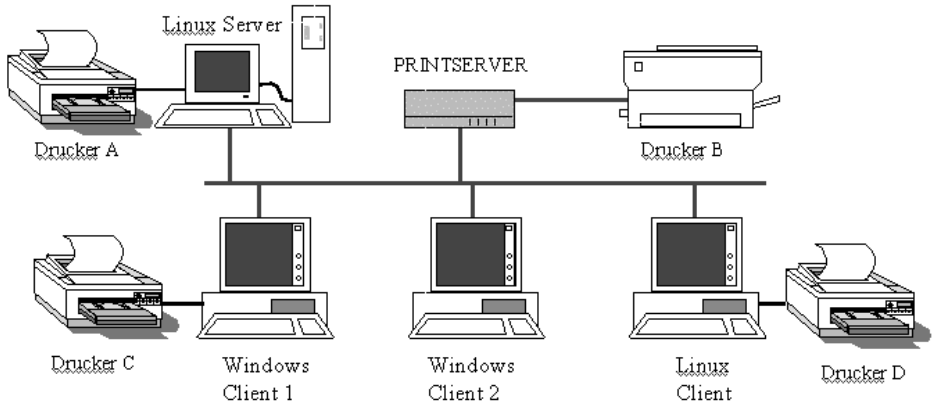


Abbildung 1-2 Heterogene Druckumgebung im Netzwerk

Die aktuelle Forderung der flexiblen Nutzung von Netzwerkdruckern, die entweder an Servern, an Clients oder an Printserverboxen angeschlossen sind, erfordert aufwändige Konfigurationsarbeiten an dem diese Dienste zentral im Netzwerk bereitsstellenden Server. Die Druckdaten müssen bis zum tatsächlichen Ausdruck auf Datenträger gesichert werden und es sollten Vorkehrungen getroffen werden, um fehlerhafte oder unvollständig erfolgte Ausdrucke bei Bedarf wiederholen zu können.

In räumlich verteilten Netzwerken (WAN), in denen über Weitverkehrsverbindungen entfernte Standorte miteinander verbunden sind, kann es sinnvoll sein, Arbeitsergebnisse auch direkt auf einem Druckgerät des anderen Standortes ausgeben zu können. Eine technische Zeichnung kann so beispielsweise direkt vor Ort auf dem dort verfügbaren Hochleistungsplotter ausgegeben werden; bei Ausfall eines Druckgerätes am eigenen Standort kann auf ein vergleichbares Gerät im Netzwerk zurückgegriffen werden.

1.2.4 Programmdienste

Im Gegensatz zu den als *Fileserver* optimierten Novell-Servern (zentrale Bereitstellung von Datei- und Druckdiensten) stellen Unix- und Windows-NT-Netzwerkbetriebssysteme typische *Application-Server* dar. Diese können im Netzwerk *Pro-*

grammdienste zur Verfügung stellen und sind über eine zum System gehörende Möglichkeit zur Shell-Programmierung in der Lage, ereignis- oder zeitgesteuerte Aktionen zentral zur Verfügung zu stellen. Der *Application-Server* ist die ideale Plattform für moderne Netzwerke, viele sonst nur über aufwändige Zusatzsoftware realisierbare Netzwerkfunktionen stehen direkt zur Verfügung.

Interessant ist z. B. der Aufwand, den eine Sicherung der auf dem Server gespeicherten Datenbestände zu festgelegten Zeitpunkten benötigt. Der *Application Server* stellt bereits Programme zur Verfügung, mit denen vorher ausgewählte Daten auf ein Bandlaufwerk gesichert werden können. Mit einem zweiten Programm wird dann erreicht, dass dieses Kommando zu den geplanten Zeitpunkten automatisiert immer wieder ausgeführt wird. So wird die geforderte Steuerung der Sicherung ohne Zusatzsoftware erreicht; der Systemverwalter erstellt die dafür notwendigen Steuerskripte auf dem Server. Der Fileserver verfügt vielleicht auch über ein eigenes Programm zur Erstellung eines solchen Sicherung, spätestens aber bei der freien Programmierbarkeit über Skripte ist man dann auf Zusatzsoftware angewiesen.

Je mehr Dienste erforderlich sind, die direkt auf Kommunikationsanforderungen aus dem Netzwerk reagieren und dann die angeforderten Daten zur Verfügung stellen, um so mehr wird es notwendig, eine Programmstruktur auf dem Server einzurichten, die diese komplexen Abläufe handhaben kann. Die Funktionen eines Web- Servers, der einem Internet-Browser auf Anforderung seine Seiten zur Anzeige übergibt, ist dabei heute auf allen Netzwerkbetriebssystemen Standard. *Application Server* realisieren vollständige Intranet- oder Internetstrukturen, sie zentraliseren Faxdienste und werden immer mehr zu einem echten Kommunikationsserver, der sogar Aufgaben als Telefonanlage wahrnimmt.

1.2.5 Datenbanksystem

Obwohl man in den siebziger Jahren noch keine Hochleistungs-PC-Arbeitsplätze zur Verfügung hatte, war es trotzdem möglich, auch sehr große Datenbestände zu handhaben, zu verdichten und gezielt auszuwerten. Die eigentliche Datenbank wurde nicht auf dem Netzwerk-Arbeitsplatz geführt. Dies war auch aus technischen Gründen nicht möglich, denn die zur Verfügung stehende Rechenleistung wurde ausschließlich vom Zentralrechner, dem *Mainframe*, erbracht. Der Mainframe speicherte alle Daten und führte ein Datenbankprogramm aus. Die zur Steuerung und Anzeige der Ergebnisse verwendeten Arbeitsplätze war oft nur einfache »dumme« Terminals.

Mit der zunehmenden Leistungsfähigkeit der PC-Arbeitsplätze wurde es dann möglich, die Datenbankprogramme direkt auf dem Client auszuführen; nur noch die Datenbestände wurden zentral abgelegt. Damit entstand die Situation, dass teilweise erhebliche Datenmengen, die z. B. im Rahmen einer Abfrage vom lokalen Datenbankprogramm des Clients geladen wurden, wiederholt über das Netzwerk

transportiert werden mussten. Auch bei relativ kleinen Datenbanken entstehen so erhebliche Antwortzeiten. Ein typischer Vertreter dieser Art von Datenbankstruktur ist das heute vielfach verwendete Microsoft Access.

Heute ist die Zentralisierung von Datenbeständen und Datenbankprogramm wieder eine typische Serverdienstleistung. Datenbankprogramme wie z.B. *Informix*, *db/2*, *Adabas* oder *Oracle* basieren auf dem Industriestandard *Standard Query Language* SQL. Sie stellen in Tabellen organisierte *relationale Datenbanksysteme* zur Verfügung, die über die in SQL definierten Befehle gesteuert werden:

- Alle Datenbestände sind in Form von Tabellen auf dem Datenbankserver gespeichert. Wird eine Abfrage ausgeführt, so werden alle dazu notwendigen Arbeitsschritte ohne Netzwerkbelastung auf dem Server geleistet. Nur der SQL-Code zur Steuerung der Abfrage und die Tabelle mit dem Abfrageergebnis wird über das Netzwerk transportiert.
- Der Datenbankserver kann über in SQL programmierte Trigger mit einem Programm auf das Einfügen, Ändern und Löschen von Datensätzen oder auf das Ändern einzelner Feldinhalte reagieren. SQL ermöglicht mit sehr leistungsfähigen Kontrollstrukturen auch die Programmierung sehr komplexer Routinen.
- Die auf dem Server geführten Datenbestände können sehr detailliert abgesichert werden. Über eigene, zusätzlich zur vorhandenen Serverabsicherung geführte, Benutzerverwaltungen kann die Zugriffsberechtigung bis auf das einzelne Tabellenfeld festgelegt werden.
- Auf die servergespeicherten Datenbestände kann über standardisierte Schnittstellen zugegegriffen werden. Mit der von Microsoft-Windows-Systemen bekannten Datenbankschnittstelle *Open Database Connector* ODBC kann Access als Datenbank-Clientprogramm direkt auf Tabellen des Datenbankservers zugreifen. Mit *Visual Basic* können auf die tatsächlichen Bedienungsabläufe optimierte *Datenbankfrontends* erstellt werden, die dann auch die Bearbeitung von Datenbeständen erlauben, ohne dass dafür je PC-Arbeitsplatz eine Access-Lizenz notwendig ist. *Java*-Applikationen werden mit JDBC zum vollwertigen Datenbank-Client. Fast alle Datenbanksysteme bieten weitere Schnittstellen zum direkten Im- und Export von Daten in Web-Präsentationen, z.B. als Basis für Electronic Commerce.

Die Verfügbarkeit und vor allem die Kosten leistungsfähiger Datenbankserver bezogen auf ein Netzwerkbetriebssystem stellen damit ein wesentliches Entscheidungskriterium für die Auswahl des passenden Systems dar. Steht für Novell- und Windows NT-Systeme schon seit Jahren eine große Auswahl zur Verfügung, so haben fast alle namhaften Datenbankanbieter ihre Programme innerhalb der letzten Monate auch auf Linux-Plattformen portiert und bieten vergleichsweise preiswerte Lizenzen auch für die kommerzielle Nutzung an. Für den privaten Anwender sind viele Datenbanken unter Linux sogar kostenlos erhältlich und im Standardumfang vieler Distributionen bereits enthalten.

1.2.6 Mainframe-Zugang

Sind im eigenen Netzwerk Großrechner (Mainframes) eingesetzt, so werden Gateway-Funktionen zum Zugriff auf die dort gespeicherte Datenbestände benötigt. Da Mainframe-Applikationen sehr häufig reine Datenbankanwendungen darstellen, werden dafür speziell programmierte Oberflächen benötigt, die in der Regel als in den Funktionen erweitertes Terminal ausgeführt sind. Diese oft noch als DOS-Anwendung programmierten Softwarepakete nutzen TCP/IP, DLC oder andere herstellerspezifische Protokolle.

Die Gateway-Funktionen zum Zugriff auf Großrechner müssen in der Lage sein, nur dafür benötigte Protokolle in das standardmäßig verwendete TCP/IP »umzusetzen«, damit Netzwerk-Clients, ausgestattet mit der passenden Terminalsoftware, direkt zugreifen können.

An den Netzwerkarbeitsplätzen können Terminalprogramme eingesetzt werden, die meisten hierzu geeigneten Programme stellen Emulationen der Standards *ANSI* oder *VT100* bereit. Nur in seltenen Fällen ist es notwendig, weitere Anpassungen an den verwendeten Steuercodes vorzunehmen. In einem solchen Fall ist dann allerdings eine möglichst freie Konfigurationsmöglichkeit notwendig, z. B. über Steuerdateien.

1.2.7 Datensicherung

Jeder Server muss technische Möglichkeiten zur Gesamtsicherung der eigenen Datenbestände erhalten. Bei Bedarf sollte es möglich sein, auch ausgewählte Datenbestände anderer Server oder auch von Netzwerk-Clients mitzusichern. Die tatsächlich an den Servern und an den Clients eingesetzten Betriebssysteme sollten dabei keine Rolle spielen, alle Sicherungsabläufe sollten über eine grafisch orientierte Oberfläche administriert werden können. Idealerweise können Web-Oberflächen im Netzwerk eingesetzt werden, die die Steuerung aller Funktionen von jedem Arbeitsplatz aus ermöglichen.

Serversicherungen ermöglichen nach einem Funktionsausfall die schnelle Wiederherstellung der Netzwerkfunktionen. Serverdatenbestände sind, stabile Hardware und ein ausreichend fehlertolerantes Festplattensubsystem vorausgesetzt, im Wesentlichen durch drei Gefahren bedroht:

- Diebstahl der Serverkomponenten
- Serververlust durch Brand oder andere externe Einflüsse
- Viren

Die regelmäßige Sicherung der Datenbestände sollte unter Berücksichtigung dieser Gefahrenquellen geplant und durchgeführt werden. Zur Sicherung wird immer ein Satz von mehreren Bändern verwendet, die getrennt vom Server in einem

getrennten Brandabschnitt aufzubewahren sind. Sehr kritisch sind in diesem Zusammenhang Bandlaufwerke zu sehen, die mehr als ein Band aufnehmen können. Diese Geräte können in einem »Karussel« alle Bänder aufnehmen, die während einer Woche nacheinander zur Sicherung benötigt werden. Bei Diebstahl verschwinden mit ihm alle Bänder, weswegen Sicherungsbänder täglich gewechselt und an einen sicheren Ort gebracht werden müssen!

Die zur Sicherung eingesetzte Hardware, heute vor allem die digitalen Bandlaufwerke (DAT) mit Kapazitäten im Bereich um 10 GByte, kann alle auf dem Server gespeicherten Datenbestände so speichern, dass mit einer zu diesem System passenden Bootdiskette die gesamte Wiederherstellung des Systems in einem Arbeitsgang möglich ist. Für die Sicherung der Servergrundkonfiguration kann es zusätzlich sinnvoll sein, das Festplattenimage mit Programmen wie z.B. *Drive Image* (Hersteller: *PowerQuest*) auf CD-ROM oder einem anderen ausreichend großen Medium dauerhaft zu sichern. Empfehlenswert ist dies auch für fertig konfigurierte PC-Netzwerkarbeitsplätze, die dann völlig unabhängig vom eingesetzten Betriebssystem nach einem Ausfall schnell wiederhergestellt werden können.

Für die Sicherung der Datenbestände können im gewissen Umfang auch einmal oder mehrfach beschreibbare CD-ROMs oder die wegen der höheren Speicherkapazität wesentlich besser geeigneten DVD-Medien genutzt werden. Der Preisverfall wird diese Hardware in Zukunft noch stärker in Bereiche der Datensicherung vordringen lassen.

Die zur Durchführung der Bandsicherung eingesetzte Software sollte so programmiert werden können, dass der im Umfang frei zu definierende Sicherungslauf zu vorgeplanten Zeitpunkten automatisch ausgeführt wird. Gesichert werden müssen dabei nicht nur Dateien und Verzeichnisse, sondern es sollte auch möglich sein, laufende Dienste wie z.B. einen Datenbankserver vollständig zu sichern, ohne dafür den Dienst temporär beenden zu müssen. Gerade zur Erfüllung der letzten Forderung wird es in der Regel notwendig sein, spezielle Sicherungssoftware zu beschaffen.

1.2.8 Netzwerkverwaltung

Bereits in kleinen Netzwerken können automatisierte Werkzeuge zur Netzwerkverwaltung sehr wichtig werden. Dabei sind die Leistungsanforderungen je nach Netzwerktyp, physikalischer Ausdehnung, Art der verwendeten Komponenten, Größe und zu erwartenden netzwerktypischen Problemem sehr unterschiedlich:

- Prüfung der technischen Netzwerkfunktionen, z.B. die Kontrolle der aktuellen Auslastungen, nutzbarer Bandbreiten und das Auffinden von Unterbrechungen und anderen temporären Engpässen im Netz
- Netzwerkstatistiken; Protokollierung und statistische Auswertung der wichtigsten Netzwerkparameter über die gesamte Nutzungszeit

- Protokollanalyse: Aufzeichnung von allen oder ausgewählten Datenpaketen mit der Möglichkeit, die erfassten Datenmengen über Filter weiter zu konzentrieren. Bei Bedarf sollten die zu protokollierenden Kommunikationsbeziehungen gezielt ausgewählt werden können. Wichtig ist auch die Möglichkeit zur Analyse der Dateninhalte im Klartext.
- Automatische Inventarisierung: Erfassung der im Netzwerk verwendeten Hard- und Software. Diese Funktion sollte möglichst unabhängig von den eingesetzten Betriebssystemen sein und auch einen vollständigen, bei Bedarf sogar dynamisch geführten Lizenznachweis ermöglichen.
- Grafische Darstellung der gesamten Netzwerkorganisation mit ständig aktualisierter Anzeige wichtiger Netzwerkparameter: Einbezogen werden alle Server und Netzwerk-Clients; die um statistische Zusatzinformationen erweiterte Darstellung ermöglicht die Erfassung des aktuellen Netzwerkzustandes auf einen Blick.

Die zur Netzwerkverwaltung eingesetzte Software sollte frei programmierbar und universell einsetzbar sein. Wichtig ist die Integration vorhandener Komponenten über Standards wie z. B. das *Simple Network Management Protocol* SNMP. Herstellerspezifische Lösungen, die in der Regel nur auf die Verwaltung der eigenen Komponenten optimiert sind, sollten nur im Ausnahmefall eingesetzt werden.

1.2.9 Internetdienste

Die Bereitstellung von Internetdiensten in Intranet- oder Internetstrukturen wird immer mehr zur Standardaufgabe im Netzwerk. Auch wenn aktuell noch keine direkte Anbindung des Firmennetzwerkes an das eigentliche *Internet* geplant ist, kann es sehr wichtig sein, bereits die grundlegenden Funktionen zu realisieren und dann zunächst im eigenen Intranet mit *WWW*- und *E-Mail*-Server die eigenen Arbeitsabläufe zu optimieren.

Immer mehr Firmen gehen dazu über, ihre Warenangebote und Dienstleistungen über datenbankgestützte Systeme direkt den Kunden anzubieten. Das Internet ist dafür eine ideale Plattform, die so zumindest oft beabsichtigte *E-Commerce*-Lösung kann die laufenden Betriebskosten drastisch reduzieren.

Mit diesen Anforderungen ist auch *TCP/IP* zum Standardprotokoll in Netzwerken geworden, obwohl es im direkten Vergleich zu anderen in PC-Netzwerken genutzten Protokollen , wie *NetBIOS/NetBEUI* oder *IPX/SPX* , die mit Abstand höchsten Anforderung an das für Konfiguration und Betrieb notwendige Fachwissen stellt.

Die Komplexität der eingesetzten Internetdienste nimmt noch schlagartig zu, wenn eine Verbindung in das *Internet* benötigt wird. Die Kommunkationsabläufe werden zentral über einen *Proxy-Server* geführt, eine *Firewall* sichert das eigene Netzwerk gegen unbefugten Zugriff. Vor allem Sicherheitsaspekte führen dann

zum Einsatz komplexer Softwarepakete, die über aufwändig programmierte Regeln den gesamten Datenverkehr überwachen und bei Bedarf unterbinden. Kommerzielle Pakete, die einen hohen Absicherungsgrad garantieren, erreichen Preisgrößenordnungen von 100.000 DM und mehr.

Alle Netzwerkbetriebssysteme bieten heute zumindest als Standard und ohne Nebenkosten einen WWW-Server. Alle weiteren Internetdienste werden dann aber in der Regel über Zusatzprogramme, oft von Zweitanbietern, realisiert und verursachen hohe Kosten. Da nahezu alle Linux-Distributionen die beschriebenen Internetdienste vollständig enthalten und damit keine Zusatzkosten entstehen, ist der Netzwerk-Server unter Linux besonders geeignet, als Server für Intranet- oder Internetstrukturen eingesetzt zu werden. Was in jedem Fall bleibt (und auch unabhängig vom eingesetzten Serverbetriebssystem ist), ist die hohe Anforderung an das Fachpersonal, das die komplexen Internetdienste konfiguriert und im Betrieb überwacht.

1.2.10 E-Mail

Die elektronische Kommuniaktion über *E-Mail*-Systeme ist auch dann schon Standardanforderung im Netzwerk, wenn Intranet- oder Internet-Strukturen keine Rolle im eigenen Netzwerk spielen. Selbst wenn es sich zu Beginn nur um ein einfaches, vielleicht auf Basis von *Microsoft Mail* konzipiertes System handelt, das ohne Server erst einmal ausschließlich auf PC-Netzwerkarbeitsplätzen realisiert wird, kann die Akzeptanz dieses Arbeitsmittels sehr schnell erreicht werden.

Serverbasierte *E-Mail*-Systeme verwalten alle Informationen zentral, organisieren Gruppen, die die Zustellung an viele Benutzer gleichzeitig ermöglichen, und stellen bei Bedarf eine Internetanbindung her. Dazu werden Serverdienste wie *POP3* oder *SMTP* eingerichtet, z. B. realisiert über kommerzielle Systeme *wie Microsoft Exchange* oder *Lotus Notes*. Auch hier bieten nahezu alle Linux-Distributionen ohne zusätzliche Kosten *E-Mail*- Systeme an, die sofort im kommerziellen Bereich eingesetzt werden können und sehr professionelle Lösungen darstellen.

1.2.11 Router zur Netzwerkverbindung

Sollen geografisch verteilte lokale Netzwerke bei Bedarf automatisch miteinander verbunden werden, so sind *Router* die hierfür geeigneten Netzwerkkomponenten. Die technische Realisierung der Weitverkehrsverbindungen erfolgt typischerweise durch ISDN-Leitungen (StandardAnschluss: *S0-Bus*). Bei Kanalbündelung können bis zu 128 kBit/s erreicht werden. Die so für die Weitverkehrsverbindung nutzbare Übertragungsbandbreite ist mit 0,1 MBit/s aber um den Faktor 1000 langsamer als die theoretisch erreichbare Bandbreite im hausinternen 100-MBit/s-Netzwerk. Dieser große Unterschied wird bei der Arbeit im Netzwerk deutlich

spürbar, umso wichtiger wereden im Rahmen der Netzwerkplanung Überlegungen, die geforderten Ressourcen ausreichend nahe an den Ort der tatsächlichen Nutzung zu plazieren.

Router müssen sehr leistungsfähige und vor allem frei programmierbare Netzwerkkomponenten sein. Ein Router soll

- die Verbindung automatisch nach Auswahl der im entfernten Netzwerk (*Remote Network*) vorhandenen Ressource hergestellen,
- unnötige Verbindungen vermeiden,
- bei mehreren möglichen Verbindungen die Leitung nutzen, die aktuell verfügbar und nach vorgegebenen Kriterien am besten geeignet ist,
- während der Verbindungszeit ständig die Leitungsqualität überwachen und bei Bedarf unvollständige oder fehlerhafte Datenübertragungen wiederholen,
- nach Abschluß dieser und eventuell weiterer Verbindungsanforderungen die hergestellte Verbindung nach Ablauf einer frei einstellbaren Wartezeit automatisch wieder beenden.

Diese Komplexität der Routerfunktionen macht deutlich, dass z.B. der im Lieferumfang von Windows-NT-Systemen enthaltenen *RAS-Server-Dienst* nicht zur Realisierung eines solchen Routers geeignet ist. Der RAS-Dienst ist, mit den entsprechenden *DFÜ-Clients* unter Windows oder OS/2 allerdings eine einfach zu konfigurierende Möglichkeit, einzelnen Rechnern die Einwahl in ein Firmennetz zu ermöglichen, z.B. im Rahmen der Realisierung von Telearbeitsplätzen oder zu Fernwartungsaufgaben.

Zur Kopplung lokaler Netzwerke sind, wie auch bei Novell-Systemen, die allerdings noch nicht einmal über einer vergleichbaren RAS-Dienst verfügen, entweder zusätzliche Software (ein *Multiprotokollrouter*) oder externe Netzwerkkomponenten notwendig.

Auch hier bietet das Netzwerkbetriebssystem Linux Router-Funktionen, die alle oben gestellten Anfordungen vollständig erfüllen, ohne weitere Kosten entstehen zu lassen.

1.3 Serverhardware

Unter Linux kann auch der »bereits seit langem ausgemusterte 486er« noch als Hochleistungsserver eingesetzt werden – solche und ähnliche Argumente werden immer wieder angebracht, wenn es darum geht, die Vorzüge von Linux-basierten Servern aufzuzeigen. Ist Linux tatsächlich so gut, dass nahezu keine besonderen Anforderungen an die Serverhardware zu stellen sind, liegt hier nicht die Möglichkeit, viel Geld für Neuanschaffungen zu sparen?

Die Hardware ist die Basis für das Serversystem. Im kommerziellen Bereich bestehen, unabhängig vom eingesetzten Betriebssystem, hohe Anforderungen an die Zuverlässigkeit der eingesetzten Anlagen. Die Technik muss diesen Anforderungen entsprechen und das heißt auch, möglichst ganzjährig und »rund um die Uhr« verfügbar sein. Für den Arbeitsplatz-PC kann dagegen auf ein anderes Gerät gewechselt werden oder es wird ein temporärer Ausfall toleriert. Für einen Server lassen sich solche Einschränkungen in der Regel nicht hinnehmen, die *Server-Up-Time* sollte weit über 90 Prozent liegen.

Damit stehen bei der Auswahl von Serverhardware Überlegungen zur Zuverlässigkeit im Vordergrund, die Anforderungen an die Hardware sind hoch.

Die *Zuverlässigkeit* (*Reliability*) ist die Wahrscheinlichkeit, mit der ein nichtreparierbares System oder ein System, für das keine Wartungs- oder Reparaturmaßnahmen vorgesehen sind, während einer festgelegten Zeitspanne nach Inbetriebnahme nicht ausfällt. Betrachtet wird also die Wahrscheinlichkeit, mit der ein Netzwerk eine bestimmte Lebensdauer erreicht. Jeder Ausfall entspricht dem Totalausfall.

Der Server muss also »fehlertolerant« sein, d.h. Hardware-Ausfälle müssen in gewissem Umfang toleriert werden können. Richtig ist zwar, dass Linux gerade im Vergleich zu Novell oder Windows-NT-Systemen vergleichsweise niedrige Hardwareanforderungen stellt, wie bei jedem anderen Serverbetriebssystem sind aber ganz bestimmte Zuverlässigkeitsanforderungen zu erfüllen. Im Bereich der Serverhardware darf und kann nicht gespart werden, PCs »von der Stange« sind in der Regel nicht einsetzbar.

1.3.1 Die Zentraleinheit (CPU)

Die heute sehr häufig im Server-Bereich eingesetzten Intel-basierten Systeme mit Pentium-CPU und Taktfrequenzen im Bereich von mindestens 500 MHz stellen bereits eine sehr leistungsfähige Serverplattform zur Verfügung. Wird mehr Leistung benötigt, können auf der gleichen Basis Mehrprozessorsysteme realisiert werden. In der Regel ist es einfach, zumindest ein Doppelprozessorsystem zu realisieren. Systeme mit 4, 8 oder mehr CPUs erfordern dann eine herstellerspezifische Erweiterung der Serversoftware im Bereich des Betriebssystemkerns. Die Grenze liegt aktuell bei 32 oder 64 CPUs in einem Server.

Jeder Server sollte möglichst »zukunftssicher« gestaltet werden. Wird später mehr Leistung gefordert, muss es auf einfachem Wege möglich sein, weitere CPUs (zumindest die zweite) nachzurüsten. Diese »Skalierbarkeit« der eingesetzten Hardwareplattform muss sehr kritisch betrachtet werden. Damit sind jetzt schon wichtige Kriterien definiert, die für die Auswahl des Mainboards anzusetzen sind. Das Doppelprozessorboard sollte die Einstiegslösung sein, auch wenn zunächst nur ein Prozessor bestückt wird.

Bei gestiegenen Leistungsanforderungen sollte ein Wechsel auf eine andere Hardwareplattform ohne großen Aufwand möglich sein. Hier haben die meisten Serverbetriebssysteme ihre Schwächen, denn außerhalb der Intel-Welt bleiben wenige Möglichkeiten für Betriebssysteme wie Novell oder Windows NT (wobei das letztgenannte schon deutlich mehr Möglichkeiten bietet). Unix-basierte Systeme laufen seit Jahrzehnten auf fast allen bekannten Hardwareplattformen. Zusätzlich wurde etwas erreicht, wovon andere moderne Serverbetriebssysteme weit entfernt sind: Software, die vor vielen Jahren für Unix geschrieben wurde, läuft auch heute noch, jedoch um ein Vielfaches schneller, auf modernen 32- oder 64-Bit Unix-Plattformen. (Wer kennt denn schon noch die nicht noch einmal halb so alten Betriebssysteme wie *CP/M*, *MP/M* und darauf basierende Programme oder kann noch damals für diese Systeme entwickelte Software einsetzen?) Was aus den anderen, heute aktuellen Server-Betriebssystemen wird, bleibt abzuwarten.

Linux führt die Unix-Tradition mit gleicher Zielsetzung fort. Kurz nach Erscheinen neuer Hardware sind schnell entsprechende Anpassungen verfügbar, die Realisierung von 32- und 64-Bit-Serverplattformen ist bereits nahezu uneingeschränkt möglich. Linux unterstützt bereits seit längerer Zeit 64-Bit-Prozessoren wie *Alpha und Ultr-SPARC*, an der Portierung auf *Intels Merced* wird gearbeitet. Immer mehr Anbieter von Hochleistungshardware (*High-End-Systeme*) bieten ihre Server auch für Linux an und liefern entsprechende Kernel-Anpassungen zu einem entsprechenden Preis.

Wenn heute sehr hohe Anforderungen an die Zuverlässigkeit eines Servers gestellt werden, so kommen immer sogenannte *Mainframe*-Systeme zum Einsatz. Diese »Großrechner« müssen also noch über ein Merkmal verfügen, das entweder PC-basierte Server noch nicht haben oder das man diesen Rechnern nicht zutraut. Ein Mainframe arbeitet in der Regel in einer *Cluster*-Struktur, wobei ein Cluster als Block von mehreren Rechnern zu verstehen ist. Ein Cluster, in Abbildung 1-3 als Struktur mit drei CPUs dargestellt, verhält sich für den Netzwerk-Client, also von außen gesehen, wie ein einzelner Rechner: die Last wird vom Cluster-Betriebssystem möglichst gleichmäßig auf die zur Verfügung stehenden CPUs verteilt. Zwar bringt das eine höhere Rechenleistung, im Vordergrund steht aber die Fähigkeit dieses Clusters, den Ausfall einer oder sogar mehrerer CPUs zu »tolerieren«.

Fällt in dem in Abbildung 1-3 dargestellten Cluster die CPU B aus, so arbeitet das Gesamtsystem weiter, wenn auch mit verringerter Leistung. Die ausgefallene Komponente kann dann ohne Unterbrechung des Serverbetriebes gewechselt werden.

Zur technischen Realisierung solcher Systeme werden die einzelnen CPUs mit der direkt dazu gehörigen Hardware und Arbeitsspeicher als Einschub ausgeführt. Mehrere Einschübe werden in einem Geräteträger zusammengefasst und bilden das Cluster, in dem sich alle CPUs gegenseitig überwachen. Fällt eine CPU aus, so wird dieser Fehler angezeigt, der entsprechende Einschub kann während des Be-

triebes gewechselt werden (H*ot Plug*). Diese bisher nur in der Mainframe-Technik anzutreffende Hardware bietet zusätzlich auch den Vorteil, die tatsächliche Rechenleistung bei Bedarf durch das Hinzufügen eines weiteren Einschubs zu erhöhen. Diese »frei skalierbare Hardwareplattform« hat dann natürlich ihren Preis – ein einzelner Einschub kann über 70.000 DM kosten, dazu kommen die Kosten für das Trägersystem, den Massenspeicher und andere Komponenten.

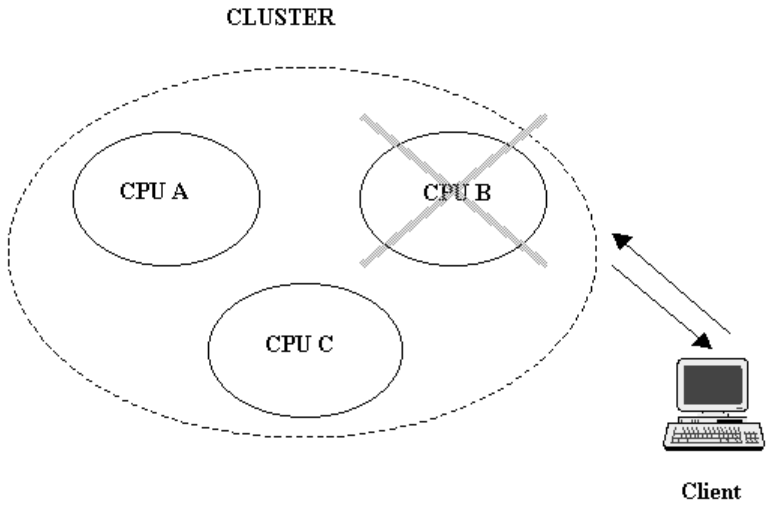


Abbildung 1-3 Zuverlässigkeitsgerichtetes Cluster

1.3.2 Arbeitsspeicher

Der Arbeitsspeicher (RAM) eines Servers umfaßt, unabhängig vom verwendeten Betriebssystem, mindestens 128 MByte. Warum die Systeme so extrem speicherintensiv sind, läßt sich an einem kleinen Beispiel verdeutlichen:

Moderne Server arbeiten nach dem Prinzip des präemptiven Multitaskings, d.h. das immer die Kontrolle behaltende Betriebssystem verteilt die zur Verfügung stehende Rechenzeit auf die angeforderten Prozesse und scheint damit eine Vielzahl von Aufgaben nahezu gleichzeitig auszuführen. Abbildung 1-4 zeigt ein einfaches Beispiel für drei laufende Prozesse P1, P2 und P3.

Betrachten wir den Umschaltzeitpunkt »Taskumschaltung« zwischen den Prozessen P1 und P2: Wird P1 beendet, so müssen alle den aktuellen Prozesszustand beschreibenden Daten gesichert, danach alle zu P2 gehörenden Daten wiederhergestellt werden. Diese extrem zeitkritische Aktion kann sehr viel Speicher erfordern. Zusätzlich nehmen diese Speicheraktionen mit der Zunahme der vom System geführten Prozesse auch immer weiter an Umfang zu. Gespeichert werden sollten

diese Informationen im Arbeitsspeicher, nicht in der deutlich langsameren »Auslagerungsdatei«, der Swap-Datei, auf der Festplatte. Sollte dies aus Mangel an Arbeitsspeicher aber erforderlich werden, wird der Server erheblich langsamer.

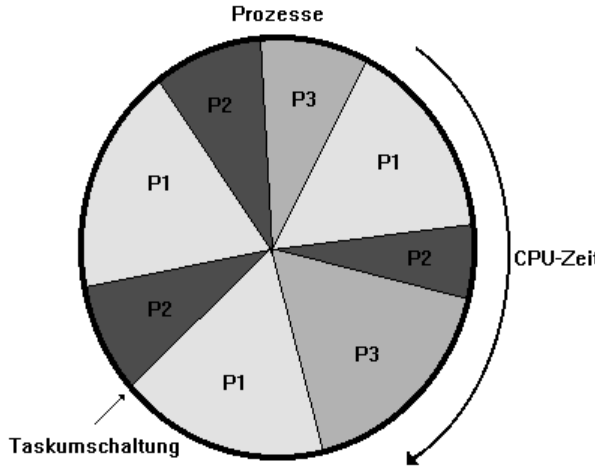


Abbildung 1-4 Zeitscheibe der CUP-Rechenzeit

Allein aus diesem Grund wird schon deutlich, warum jeder Server sehr viel Arbeitsspeicher braucht.

Der virtuelle Adressraum einer Intel-CPU ist auf 2 GByte begrenzt, was für herkömmliche Anwendungen ausreichend ist. In kommerziellen Servern kann diese Grenze durchaus erreicht werden.. Unix-Systeme und Windows NT können mit noch deutlich größeren Arbeitsspeichern arbeiten, für Linux ist zumindest schon ein Kernel-Patch verfügbar, der die Grenze auf 4 GByte RAM schiebt.

Wachsende Arbeitsspeicher bringen aber auch eine Reihe von Problemen mit sich:

- Immer mehr Daten werden im flüchtigen Arbeitsspeicher (RAM) gehalten. Bei einem Ausfall oder Spannungsschwankungen sind diese Informationen unwiederbringlich verloren. Server müssen also nicht nur sauber herauf- und heruntergefahren werden, um die im RAM liegenden Informationen auf die Festplatte schreiben zu können, sie benötigen auch ausreichend stabile Netzteile und eine zusätzlich geschützte Stromversorgung. Insbesondere der Schutz vor Unter- und Überspannungsereignissen sowie vor Netzausfällen erfordert zusätzliche Hardware.
- Die Wahrscheinlichkeit eines Fehler steigt mit zunehmender Größe immer weiter an. Eine *Checksum-* (*Parity-Bit-*) Prüfung kann einen solchen Fehler zumindest erkennen. Das System wird im Fehlerfall aber angehalten werden

müssen, es kommt zum sofortigen Totalausfall des Servers! Besser sind Arbeitsspeicher, die solche Einzelbitfehler nicht nur erkennen, sondern auch automatisch korrigieren. Diese Error-Correcting-Code (ECC) Speicher verfügen dazu über eine zusätzliche Hardware, die typischerweise bis zu 4 Bitfehler je Wort erkennen und beheben kann, ohne die Server-CPU zu belasten. Der ECC-Speicher ist heute nur noch ca. doppelt so teuer wie der einfachere Speicher, er sollte unbedingt bei kommerziellen Servern eingesetzt werden.

1.3.3 Festplattensystem

Die in Servern verwendeten Festplatten mit Kapazitäten im mehrfachen Gigabytebereich stellen Speicherkapazitäten zur Verfügung, die auch für die modernen, stark grafikorientierten Datenbestände ausreichend groß sind. Die zur Verfügung stehenden Speicherkapazitäten stellen also kaum noch ein Problem dar, vielmehr wird es immer wichtiger, die Festplattensysteme unter den Kriterien Zugriffsgeschwindigkeit (*Performance*) und Zuverlässigkeit zu untersuchen.

Zugriffsgeschwindigkeit

Die Geschwindigkeit, mit der Daten im Festplattensystem eines Servers geschrieben und gelesen werden können, begrenzt die Gesamt-Systemleistung. Zusätzlich haben Untersuchungen von Dateisystemen eindeutig ergeben, dass die »Belastung« innerhalb des Gesamtsystems ungleichmäßig verteilt ist. Es existieren auf der einen Seite sogenannte Brennpunkte (*Hot Spots*) als Bereiche mit sehr hoher Anzahl an Zugriffen (z.B. Auslagerungsdateien, Swap-Areas) und im Gegensatz dazu Bereiche mit geringer bis keiner Aktivität (z.B. Druckwarteschlangen, Spool-Verzeichnisse, Benutzerverzeichnisse). Teilweise bestehen die Brennpunkte sogar nur aus einer einzigen Datei oder aus einem Satz von Dateien, die standardmäßig auf derselben Platte liegen. Auf diesen Betrachtungen basiert die sogenannte 80/20-Regel. Sie besagt, dass auf den meisten Systemen auf 20% der Plattenkapazität 80% der I/O-Anfragen anfallen.

Eine kleine und zugegebenermaßen sehr grobe Abschätzung soll jetzt die grundsätzliche Problematik der Zugriffsgeschwindigkeit verdeutlichen:

> Nehmen wir die mittlere Zugriffszeit im Arbeitsspeicher (RAM) an mit 10 ns (1x 10^{-9} s), die mittlere Zugriffszeit auf eine Festplatte mit 10 ms (1x 10^{-3} s), so beträgt dann die Differenz zwischen diesen beiden Werten 1x 10^{6}, *also eine Million*!

Der erhebliche Geschwindigkeitsnachteil der Festplatte hat im Wesentlichen mechanische Ursachen. Entscheidend ist der hohe Zeitbedarf für die Positionierung des Schreib-Lesekopfes inkl. der Beruhigungszeit. Der Abschätzung folgend sollten also alle Schreib- und Leseaktionen für Daten optimalerweise im Arbeitsspeicher erfolgen, was allerdings in der Praxis kaum zu realisieren ist. Die Daten müssen dauerhaft auf einem Datenträger gespeichert werden, damit wird der Zugriff auf das Festplattensystem zumindest nicht ganz zu vermeiden sein.

Um einen Geschwindigkeitszuwachs für das Festplattensystem zu erreichen, kann mit unterschiedlichen Ansätzen gearbeitet werden:

1) Möglichst viel Arbeitsspeicher: Nahezu alle Serverbetriebssysteme verwalten dynamische Puffer mit Festplattendaten im Arbeitsspeicher (*Cache*). Mit Techniken wie R*ead Ahead* (Daten im Voraus, also »auf Verdacht« lesen) und *Delayed Write* (Daten erst dann auf die Festplatte schreiben, wenn »Zeit« dafür ist oder die Datenmenge es erfordert) können die mittleren Festplatten-Zugriffszeiten erheblich verbessert werden.

2) »Parallelisieren«, d.h. Verteilen der Last auf mehrere Festplatten: Dies läßt sich z.B. mit dem Feature »Disconnect« guter SCSI-Systeme erreichen. Der Controller sendet an die angeschlossene Festplatte den Befehl »Gehe zu Track x, Sektor y«. Unmittelbar danach, also noch zum Beginn der Kopfpositionierungszeit, schickt der Controller den Befehl »Disconnect« und trennt diese Platte vorübergehend vom SCSI-Bus ab, sodass andere SCSI-Geräte ihre Datenübertragungen ausführen können. Später wird die Verbindung zur Platte wieder aktiviert, optimalerweise genau dann, wenn die Positionierung abgschlossen ist.

3) Ein RAID-System mit eigenem Controller bringt den größten Geschwindigkeitszuwachs, da jetzt die Daten tatsächlich parallel von den Festplatten gelesen und geschrieben werden können. RAID-Controller verfügen in der Regel über einen eigenen großen Arbeitsspeicher, der die Zugriffsgeschwindigkeit noch weiter erhöht. Diese als *RAID-0*-.Systeme bezeichneten Lösungen werden durch spezielle SCSI-Controller realisiert, die dann ein aus mehreren Festplatten gebildetes *Stripe-Set* realisieren. Auf diese Weise werden die I/O-Anfragen weitgehend gleichmäßig auf alle beteiligten Platten verteilt (*Load-Balancing*). Die Performance insbesondere der oben genannten Brennpunkte wird erheblich gesteigert. Gerade dann, wenn der Plattentreiber zusätzlich das sogenannte *Command-Queueing* unterstützt (z.B. Linux SCSI-Treiber) bzw. die Platten an unterschiedlichen Adaptern hängen, erhält man einen sehr hohen Performancegewinn. Es ist also von Vorteil, mehrere kleine Platten zu einer großen zusammenzusetzen und diese dann an entsprechend leistungsfähigen Controllern zu betreiben. Die für das Führen des Strip-Set benötigte Rechenleistung wird komplett von dieser Hardware übernommen und der Server muss keine zusätzliche Rechenzeit aufwenden. Der mit der Anzahl der Festplatten zunehmenden Gefahr eines Ausfalles wird bei RAID-0 allerdings nicht begegnet.

Zuverlässigkeit

Bereits 1987 veröffentlichte ein Team an der *University of California, Berkeley* (UCB), dass die Performance eines Festplattensystems zwar gesteigert werden kann, wenn die Daten auf mehrere kleinere Platten und auf einer großen verteilt werden. Dabei sinkt allerdings die *Mean Time Between Failure* MTBF für das gesamte System allerdings dramatisch.

Obwohl Festplattenausfälle (z.B. durch den gefürchteten *Head-Crash*) heute eher selten geworden sind, darf diese Gefahr nicht unterschätzt werden. In einem kommerziellen System ist es nicht zu akzeptieren, dass der Ausfall nur einer Festplatte bereits einen Serverstillstand nach sich zieht und Ausfallzeiten von mehreren Stunden auftreten, um die Hardware zu ersetzen und die hoffentlich gesicherten Datenbestände wieder zurückzuspielen.

Um dieser Gefahr zu begegnen und die geforderte Zuverlässigkeit zu erreichen, wurden mehrstufige RAID-Systeme (RAID-1 bis RAID-5) entwickelt. Die Industrie hat später noch zwei weitere (das bereits besprochene RAID-0 und zusätzlich RAID-6) hinzugefügt. Ein *Redundant Array of Inexpensive Disks* RAID verwendet mehr Platten, als für das Erreichen der gewünschten Speicherkapazität nötig wären (Redundanz). Auf diese Weise wird der gesunkenen MTBF entgegengesteuert, die Zuverlässigkeit wird erhöht.

Die an der UCB spezifizierten RAID-Stufen haben folgende Eigenschaften:

- Mehrere physikalische Platten, die nach außen hin wie eine gesehen werden. Unter Linux ist dieses das MD-Device, */dev/md*.
- Die Daten werden auf definiertem Weg auf die verschiedenen Platten verteilt.
- Redundanter Plattenplatz wird benutzt, damit Daten auch bei Ausfall einer oder mehrerer Platten wiederhergestellt werden können.

In der Praxis sind vor allem die RAID-Level 1 und 5 anzutreffen, die hier kurz vorgestellt werden sollen:

- RAID-1: Hier werden die Platten einfach gespiegelt. Damit erhöhte sich zwar die Zuverlässigkeit erheblich, es wurde jedoch nichts gegen das ursprüngliche Brennpunkteproblem unternommen. Ein RAID-1-System kann zusätzlich auch wie ein RAID-0-System als Striping-System ausgelegt werden. So wird gleichzeitig ein Höchstmaß an Performance und sehr gute Datensicherheit erreicht, allerdings auch die höchsten Kosten, denn man muss den gleichen Plattenplatz genau zweimal einrichten. Wenn das RAID-System auch die Lesezugriffe auf die gespiegelten Platten verteilt, wird das Lesen von Daten noch schneller.
- RAID-5: wird typischerweise aus einem Array von fünf gleich großen Festplatten gebildet. Es werden mit XOR Checksummen über beliebig große Stücke berechnet, die auf verschiedenen Festplatten gespeichert werden. Der erste Checksummen-Block liegt auf der letzten, der nächste auf der vorletzten Platte und so weiter. RAID-5 erhöht nicht nur die Zuverlässigkeit, sondern bringt auch einen deutlichen Geschwindigkeitsgewinn, da die Daten am Stück geschrieben werden können. Die I/O-Anfragen werden gleichmäßig auf die Platten verteilt, sodass man auch eine gute I/O-Performance erhält. Deutlich wird, dass man hier die größte Ausfallsicherheit bei gleichzeitiger Performancestei-

gerung erhält. Selbst, wenn zwei Platten ausfallen, können die Daten komplett aus den verbliebenen rekonstruiert werden. Erst beim Ausfall der dritten Festplatte sind die Daten unwiderruflich verloren. RAID-5-Systeme ermöglichen in der Regel den Austausch ausgefallener Festplatten während des Betriebes (*Hot Swap*), sodass nach einem Festplattenausfall auch die Reparatur ohne Serverstillstand erfolgen kann.

Bei den hohen Zuverlässigkeitsanforderungen kommerzieller Serversysteme gehören RAID-Systeme heute zum Standard. Einige Serverbetriebssysteme enthalten dafür Softwarelösungen; eindeutig zu bevorzugen sind aber immer Lösungen auf Basis zusätzlicher RAID-Controller, die dann die Server-CPU(s) von den zusätzlich notwendigen Rechenleistungen entlasten und so auch einen erheblichen Performancegewinn bringen.

1.3.4 Diskettenlaufwerk, CD-ROM, DVD

Die Ausstattung eines Servers mit Disketten- und CD-ROM-Laufwerk(en) gehört heute zum Standard. Wird das Diskettenlaufwerk aber nur zum Einspielen kleiner Patches oder vielleicht noch zur Sicherung kleiner Datenmengen durch den Systemverwalter genutzt, so werden ein oder mehrere CD-ROM-Laufwerke auch immer mehr zur Bereitstellung von Daten eingesetzt.

Viele Server übernehmen so auch die Funktion eines CD-ROM- und/oder DVD-Servers, wobei die Bereitstellung der CDs durch den Systemverwalter erfolgt. Da CD-ROM-Laufwerke noch langsamer sind als Festplattenlaufwerke, muss hier entweder ein erheblicher Arbeitsspeicherbedarf zum Zwischenspeichern abgerufener Informationen einkalkuliert werden oder die Daten werden auf schnellen Festplatten zwischengespeichert und von dort aus im Netzwerk zur Verfügung gestellt.

Unter Novell und Windows NT-Systemen können CD-ROMs nur unter den zugewiesenen Laufwerkbuchstaben genutzt werden. Der Vorrat an einzelnen, frei verfügbaren Buchstaben ist aber begrenzt. Unix-Systeme ermöglichen es, beliebig viele CD-ROM-Laufwerke unter einem gemeinsamen Verzeichnis zu »mounten«, d.h. der User sieht später in seinem Dateisystem vielleicht ein Verzeichnis *CD-ROM* und darunter alle CDs mit Namen, die den Inhalt eindeutig kennzeichnen. Windows 2000 wird eine solche Möglichkeit anbieten, unter Unix-Systemen ist dieses Verfahren Standard.

1.3.5 Bandlaufwerk zur Datensicherung

Zur Sicherung von Serverdatenbeständen im Mehrfach-Gigabyte-Bereich werden heute vorzugsweise digitale Bandlaufwerke (DAT) eingesetzt. Optische Speichermedien erreichen diese Kapazitäten auch, schneiden aber im Preis-Leistungsvergleich noch deutlich schlechter ab.

Ein DAT-Bandlaufwerk ausreichender Kapazität gehört zur Ausstattung jedes kommerziellen Servers. Ob das Gerät am Server selber eingebaut wird oder ob die Sicherung über einen Arbeitsplatz-PC erfolgt, hängt von den technischen und räumlichen Gegebenheiten ab. Auch ist es möglich, mit der entsprechenden Sicherungssoftware mehrere Server gemeinsam auf ein Band zu sichern.

Auf die in der Literatur ausreichend behandelten Sicherungsverfahren soll an dieser Stelle nicht weiter eingegangen werden, erlaubt sei aber ein Hinweis auf die möglichen Gefahren, die die selbsterstellte Sicherungsstrategie unbedingt berücksichtigen sollte:

- Serververlust durch Brand: Die Bänder sollten zumindest in einen Stahlschrank oder Tresor in einem anderen Brandabschnitt gelagert werden, besser noch weiter vom Serverstandort entfernt in einem Bankschließfach.
- Serververlust durch Diebstahl: Bei Beachtung der vorstehenden Absicherungsmaßnahmen ist auch ein Diebstahl der Serverhardware kein Grund, nicht innerhalb weniger Stunden alle Serverfunktionen wiederherzustellen.

Bandsicherungen werden wesentlich häufiger dazu genutzt, um partiell beschädigte Datenbestände wiederherzustellen oder um virusverseuchte Dateien durch ältere, aber vielleicht virenfreie Kopien zu ersetzen. Die dafür eingesetzte Steuerungs-Software sollte ausreichend stabil und eindeutig in der Bedienung sein, da es sich hier aus Sicht des Systems immer um sehr kritische Arbeitsgänge handelt.

1.3.6 Netzwerkkarte(n)

Der Anschluss an das Netzwerk, also die Verbindung zu den Clients, erfolgt über eine oder mehrere Netzwerkkarten im Server. Alle Server-Betriebssysteme sind in der Lage, unterschiedliche Netzwerkkarten anzusteuern. Eine Mischung aus Ethernet-Systemen (100BaseT, 10BaseT) und anderen Netzwerken (z.B. Token-Ring, TPDDI oder FDDI) ist sehr häufig anzutreffen. Der Server übernimmt dann auch Routing-Aufgaben und verbindet bei Bedarf die Teilnetze. Als Netzwerkprotokoll kommt heute fast ausnahmslos TCP/IP zum Einsatz.

Ein Server sollte immer eine möglichst schnelle Netzwerkanbindung erhalten. Am weitesten verbreitet ist heute Ethernet 100BaseT, das Daten mit 100 MBit/s (vollduplex 200 MBit/s) überträgt.

Auch hier wieder eine Überlegung zur Zuverlässigkeit: Fällt diese Netzwerkkarte aus, so ist dies aus Sicht der Clients auch sofort der Server-Totalausfall! Um hier vorzubeugen, ist auch hier redundante Hardware erforderlich, d.h. es wird eine zweite Netzwerkkarte eingebaut, die parallel betrieben wird und damit bei Ausfall der ersten Netzwerkkarte die Kommunikation mit den Clients übernimmt bis die defekte Hardware ausgetauscht werden kann.

1.3.7 Servergehäuse

Das Servergehäuse muss ausreichend Platz für alle benötigten Komponenten bieten, ein »Big-Tower«-Standardgehäuse ist hier aber in der Regel nicht ausreichend.

Zunächst muss berücksichtigt werden, dass eine Reihe von Zusatzgeräten, wie z. B. SCSI-Karten, Festplatten- und CD-ROM-Laufwerke, nicht nur mechanisch befestigt werden, sondern auch mit Energie versorgt werden müssen. Die Mindestanforderung hier ist ein 300-W-Netzteil. Mehr Leistung bedeutet auch mehr Verlustwärme. Es werden zusätzliche Lüfter benötigt, um die notwendige Kühlung sicherzustellen. Servergehäuse sollten auch aus Zuverlässigkeitssicht immer über mehrere Gehäuselüfter verfügen, die zusätzlich zu Netzteil-, CPU- und Festplattenlüftern betrieben werden. Die zu erwartende Geräuschentwicklung verhindert jedoch den Einsatz direkt im Bürobereich.

Sehr häufig werden für den mechanischen Aufbau von Serversystemen auch 19«-Gehäuse verwendet, die dann in »Racks« montiert werden. Diese mechanisch sehr stabilen Gehäuse sind deutlich teurer als vergleichsweise Server-Cases; das Problem liegt aber oft in der mangelnden Erweiterbarkeit. Müssen zusätzliche Laufwerke eingebaut werden, ist es häufig notwendig, entweder ein völlig neues, größeres Gehäuse zu beschaffen oder die Peripherie in ein zweites Gehäuse »auszulagern«. Werden 19«-Gehäuse verwendet, sollte der Platzbedarf also sorgfältig und weitsichtig geplant werden.

1.3.8 Netzteil(e)

Server benötigen eine redundante Stromversorgung mit einer Leistung von mindestens 300 W. Optimal ist ein Servergehäuse mit zwei oder sogar drei *redundanten* Netzteilen (jeweils mit Lüfter), die über eine Ausfallanzeige verfügen und im laufenden Betrieb gewechselt werden können.

Zusätzlich sollte der Anschluss an das 230-V-Netz so erfolgen, dass alle drei Phasen wahlweise genutzt werden können. Fällt nur eine Phase aus, so kann der Server in diesem Fall nach Umschalten bzw. Umstecken der Versorgungsspannungsleitung weiter betrieben werden.

1.3.9 Unterbrechungsfreie Stromversorgung

Jede Spannungsschwankung, insbesondere auch der kurzfristige Wegfall der Netzspannung führen ohne weitere Maßnahmen in der Regel immer zum Ausfall des Servers. Auch wenn keine direkten Schäden entstehen, sind zumindest die im Arbeitsspeicher gehaltenen Daten verloren, der Server ist »abgestürzt« und muss neu gestartet werden.

Unterbrechnungsfreie Stromversorgungen (*USV, UPS*) schützen Server vor diesen Gefahren. Sie puffern mit den eingebauten Akkusätzen die Versorgungsspannung und fahren den Server über eine eingebaute Elektonik dann so herunter, dass keine Daten verlorengehen. Nach dem Wiederherstellen der Versorgungsspannung wird der Server automatisch neu gestartet.

Eine USV ist im kommerziellen Einsatz immer ein externes Gerät, das entweder direkt neben den Server gestellt wird oder im 19«-Rack (Abbildung 1-5) mit montiert wird. Die Kapazität der USV und die Leistungsaufnahme des Servers bestimmen die maximale Pufferzeit. Da die Akkus einer nicht unerheblichen Alterung unterliegen, nimmt dieses Pufferzeit über die zu erwartende Lebensdauer von ca. 2 Jahren ständig ab. Fast jede USV ermöglicht aus diesem Grund einen einfachen Wechsel des Akkusatzes.

Abbildung 1-5 USV für Montage im 19«-Rack (Quelle: www.apcc.com)

Eine USV ausreichender Kapazität sollte eine Pufferzeit von ca. 15 Minuten erreichen können, bei typischen Servern erfordert das Leistungen im Bereich von 1000 bis 1500 VA. Die USV muss über eine Kommunikationsverbindung zum Server verfügen (typischerweise eine serielle Verbindung), die im Zusammenwirken mit der auf dem Server installierten Software zumindest das Herunterfahren am Ende der Pufferzeit ermöglicht. Die meisten USV ermöglichen aber auch die Abfrage des aktuellen Systemzustandes und die Steuerung der Systemparameter über die gleiche Software.

USV-Geräte gibt es in einer preiswerten *Offline*- (Abbildung 1-6) und in der bei vergleichbarer Leistung fast doppelt so teuren *Online*-Ausführung (Abbildung 1-7).

Bei vorhandener Netzspannung versorgt die Offline-USV den Server direkt über das 230-V-Netz (Schalter S1 offen). Parallel dazu wird der Akkusatz geladen. Fällt die Netzspannung aus, schaltet die USV auf Akkubetrieb um (Schalter S1 geschlos-

sen) und versorgt den Server für begrenzte Zeit mit Energie. Dadurch entstehen entsprechende Schaltzeiten, die im Bereich von wenigen Millisekunden liegen.

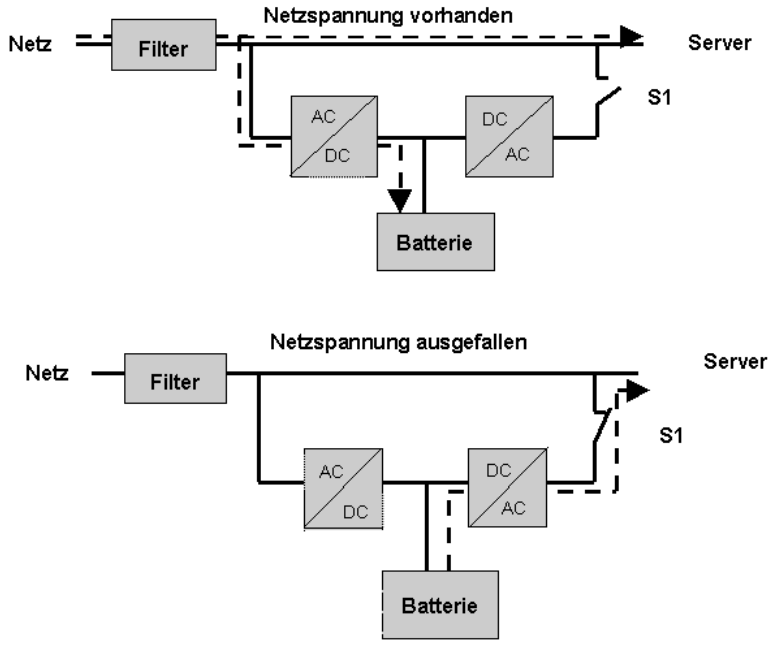


Abbildung 1-6 Prinzipschaltbild einer Offline-USV

Die *Line-Interaktive USV* ist eine Weiterentwicklung der Offline-Technologie. Durch einen parallel zur Spannungsversorgung der USV geschalteten Regelkreis werden die Schwankungen der Netzspannungen auf einen für den Verbraucher tolerierbaren Wert reguliert und Netzressourcen optimal genutzt, da bei Spannungseinbrüchen nicht gleich auf Batteriebetrieb geschaltet wird. Dies erhöht die Batterielebensdauer.

Im Vergleich zur Offline-USV enthält die Online-USV zunächst offensichtlich weniger Komponenten. Der Umschalter und die Kabelverbindung zwischen Ein- und Ausgang sind entfallen. Dieser USV-Typ versorgt jetzt den Server ständig über den Pfad *Netz -> Ladeteil (AC/DC) -> Batterie -> Spannungswandler (DC/AC) -> Server* »online« mit Energie. Die eingehende Netzspannung wird galvanisch getrennt und gleichgerichtet. Die gleichgerichtete Spannung wird geglättet, stabilisiert und wieder in eine saubere Wechselspannung gewandelt. Dieses Verfahren gewährleistet für die Verbraucher weitgehende Abschirmung von Spannungsschwankungen, Unterbrechungen, Rauschen oder Spikes. Sowohl Ladeteil wie auch Spannungswandler müssen damit wesentlich mehr Dauerleistung aufbringen als in der Offline-USV, die Bauteile werden deutlich größer und teurer. Für den kommerziellen Einsatz sollte unbedingt eine Online-USV zum Einsatz kommen.

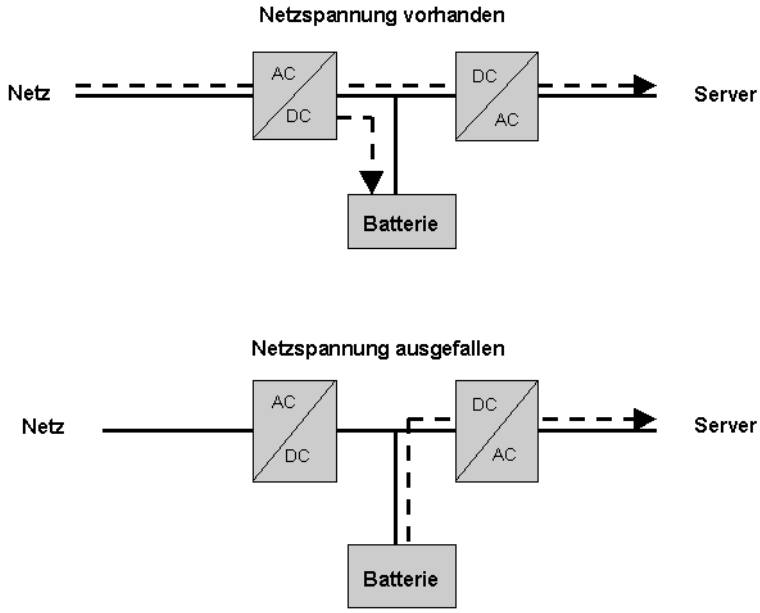


Abbildung 1-7 Prinzipschaltbild einer Onliner-USV

1.3.10 Überspannungsschutzmaßnahmen, EMV

Der Stellenwert der Elektromagnetischen Verträglichkeit (EMV) hat in der vergangenen Jahren erheblich an Bedeutung gewonnen. Die Elektromagnetische Verträglichkeit wird nach DIN 0870 definiert als

die Fähigkeit einer elektrischen Einrichtung – Anlage, Gerät, Baugruppe – in ihrer elektromagnetischen Umgebung zufriedenstellend zu funktionieren, ohne diese Umgebung unzulässig zu beeinflussen.

Das größte Gefährdungspotential für Netzwerksysteme zeigt sich bei direktem bzw. indirektem Blitzeinschlag und den unmittelbar damit zusammenhängenden elektrischen Effekten. Dabei treten die größten Schäden und Zerstörungen häufig gerade an den Geräten auf, bei denen man auf eine permanente Betriebsbereitschaft angewiesen ist.

Die primären Maßnahmen gegen Überspannungseffekte sind Potentialtrennung, Schirmung, Erdung, Potentialausgleich sowie das getrennte Installieren von sich möglicherweise gegenseitig beinflussenden Leitungen.

Mit dem Einsatz einer Online-USV können weniger energiereiche Störungen bereits oft erfolgreich vom Server oder anderen aktiven Netzwerkkomponenten ferngehalten werden. Zusätzlich ist aber in jedem Fall ein leistungsfähiger Überspannungsschutz erforderlich, der Netzversorgung, Netzwerk und andere Kom-

munikationsleitungen (z.B. analoges Telefon, ISDN) umfasst. Dementsprechend sind, wie mit dem wirkungsvollen *Überspannungs-Schutzkreis* dargestellt, vor jedem dieser leitungsgebundenen Übergabepunkte Überspannungsableiter zu installieren.

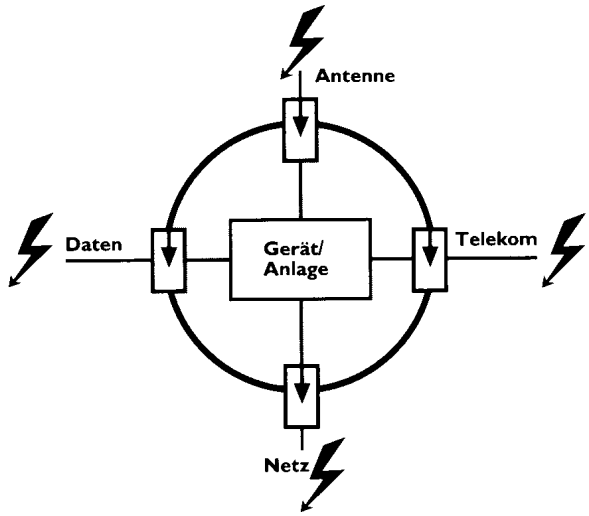


Abbildung 1-8 Überspannungsschutzkreis
(Quelle: Phoenic Contact, Blomberg)

Für die Stromversorgung reicht ein Schutzgerät allein unmittelbar vor der Einspeisung des Gerätes nicht aus, zusätzlich muss ein sogenanntes Blitz-Schutzzonenkonzept realisiert werden, das fester Bestandteil jeder Server-Energieversorgung sein sollte. Im Rahmen eines solchen Konzeptes, dass bereits Bestandteil der Planung der Energievorsorgung sein sollte, werden *Grob-, Mittel- und Feinschutzelemente* so in die Elektroinstallation eingebaut, dass sonst schädliche Überspannungsseffekte auf ein für den Server ungefährliches Maß reduziert werden.

1.3.11 Mechanische Absicherung

Server müssen vor unbefugtem Zugriff geschützt werden. Diese Forderung folgt nicht nur auf den bereits getroffenen Überlegungen zur Systemzuverlässigkeit, sondern ergeben sich auch aus den Regeln zur Datensicherheit.

Ein Server sollte nie ungeschützt unter dem Schreibtisch des Systemverwalters stehen, der Verschluß in einem 19«-Schrank stellt die absolute Minimalanforderung dar, um zumindest den Schutz vor mechanischem Zugriff zu gewährleisten. Die USV gehört mit in diesen Schrank hinein.

Noch besser geeignet sind zusätzlich abgesicherte Räume, die möglichst alle wichtigen aktiven und passiven Netzwerkkomponenten aufnehmen sollten. Damit dieser oft im Keller liegende Serverraum nicht zum Administratorarbeitsplatz wird, sollte frühzeitig an Möglichkeiten zur Fernsteuerung der Server und der dazu gehörenden Komponenten gedacht werden.

Zumindest die physikalischen Größen Temparatur und Luftfeutigkeit sollten im Serverraum überwacht werden, bei Bedarf ist eine Klimatisierung vorzusehen.

1.3.12 Schranküberwachung

Wichtige Umgebungsparameter sollten von einer *Schranküberwachung* erfasst, protokolliert und weitergemeldet werden. Dazu gehören z.B. die Überwachung der Umgebungstemperatur oder auch Alarmmeldungen bei Feuer, Wassereinbruch oder gewaltsamem Eindringen in den abgesicherten Bereich.

Schranküberwachungssysteme (*Rack Monitoring Systems*) werden zweckmäßigerweise als 19«-Komponente mit externen Sensoren ausgeführt und zusammen mit den übrigen Netzwerkkomponenten im Schaltschrank montiert. Fast alle Hersteller von 19«-Schranksystemen bieten solche Lösungen an; teilweise ist der aktuelle Zustand auch bereits über das Netzwerk abrufbar (z.B. über SNMP, *Simple Network Management Portocol*).

Einfache Schranküberwachungssysteme mit einer begrenzten Anzahl an Sensoren, die lediglich die Umgebungstemperatur und kontaktgesteuerte Ereignisse überwachen (Türkontakte, Bewegungsmelder, Rauchmelder, Chipkartenlesegerät), können auch direkt am Server betrieben werden. Ein solches System, das an einer freien parallelen Schnittstelle des Linux- oder NT-Servers angeschlossen betrieben wird, kann unter *www.samulat.de* bezogen werden.

1.3.13 Netzwerkverkabelung

Die Komplexität und die Anforderungen an die Leistungsfähigkeit der lokalen Netzwerke (LAN) steigen kontinuierlich. Neue Normen, vor allem die im Juli 1995 vom Europäische Komitee für Elektrotechnische Normung (CENELEC) verabschiedete europäische Norm *Strukturierte Verkabelung* DIN 50173 ermöglichen die Umsetzung allgemeingültiger, herstellerneutraler Konzepte in der Gebäudeverkabelung. Die so entstehenden Netzwerkverkabelungen sind gleichermaßen geeignet für Datennetzwerke und für analoge/digitale Telefonsysteme, sodass die Planung dieser Systeme nicht mehr getrennt erfolgen darf.

In modernen Netzwerken dürfen damit die sehr fehlerträchtigen Koaxialsysteme (Ethernet 10Base2) nur noch im Ausnahmefall eingesetzt werden. DIN 50173 definiert eine hierarchische, in drei Ebenen geteilte Kommunikations-Infrastruktur:

- Primärverkabelung (Gelände- oder Campusverkabelung, typischerweise die Verkabelung zwischen Gebäuden, ausgeführt mit Lichtwellenleitern)
- Sekundärverkabelung (Etagenverkabelung, Verbindung zwischen den in einem Gebäude stehenden Konzentratoren, ausgeführt mit Lichtwellenleitern oder Twisted-Pair-Kabeln)
- Tertiärverkabelung (Anschlussverkabelung, Verkabelung zwischen den Konzentratoren und den Endgeräte-Anschlussdosen, ausgeführt mit Twisted-Pair-Kabeln).

Die Norm definiert die Topologie und die übertragungstechnischen Kenndaten für ein offenes, d.h. herstellerneutrales *In-House*-Verkabelungssystem für Anwendungen der Telekommunikation und der Informationstechnik. Daraus leiten sich die Leistungsanforderungen an die gesamte Übertragungsstrecke sowie die benötigten Komponenten wie Kabel, Steckverbinder, Datendose und Patchfelder ab. Als Übertragungsmedium sind je nach Ebene symmetrische Kupferkabel (*TP, UTP, STP*) sowie Lichtwellenleiter zugelassen.

Abbildung 1-9 zeigt die typische Struktur eines nach DIN 50173 realisierten Netzwerkes, in dem die immer achtpolig ausgeführte Verkabelung für das PC-Netzwerk und auch für den Telefonanschluss genutzt wird.

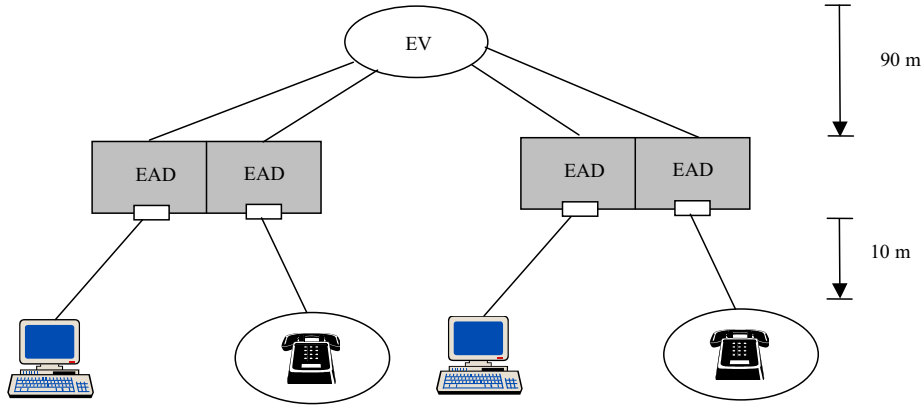


Abbildung 1-9 Aufbau der Verkabelung nach DIN 50173

Zentraler Punkt ist der Etagenverteiler *EV*, in dem alle von den Endgerätesanschlussdosen *EAD* kommenden, im Gebäude fest installierten Leitungen in einem *Patchfeld* enden. Der Etagenverteiler nimmt auch die aktiven Komponenten für das Netzwerk und für die Telefonverteilung auf, je nach aktuellem Bedarf werden die Vervindungen zwischen aktiven und passiven Netzwerkkomponenten über *Patchkabel* hergestellt.

Einige wichtige Eckdaten bzw. Forderungen :

- Die im Gebäude fest installierte Verkabelung zwischen EV und EAD sollte immer achtpolig ausgeführt werden, auch wenn zunächst nicht alle Leitungen vollständig genutzt werden. JedeVerbindung zwischen dem Patchfeld und einer EAD wird mit einem eigenen Kabel realisiert.
- Nach der Installation der Gebäude-Netzwerkverkabelung ist jede einzelne Verbindung meßtechnisch zu prüfen. Das Messprotokoll jeder Leitung ist Teil der zu erstellenden Dokumentation.
- Die gesamte Verkabelungsanlage muss so gekennzeichnet werden, dass keine Verwechslung der Anschlüsse möglich ist.
- Die maximale Entfernung zwischen zwei aktiven Netzwerkkomponenten darf 100 Meter nicht überschreiten. In diese Länge ist auch des Patchkabel im EV und das Verbindungskabels zwischen EAD und der Netzwerkkarte des PCs enthalten.
- In der Etagenverteilung werden alle aktiven und passiven Komponenten zentral geerdet. Alle Schirmungen werden hier aufgelegt, auch die Patchkabel sind geschirmt.
- Als Anschlusskabel zwischen EAD und PC können nicht geschirmte Patchkabel sinnvoll sein, um Erdschleifen in jedem Fall zu verhindern.

Es können alle bekannten Übertragungsverfahren für lokale Netzwerke bis zu 100 MBit/s (und höher) realisiert werden.

Weitere Informationen zu diesem Thema sind z.B. in [GlHL95] und [Gers95] enthalten.

2 Basiskonfiguration Linux-Server

Die bisher dargestellten Anforderungen an den Server und an die weiteren aktiven und passiven Netzwerkkomponenten stellen den Umfang einer typischen Servergrundkonfiguration dar. Je nach den speziellen Ressourcenforderungen im eigenen Netzwerk wird es zwar später notwendig sein, weitere Dienste einzurichten, der Grundumfang ist aber immer ähnlich.

An dieser Stelle soll auf die eigentliche Linux-Grundinstallation nicht eingegangen werden, denn sie ist bereits Inhalt vieler anderer Publikationen. Insbesondere die »Einsteigerliteratur« beschreibt immer wieder die gleichen Arbeitsschritte und wer einmal selbst eine Grundinstallation ausgeführt hat, weiß, dass diese Arbeiten mit den zu jeder Distribution gehörenden Tools schnell durchgeführt werden können.

Wichtig ist natürlich, was tatsächlich in der Grundinstallation eines Netzwerk-Servers an Besonderheiten zu berücksichtigen ist. Als Beispiel dazu wird hier die SuSE-Distribution verwendet; die vorgestellten Arbeitsschritte sind aber weitestgehend unabhängig von einer speziellen Distribution und können auch mit anderen Linux-Systemen nachvollzogen werden. Im Einzelfall wird es nötig sein, aktuelle Programmversionen über das Internet direkt zu beziehen. Dafür sind im Text dann die entsprechenden Bezugsquellen angegeben bzw. die Software ist auf der beiliegenden CD-ROM zu finden.

2.1 Grundinstallation

Die Server-Grundinstallation erfolgt zweckmäßigerweise mit einem Installationstool wie dem zur SuSE-Distribution gehörenden Programm YaST (*Yet another Setup Tool*). Die Installation wird dabei entweder über die mitgelieferte Bootdiskette oder über das Setup der ersten CD gestartet. Als Quellmedium dienen neben der CD-ROM auch NFS, FTP oder die Festplatte.

Die einzelnen Installationsschritte sollen hier nicht wiederholt werden, eine sehr gute Darstellung enthält z.B. das bei der SuSE-Distribution mitgelieferte Handbuch. Jedoch sollte bereits vor der Installation geprüft werden, ob die Treiber für die vom Server verwendete Hardware, insbesondere für Mainboard, Festplatten-Subsystem und Netzwerkarten, im Grundumfang enthalten sind.

In einigen Fällen ist es notwendig, zu Beginn der Installation die richtigen Treiber insbesondere für die Netzwerkkarten manuell auszuwählen. Als Bootmanager sollte LILO konfiguriert werden.

Im Gegensatz zu vielen anderen Distributionen installiert SuSE die *glibc* als */lib/libc.so*, was bei einer Reihe von Softwarepaketen für Ärger sorgt, da bei deren In-

stallation die entsprechenden Libraries nicht gefunden werden und der entsprechende Pfad bei Bedarf manuell geändert werden muss.

2.1.1 Partitionierung der Festplatte

Für die Installation eines kommerziellen Servers ist es in jedem Fall notwendig, die Partionierung der Festplatte schon während der Grundinstallation manuell vorzunehmen. Bei Festplatten mit mehr als 1024 Zylindern muss zumindest der Linux-Kernel innerhalb der ersten 1024 Zylinder liegen! Diese Restriktion ist keine Einschränkung des Betriebssystems Linux, sondern kommt aus dem BIOS.

Damit kann es günstig sein, eine kleine Start-Partition (20 MB sind bereits ausreichend) für das Verzeichnis `/boot` innerhalb dieser 1024 Zylinder anzulegen, das dann nur den Linux-Kernel enthält. Auf den weiteren Partitionen wird nun das Dateisystem von Linux *ext2fs* eingerichtet. Mindestens einer Partition muss dann der Mount-Point »/« für das Root-Filesystem zugeordnet werden. Das Root-Filesystem wird dann automatisch vom Kernel gemountet, alle anderen Dateisysteme können auch per Hand nachträglich gemountet werden:

Für die Planung der tatsächlichen Festplattenpartitionierung sollten die nachstehenden Überlegungen berücksichtigt werden:

- Jedes Betriebssystem benötigt ausreichend freie Festplattenkapazität, um überhaupt funktionieren zu können. Dieses gilt natürlich auch für eine Linux-Installation. Dieser freie Speicherplatz darf auch während des Betriebs nicht unter eine bestimmte Mindestgrenze von 100 MByte absinken! Damit ist es immer sehr gefährlich, die direkt zum Betriebssystem gehörenden Programme und Daten nicht in einer eigenen Partition zu speichern. Es muss in jedem Fall gewährleistet werden, dass die während des späteren Serverbetriebes routinemäßig auf dem Server abgelegten Daten die Betriebssystem-Partition auf keinen Fall »dichtmachen« können. Der Server sollte also mindestens zwei Partitionen (oder getrennte Festplatten) erhalten: eine für das Betriebssystem und eine für Daten. Bei Bedarf ist die oben schon beschriebene Startpartition zusätzlich notwendig.
- Hat das Betriebssystem eine eigene Partition, so ist im nächsten Schritt zu überlegen, ob noch weitere Unterteilungen notwendig sind. Eine wesentliche Rolle in diesen Überlegungen spielen die zu erwartenden Datenmengen, so z. B. auch, ob umfangreiche Druckaufträge über diesen Server abgearbeitet werden sollen oder ob ein E-Mail-System größere Datenmenge handhaben muss. Hier kann zwar in vielen Fällen bereits mit einer benutzerbezogenen Begrenzung der Speicherkapazitäten gearbeitet werden (*Quoting*), ein wirklich gute und vollständige Absicherung von Datenbeständen wird aber nur dann zu erreichen sein, wenn auch diese »Massendaten« in einem oder sogar mehreren getrennten Partitionen angeordnet werden. Im schlimmsten Fall ist dann nur ein

massendatenbezogener Dienst wegen Speicherplatzmangel temporär nicht mehr verfügbar, alle anderen Serverdienste laufen aber weiter.

Eine typische Serverkonfiguration umfaßt also immer mehrere Partitionen bzw. mehrere Festplattensysteme. Die tatsächlichen Partionsgrößen hängen dann von sehr vielen Netzwerk-spezifischen Faktoren ab. Eine beispielhafte Konfiguration auf Basis eines 10 GByte großen Festplattensystems zeigt Tabelle 2-1.

Mountpoint	Größe	Inhalt
/boot	20 MByte	Linux-Kernel
/	2,0 GByte	Betriebssystem, Programme, Auslagerungsdatei
/home	6,0 GByte	Benutzerverzeichnisse, bei Bedarf zusätzlich eine Mengenbegrenzung durch Quoting
/var	2,0 GByte	Druckwarteschlangen, Protokolle, E-Mail, ...

Tabelle 2-1 Beispiel zur Festplattenpartitionierung

Für jeden Benutzer sollten mindestens 100 MByte freier Speicherplatz eingerichtet werden. Wird eine Begrenzung durch Quoting realisiert, so müssen die Benutzer darüber informiert werden. Das Quoting sollte durch den Systemverwalter laufend überwacht werden.

Die Auslagerungsdatei (Swap-Space)

Auch ein Server mit sehr großem (> 512 MByte) Arbeitsspeicher braucht eine Auslagerungsdatei, die Swap-Datei. Unter Linux kann eine einzelne Swap-Datei oder -Partition bis zu 128 MByte groß sein. Wenn unter extremen Bedingungen mehr als 128 MByte Swap-Bereich notwendig sind, können bis zu 16 Dateien oder Partitionen angelegt werden.

Der in der Regel bereits während der Grundinstallation eingerichtete Swap-Bereich sollte typischerweise 64 oder 128 MByte umfassen.

Swap-Bereich einrichten

Besteht nach der Grundinstallation die Notwendigkeit, einen Swap-Bereich einzurichten oder zu vergrößern, so muss zunächst eine neue Datei oder Partition angelegt werden, die diesen Swap-Bereich später aufnehmen soll. Eine neue Partition kann z. B. mit dem später beschriebenen Shell-Befehl *fdisk* angelegt werden.

Für eine neue Swap-Datei wird eine neue Datei angelegt und so viele Bytes hineingeschrieben, wie die Swap-Datei groß sein soll. Eine einfache Methode hierfür ist der Shell-Befehl *dd* . Mit

```
dd if=/dev/zero of=/swap/sw2 bs=1024 count=8192
```

wird eine acht Megabytes große Swap-Datei */swap/sw2* angelegt. Es werden 8.192 Datenblöcke von */dev/zero* in die Datei */swap/sw2* geschrieben. (*/dev/zero* ist ein spezieller Gerätename, der bei Lesezugriffen immer Null-Bytes liefert-ähnlich wie */dev/null* .). Nach dem Anlegen der Datei wird mit

```
sync
```

das gesamte Dateisystem synchronisiert. Sobald eine Swap-Datei oder -Partition eingerichtet ist, wird diese mit dem Befehl

```
mkswap -c /swap/sw2 8102
```

formatiert. Der Befehl mkswap hat dabei das Format *mkswap -c Gerätename Größe, wobei* der *Gerätename* der Name der Swap-Partition oder -Datei und *Größe* die Größe des Swap-Bereichs in Blöcken ist. Der Schalter *-c* ist optional und bewirkt, dass der Swap-Bereich bei der Formatierung auf fehlerhafte Blöcke untersucht wird. Soll eine Swap-Partition formatiert werden, so ist als *Gerätename* der Name des Special Files anzugeben, z. B. */dev/hda3*.

Nach dem Einrichten einer Swap-Datei mit *mkswap* sollte immer nochmals *sync* aufgerufen werden, damit die Formatierungsinformationen auf jeden Fall physikalisch in die neue Swap-Datei geschrieben werden. Nach dem Formatieren einer Swap-Partition ist der Aufruf von *sync* in der Regel nicht notwendig.

Swap-Bereich initialisieren

Damit ein neuer Swap-Bereich auch genutzt werden kann, muss dieser mit dem Befehl *swapon* initialisiert werden. Nach dem Einrichten der oben gezeigten Swap-Datei */swap/sw2* und dem Aufruf von *mkswap* und *sync* wird mit

```
swapon /swap/sw2
```

der ursprüngliche Swap-Bereich um die hier angegebene Datei erweitert. Mit dem Shell-Befehl *free* läßt sich das überprüfen. Eine neue Swap-Partition wird mit

```
swapon /dev/hda3
```

initialisiert, als Parameter wird wieder das dazu gehörende Special File angegeben.

Ähnlich wie Dateisysteme werden auch Swap-Bereiche beim Booten initialisiert. Dies geschieht mit dem Befehl *swapon -a, der* in einer der Startdateien enthalten ist. Dieser Befehl liest die Konfigurationsdatei */etc/fstab*, die Informationen zu den Dateisystemen und Swap-Bereichen enthält. Alle Einträge in */etc/fstab* mit dem Wert *sw* im Feld *Optionen* werden mit *swapon -a* initialisiert. Enthält */etc/fstab* die Einträge

```
# device     directory    type   options
/dev/hda3    none         swap   sw
/swap/sw2    none         swap   sw
```

werden die beiden Swap-Bereiche */dev/hda3* und */swap/sw2* beim Booten initialisiert. Für jeden neu angelegten Swap-Bereich muss in */etc/fstab* ein Eintrag manuell hinzugefügt werden.

Swap-Bereich entfernen

Ein bestehender Swap-Bereich wird

```
swapoff gerätename
```

entfernt, wobei *Gerätename* den Namen der zu entfernenden Swap-Partition oder-Datei angibt. Soll eine Swap-Datei endgültig gelöscht werden, so muss die Datei zusätzlich mit dem Befehl *rm* entfernt werden (nachdem *swapoff* aufgerufen wurde).

Sobald Sie eine Swap-Partition mit *swapoff* deaktiviert wurde, kann diese wie jede andere Partition genutzt werden. Die entsprechenden Einträge in */etc/fstab* müssen abschließend manuell entfernt werden, es kommt sonst beim nächsten Bootvorgang zu Fehlermeldungen.

Zugriffsgeschwindigkeit und Zuverlässigkeit

Die bereits angestellten Überlegungen zur Optimierung der Zugriffsgeschwindigkeit und zur Zuverlässigkeit der eingesetzten Festplatten sollten auch in die Planung der tatsächlichen Verzeichnisstruktur einfließen. Es kann sinnvoll sein, ausgewählte Datenbestände mit hohen Zuverlässigkeitsanfordeungen auf entsprechend ausgelegte Datenträger zu plazieren, während andere Verzeichnisse entweder nur hohe Zugriffsgeschwindigkeiten erfordern oder keine speziellen Anforderungen stellen.

Aus wirtschaftlichen Gründen wird es nur selten ein für alle Anforderungen optimal geeignetes RAID-System geben. Je nach Struktur und Sensibilität der Daten kann es erforderlich sein, mehrere Festplattensysteme einzubauen. Dazu ein paar Überlegungen zu den eben vorgestellten Verzeichnissen:

Swapspace	muss nur schnell sein, Redundanz ist hier überflüssig. Optimal ist hier ein System mit RAID Level 0, aufgebaut aus einem Satz schneller und kleiner Platten.
/var/spool	Auf diesen Bereich wird häufig zugegriffen (z.B. */var/spool/news* oder */var/spool/proxy*). Er enthält nur mäßig wichtige Daten (z.B. */var/spool/mail*). Ihr Verlust wäre zwar schmerzlich, jedoch zu verkraften. Je nach tatsächlichen Anforderungen bietet sich RAID Level 1 oder RAID Level 5 an.
/usr	Hier liegen viele Programme und Daten, auf die schnell zugegriffen werden muss. Auch dieser Bereich ist meist vollkommen wiederherstellbar, wenn auch nur sehr zeitaufwändig. Auch hier bieten sich ebenfalls RAID Level 0 oder RAID Level 5 an.
/home	Ist der wohl wichtigste Bereich in einem Linux-System. Es enthält fast ausschließlich nicht sofort und vollständig rekonstruierbare Daten. Daher sollte hier mindestens RAID Level 1, besser noch die Verwendung von RAID Level 5 vorgesehen werden.

Grundsätzlich sollte bei jeder Serverplanung der Einsatz von RAID Level 0 und RAID Level 1 eingeplant werden, um zumindest die Minimalanforderungen zu erfüllen. Ein Server mit einem schnellen SCSI-System, in dem nicht einmal die wichtigsten Datenbestände mindestens über RAID Level 1 gesichert sind, ist für den kommerziellen Einsatz unbrauchbar. Ebenso wie bei Systemen unter Novell (SFT II) oder Windows NT (Spiegelsatz aus zwei Festplatten) muss der Systemverwalter hier die entsprechende Redundanz einbauen.

2.1.2 TCP/IP-Grundkonfiguration

Linux unterstützt mittlerweile die meisten Netzwerkkarten *(Ethernet, Arcnet, TokenRing, ...)* und kennt fast alle gängigen Netzwerkprotokolle, wie z.B. TPC/IP, IPX/SPX oder AppleTalk. In den meisten Fällen wird die Anbindung an das LAN über eine *Ethernet*-Netzwerkkarte und dem Protokoll *TCP/IP* durchgeführt. Praktisch die gesamte Netzwerk-Grundkonfiguration kann mit YaST durchgeführt werden, wobei sich die nachfolgende Darstellung zunächst nur auf die grundlegenden Arbeiten beschränkt. Im Rahmen der später erfolgenden weiteren Serverkonfiguration wird auch auf Details eingegangen.

Installation der Netzwerkkarte

Das Einbinden der ersten Netzwerkkarte erfolgt im Rahmen der Grundkonfiguration. Bei vielen Karten ist dies sogar über eine automatische Erkennung möglich, so dass keine weiteren Arbeiten erforderlich sind.

Bei Start des Servers werden Typ und MAC-Adresse der Netzwerkkarte angezeigt.

IP-Adresse vergeben

Im Rahmen der TCP/IP-Grundkonfiguration muss der Linux-Server eine im Netzwerk eindeutige IP-Adresse erhalten. Die dazu notwendigen Angaben müssen manuell erfolgen. Diese Vergabe der IP-Adressdaten ist zwar grundsätzlich auch über DHCP möglich, dies sollte allerdings bei einem Server auf keinen Fall gemacht werden. Folgende Angaben sind notwendig:

Rechnername	ein im gesamten lokalen Netzwerk eindeutiger, maximal acht Zeichen langer Name (*hostname*), z.B. *linux01*
Domänenname	Ist der Name der *Domäne* (Domain), der dieser Rechner angehören wird. Der Name ist zunächst frei wählbar; ist bereits ein echter Internet-Domänenname zugewiesen worden, wird dieser verwendet. Sollen alle Rechner z.B. in der eigenen Domäne *samulat.de* betrieben werden, so wird diese Domäne hier angegeben. Der Rechner mit dem Namen *linux01* wäre dann unter dem vollständigen Namen *linux01.samulat.de* eindeutig im Netzwerk identifiziert.

IP-Adresse	Ist die eindeutige IP-Adresse im Netzwerk. Für rein private Netzwerke sind durch RFC 1597 drei Adressbereiche definiert, bei denen sichergestellt ist, dass selbst bei einer versehentlich aufgebauten Internetverbindung keine Probleme entstehen können, da diese zwischen Internet-Systemen nicht geroutet werden. Die zugelassenen Adressbereiche sind: 10.0.0.1 bis 10.255.255.254 (Class-A-Netz) 172.16.0.1 bis 172.31.255.254 (Class-B-Netz) 192.168.0.1 bis 192.168.255.254 (Class-C-Netz) Zweckmäßigerweise wird in der Grundkonfiguration eine eindeutige Adresse aus dem C-Netz zugewiesen; in dem hier beschreibenen Beispiel erhält der Server *linux01* die IP-Adresse 192.168.100.1.
Subnet-Mask	Die Netzwerkmaske (Subnet Mask) gibt an, wo innerhalb der IP-Adresse die Trennung zwischen dem Teil *network* und *host* liegt. In der Grundkonfiguration, in der eine Adresse aus dem C-Netz eingetragen wird, ist die Subnet Mask immer 255.255.255.0, d.h. die vierte Stelle der IP-Adresse ist der *host*-Anteil.

Die Zuweisung einer Gateway-Adresse ist an dieser Stelle noch nicht notwendig. Mit YaST kann die TCP/IP-Grundkonfiguration jetzt vorgenommen werden:

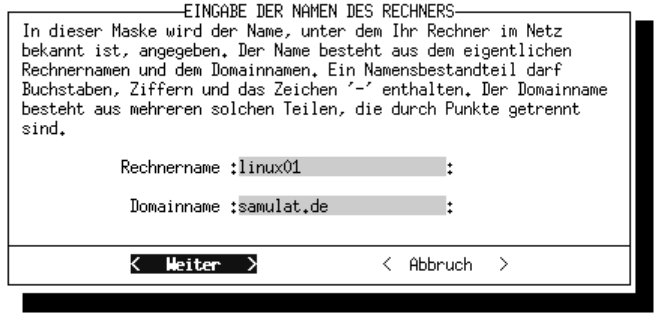


Abbildung 2-1 Rechnernamen mit YaST zuweisen

Der Server erhält den voll qualifizierten Namen *linux01.samulat.de*, der sich aus dem Rechnernamen, dem Domänennamen *samulat* und der Top-Level-Domain *de* zusammensetzt.

Alle Parameter werden in */etc/rc.config* gespeichert. Werden nachträglich Änderungen in der TCP/IP-Grundkonfiguration vorgenommen, so muss der Server abschließend neu gestartet werden. Alternativ können mit

```
rcnetwork restart
```

nur die betroffenen Netzwerkdienste neu initialisiert werden.

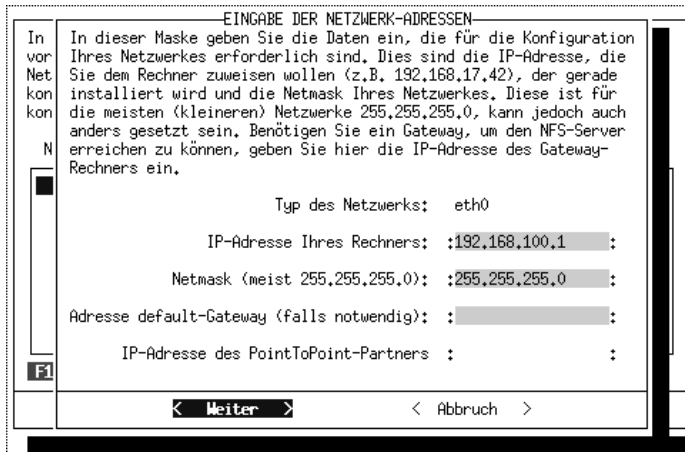


Abbildung 2-2 IP-Adressvergabe mit YaST

Prüfung der Konfiguration

Ob die Netzwerkkarte korrekt erkannt und die TCP/IP-Grundkonfiguration richtig ausgeführt wurde, mit dem Shell-Befehl *ifconfig* überprüft wird (Listing 2-1).

```
# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:80:48:C5:D2:92
          inet addr:192.168.100.1  Bcast:192.168.100.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:311876 errors:0 dropped:0 overruns:0 frame:0
          TX packets:295715 errors:0 dropped:0 overruns:0 carrier:0
          collisions:20 txqueuelen:100
          Interrupt:11 Base address:0xb800

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:3924  Metric:1
          RX packets:1374 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1374 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
```

Listing 2-1 Anzeige der TCP/IP-Grundkonfiguration mit ifconfig

Die mit *ifconfig* erstellte Liste sollte mindestens einen Eintrag für *eth0* enthalten. Der Eintrag enthält die Details der Hardwarekonfiguration (I/O-Adresse und verwendeter Interrupt), *HWaddr* ist die immer zwölfstellige MAC-Adresse der Netzwerkkarte.

Das zusätzlich immer vorhandene Interface *Local Loopback* (lo) spielt eine besondere Rolle: Es ermöglicht die Verwendung des Netzwerkprotokolls für lokale Dienste, dient also der rechnerinternen Kommunikation. Als IP-Adresse ist immer 127.0.0.1 angegeben.

Der Test des TCP/IP-Protokollstacks erfolgt mit dem Shell-Befehl *ping*, wobei zunächst mit

```
# ping 127.0.0.1
PING 127.0.0.1 (127.0.0.1): 56 data bytes
64 bytes from 127.0.0.1: icmp_seq=0 ttl=255 time=0.237 ms
64 bytes from 127.0.0.1: icmp_seq=1 ttl=255 time=0.081 ms

--- 127.0.0.1 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 0.081/0.159/0.237 ms
```

die grundlegenden Funktionen überprüft werden. Für diesen Schritt wird das immer vorhandene *Local Loopback* 127.0.0.1 als Adresse angegeben; die eigentliche Netzwerkkarte spielt an dieser Stelle noch keine Rolle Der unter Linux verfügbare Befehl *ping* wird nicht nach vier Testläufen, wie z. B. bei Windows-Systemen üblich, automatisch nach vier Durchläufen abgebrochen, sondern läuft endlos. Er muss mit der Tastenkombination Strg C beendet werden. Mit

```
ping 192.168.0.1
```

wird jetzt geprüft, ob bei Angabe der eigenen IP-Adresse ebenfalls eine Reaktion erfolgt. Ist auch dieser Test erfolgreich, kann die Verbindung zu einem *Remote-Host*, also zu einem im Netzwerk bereits verfügbaren entfernten Rechnern unter TCP/IP geprüft werden.

2.1.3 X-Windows-Grundkonfiguration

Das grafische X-Windows-System dient dem Ziel, eine intuitiv bedienbare Oberfläche für UNIX-Anwendungen zur Verfügung zu stellen, unabhängig von der Grafikhardware, Netzwerkauslegung und eventuell sonst noch einbezogenen Betriebssystemen. XWindows wurde von einem Konsortium unter Federführung des MIT entwickelt und steht mittlerweile in der Version X11R6.1 zur Verfügung. Mit Xfree86 ist eine Implementierung für PC-Unix verfügbar, die heute in allen gängigen Linux-Distributionen enthalten ist. Da heute eigentlich alle Netzwerkbetriebssysteme über grafische Oberflächen administriert werden können, sollte die Installation und Konfiguration von X11 in jedem Fall als Teil der Grundkonfiguration angesehen und bereits jetzt durchgeführt werden.

Bei Windows 9x/NT bilden das eigentliche Betriebssystem und die grafische Bedieneroberfläche eine auf den PC begrenzte Einheit. Kann die grafische Oberfläche nicht mehr bedient werden, so kann das gesamte System allein deswegen ausfallen, weil keine Eingaben mehr vorgenommen werden können. Die Idee unter XWindows weicht davon ab: Diese grafische Betriebssystemerweiterung läuft zwar auf Basis des Rechnerbetriebssystems, bildet aber ansonsten eine davon weitgehend unabhängige Client-Server-Struktur. Damit sind Ideen realisierbar, die

unter Windows 9x/NT nur sehr aufwändig und mit Zusatzsoftware realisiert werden können: Der Grafik-Server auf dem einen PC arbeitet im Netzwerk mit einem oder mehreren Grafik-Clients zusammen. Damit können nicht nur die Arbeitsergebnisse einer Anwendung auf dem Grafik-Server auf beliebigen Client-PCs grafisch präsentiert werden, auch die Ideen zur Fernbedienung von Server-Prozessen können dabei anders und im Vergleich zu Windows9x/NT wesentlich einfacher realisiert werden.

X11 ist ein Netzwerkprotokoll für grafische Informationen, das Programm X11 heißt *X-Server* und setzt das Netzwerkprotokoll in ein Bild auf dem Monitor um; dazu bedient sich das X11 einer freien virtuellen Konsole (ohne *login*-prompt). Der Monitor erhält eine Netzwerkadresse; dazu wird an den Maschinennamen ein *:0.0* angehängt, z. B. *linux01:0.0* oder *linux01.samulat.de:0.0*. Mit der Umgebungsvariablen *DISPLAY* wird Programmen mitgeteilt, welchen Monitor sie benutzen sollen.

Das reine X11 stellt die grafischen Informationen lediglich dar, verwaltet die Fenster aber nicht im eigentlichen Sinne Die Verwaltung der Fenster, deren Gestaltung und Aktionen werden vom *Windowmanager* (Fenstermanager) gesteuert. Virtuelle Fenstermanager vergrössern die nutzbare Oberfläche; die eigentliche Bildschirmgröße legt den maximal sichtbaren Ausschnitt fest. Zum Standard-Lieferumfang der meisten Linux-Distributionen gehören mehrere Fenstermanager:

fvwm2	Der *Free Virtual Window Manager* ist der Standard unter Linux. Er ist klein, schnell und vielseitig (*mwm*-Emulation, *twm*-Emulation); die Konfigurationsdateien sind *~/.fvwmrc* (privat) und */etc/X11/fvwm/system.fvwmrc* (global) *.
fvwm95	ähnlich *fwm2*, aber der Bedieneroberfläche von Windows 95 nachempfunden
mvm	Der *Motif Window Manager* ist ein kommerzielles Produkt und deutlich größer und langsamer als *fvwm*.
kde	*K Desktop Environment*, die moderne und immer weiter Anwendung findende Alternative zu *fvwm2*.

Die meisten Fenstermanager legen ihre Konfigurationsdateien und die dazu gehörenden Daten in den Unterverzeichnissen */usr/X11R6/lib/X11* ab.

Grundkonfiguration über sax

Die Grundkonfiguration von X11 kann schnell und unkompliziert mit dem Programm *sax* erfolgen. Mit

```
sax
```

oder über die entsprechende Funktionen im Menü *Administration* von YaST wird die Konfiguration gestartet.

Start des Fenstermanagers

Der Start des Fenstermanagers erfolgt manuell mit dem Shell-Befehl

```
startx fvwm95
```

wobei der optionale Parameter (im Beispiel *fvwm95*) den zu verwendenden Fenstermanager angibt. Als weiterer Parameter kann z. B. die Farbtiefe vorgegeben werden. Mit

```
startx fvwm95 -- -bpp 16
```

startet das X-Windows-System mit einer Farbtiefe von 16 Bit. Der standardmäßig zu verwendende Fenstermanger wird über die Variable *WINDOWMANGER* festgelegt:

```
# ECHO $WINDOWMANAGER
/usr/X11R6/bin/kde
```

Soll diese Zuordnung bei Bedarf für bestimmte Benutzer geändert werden, so sollte in der Datei *~/.bashrc* der Eintrag

```
export WINDOWMANAGER=kde
```

ergänzt werden. Für diesen Benutzer ist dann der *kde* der Standard-Fenstermanager.

Abbildung 2-3 zeigt die grafische Oberfläche des Fenstermanagers KDE. Diese Oberfläche hat in kurzer Zeit eine sehr weite Verbreitung gefunden; sie ist durch die klar strukturierte und einfache Bedienung auch sehr gut für den Umsteiger aus der Windows-Welt geeignet. KDE wird für den hier vorgestellten Linux-Server als Standard-Arbeitsoberfläche eingesetzt.

KDE ist vollständig URL-basiert, so dass alle Pfadangaben und Verweise auf Dateien in einem einheitlichen Format verarbeitet werden. Viele Aktionen können per *Drag & Drop* gesteuert werden; das in HTML verfügbare und leistungsfähige KDE-Hilfesystem stellt umfangreiche Dokumentationen zur Verfügung.

Grafisches Login

Soll an dem Linux-Server ausschließlich unter X11 gearbeitet werden, so sollte auch der Anmeldevorgang direkt unter *KDE* erfolgen. Mit Hilfe von YAST kann dies unter *Administration des Systems -> Login-Konfiguration* eingestellt werden. Soll die grafische Anmeldung erfolgen, kann dies entweder mit dem einfachen *xdm* (*X Display Manager*) oder mit dem leistungsfähigerem *kdm* (*KDE Display Manager*) erfolgen.

kdm bietet die Möglichkeit, Berechtigungen zuzuweisen. Festgelegt werden kann, welche Benutzer diesen Rechner neu starten oder herunterfahren dürfen.

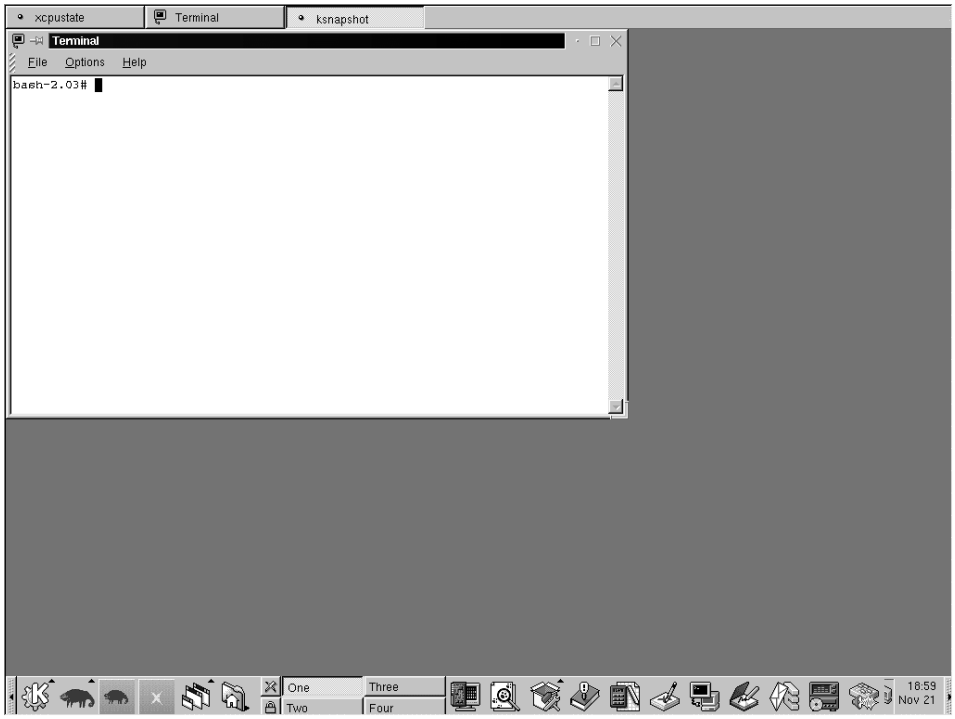


Abbildung 2-3 Die grafische Oberfläche des Fenstermanagers kde

Über YaST kann über den Menüpunkt *Administration -> Login-Konfiguration* festgelegt werden, dass die Anmeldung bereits über die grafische Oberfläche erfolgt (Abbildung 2-4).

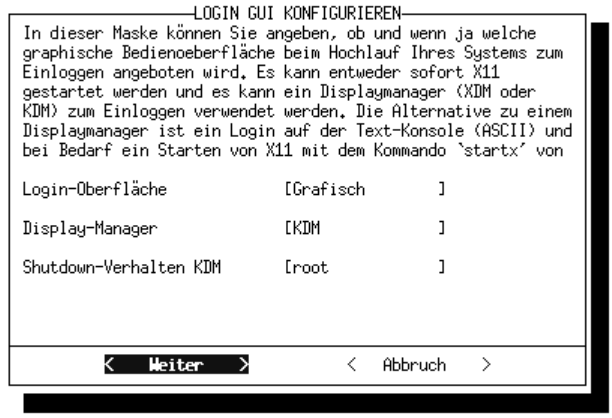


Abbildung 2-4 Die Anmeldung soll über eine grafische Oberfläche erfolgen

Im Fenstermanager *kdm* (Abbildung 2-5) können über Schaltflächen der zu startende Fenstermanager (*Session Type*) und die Menüsprache (*Language*) ausgewählt werden. Bei Bedarf wird der Rechner über den Schalter *Shutdown* heruntergefahren.

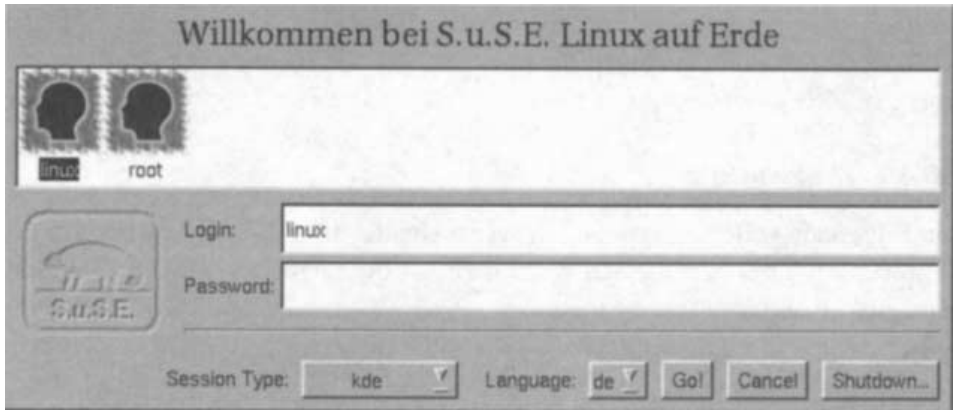


Abbildung 2-5 Anmeldung über den Fenstermanager kdm

Die hier getroffene Sprachauswahl gilt nur für den Login-Manager und hat keine Auswirkung auf die später gestartete Sitzung.

Unter KDE kann der *kdm* vom Systemverwalter *root* unter *Anwendungen -> Login-Manager* konfiguriert werden (Abbildung 2-6).

Verändert werden können: Aussehen, Begrüßungstext beim Login, Logo, Windows- oder Motiv-Stil, Schriftarten, Sprachauswahl. Unter *users* wird festgelegt, welche im System bekannten Benutzer während der Anmeldung ausgewählt werden können.

Grundkonfiguration KDE

Das KDE-Projekt (*www.kde.org*) wurde im 1996 vom Lynx-Entwickler Matthias Ettrich ins Leben gerufen, um Linux eine moderne, einfach zu bedienende und freie Benutzeroberfläche zu geben. KDE soll keine Kopie bekannter Oberflächen darstellen, sondern versucht, bewährte Attribute zu integrieren. So kennt KDE die Vorlagen-Funktion von OS/2 (Templates); die Startleiste (K-Panel) und die Gestaltung der Fenster orientieren sich an Windows-Systemen. Drag & Drop ist durchgängig möglich.

KDE funktioniert nicht nur unter Linux, es ist auch für viele Unix-Versionen, wie *FreeBSD*, *IRIX*, *HP-UX* und viele andere verfügbar.

Bei SuSE-Linux wird KDE als Standard-Fenstermanager installiert, die Standardeinstellungen sind für eine 1024er Auflösung gedacht. Wird KDE erstmalig gestartet, so werden, nach Bestätigung mit »Yes« zunächst die Standard-Templates installiert.

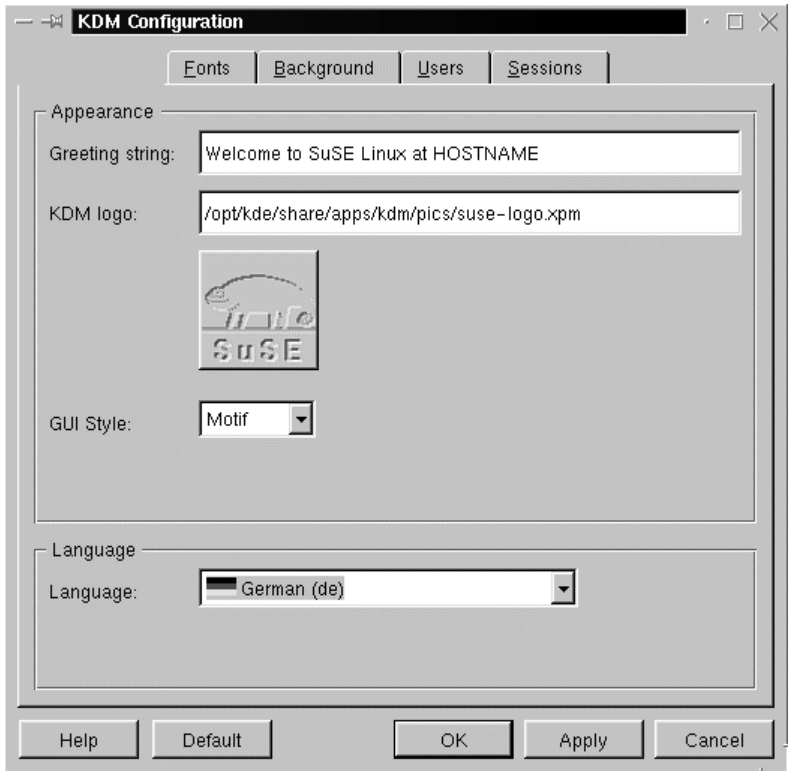


Abbildung 2-6 KDM-Konfiguration

Der KDE Bildschirm (Abbildung 2-3) ist in drei Bereiche unterteilt:

- Der eigentliche Desktop, der zunächst nur drei Symbole (Icons) enthält: den Papierkorb, den Vorlagen- und den Autostart-Ordner. Standardmäßig ist ein *xterm*-Fenster geöffnet, an dem Shell-Befehle direkt angegeben werden können.
- Die Taskleiste am oberen Bildschirmrand, sie zeigt die aktuell ausgeführten X-Anwendungen. Mit einem Klick auf den Namen wird das Programm in den »Vordergrund« geholt.
- Das K-Panel (K-Leiste) am unteren Bildschirmrand wird vor allem zum Start von Programmen verwendet.

Das K-Panel (Abbildung 2-7) enthält viele Buttons, die jeweils ein Programm oder eine Menüstruktur enthalten. Menüstrukturen sind an den kleinen aufwärts gerichteten Pfeilen zu erkennen, die einzelnen Buttons können mit einem rechten Mausklick verschoben, gelöscht oder konfiguriert werden.

Abbildung 2-7 K-Panel

In der Mitte des K-Panels sind vier Auswahl-Buttons für virtuelle Bildschirme, die jeweils einzeln konfiguriert werden können. Durch einen Klick auf einen bereits aktivierten Bildschirm kann der angezeigte Name geändert werden.

Zur schnellen Ausführung von Linux-Befehlen stellt KDE eine hilfreiche Funktion zur Verfügung, die an das unter Windows bekannte *Start -> Ausführen* angelehnt ist: Die Tastenkombination Alt F2 öffnet ein kleines Eingabefenster, in dem ein Shell-Befehl direkt eingegeben werden kann.

Die Standardsprache von KDE ist Englisch, bei Bedarf sollte also zuerst diese Einstellung geändert werden. Dies ist möglich über *K-Menü -> Settings -> Desktop -> Language*.

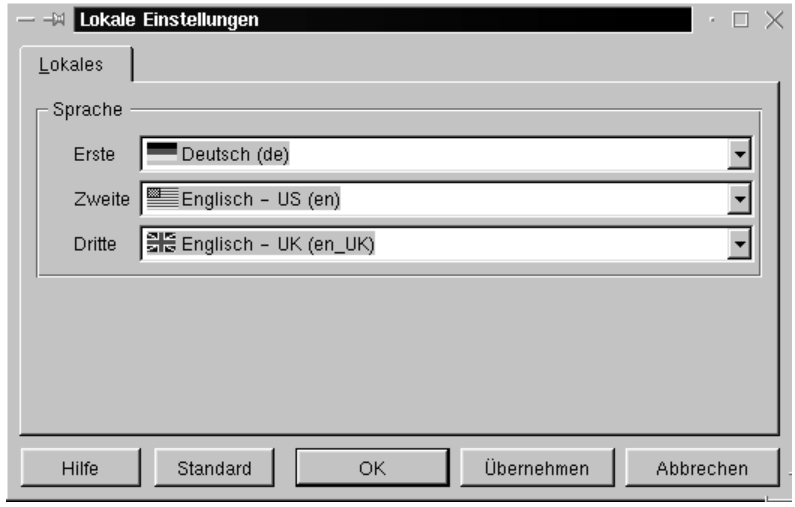


Abbildung 2-8 Grundkonfiguration KDE: Sprachauswahl

Die Sprachauswahl sollte gemäß Abbildung 2-8 als erste Sprache Deutsch, danach zwei weitere Möglichkeiten enthalten. Die zweite oder dritte Sprache wird nur dann verwendet, wenn die vorhergehende Auswahl nicht unterstützt wird. Nach dem nächsten Start von KDE wird die Änderung der Sprachauswahl wirksam.

Soll KDE beendet werden, so erfolgt dies über *K-Menü -> Logout.*

KDE Hilfe – khelp

Alle KDE-Programme verfügen über eine vollständig in HTML geschriebene Hilfe. Die Hilfeseiten werden mit dem Programm *khelp* angezeigt, schnell erreichbar z. B.

über *K-Panel -> KDE Hilfe*. Zusätzlich zu den KDE-Hilfeseiten kann *khelp* auch die *Linux Man-Pages* und *info-Seiten* anzeigen (Aufruf mit *man(Index)* oder *info(dir)*) (siehe Abbildung 2-9).

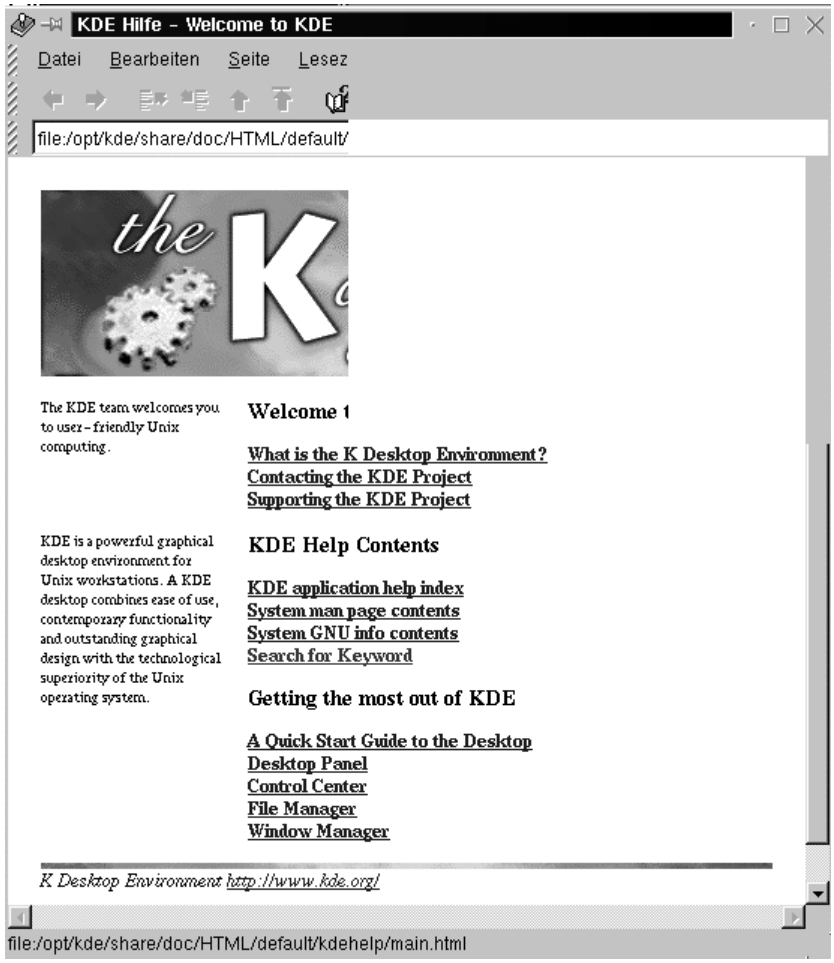


Abbildung 2-9 KDE-Hilfe

2.1.4 Einsatz von USV-Anlagen

Ein Netzausfall oder Spannungsschwankungen stellen für jeden Server eine große Gefahr dar. Mit dem Einsatz einer entsprechend dimensionierten USV kann diesen Gefahren im Wesentlichen begegnet werden. Wichtig ist, dass ein Gerät eingesetzt wird, dass über eine Kommunikationsverbindung und die auf dem Server installierte Software den Server nach Ablauf der USV-Pufferzeit rechtzeitig den Befehl zum zwangsweisen Herunterfahren geben kann.

Die Kommuniaktionsverbindung wird in der Regel über eine der seriellen Schnittstellen realisiert, wobei entweder nur die von der USV erzeugten Schaltzustände an den Server weitergegeben werden oder sogar eine echte Datenverbindung realisiert wird.

Auf dem Markt sind zwischenzeitlich eine Reihe von USV-Systemen verfügbar, die speziell für den Betrieb unter Linux geeignet sind. Zwei dieser Geräte, ihre Installation und das Betriebsverhalten sollen im Folgenden vorgestellt werden.

Beispiel 1: APC

Das nachfolgende Beispiel zeigt die Installation und den Betrieb eines Servers mit einer USV vom Typ APC Smart-UPS V/S 650 (Abbildung 2-10).

Abbildung 2-10 APC Smart UPS

Es handelt sich bei diesem Gerät um eine preiswerte, unterbrechende USV, die über ein serielles Kabel mit einem Server kommunizieren kann. Ein entsprechendes RS232-Kabel ist zwar im Lieferumfang enthalten, kann aber leider für den Linux-Server nicht verwendet werden. Für Unix- und Linux-Systeme muss ein Spezialkabel mit der Bestellbezeichnung

APC-Nr 740-0023A (Unix Signaling Cable), BestNr. AP9823

beschafft werden. Dieses Kabel verbindet dann eine beliebige serielle Schnittstelle des Linux-Servers mit der USV, für die beschriebene Installation wird COM1 verwendet.

Die zur Installation notwendige Software *Powerchute Plus v4.5.2 for linux* (5 MByte) kann direkt über *www.apcc.com* bezogen werden (In der CD-ROM finden Sie diese Software unter *usv/apc/pc452_libc.tar*). Zweckmäßigerweise wird diese Datei zunächst in ein Verzeichnis auf dem Linux-Server kopiert, im Beispiel wird dazu */apc_sw* neu angelegt:

```
# md /apc_sw
# cp pc452_libc.tar /apc_sw
# cd /apc_sw
```

Die Datei *pc452_libc.tar* wird dann mit

```
# tar xvf pc452_libc.tar
```

entpackt, der Inhalt von */apc_sw* sollte danach so aussehen:

```
# ls -l
total 10867
drwxr-xr-x   2 root   root     1024 Nov 22 20:47 .
drwxr-xr-x  24 root   root     1024 Nov 22 20:46 ..
-rw-r--r--   1 root   root   604160 Oct  5 21:56 BI_LINUX
-rw-r--r--   1 root   root    51200 Oct  5 21:56 CI_LINUX
-rw-r--r--   1 root   root    40960 Oct  5 21:56 COMMON
-rw-r--r--   1 root   root  3389440 Oct  5 21:56 FI_LINUX
-rw-r-----   1 root   root  1351680 Oct  5 21:56 HELP
-rwxr-x--x   1 root   root    95574 Oct  5 21:56 INSTALL
-rwxr-----   1 root   root  5539840 Nov 22 20:46 pc452_libc.tar
```

Die eigentliche Installation wird jetzt mit

```
# ./INSTALL
```

gestartet. Alle weiteren Arbeitsschritte erfolgen dann menügesteuert:

```
-------------------------------------------------------------
      PowerChute Plus for Unix v4.5.2 Installation Script
          Copyright American Power Conversion 1999
-------------------------------------------------------------
If you quit this script at any time before committing your choices, the
installation will not take place, and no modifications will be made to your
currently installed PowerChute products

         1) CD-ROM
         2) Floppy
         3) Tape

Select the media type from which you will install: [?] 1
Enter path to mounted CD ROM [/apc_sw]
Would you like to see an overview of the Installation?[y/n,q] n
-------------------------------------------------------------
The User Interface module of this product is capable of monitoring any system
on your network running the Daemon module of PowerChute Plus for Unix.
For this reason, you may feel it is only appropriate to install the entire
distribution on certain systems.
-------------------------------------------------------------
```

```
NOTE: In order to perform any sort of UPS monitoring, including shutdown due to
power failure, you must install at least the Daemon module of the product!
-------------------------------------------------------------
      1) User Interface Module Only
      2) Daemon Module Required for UPS Monitoring Only
      3) Both the User Interface and Daemon Modules

Which Parts of PowerChute Plus for Unix do you wish to install? [?] 3

               1) Matrix-UPS
               2) Smart-UPS
               3) Back-UPS
               4) Back-UPS Pro
               5) Symmetra Power Array
               6) Smart-UPS DP

Which APC Hardware will PowerChute Plus for Unix be running with [?] 2
-------------------------------------------------------------
The Measure-UPS is a device which is designed to perform environmental
monitoring in conjunction with PowerChute Plus for Unix
-------------------------------------------------------------

Do you currently have a Measure-UPS attached to the UPS? [y/n,q] n
               1)  Solaris 2.X for SPARC
               2)  SunOS 4.X (Solaris 1.X) for SPARC
               3)  Solaris 2.X for Intel
               4)  AIX for IBM RS6000
               5)  HPUX 10.X/11.X  (700/800 series)
               6)  SCO Unix/Unixware 7.X
               7)  NCR (AT&T UNIX)
               8)  SGI Irix
               9)  UnixWare 2.X
               10) Olivetti Unix Sys V Rel.4.0
               11) Siemens Unix (RISC)
               12) Unisys Unix Sys V Rel.4.0
               13) DEC OSF/1
               14) Linux

On which Operating System are you installing? [?] 14
-------------------------------------------------------------
PowerChute Plus for Unix is able to monitor other hosts.  However, in order to
monitor other hosts TCP/IP must be installed. If you do not have TCP/IP
installed, answer 'n' to the following question.
-------------------------------------------------------------

Do you currently have TCP/IP Installed? [y/n,q] y
-------------------------------------------------------------
If you will be using the Motif version of the User Interface on a monochrome
monitor, using the Monochrome Coloring scheme is recommended.
```

```
-------------------------------------------------------------
      1) Use Default Color scheme
      2) Use Monochrome Color scheme

Which color scheme do you wish to use [1]? 1
-------------------------------------------------------------
PowerChute Plus for Unix requires complete control of the serial port. No
processes, including gettys, are allowed to be accessing the port. Therefore,
the serial port you select must NOT be enabled for logins. To ensure that
PowerChute Plus for Unix has control of the serial port, make sure that it is
not enabled for logins.  To disable the port for logins consult the PowerChute
Plus for Unix manual.
-------------------------------------------------------------
      1) /dev/ttyS0
      2) /dev/ttyS1
      3) Other

Which serial device will be dedicated to PowerChute Plus for Unix [?] 1
-------------------------------------------------------------
You should have the black cable, #940-0024C attached to /dev/ttyS0
Please verify.
-------------------------------------------------------------
pcp_unix.apc, version 4.5.1 is already installed in:
/usr/lib/powerchute
-------------------------------------------------------------
Where do you wish to install PowerChute Plus for Unix?
[/usr/lib/powerchute] /usr/lib/powerchute
-------------------------------------------------------------
Command files may be executed with root privileges or with the privileges you
assign to the pwrchute account (allowing you to customize command file
execution according to your system requirements).
-------------------------------------------------------------
Do you want to execute command files as root? [y/n,q] y
-------------------------------------------------------------
E-mail may be sent with root privileges or with the privileges you assign to
the pwrchute account.
-------------------------------------------------------------
Do you want to send e-mail as root? [y/n,q] y
-------------------------------------------------------------
Pwrchute help now comes in an HTML format.
-------------------------------------------------------------
Would you like to install HTML help? [y/n,q] Please enter the name of your web
browser (case sensitive): netscape
Web browser executable found at /usr/X11R6/bin/netscape
-------------------------------------------------------------
PRODUCT                  : PowerChute Plus for Unix
INSTALL USER INTERFACE   : TRUE
INSTALL DAEMON           : TRUE
OPERATING SYSTEM         : Linux
```

```
INSTALL PATH               : /usr/lib/powerchute
PATH TO MOUNTED CD ROM     : /home/install/linux/apc
DEDICATED TTY              : /dev/ttyS0
UPS TYPE                   : Smart-UPS
Measure-UPS INSTALLED      : FALSE
PREVIOUS VERSION FOUND     : TRUE
REMOVE PREVIOUS VERSION?   : FALSE
INSTALLING AS ROOT         : TRUE
TCP/IP Installed           : TRUE
RUN COMMAND FILES AS ROOT  : TRUE
SEND E-MAIL AS ROOT         : TRUE
WEB BROWSER                : /usr/X11R6/bin/netscape
-------------------------------------------------------------
Are the above selections correct? [y/n,q] y
Installing PowerChute Plus for Unix from /home/install/linux/apc...
Extracting files.....

Remove.sh
what_os.sh
(...)
apachesh.pdf
readme_apache
language.txt

Checking for presence of binary files...
Done

Checking for presence of scripts...
Done

Checking for binary compatibility...
binary compatibility VERIFIED

/dev/ttyS0 verified as a valid tty

The following Port validations for /dev/ttyS0 may take a few moments....
/dev/ttyS0 appears to be a local control port

make backup copy of startup files...
modifying startup files...
making backup copy of shutdown files...
modifying shutdown files....
modifying /etc/apc_repository

UPS communications on /dev/ttyS0 verified
Done.

Modifying powerchute.ini file
Eeproms okay.
```

```
-------------------------------------------------------------
PowerChute Plus for Unix Installation complete.  You will need to reboot in
order to start the application
-------------------------------------------------------------
NOTE: For every machine on which you run the UPS monitoring daemon, you MUST
create a "pwrchute" user account. Otherwise, the User Interface will show you a
list of machines which are running the UPS monitoring daemon but you will be
unable to monitor or configure them.
-------------------------------------------------------------
```

Nach erfolgreichem Abschluß der Installation ist die USV-Software im Verzeichnis */usr/lib/powerchute* gespeichert. Nach dem Wechsel in dieses Verzeichnis sollte zunächst der Inhalt der Konfigurationsdatei *powerchute.ini* überprüft werden. In jedem Fall muss dort der Eintrag

```
SignallingType = Simple
```

stehen, um den Kabeltyp richtig zu spezifizieren. Der USV-Daemon[1] *upsd kann jetzt mit*

```
# ./upsd
```

erstmalig testweise gespeichert werden. Mit

```
# ./xpowerchute
```

wird die Bedieneroberfläche unter KDE (!) gestartet. Das Programm zeigt zunächst die Auswahl der zur Adminstration verfügbaren USV-Geräte (Abbildung 2-11).

Nach Auswahl des Eintrages linux01 und Angabe des Standard-Kennwortes *apc* wird der aktuelle USV-Status angezeigt (Abbildung 2-12). Weitere Konfigurationen sind nicht notwendig, beim nächsten Systemstart sollte der erfolgreiche Start des Daemons *upsd* angezeigt werden, die Meldung

```
Power Chute Plus for Unix, v4.5.2 (libc5):
Copyright © 1999. American Power Conversion
```

wird auf dem Bildschirm ausgegeben. Der Startvorgang und alle weiteren Ereignisse werden in */var/lib/powerchute/powerchute.log* protokolliert.

Bei einem Ausfall der Netzspannung erzeugt die USV ein akustisches Warnsignal und eine Warnmeldung informiert die Benutzer darüber, dass die Netzspannung ausgefallen ist und dass der Server in *x* Minuten heruntergefahren wird. Wird die Netzspannung innerhalb dieser Warnfrist wiederhergestellt, wird auch dies an alle Benutzer gemeldet; der Server wird dann nicht heruntergefahren.

1. Hier wird die englische Schreibweise Daemon verwendet, um einen unter Linux laufenden Hintergrundprozess zu bezeichnen.

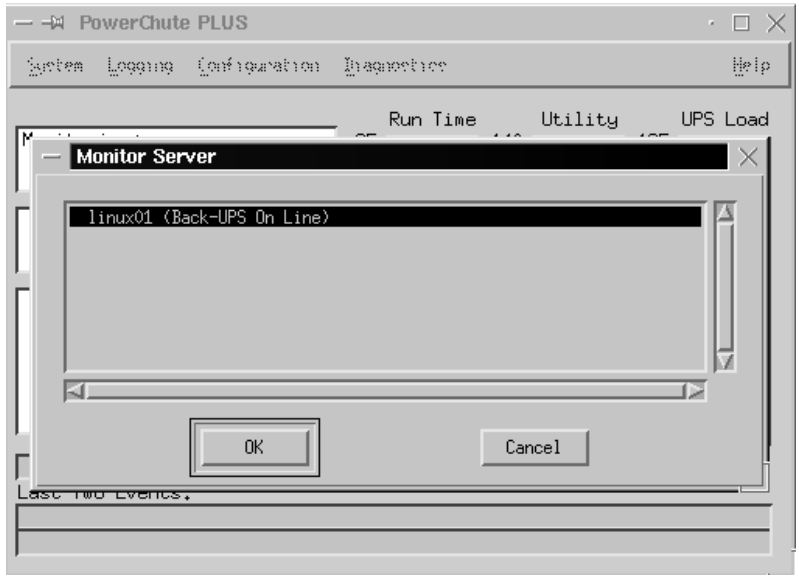


Abbildung 2-11 Programm xpowerchute: Auswahl der USV

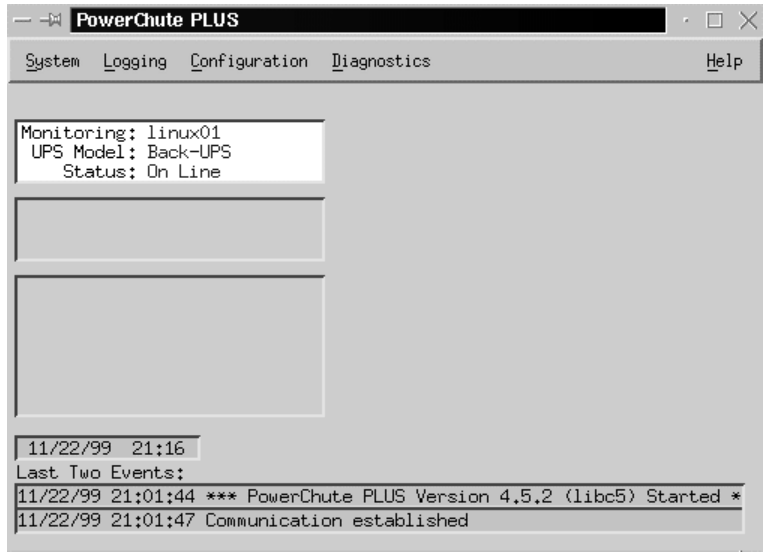


Abbildung 2-12 Programm xpowerchute: Statusanzeige der ausgewählten USV

Beispiel 2: IMV

Einen anderen Weg, der einen nahezu betriebssystemunabhängigen Betrieb ihrer USV-Systeme ermöglicht, geht die Firma IMV (*www.imv.de*). Die gesamte Software läuft unter *Java*, die Einrichtung der Software erfolgt über ein einzelnes *class*-file.

Die Installation wurde getestet für eine USV vom Typ IMV Match 750 (Abbildung 2-13). Das benötigte Anschlusskabel für die Verbindung der USV mit einer seriellen Schnittstelle ist im Lieferumfang enthalten.

Abbildung 2-13 IMV Match 750

Voraussetzung für die Installation ist eine Java-Laufzeitumgebung (*java runtime environment* jre) ab Version 1.1. Unter SuSE-Linux wird dazu das Paket *javarunt* aus der Serie *d Programmentwicklung* benötigt, installiert wird die Version 1.1.7v3-5.

Die USV-Setup-Software besteht aus der einzelnen Datei *setup.class*, sie ist auf der Heft-CD im Verzeichnis */usv/imv* zu finden und wird zunächst in ein beliebiges Verzeichnis auf dem Linux-Server kopiert (im nachfolgenden Beispiel in */tmp/imv*). Mit

```
# /usr/lib/jre/bin/jre -cp /tmp/imv setup
```

wird dann der *Java Install Shield* mit dem IMV Setup gestartet (Abbildung 2-14).

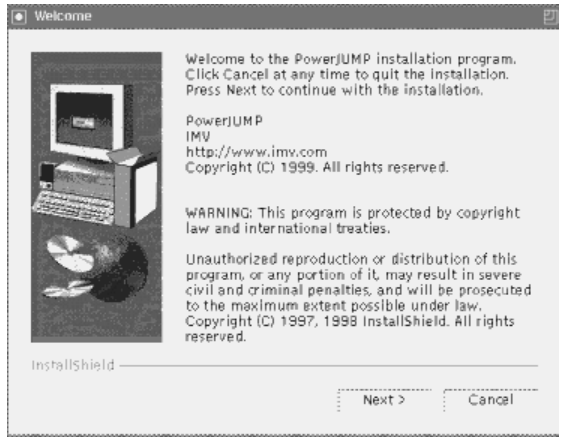


Abbildung 2-14 IMV Setup

Das IMV-Setup führt schrittweise durch die weitere Installation, abgefragt wird z.B. das Verzeichnis, in dem die USV-Software installiert werden soll (in diesem Beispiel */PowerJUMP)*. Danach kann der USV-Daemon mit *runjumpu* gestartet werden:

```
Linux05:/PowerJUMP # ./runjumpu

  PowerJUMP - JAVA UPS monitor version 1.1 (build 12.7.1999)
  (c) IMV, 1999

         User: root
       System: Linux
      Version: 2.2.13
 Architecture: x86

Configured UPSes:1
UpsWeb is listening in port: 2161
Internal communication port: 2160
Started serial monitoring on port COM2
```

Das Programm *runviewu* ermöglicht dann die Konfiguration und Anzeige der aktuellen USV-Parameter in einer grafischen Oberfläche (Abbildung 2-15). Das voreingestellte Kennwort zur Bedienung dieser Oberfläche ist »imv«, es kann bei Bedarf über den Menüpunkt *Authentification* jederzeit geändert werden.

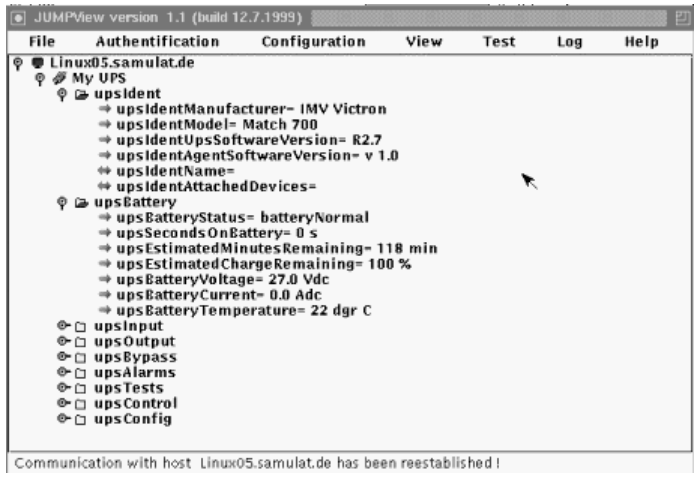


Abbildung 2-15 Bedieneroberfläche des Programms runviewu

Der USV-Daemon der Firma IMV enthält auch einen einfachen WWW-Server, der Informationen über Konfiguration, den aktuellen Systemzustand und Dokumentationen unter der Portadresse 2161 bereitstellt (Abbildung 2-16).

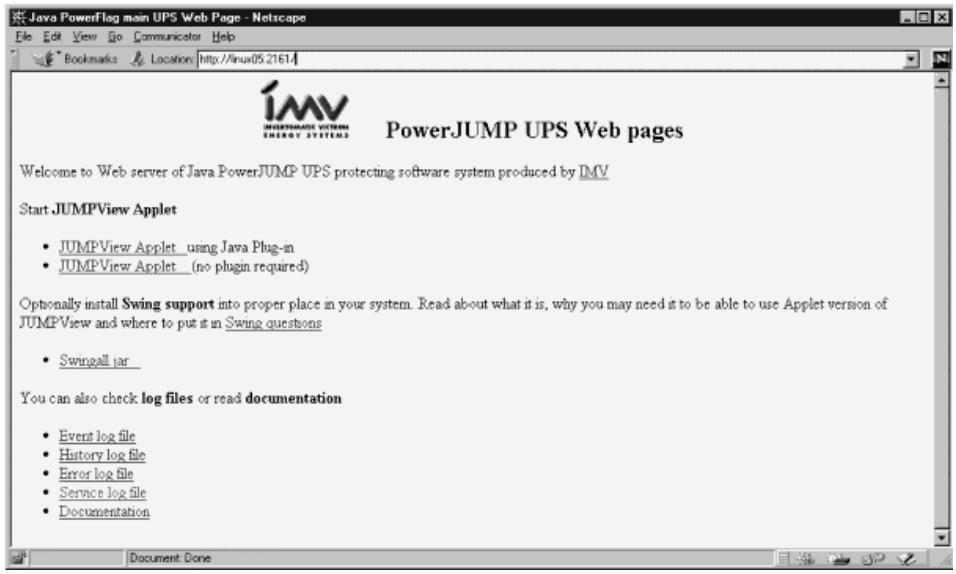


Abbildung 2-16 Anzeige von USV-Informationen über einen Internet-Browser

Der Zugriff auf diese Informationen ist unter der Adressangabe *http://linux05:2161* mit jedem Internet-Browser möglich, wobei hier *linux05* der Name des Servers ist, auf dem der USV-Daemon mit *runjumpu* gestartet wurde.

2.2 Systemsicherheit

Die auf einem Server gespeicherten Datenbestände und die von Diensten zur Verfügung gestellten Funktionen müssen im Netzwerk vor unbefugtem Zugriff geschützt werden. Bereits bei der Serverinstallation werden damit Überlegungen zur Realisierung einer Mindestsystemsicherheit sehr wichtig. Mit späteren Nachbesserungen lassen sich die Anforderungen aus Strafgesetzbuch und Bundesdatenschutzgesetz nur noch mit sehr hohem Arbeitsaufwand erfüllen.

Die erste Überlegung ist, ob das eingesetzte Serversystem überhaupt geeignet ist, höhere Sicherheitsanforderungen zu erfüllen. Als Maßstab kann hier eine Klassifizierung betrachtet werden, die vom *Department of Defense* (DoD) der Vereinigten Staaten von Amerika im *Orange Book* vorgenommen wurde. Danach existieren für die Einstufung sieben Klassen D, C1, C2, B1, B2, B3 und A1. Alle Systeme, die nicht in C1 bis A1 eingereiht werden können, sind der Sicherheitsklasse D zugeordnet. Diese stellt den geringsten Sicherheitsstandard dar und schließt alle DOS- und DOS-ähnlichen Systeme von der Teilnahme in Netzwerken aus.

Für die Installation und den Betrieb des Linux-Servers sind die Anforderungen der Klasse C wichtig [BoDE99]:

Soll ein Betriebssystem der C-Sicherheitsklasse entsprechen, so ist zwingend Voraussetzung, dass vom Benutzer bestimmbare Zugangsbeschränkungen existieren. Der Benutzer muss die Rechte seiner Dateien und ausführbaren Programme selbst vergeben können:

- C1-Systeme eignen sich für Benutzer, die alle dem gleichen Sicherheitsniveau entsprechen. Zwischen Daten und den Benutzern muss streng getrennt werden.
- In C2-Systemen müssen die vom Benutzer bestimmbaren Einschränkungen so realisierbar sein, dass die Aktionen der einzelnen Benutzer abgespeichert und überwacht werden können. Die einzelnen Benutzer müssen identifizierbar sein, z.B. über die eindeutige Authentifizierung mit Anmeldename/Kennwort. Die Kontrolldaten müssen zusätzlich vor unbefugtem Zugriff geschützt sein. Überwacht werden müssen die Benutzung des Authentifikationsmechanismus, das Löschen und das Öffnen von neuen Objekten durch den Benutzer (Programmstarts, Anlegen neuer Dateien usw.).

Praktisch alle entsprechend konfigurierten Unix-, Novell- und Windows-Systeme erfüllen die C2-Voraussetzungen. Ein Windows-NT-Server darf dann aber nicht an das Netzwerk angeschlossen werden (!) und auch kein Diskettenlaufwerk besitzen.

Linux ist (bisher) nicht zertifiziert, aber wie bei vergleichbaren Unix-Systemen ist es so sicher, wie der Systemverwalter es konfiguriert. Bei allen Unix-System lässt sich die für ein C2-System geforderte Absicherung und Überwachung verleichsweise einfach einrichten.

2.3 Hochverfügbarkeitssysteme

Der Ausfall eines Servers kann in sehr kurzer Zeit enorme Kosten verursachen. Was helfen Wartungsverträge mit kurzen Reaktionszeiten von z.B. 6 oder 12 Stunden, wenn trotzdem die Arbeit eines gesamten Tages nicht mehr geleistet werden kann oder gerade erfaßte wichtige Datenbestände zerstört oder zumindest in diesem Moment nicht mehr vollständig verfügbar sind? Auch eine gute und vollständige Bandsicherung muss nach einem Serverausfall wieder aufgespielt werden, was Zeit und Nerven kostet. Ist vielleicht sogar ein neuer Server erforderlich? Es kostet viel Zeit, dieses Gerät neu einzurichten und die gesicherten Datenbestände wiederherszustellen.

Hochverfügbarkeitssysteme arbeiten nach dem Ausfall eines Servers weiter, sie verwenden redundante Strukturen; d.h. beim Ausfall eines Systems übernimmt ein gleichwertiger zweiter Server automatisch alle Funktionen und verhält sich nach außen hin wie das ursprüngliche System (*Hot Standby*).

In der Mainframe-Technik geht man noch einen wichtigen Schritt weiter, hier hat sich der Begriff des *Clusters* eingebürgert. Der Mainframe wird dabei aus vielen, möglichst unabhängigen »CPU-Modulen« gebildet, die sich nach außen wie ein einzelner Rechner darstellen. Fällt eines dieser Module aus, so wird es automatisch isoliert. Das verbleibende Cluster arbeitet dann zwar etwas langsamer weiter, die Mainframe-Funktionen bleiben aber insgesamt erhalten. Die CPU-Module werden in Cluster-Systemen in der Regel mechanisch als Einschub ausgeführt, der sogar während des Betriebes gewechselt werden kann.

Hochverfügbarkeit beginnt schon bei der Planung des einzelnen Servers mit der Auswahl der Hardware. Nur mit qualitativ hochwertigen Komponenten läßt sich das Restrisiko weiter minimieren.

Unter Linux existieren heute bereits ganz eine ganze Reihe unterschiedlicher Ansätze, hohe Verfügbarkeitsanforderungen zu erfüllen.

2.3.1 Backup-Server-System

Bei Ausfall des Primär-Servers übernimmt ein Backup-Server mittels ARP-Spoofing die IP-Adresse (mit Alias oder echter Hardware) des Primär-Servers und somit automatisch alle wichtigen Dienste. Eine solche Lösung ist z. B. mit dem Programmpaket fake (*http://linux.zipworld.com.au/fake/)* realisierbar.

Dieses in der Regel sehr preiswerte Verfahren ist dann kaum noch einsetzbar, wenn der Backup-Server auch z. B. Datenbankfunktionen mit übernehmen muss. Die beiden Server werden nicht ausreichend synchronisiert, d. h. alle Änderungen in Datenbeständen, die nicht auf anderem Wege auf den Backup-Server kopiert wurden, sind nach dem Ausfall des Primär-Servers verloren.

2.3.2 High-Available-System

Von der Wizard Software Enginieering GmbH Müchen (*www.wizard.de*) ist ein echtes »High-Available-System« auf Linux-Basis verfügbar, das auch die Datensynchronisierung umfasst (siehe Abbildung 2-17):

Das System von *Wizard/bee* besteht aus zwei beinahe identischen Rechnern, die jeweils über einen *Private Link* und einen *Publik Link* per Ethernet miteinander verbunden sind. Damit erhält man redundante Kommunikationswege zwischen den beiden Servern. Der jeweils aktive Server besitzt neben seiner eignen IP-Adresse (hier z. B. 10.0.0.51) die des »virtuellen Servers« (10.0.0.50). Bemerkt der inaktive Server (hier z. B. 10.0.0.52) den Ausfall des aktiven Servers, so übernimmt er zunächst das Shared Storage, danach die IP des virtuellen Servers und konfiguriert eine entsprechende virtuelle Netzwerkkarte (*eth0:0*), um schließlich die geforderten Dienste zu starten. Die HA-Software von Wizard ist modular aufgebaut und die Dienste eines virtuellen Servers lassen sich in mehrere Serviceobjekte un-

terteilen, die gezielt an- und abgeschaltet werden können. Jedem Serviceobjekt ist ein Watchdog-Prozess zugeordnet, der mit einem Satz von Prüffunktionen das Wohlbefinden der aktiven Dienste testen kann [KuSc99].

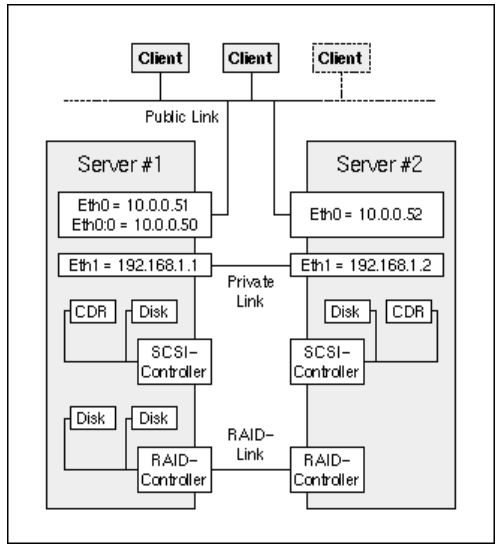


Abbildung 2-17 Hochverfügbarkeitssystem von Wizard/bee [KuSc99]

Ein *Failover* des Wizard-Systems benötigt unter Linux derzeit etwa zwei Minuten, wobei die meiste Zeit für das Laden des Kernelmoduls für den RAID-Controller und dem damit verbundenen SCSI-Reset bzw. Scan verloren geht. Mit weiter optimierten Gerätetreibern sind wesentlich kürzere Failover-Zeiten möglich.

2.4 Grundlagen der Systemverwaltung

Unix-Systeme gibt es seit fast 30 Jahren. Entwickelt wurde Unix ursprünglich auf einer PDP-4, einem 4-Bit-Rechner. Die Befehle zur Systemsteuerung mussten sehr kurz sein, teilweise haben diese Buchstabenkombinationen keinen direkten Bezug zur tatsächlichen Funktion (es sind tatsächlich manchmal die Initialen der Namen der Programmierer). Eine große Anzahl von Optionen erweitert diese Befehle, geben ihnen manchmal sogar eine völlig andere Funktion.

Unix wurde in den nachfolgenden Jahren auf fast alle gängigen Controller- und Computer-Plattformen portiert. Bei allen diesen Anpassungen und Erweiterungen des Betriebssystems wurde etwas Erstaunliches realisiert: Unix blieb fast uneingeschränkt »abwärtskompatibel«. Programme, die zu Beginn der Unix-Entwicklung geschrieben wurden, laufen heute noch auf den modernsten Rechnersystemen!

Deswegen ist es aber auch heute noch für den Systemverwalter von Unix- und Linux-Rechnern unumgänglich, sich mit der textortierten Bedienung der Benutzerschnittstelle auseinanderzusetzen. Nur auf diesem Wege können die tatsächlichen Leistungsmerkmale optimal genutzt werden.

Unix-Systemverwaltung heißt Shell-Programmierung. Der Systemverwalter lernt, Konfigurationen auch komplexer Systeme in einfachen Textdateien zu erstellen und Systemmeldungen, die ebenfalls als einfache ASCII-Texte ausgegeben werden, mit eigenen Programmen zu sammeln, zu verdichten und auszuwerten. Dies setzt Programmierkenntnisse voraus; der Rahmen reicht von der einfachen Anwendung der Kommandos zur Textbearbeitung bis zur echten C-Programmierung. Spätestens jetzt wird klar, warum die Ausbildung eines Unix Systemverealters viele Monate dauert.

Mit dieser Idee der Systemverwaltung kann Erstaunliches geleistet werden: Viele Arbeiten, die unter Novell- oder Windows-NT-Netzwerken nicht möglich sind oder teure Zusatzprogramme voraussetzen, werden elegant durch direkte Programmierung gelöst. Selbst im Umgang mit MSDOS- oder Windows-9x/NT-Oberflächen sind dann die Möglichkeiten zur skriptgesteuerten Automatisierung vieler Funktionen besser erkennbar.

Wer ärgert sich als Systemverwalter eines Windows NT Netzwerkes nicht darüber, wenn eine große Anzahl von neuen Benutzer manuell im Benutzermanager eingepflegt werden muss? Warum kann dies nicht automatisch per Skript, z. B. aus einer einfachen ASCII-Liste heraus, erfolgen? Auch Windows NT bietet solche Möglichkeiten zur Shell-Programmierung, nur die meisten Systemverwalter kennen sie nicht!

Leider (zumindest unter diesem Aspekt) treten auch für Unix/Linux immer mehr die grafisch orientierten Oberflächen zur Installation und Systemverwaltung in den Vordergrund. Sie vereinfachen, wie auch bei den aktuellen Novell- oder Windows-NT-Systemen, diese Arbeiten zunächst erheblich, erfordern aber keine tiefergehenden Systemkenntnisse mehr. Die Fehlersuche reduziert sich auf den festen Entschluß zur Neuinstallation, langsame und träge reagierende Netzwerke »müssen dann so sein«. Vielleich bringt die nächste Rechnergeneration ja noch mehr Leistung und löst so das Problem.

Professionelle Netzwerklösungen verlangen sehr tiefgehende Kenntnisse der wichtigsten Mechanismen, auch wenn dies unter Windows NT nicht so offensichtlich zu sein scheint. Ein Linux-Systemverwalter muss sich mit möglichst vielen Details der Konfiguration ganz unterschiedlicher Netzwerkdienste beschäftigen, nur so kann ein optimales Ergebnis erzielt werden. Um so verständlicher ist, dass gerade Linux unter Systemspezialisten in der universitären Ausbildung verbreitet ist.

Dieses Buch bietet keine Grundlagenausbildung für einen Unix-Systemverwalter was in dieser Kürze auch nicht möglich. Für alle, die abr schon über fundierte Kenntnisse im Betrieb anderer PC-Netzwerkbetriebsysteme verfügen, sollen hier die

wichtigsten Werkzeuge vorgestellt werden. Die typischen administrativen Aufgaben werden über Shell-Befehle und auch mit grafischen Oberflächen durchgeführt.

Die vorgestellten Arbeitsgänge orientieren sich an der Idee, einen kommerziell eingesetzten Linux-Server richtig zu verwalten, im Betrieb zu optimieren und vor allem auch die dort bereitgestellten Netzwerkressourcen wirkungsvoll vor unbefugtem Zugriff zu schützen. Dazu gehören die Werkzeuge zur Kontrolle des aktuellen Systemzustands, die Einrichtung und Konfiguration der Server-Dienste (fast ohne eine einzige Bootsequenz!) und weitere Ausbaumöglichkeiten des Gesamtsystems.

2.4.1 Starten und Herunterfahren

Bootmanager LILO

Die Bootsequenz des Linux-Servers wird gesteuert über den Bootmanager *LILO*, der bei Bedarf auch wahlweise andere PC-Betriebssysteme bzw. andere Linux-Kernel starten kann. Der Bootmanager wird während der Grundinstallation über YaST eingerichtet und konfiguriert, die dazu gehörende Konfigurationsdatei ist */etc/lilo.conf* (Listing 2-2).

```
# LILO Konfigurations-Datei
# Start LILO global Section
boot=/dev/sda
vga=normal
read-only
prompt
timeout=50
# End LILO global Section
#
image = /boot/vmlinuz
root = /dev/sda3
label = linux
```

Listing 2-2 Beispiel für eine LILO-Konfigurationsdatei

Die Datei beginnt mit einem globalen Abschnitt, gefolgt von einem oder mehreren System-Abschnitten; das Zeichen »#« leitet eine Kommentarzeile ein. Der Eintrag *image* gibt den zu ladenden Linux-Kernel an, *root* enthält die Linux-Root-Partition. Standardmäßig nicht vorhanden, aber sehr empfehlenswert, ist die Ausgabe einer »Begüßungsmeldung«, dies erfolgt mit der zusätzlichen Zeile

```
message = <textdatei>
```

im globalen Abschnitt von */etc/lilo.conf*, wobei <*textdatei*> den Text der anzuzeigenden Meldung enthält. Startet LILO das System, so wird der Text *LILO* und die optionale Begrüßungsmeldung auf dem Bildschirm ausgegeben; danach erscheint die Eingabeaufforderung

```
boot:
```

Hier wird durch Eingabe des Names das gewünschte Betriebssystem ausgewählt. Die Liste aller Namen kann mit der Taste ⇆ abgerufen werden. Nach Ablauf der über *timeout=50* eingestellten Wartezeit wird das Standardsystem automatisch gestartet, wenn keine manuelle Auswahl erfolgte.

LILO bietet vielfältige Möglichkeiten, die Startkonfiguration den individuellen Anforderungen anzupassen. Dazu steht eine Reihe von weiteren Parametern zur Verfügung, die in */etc/lilo.conf* verwendet werden kann. Vor allen Arbeiten in diesem Bereich ist es allerdings sehr empfehlenswert, eine Sicherung der ursprünglichen Konfigurationsdatei anzulegen. Weitere Details zur LILO-Konfiguration können z.B. dem SuSE-Handbuch entnommen werden.

Booten des Systems

Während der eigentlichen Linux-Systemstartprozedur (*Bootsequenz*) werden zwei grundlegende Prozesse erstellt:

- Prozess 0: der *swapper*-Prozess (Auslagerungsprozess)
- Prozess 1: der *init*-Prozess (Initialisierungsprozess)

Beide Prozesse bleiben solange existent, wie das Linux-System aktiv ist. Der *swapper* unterstützt Speicherverwaltungs-Operationen; der *init*-Prozess ist direkt oder indirekt für den Start aller anderen Benutzer-Prozesse verantwortlich. Andere Prozesse können vom System erstellt werden, um Aktivitäten wie z.B. das *Paging* (die Seitenüberlagerung) zu bedienen.

In der System-Anlaufzeit erstellt *init* typischerweise mehrere Hintergrundprozesse (sogenannte *daemons*, d.h. im »Hintergrund« laufende System- oder Dienstprogramme) und die *gettys*, die jedes interaktiveTerminal auf Benutzeranmeldungen (*Logins*) überwachen.

Der Prozess *init* fährt das System so weit hoch (-> *Runlevel*), wie in der System-Konfigurationsdatei */etc/inittab* durch den Eintrag *initdefault* festgelegt.

Linux kennt verschiedene *Runlevel*, die den Zustand des Systems definieren (Tabelle 2-2).

Runlevel	Beschreibung
0	Systemhalt
S	Single User
1	Multi User ohne Netzwerk
2	Multi User mit Netzwerk
3	Multi User mit Netzwerk und X-Windows

Tabelle 2-2 Linux Runlevel

Runlevel	Beschreibung
4	Reserviert
5	Reserviert
6	Neustart des Systems

Tabelle 2-2 Linux Runlevel

Um zu einem späteren Zeitpunkt das *Runlevel* zu wechseln, wird der Shell-Befehl *init* mit der Nummer des zugehörigen Runlevels verwendet:

Befehl	Aktion
init S	wechselt in den »Single-User-Mode«.
init 0	hält das System an.
init 3	Bei fertig konfigurierter X-Windows-Umgebung wird mit der grafischen Oberfläche gestartet.
init 6	startet das System neu.

Tabelle 2-3 Der Shell-Befehl init

Die Einträge in der System-Konfigurationsdatei */etc/inittab* haben das Format

```
name:ebene:aktion:kommando
```

mit:

name	ein eindeutiger Name aus einem oder zwei Buchstaben, der jeden Eintrag in dieser Datei identifiziert. Einige dieser Einträge müssen einen bestimmten *name* haben, damit sie korrekt funktionieren.
ebene	eine Liste der *Runlevels*, auf denen der betreffende Eintrag ausgeführt werden soll
aktion	Welche Aktion soll das unter *kommando* angegebene Programm ausführen?
kommando	Linux-Shell-Kommando mit Parametern

Eine *ebene*, das *Runlevel*, ist eine Ziffer oder ein Buchstabe zur Bezeichnung des aktuellen Systemzustands bei der Ausführung von *init*. Ein Beispiel: Wenn der Runlevel des Systems auf 3 geändert wird, werden diejenigen Einträge in */etc/inittab* ausgeführt, die im Feld *Runlevel* eine 3 haben. Die Runlevel bieten eine einfache Methode, die Einträge in */etc/inittab* gruppenweise zusammenzufassen. So kann z.B. festgelegt werden, dass Runlevel 1 nur das notwendige Minimum an Skripts ausführt, Runlevel 2 alles aus Runlevel 1 und zusätzlich die Netzwerkkonfiguration durchführt, während Runlevel 3 alles aus Runlevel 1 und 2 sowie den Wählzugang zum System konfiguriert.

Dazu ein Beispiel (Listing 2-3): Ein Auszug aus der Datei */etc/inittab* (Linux SuSE v6.2).

```
# /etc/inittab
#
# Copyright (c) 1996 SuSE GmbH Nuernberg, Germany.  All rights reserved.
#
# Author: Florian La Roche <florian@suse.de>, 1996
#
# This is the main configuration file of /sbin/init, which
# is executed by the kernel on startup. It describes what
# scripts are used for the different run-levels.
#
# All scripts for runlevel changes are in /sbin/init.d/ and the
# main file for changes is /etc/rc.config.
#
# default runlevel
id:2:initdefault:

# check system on startup
# first script to be executed if not booting in emergency mode
si:I:bootwait:/sbin/init.d/boot

# /sbin/init.d/rc takes care of runlevel handling
#
# runlevel 0 is halt
# runlevel S is single-user
# runlevel 1 is multi-user without network
# runlevel 2 is multi-user with network
# runlevel 3 is multi-user with network and xdm
# runlevel 6 is reboot

l0:0:wait:/sbin/init.d/rc 0
l1:1:wait:/sbin/init.d/rc 1
l2:2:wait:/sbin/init.d/rc 2
l3:3:wait:/sbin/init.d/rc 3
#l4:4:wait:/sbin/init.d/rc 4
#l5:5:wait:/sbin/init.d/rc 5
l6:6:wait:/sbin/init.d/rc 6

# what to do in single-user mode
ls:S:wait:/sbin/init.d/rc S
~~:S:respawn:/sbin/sulogin

# what to do when CTRL-ALT-DEL is pressed
ca::ctrlaltdel:/sbin/shutdown -r -t 4 now
```

```
# special keyboard request (Alt-UpArrow)
# look into the kbd-0.90 docs for this
kb::kbrequest:/bin/echo "Keyboard Request -- edit /etc/inittab to let this
work."

(...)
# end of /etc/inittab
```

Listing 2-3 Beispiel für /etc/inittab

Die Zeile *ca::ctrlaltdel:/sbin/shutdown -r -t 4 now* legt fest, was passiert, wenn an der Konsole die Tastenkombination Strg Alt Entf gedrückt wird. Diese Tastenkombination erzeugt einen Interrupt, der normalerweise das System neu starten würde. Unter Linux wird dieser Interrupt abgefangen und an *init* weitergeleitet, das dann den Eintrag mit dem *Aktion* -Feld *ctrlaltdel* ausführt. Der Befehl, der hier ausgeführt wird,

```
/sbin/shutdown -r -t 4 now
```

fährt das System herunter und startet es anschließend neu.

Bei jedem Wechsel eines Runlevels werden zunächst die Stop-Skripte des gegenwärtigen Levels abgearbeitet. Danach werden die Start-Skripte des neuen Runlevels ausgeführt (die Zuordnungen enthält */etc/inittab*).

Konfigurationsanpassung bei Wechsel des Runlevel

Das Programm */sbin/init.d/rc* steuert bei S.u.S.e Linux v6.2 die Abarbeitung der Start- und Stop-Scripte bei einem Wechsel des Runlevel. Dazu wird */sbin/init.d/rc* mit dem gewünschten Level-Bezeichner aus */etc/inittab* aufgrufen. Das nachfolgende Beispiel (Listing 2-4) zeigt einen Auszug aus */sbin/init.d/rc*:

```
#! /bin/bash
# Copyright (c) 1996-98 SuSE GmbH Nuernberg, Germany.  All rights reserved.
#
# Author: Florian La Roche <florian@suse.de>, 1996
#      Werner Fink <werner@suse.de>, 1994,96,98
#
# /sbin/init.d/rc   --    The Master Resource Control Script
#
# This file is responsible  for  starting/stopping  services
# when the runlevel changes.  If the action for a particular
# feature in the new run-level is the same as  the action in
# the previous run-level, this script will neither start nor
# stop that feature.
#
# avoid being interrupted by child or keyboard
trap "echo" SIGINT
```

```
. /etc/rc.config

# set onlcr to avoid staircase effect and do not lock scrolling
stty onlcr -ixon 0>&1

#     \033[1m       switch bold on
#     \033[31m      switch red on
#     \033[32m      switch green on
#     \033[33m      switch yellow on
#     \033[m        switch color/bold off
extd="\033[1m"
warn="\033[1m"
norm="\033[m"
stat="\033[71G"

echo -n "Master Resource Control: "
echo -n "previous runlevel: $PREVLEVEL, "
echo -e "switching to runlevel: ${extd}${RUNLEVEL}${norm}"

curdir=/sbin/init.d/rc$RUNLEVEL.d
prevdir=/sbin/init.d/rc$PREVLEVEL.d
rex="[0-9][0-9]"

failed=""
#
# run the KILL scripts of the previous runlevel
#
for i in $prevdir/K*; do
    test -x "$i" || continue
(...)
exit 0
```

Listing 2-4 Auszug aus /sbin/init.d/rc

Boot-Meldungen des Kernels

Während der Kernel in den Arbeitsspeicher geladen wird, erscheinen auf der Systemkonsole Meldungen wie:

```
Linux version 2.2.10 (root@Mandelbrot.suse.de) (gcc version 2.7.2.3) #4 Tue Jul
20 17:01:36 MEST 1999
Detected 350799293 Hz processor.
Console: colour VGA+ 80x25
Calibrating delay loop... 349.80 BogoMIPS
Memory: 62868k/65472k available (1260k kernel code, 404k reserved (endbase
0xa0000), 896k data, 44k init)
VFS: Diskquotas version dquot_6.4.0 initialized
CPU: Intel Pentium II (Deschutes) stepping 02
(...)
```

```
IP Protocols: ICMP, UDP, TCP, IGMP
Initializing RT netlink socket
Starting kswapd v 1.5
Detected PS/2 Mouse Port.
pty: 256 Unix98 ptys configured
Real Time Clock Driver v1.09
RAM disk driver initialized:  16 RAM disks of 20480K size
PIIX4: IDE controller on PCI bus 00 dev 21
PIIX4: device not capable of full native PCI mode
PIIX4: device disabled (BIOS)
Floppy drive(s): fd0 is 1.44M
FDC 0 is a post-1991 82077
md driver 0.36.6 MAX_MD_DEV=4, MAX_REAL=8
linear personality registered
raid0 personality registered
raid1 personality registered
raid5 personality registered
(scsi0) <Adaptec AIC-7890/1 Ultra2 SCSI host adapter> PCI 6/0
(scsi0) Wide Channel, SCSI ID=7, 32/255 SCBs
(scsi0) Downloading sequencer code. 374 instructions downloaded
scsi0 : Adaptec AHA274x/284x/294x (EISA/VLB/PCI-Fast SCSI)
       <Adaptec AIC-7890/1 Ultra2 SCSI host adapter>
scsi : 1 host.
(scsi0:0:0:0) Synchronous at 80.0 Mbyte/sec, offset 15.
  Vendor: IBM       Model: DDRS-34560D   Rev: DC1B
  Type:   Direct-Access                  ANSI SCSI revision: 02
Detected scsi disk sda at scsi0, channel 0, id 0, lun 0
(scsi0:0:2:0) Synchronous at 20.0 Mbyte/sec, offset 16.
  Vendor: PIONEER Model: CD-ROM DR-U16S  Rev: 1.01
  Type:   CD-ROM                         ANSI SCSI revision: 02
(...)
Partition check:
 sda: sda1 sda2 sda3
VFS: Mounted root (ext2 filesystem) readonly.
Freeing unused kernel memory: 44k freed
Adding Swap: 128516k swap-space (priority -1)
Kernel logging (proc) stopped.
Kernel log daemon terminating.
```

Listing 2-5 Boot-Meldungen des Kernels

Es ist der Linux-Kernel selbst, der diese Meldungen ausgibt, wenn die Gerätetreiber initialisiert werden. Welche Meldungen ausgegeben werden, hängt von den im Kernel vorhandenen Treibern (einkompiliert) sind und tatsächlich im System eingebunden Hardware ab.

So wird z.B. der Treiber für die seriellen Schnittstellen initialisiert, der für jede gefundene Schnittstelle einige Informationen ausgibt. Eine Zeile wie:

```
ttyOO at 0x03f8 (irq = 4) is a 16450
```

besagt, dass die erste serielle Schnittstelle (*/dev/tty00* oder COM1) unter der Adresse *0x03f8* gefunden wurde, IRQ 4 benutzt und einen UART 16450 enthält.

Eine weitere Meldung zeigt die Berechnung der *BogoMips* für den Prozessor. Es handelt sich um eine falsche Berechnung (*bogus*: = Schwindel-daher der Name), die zur optimalen Gestaltung der Warteschleifen in einigen Gerätetreibern benutzt wird. Der Kernel gibt außerdem Informationen zum Arbeitsspeicher aus:

```
Memory: 62868k/65472k available (1260k kernel code, 404k reserved (endbase
0xa0000), 896k data, 44k init)
```

In diesem Beispiel stehen dem System 62.868 Kilobytes an RAM zur Verfügung, der Kernel selbst belegt davon 2.604 Kilobytes.

Alle Meldungen werden beim Booten auf der Konsole ausgegeben und in der Regel auch in den System-Logdateien mitgeschrieben, z.B. in */var/log/boot.msg*.

Anmelden als Systemverwalter

Der Systemverwalter (*Superuser*) eines Unix/Linux-Servers trägt den Namen *root*, sein Kennwort wurde bereits während der Grundinstallation festgelegt. Der *root* hat sehr weitreichende Berechtigungen zur Systemverwaltung; Routineaufgaben sollten, wenn möglich, mit anderen Benutzerkonten abgearbeitet werden. Wenn der Systemverwalter in der aktuellen Shell angemeldet ist, zeigt der Systemprompt »#«. Damit ist auch für einen zufälligen »Beobachter« schnell klar, wer an diesem Server arbeitet.

Jeder Benutzer kann sich mit dem Shell-Befehl *su* als Systemverwalter anmelden, wenn er das root-Kennwort kennt.

```
Welcome to SuSE Linux 6.2 (i386) - Kernel 2.2.10 (pts/0).
login: samulat
Password:
Last login: Mon Nov 22 18:23:25 from nt40ws.samulat.de
Have a lot of fun...
samulat@linux01:~ > su
Password:
root@linux01:/home/samulat >
```

Im o.a. Beispiel hat sich zunächst der Benutzer *samulat* angemeldet, nach Eingabe von *su* und dem richtigen *root*-Kennwort arbeitet dieser als Systemverwalter.

Herunterfahren des Systems

Jeder kommerzielle Server soll möglichst lange Zeit unterbrechungsfrei laufen, die *Server Up Time* soll möglichst groß werden. Gerade Linux-Server erreichen dies oft sehr einfach, selbst nach umfangreichen Änderungen in der Systemkonfigurationen sind keine Bootvorgänge notwendig.

Sollte es doch einmal notwendig werden, einen Linux-Server herunterzufahren, so kann der Systemverwalter *root* dies über den Shell-Befehl

```
# shutdown -h now
```

oder

```
# shutdown -r now
```

auslösen. Die Option »-h« hält den Rechner nach dem Herunterfahren an; »-r« erzwingt einen Neustart des Systems. Direkt vor dem Herunterfahren wird eine entsprechende Warnmeldung an der Systemkonsole und an allen angeschlossenen Terminals ausgegeben.

Soll das Herunterfahren nicht sofort (*now*) erfolgen, wird mit der Option »-t« die Anzahl der Sekunden angegeben, die zwischen Warnmeldung und dem tatsächlichen Herunterfahren gewartet werden soll. Mit der Tastenkombination

(Strg)(Alt)(Entf)

kann der Linux-Server ebenfalls gebootet werden. Das Herunterfahren des Linux-Servers über »shutdown« oder über die Tastenkombination ist grundsätzlich nur dem Systemverwalter erlaubt. Bei Bedarf können weitere Benutzer in der Datei */etc/shutdown.allow* als berechtigt eingetragen werden.

Bootdiskette erstellen

Jedes Serversystem sollte im Notfall auch mit einer Bootdiskette gestartet werden können. Die Linux-Bootdiskette ermöglicht den Systemstart auch dann, wenn die LILO-Konfiguration nicht stimmt oder ein neuer Kernel erstellt wurde, mit dem der Server nicht mehr bootet.

Bootdiskette mit YaST erstellen

Über YaST wird die Bootdiskette mit der Funktion *Administration des Systems -> Kernel- und Bootkonfiguration -> Boot-Diskette erzeugen* erstellt (Abbildung 2-18).

Bootdiskette manuell erstellen

Benötigt wird eine leere, *FAT*-formatierte Diskette. Mit

```
root@linux01:/ > cp /boot/vmlinuz /dev/fd0
root@linux01:/ > rdev
/dev/sda3 /
root@linux01:/ > rdev /dev/fd0 /dev/sda3
root@linux01:/ > rdev -R /dev/fd0 1
root@linux01:/ >
```

wird diese Diskette zur Bootdiskette dieses Linux-Servers. Der erste Befehl *cp* kopiert den Kernel */boot/vmlinuz* auf die Diskette. Mit den drei *rdev*-Befehlen wird dann die Bootpartition des Servers ermittelt (im o.a. Beispiel */dev/sda3*) und auf der Bootdikette eingetragen.

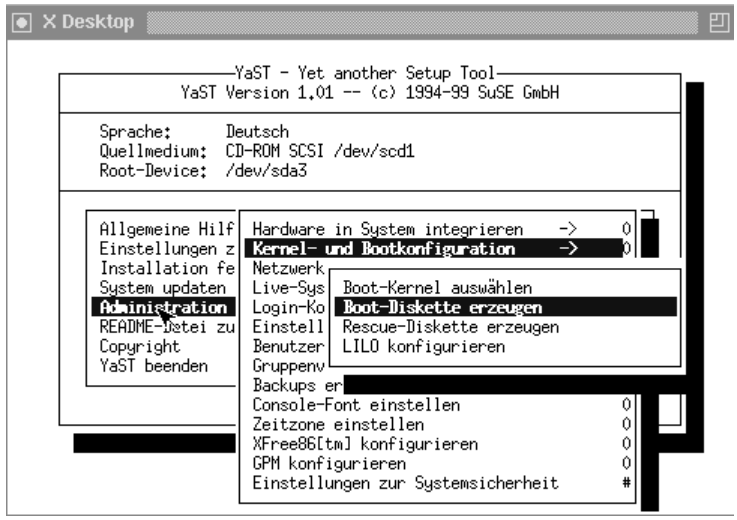


Abbildung 2-18 Bootdiskette erstellen mit YaST

2.4.2 Das Linux-Dateisystem

Datei- und Verzeichnisnamen

Das Linux-Dateisystem erlaubt die Verwendung von bis zu 255 Zeichen langen Namen für Dateien und Verzeichnisse. Das Leerzeichen ist zugelassen, Satz- oder Sonderzeichen sollten nicht verwendet werden, eine Ausnahme stellen die Zeichen »_ – .« dar. Da Unix ursprünglich nur einen 7-Bit-Zeichensatz unterstützte, kann es auch heute noch zu Problemen mit Zeichen außerhalb dieses Vorrates kommen, dies gilt besonders für die deutschen Umlaute. Hier sollte im Einzelfall überprüft werden, ob eine Anwendung diese Zeichen richtig handhaben kann. Die Beschränkung auf a-z, A-Z, 0-9,- ist nicht notwendig, aber sinnvoll.

Für die Bildung von Dateinamen gibt es keine speziellen Regeln. Die aus DOS-Systemen bekannten, meistens drei Zeichen langen Dateinamenendungen sind nicht bekannt. Unix/Linux unterscheidet zwischen Klein- und Großbuchstaben, die Datei *Text1* ist also eine andere als *text1*.

Als Trennzeichen zwischen Verzeichnissen und Verzeichnissen/Dateien wird der Slash »/« verwendet, der Backslash »\« ist dafür nicht zugelassen. Dieser stellt ein »Fluchtzeichen« dar, das zur Steuerfunktionen verwendet wird.

Wechselt der Benutzer unter DOS oder Windows 9x/NT in ein Verzeichnis, so kann er ein dort gespeichertes Programm direkt über dessen Namen aufrufen. Unter Linux ist dies grundsätzlich nicht möglich, da die Benutzer-Shell den angegebenen Befehl nicht zuerst im aktuellen Verzeichnis sucht, sondern nur die Systempfade auswertet. Trotzdem kennt Linux den Begriff des aktuellen Verzeichnisse (*Working*

Directory), dieses kann bei der Angabe von Pfaden direkt durch einen Punkt angegeben werden. Dazu ein Beispiel:

Der Benutzer wechselt mit dem Befehl *cd /daten/texte* in das angegebene Verzeichnis, indem sich das ausführbare Programm *xtest befindet*. Mit dem Befehl

```
max@linux01> pwd
/daten/texte
```

wird dieses Verzeichnis auch als aktuelles Verzeichnis bestätigt. Die Eingabe von *xtest* führt an dieser Stelle nicht zum gewünschten Start des Programmes. Mit

```
max@linux01> ./xtest
```

ist dies möglich, der Punkt wird von der Shell durch */daten/texte* ersetzt, das Programm *xtest* wird gestartet. Linux kennt noch ein weiteres Kürzel für einen Verzeichnispfad: Die Tilde »~« wird von der Shell immer als das Benutzer-Homeverzeichnis interpretiert.

Das Linux-Dateisystem kennt keine DOS-Dateinamenerweiterung, trotzdem haben sich eine Reihe von Kennzeichnern, die durch einen Punkt abgetrennt werden, zur Angabe der Dateityps bewährt:

.rc	Konfigurationsdatei (beginnen i.d.R. mit einem Punkt)
.c, .cc	C- oder C++-Programmtext
.f	Fortran-Programme
.tar	tar-tape-Archive
.gz	mit gzip komprimierte Dateien
.tmp	temporäre Dateien

Auch Unix/Linux kennt versteckte Dateien, z.B. die Konfigurationsdateien im Home-Verzeichnis jedes Benutzers. Da diese nur selten bearbeitet werden, sind sie durch einen dem Namen vorangestellten Punkt »versteckt«. Unter anderem sind dies

.profile	Benutzer-Anmeldeskript (bei Verwendung der bash)
.bashrc	Konfiguration der bash
.exrc	Konfiguration des vi, ex
.xinitrc	Startskript des X-Windows-Systems

Wird ein neuer Benutzer angelegt, so werden diese Dateien aus dem Verzeichnis */etc/skel* kopiert.

Format der Textdateien

Unter Unix/Linux werden Konfigurationsdateien und alle Statusmeldungen als ASCII-Texte dargestellt. Sollen Linux und DOS-Systeme gemeinsam mit diesen Dateien arbeiten, so ist festzustellen, da dies nicht direkt möglich ist. Die Zeichen, die ein Zeilenende darstellen, sind unterschiedlich:

- Linux: *linefeed* (LF), einfacher Zeilenvorschub, »^J«
- DOS: *carriage return* + *linefeed* (CRLF) , Wagenvorlauf, »^M«

Auch die verwendeten Zeichensätze sind unterschiedlich:

- Linux: ASCII + ISO-LATIN1 (für Umlaute und Rahmen)
- DOS: ASCII + IBMPC

Die wechselseitige Umwandlung von Textdateien erfolgt mit den Programmen *dos2unix* und *unix2dos*, bzw. *recode ibmpc:lat1* und *recode lat1:ibmpc*. Dabei sollte sehr vorsichtig vorgegangen werden: Ausführbare Programme und Daten in Nicht-Textform werden durch diese Formatumwandlungen unbrauchbar!

Standard-Verzeichnisstruktur

Linux-Systeme verwenden eine Verzeichnisstruktur, die auch an vielen anderen UNIX Anlagen zu finden ist. Linux kennt keine Laufwerke im DOS-Sinne; alle Geräte werden konsequent über Dateien angesteuert; Lese- und Schreibzugriffe auf die zum System gehörende Hardware erfolgen über *Special Files*, die im Verzeichnis */dev* verfügbar sind.

Der Start des lokalen Dateisystems ist das »/«, hier beginnt die einzige Verzeichnisstruktur des Linux-Servers. Der Beginn des Dateisystems »gehört« dem Systemverwalter *root*, nur er hat hier alle Rechte. Interessant ist der Anzatz der Benutzer-Homeverzeichnisse: Wird ein Benutzerkonto angelegt, so erhält der Benutzer ein privates Verzeichnis, standardmäßig liegt dieses in */home*. Der Verzeichnisname ist der Anmeldeaname des Benutzers, nur er und der Systemverwalter haben dort Rechte.

Die nachfolgende Aufstellung (Tabelle 2-4) zeigt eine Auswahl der standardmäßig unter Linux vorhandenen Verzeichnisse.

Verzeichnis	Inhalt
/	root- oder Stammverzeichnis. Es ist das oberste Verzeichnis des gesamten Server-Dateisystems.
/bin	Programme und Befehle, die zum Systemstart benötigt werden
/boot	Betriebssystemkernel

Tabelle 2-4 Linux Verzeichnisstruktur

Verzeichnis	Inhalt
/dev	Die Gerätedateien (*Special Files*) im Verzeichnis */dev* enthalten keine Informationen, sie stellen dem Benutzer Ein-/Ausgabekanäle zur Verfügung. Jedes an den Rechner angechlossene Terminal, jeder Drucker und jedes verfügbare andere Gerät hat in */dev* einen oder mehrere Einträge.
/etc	Skripte zur Systemkonfiguration
/home	Startverzeichnis für die Benutzerverzeichnisse
/lib	Bibliotheken (*Libraries*), die von Programmen zur Laufzeit benötigt werden
/proc	*/proc* ist ein virtuelles Dateisystem, ihm ist kein Speicherplatz zugeordnet. Im Verzeichnis */proc* befindt sich eine Reihe von *Dateien* und *Verzeichnissen*, deren Inhalt sich im Laufe der Zeit verändert. Der Kernel stellt durch das Dateisystem */proc* statistische Informationen über das System und die Prozesse bereit. Wird auf eine Datei im Dateisystem */proc* zugegriffen, erkennt der Kernel dies und gibt aktuelle Daten aus, um die Leseanforderung zu erfüllen. Die *Dateien* und *Verzeichnisse* existieren nicht auf der Festplatte, sie werden vom Kernel erzeugt, um z.B. Programmen wie *ps* und *top* den Zugriff auf Informationen zu gewähren.
/sbin	Programme und Daten, die zum Systemstart benötigt werden und dem Systemverwalter vorbehalten sind
/tmp	temporäre Dateien
/usr	alle für Benutzer vorgesehenen Programme und Daten
/usr/bin	allgemein verfügbare Befehle
/usr/doc	Systemdokumentation
/var	variable Systemdaten, die zur Laufzeit geändert bzw. ergänzt werden
/var/log	Log-Dateien
/var/spool	temporärer Zwischenspeicher für Systemprozesse (z.B. E-Mail-Dateien)

Tabelle 2-4 Linux Verzeichnisstruktur

Für die Navigation im Dateisystem des Linux-Servers steht der auch unter DOS bekannte Befehl *cd* zur Verfügung. Für die Angabe von Datei- und Verzeichnisnamen verfügt die Linux-Shell über sehr leistungsfähige Mechanismen zur Ersetzung von Platzhaltern in Dateinamen. Über die bekannten Zeichen »*« und »?« hinaus (die hier auch mehrfach und in beliebigen Kombinationen auftreten können), sind auch Angaben von Wertebereichen für einzelne Zeichen möglich, z.B steht die Angabe »[a-f]« für ein einzelnes Zeichen im Bereich »a« bis »f«.

Special Files

Die unter Linux verwendte Dateien zum Lese- und Schreibzugriff auf die Hardware, die *special files*, sind im Verzeichnis */dev* gespeichert. Der Auszug aus den Dateilisten in den Verzeichnissen */etc* und */dev* zeigt markante Unterschiede in den dargestellten Informationen. Die »Special Files« (Gerätetreiber) haben anstelle der Dateigröße eine Angabe zu *major-* und *minor-device-number*:

```
mitte2:/dev # ls -l /etc/t*
-rw-r--r--   1 root  root      580426 Nov 11  1996 /etc/termcap
-rw-r--r--   1 root  root         258 Feb 20  1995 /etc/ttytype

mitte2:/dev # ls -l /dev/hda[1-5]
brw-------   1 root  root      3,   1 Aug 26  1996 /dev/hda1
brw-------   1 root  root      3,   2 Aug 26  1996 /dev/hda2
brw-------   1 root  root      3,   3 Aug 26  1996 /dev/hda3
brw-------   1 root  root      3,   4 Aug 26  1996 /dev/hda4
brw-------   1 root  root      3,   5 Aug 26  1996 /dev/hda5
```

Die *major-device-number* verweist auf den entsprechenden Gerätetreiber im Linux-Kernel. Die *minor-device-number* wird als Parameter bei Aufruf des Linux-Kern-Gerätetreibers übergeben.

Dazu ein Beispiel: Die *Major-number* = 3 identifiziert den Kernel-Treiber zum Zugriff auf IDE-Festplatten. Über diesen Treiber können grundsätzlich mehrere Festplatten angesteuert werden. Welche Platte ausgewählt werden soll, ist über die *minor-device-number* festgelegt.

Das Anlegen eines neuen »Special Files« erfolgt über den Befehl *mknod*. Dazu müssen *major- und minor-device-number*, der Gerätetyp (b, c oder p) und der Name angegeben werden:

Gerätetyp =	b	is blockdevice	(Festplatten, Streamer, ...)
	c	is character device	(Terminal, Tastatur, ...)
	p	is named pipe	(Prozesssynchronisierung)

Systeminformation in /proc

Informationen über den aktuellen Systemzustand stellt der Kernel im virtuellen Verzeihnis */proc* bereit. Die hier enthaltenen ASCII-Dateien existieren nicht auf der Festplatte, sondern werden erst im Moment des Lesezugriffs vom Kernel erstellt. Viele Shell-Befehle verwenden Informationen aus diesem Verzeichnis.

Über */proc* können z. B. viele Informationen zur aktuellen Hardwarekonfiguration abgerufen werden, z. B.:

cpuinfo	Typ und Leistungsdaten der CPU
interrupts	belegte Interrupts
ioports	belegte I/O-Adressen
meminfo, memstat	aktuelle Speicherbelegung

2.4.3 Systemsteuerung

Tastenkombinationen

Jedes Linux-System verfügt nach der Grundinstallation bereits über sechs (!) Text- und eine Grafikkonsole (X11). Alle sieben Konsolen können auf dem gleichen PC unabhängig voneinander zur Bedienung des System genutzt werden, ein Wechsel zwischen den Konsolen ist jederzeit möglich. Die Auswahl der gewünschten Konsole und die grundlegende Steuerung erfolgt dabei über Tastaturkommandos (Tabelle 2-5), wobei einige Befehle nur unter X11, einige nur in den Textkonsolen funktionieren.

Tastenkombination	Funktion
Alt F1 bis Alt F6	Wechsel zwischen den Textkonsolen 1 bis 6 (funktioniert nicht unter X11)
Alt → und Alt ←	Wechsel zur nächsthöheren oder niedrigeren Textkonsole (funktioniert nicht unter X11)
Alt Strg F1 bis Alt Strg F6	Wechsel zu einer Textkonsole aus X11
Alt F7	Wechsel von einer Textkonsole auf die Grafkkonsole (X11)
Alt Strg Entf	Neustart des Linux-Systems, Reboot. Kann von jedem Benutzer ausgeführt werden (funktioniert nicht unter X11)
Strg D	Dateiendezeichen, »EOF«. Erwartet ein Shell-Befehl weitere Parameter über die Kommandozeile, so wird die Eingabe mit dieser Tastenkombination abgeschlossen.
Strg C	Programmausführung abbrechen (funktioniert nicht unter X11)
Strg L	Bildschirm löschen
↑ und ↓	Zeilen im Bildschirmpuffer blättern, »scrollen«
⇧ Bild↑ und ⇧ Bild↓	Seitenweise im Bildschirmpuffer blättern (funktioniert nur bis zum Konsolenwechsel)

Tabelle 2-5 Steuerung über Tastenkombinationen (Auswahl)

In jeder Textkonsole werden von der Bedienerschnittstelle (Shell) grundsätzlich die letzten 200 eingegebenen Befehle in der Datei *~/bash_history* gespeichert und können mit den Cursortasten »durchsucht« werden (die genaue Anzahl der gespeicherten Zeilen ist einstellbar). Die gezielte Suche ist mit der Tastenkombination Strg R möglich, der Stern »*« vervollständigt als Joker unvollständige Dateinamen. Die Tabulatortaste ⇆ ergänzt Befehls- und Dateinamen; ist dies nicht eindeutig, so wird ein Warnton ausgegeben. Nach zweimaligem Betätigen von ⇆ werden alle möglichen Alternativen angezeigt.

Das erste Wort der Kommandozeile wird immer als Programm interpretiert (somit auch als solches vervollständigt), das zweite und alle folgenden Wörter auf der Kommanozeile werden als Dateinamen interpretiert (bezogen auf das aktuelle Verzeichnis).

Die wichtigsten Befehle

Die nachfolgende Aufstellung soll lediglich eine kurze Einführung in eine Auswahl von Linux-Shell-Kommandos geben, weitere Befehle zur Systemverwaltung werden dann an jeweils passender Stelle in den nachfolgenden Abschnitten beschrieben.

Zusätzliche Informationen zu den hier vorgestellten Befehlen können den entsprechenden man-Pages oder der Fachliteratur, z.B. [Kofl98] oder [Hein98], entnommen werden.

Befehlsbeschreibung – man
gibt eine ausführliche Beschreibung (*man-Page*) des als Parameter übergebenen Befehls inklusive aller Optionen und Parameter. Die Darstellung kann mit [PgUp] und [PgDwn] gesteuert werden, mit der Taste [Q] wird die Anzeige beendet. Mit

```
man mkdir
```

wird die Beschreibung des Befehles *mkdir* auf dem Bildschirm ausgegeben. Die man-Pages sind in verschiedenen Klassen (*Sections*) organisiert. Zu einem Stichwort können ohne weiteres Informationen in mehreren Sektionen vorhanden sein.

Sektion	Inhalt
1	Programme oder Shell-Kommandos
2	Systemaufrufe (allgemeine Funktionen)
3	Library-Routinen (Funktionen der System-Bibliotheken)
4	Special Files (Gerätedateien, meist in */dev*)
5	Dateiformate und Konventionen z.B. für */etc/passwd*
6	Spiele
7	Makro-Pakete und deren Konventionen
8	Systemverwaltungskommandos
9	Kernel-Routinen (spezielle Routinen, kein Standard)

Tabelle 2-6 Sektionen der man-Pages

Soll in allen man-Pages nach einem bestimmten Schlüsselwort gesucht werden, so erfolgt dies mit dem Befehl *apropos*. Diese Suche ist auf die Kurzbeschreibungen der man-Pages begrenzt, *apropos* wird normalerweise orientierend vor dem konkreten Aufruf einzelner man-Pages eingesetzt.

Die in einer man-Page dargestellten Informationen sind üblicherweise in mehreren Teilen (*parts*) gegliedert:

NAME	Name der Struktur
SYNOPSIS	Syntax
DESCRIPTION	kurze Funktionsbeschreibung
OPTIONS	Optionen des Befehls
FILES	für die Anwendung relevante Dateien
SEE ALSO	Querverweise
BUGS	bekannte Fehler
AUTHORS	Autoren
EXAMPLES	Beispiele für die Anwendung
DEFAULTS	Voreinstellungen
ENVIRONMENT	relevante Umgebungsvariablen
EXIT STATUS	Rückgabewerte und ihre Bedeutung
HISTORY	Zusammenfassung

Tabelle 2-7 Teile einer man-Page

Innerhalb des SYNOPSIS-Teils gelten die nachfolgenden Darstellungsregeln (sie sollten auch in den anderen Teilen eingehalten werden):

- Fett gesetzte Passagen sollen exakt eingegeben werden. Sie beziehen sich meistens auf den Programmnamen.
- Kursiv werden Argumente dargestellt, die entsprechend anzupassen sind (formale Parameter).
- Optionale Argumente werden in eckigen Klammern dargestellt, diese Klammern sind in der Befehlszeile nicht mit einzugeben.
- Einander ausschließende Angaben werden durch ein »|« getrennt.
- Wiederholt anwendbare Argumente werden durch drei abschließende Punkte gekennzeichnet.

Die in fast allen Linux-Distributionen enthaltene X-Windows-Variante von *man* ist *xman* (Abbildung 2-19) In dieser grafischen Oberfläche erfolgt der Zugriff auf die gewünschten man-Pages über eine nach Sektionen organisierte Auswahlliste (in Abbildung 2-19 ist links die Liste der Sektion 1 zu sehen).

Noch leistungsfähiger ist das ebenfalls unter XWindows laufende Programm *tkman*, dass ebenso wie *man* mit einem Suchbegriff als Parameter gestartet wird, dann aber bei Bedarf auch sofort die »Treffer« aus mehreren Sektionen anzeigt.

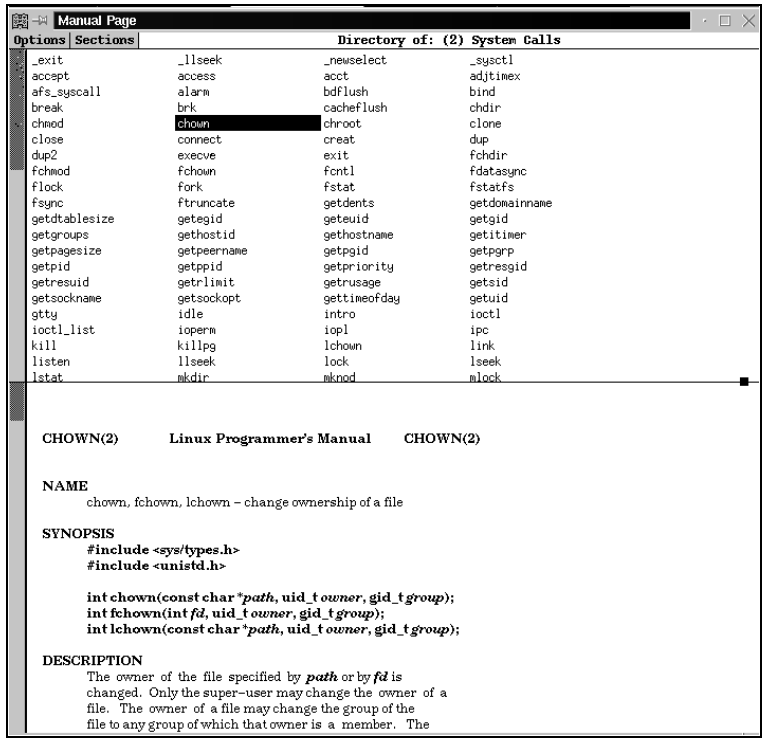


Abbildung 2-19 Bedieneroberfläche von xman

Wie auch bei *xman* stehen bei der Ausgabe der dargestellten man-Pages eine Reihe von (einstellbaren) Tastenkommandos zur interaktiven Arbeit zur Verfügung.

Terminaleinstellungen verändern -setterm

setterm ist ein Programm, mit dem verschiedene Eigenschaften des Terminals eingestellt werden (z.B. für die Textkonsolen 1 bis 6). Dazu gehören z.B. die Wiederholungsrate der Tastatur, die Schrittweite der ⇆-Taste und die Farben. Mit

```
setterm -foreground white -background blue
```

wird »weißer Text vor blauem Hintergrund« für das gerade aktive Terminal eingestellt. Soweit möglich, verwendet *setterm* Einstellungen aus der »Datenbank« *terminfo*, die den Funktionsumfang und die Steuerbefehle einer Vielzahl verschiedener Terminal-Emulationen enthält. Mit dem Befehl:

```
setterm -store
```

werden die aktuellen Einstellungen dauerhaft gesichert. Von besonderem Interesse sind *setterm* und *terminfo* immer dann, wenn es darum geht, spezielle Terminal-Anpassungen für den Mainframe-Zugriff zu definieren.

Verzeichnis anlegen – mkdir

Mit *mkdir* wird ein neues Verzeichnis angelegt. Als Parameter wird lediglich der Name des zu erstellenden Verzeichnisses angegeben, wobei absolute wie auch relative Adressen erlaubt sind. Der Befehl

```
# mkdir texte
```

erzeugt ein Verzeichnis mit dem Namen *texte* im aktuellen Arbeitsverzeichnis.

Verzeichnis löschen – rmdir

Der Befehl *rmdir* löscht ein leeres Verzeichnis. Enthält das angegebene Verzeichnis noch eine Datei oder weitere Unterverzeichnisse, erfolgt eine Fehlermeldung. Die aus DOS-Anwendungen bekannte Kurzform *rm* wird in den meisten Linux-Distributionen nicht unterstützt. Mit

```
# rmdir texte
```

wird das im aktuellen Arbeitsverzeichnis liegende Unterverzeichnis *texte* gelöscht.

Verzeichnisinhalt auflisten – ls

Der Befehl *ls* zeigt den Inhalt des Arbeitsverzeichnisses. Der Befehl hat eine Reihe von Parametern, mit denen das anzuzeigende Verzeichnis (absolut oder relativ) und der Umfang der angezeigten Details angegeben werden kann. Einzelheiten dazu können den umfangreichen man-Pages des Befehls *ls* entnommen werden. Mit

```
# ls -l texte
```

wird der Inhalt des Verzeichnisses *texte* im »Langformat« ausgegeben. Diese Anzeige enthält alle wichtigen Informationen zu den angezeigten Files, so z.B. auch Angaben über Dateityp, Berechtigungen, Besitzer und Speicherplatzbedarf:

```
drwxr-xr-x   2 samulat  users        1024 Aug 30 10:38 draeger
-rw-r--r--   1 root     root         1352 Aug 14 14:13 smb.conf
-rwxr-----   1 samulat  users          61 Aug 13 18:10 s.bat
...
```

Dateiinhalt anzeigen – cat

Der Befehl *cat* zeigt den Inhalt der als Parameter übergebenen Datei an. Die Ausgabe erfolgt auf der »Standardausgabe«, normalerweise dem Bildschirm. Mit

```
# cat /etc/smb.conf
```

wird der Inhalt der Datei */etc/smb.conf* auf dem Bildschirm ausgegeben.

Dateiinhalt anzeigen – less

Etwas komfortabler als der Befehl *cat* erlaubt es *less*, den Inhalt von Dateien bildschirmseitenorientiert ausgeben zu lassen. Mit

```
# less /etc/smb.conf
```

wird die erste »Seite« von */etc/smb.conf* auf dem Bildschirm ausgegeben. Mit einfachen Tastenkommandos wird jetzt die weitere Anzeige gesteuert:

⏎	eine Bildschirmseite weiterblättern
J	eine Zeile weiter scrollen
K	eine Zeile zurück scrollen
u	halbe Bildschirmseite zurück scrollen
/*muster*	sucht die Zeichenkette *muster* und springt dorthin.
q	*less* beenden

Dateien kopieren – cp

Der Befehl *cp* kopiert eine oder mehrere Dateien, wobei Quell- und Zielangabe wiederum absolut oder relativ erfolgen können. Sollen mehrere Dateien kopiert werden, kann mit den Wildcards »?« und »*« gearbeitet werden. Mit

```
# cp /etc/smb.conf /texte
```

wird die Datei */etc/smb.conf* in das Verzeichnis */texte* kopiert. Der Befehl

```
# cp -R /etc/* /texte
```

kopiert alle Dateien und alle Unterverzeichnisse mit Inhalt aus */etc* in das Zielverzeichnis */texte*.

Dateien verschieben oder umbenennen – mv

Der Befehl *mv* kopiert den Inhalt der angegebenen Quelldatei in die Zieldatei und löscht danach die Quelldatei. Im Gegensatz zum Befehl *cp* wird die Quelldatei also nicht kopiert, sondern verschoben. Mit

```
# mv /texte/smb.conf /texte/t1.txt
```

wird die Datei */texte/smb.conf* in */texte/t1.txt* »verschoben«, also umbenannt. Auch der Befehl *mv* kennt Wildcards und die Option -R.

Dateien löschen – rm

Um Dateien aus dem Filesystem zu entfernen, kann der Befehl rm verwendet werden. Als Parameter wird der Name der zu löschenden Datei übergeben (relativ oder absolut), Wildcards sind erlaubt. Mit

```
# rm /texte/t*
```

werden alle Dateien aus dem Verzeichnis */texte* gelöscht, die mit einem »t« beginnen. Mit

```
rm -r /texte/
```

werden alle Dateien und Unterverzeichnisse in */texte* gelöscht. Im Gegensatz zum Befehl *rmdir* werden so auch nicht leere Unterverzeichnisse entfernt!

Verknüpfungen erstellen – ln

Der Befehl *ln* erstellt Verknüpfungen von Dateien, die sogenannten Links. Ein Link ist eine Referenz auf eine Datei, wobei die Referenz selbst wie eine Datei behandelt werden kann. Mit

```
# ln /etc/smb.conf /texte/link_smb
```

wird der Link */texte/link_smb* auf die Datei */etc/smb.conf* erstellt.

Arbeiten mit einem Dateimanager – mc

Die meisten der bisher beschriebenen kommandozeilenorientierten Aktionen mit Dateien und Verzeichnissen werden unter anderen Betriebssystemen mit »Dateimanagern« wie z.B. dem guten alten *Norton Commander* oder dem *Windows Explorer* durchgeführt. Ein vergleichbares Werkzeug ist auch unter Linux verfügbar: Es ist der *mc* (*midnight commander*), der auf einer Textkonsole läuft (Abbildung 2-20).

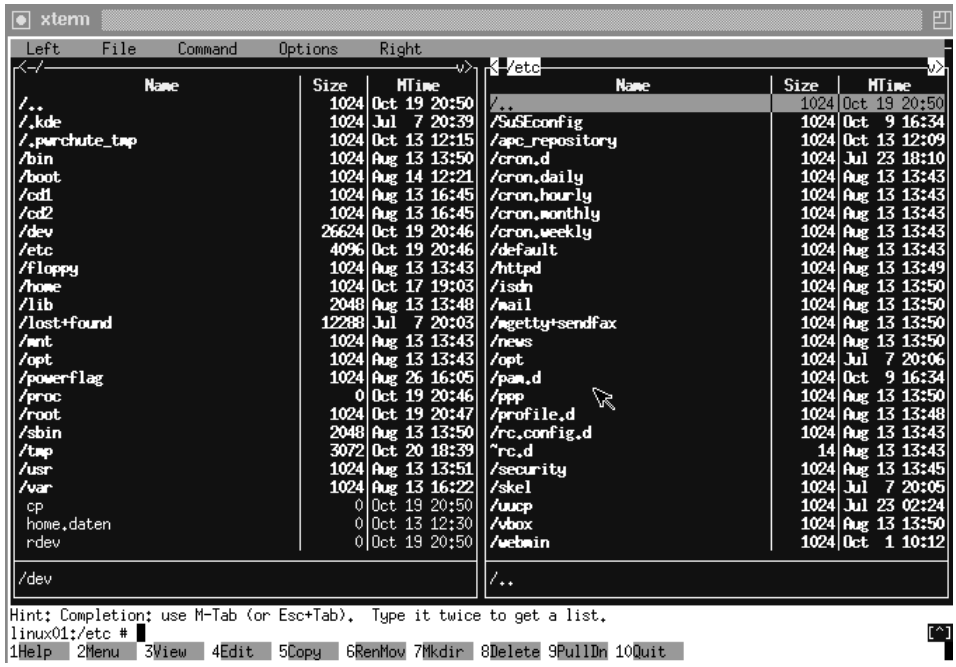


Abbildung 2-20 Das Programm mc

Das Programm *mc* ermöglicht das Anlegen, Löschen, Umbenennen und Kopieren von Verzeichnissen und Dateien. Textdateien können angesehen (*View*) oder bearbeitet werden (*Edit*), so dass Anpassungen an Konfigurationsdateien und Shell-Skripte schnell und unkompliziert erstellt werden können.

Dateien suchen – find

Der Befehl *find* ist das universelle Werkzeug zum Auffinden von Dateien und Verzeichnissen. Das Festlegen des Suchmusters, der Spezifikation, erfolgt über einen oder mehrere Tests, die wiederum über Operatoren zusammengefasst werden können. Die Tests werden wie Boolesche Ausdrücke behandelt, sie liefern entweder *true* oder *false*. Liefert der Test insgesamt *true*, so kann eine optional anzugebende Aktion ausgeführt werden. Der Befehl

```
find /usr -name "xinitrc" -print
```

sucht ab dem Verzeichnis */usr* nach der Datei *xinitrc* und zeigt alle Vorkommen mit dem kompletten Pfadnamen an.

Die Vielzahl der möglichen Testfunktionen, so z. B. Alter der Datei, Stand der Zugriffsrechte, Dateityp und viele andere mehr bis hin zu *find* machen es zu einem sehr mächtigen Werkzeug.

Dateien nach Ausdrücken durchsuchen – grep

Der Befehl *grep* sucht die vorgegebene Zeichenfolge in einer Datei und zeigt die »Treffer« an. Als Parameter werden *grep* die zu suchenden Zeichenfolge und die Liste der zu durchsuchenden Dateien übergeben. Mit

```
# grep "muster" /texte/*.txt
```

werden alle Dateien mit dem Namen ***.txt* im Verzeichnis */texte* nach der Zeichenfolge *muster* durchsucht. Der Befehl *grep* verfügt nicht nur über eine Anzahl von möglichen Steuerparametern, sondern er kann auch vielseitig bei der »Verdichtung« der von Shell-Kommandos gelieferten Ergebnissen verwendet werden. Mit

```
# ls -l /texte | grep txt
```

wird die Ausgabe des Befehls *ls* »umgeleitet« auf *grep* und dann gefiltert. Angezeigt werden dann nur noch die Zeilen, die die Zeichenfolge *txt* enthalten, hier also z. B. die Zeilen für alle Dateien mit dem Namen *.txt.

Zugriffsrechte setzen – chmod

Der Befehl *chmod* ändert die Zugriffsrechte der angegebenen Datei(en). Die drei möglichen Zugriffsrechte Lesen (= r, *r*ead) Schreiben (= w, *w*rite) und Ausführen (= x, e*x*ecute), können für alle Benutzer(= o, *o*ther), für den Besitzer (= u, *u*ser) und für die Gruppe (= g, *g*roup) geändert werden. Wird mit dem Befehl *ls – l* eine Dateiliste im Langformat erzeugt, so wird der aktuelle Stand der Zugriffsrechte jeweils am Zeilenanfang ausgegeben:

Typ	Besitzer			Gruppe			Alle Benutzer		
-	r	w	x	r	w	x	r	-	-

Tabelle 2-8 Dateizugriffsrechte

Sollen die aktuellen Berechtigungen mit *chmod* geändert werden, so sind grundsätzlich zwei Schreibweisen möglich: Es kann mit den eben beschriebenen Buchstabenkürzeln gearbeitet werden, wobei im ersten Parameter mit dem ersten Zeichen immer angegeben wird, für wen die Änderung gelten soll, danach folgt mit vorangestelltem »+« oder »-« die Berechtigung, die gesetzt (+) oder entzogen (-) werden soll. Mit

```
chmod o+w ps*
```

erhalten alle Benutzer für alle Dateien im aktuellen Verzeichnis, die mit »ps« beginnen, das Leserecht. Die zweite Schreibweise, die vor allem dann sinnvoller ist, wenn mehrere Berechtigungen gleichzeit geändert werden sollen, verwendet zur Kennzeichnug der Berechtigungen drei Zahlenwerte aus dem Oktalsystem. Mit der Zuordnung der Wertigkeiten r = 4, w = 2 und x = 1 entstehen damit für Besitzer, Gruppe und alle Benutzer insgesamt drei Zahlen in einer festen Reihenfolge, die jeweils zwischen 0 und 7 variieren können. Der Befehl

```
chmod 774 ps.x
```

setzt für die Datei *ps.x* die in Tabelle 2-7 dargestellten Berechtigungen. Weitere Informatioen zum Befehl *chmod* können den man-Pages entnommen werden.

Verzeichnisse einbinden -mount, umount

Damit unter Linux auf ein beliebiges Dateisystem zugegriffen werden kann, muss es auf ein bestimmtes Verzeichnis *aufgesetzt werden* (*mounten*). Dies läßt die Dateien in diesem Dateisystem so erscheinen, als ob sie in diesem Verzeichnis stünden, sodass direkt darauf zugegriffen werden kann.

> Es ist wichtig, dass Wechselmedien wie Disketten oder CD-ROM nicht aus dem Laufwerk entfernt oder gegen andere ausgetauscht werden, solange sie aufgesetzt sind.

Mit dem Befehl *mount* ohne Argumente kann festgestellt werden, welche Geräte wo aufgesetzt sind. Mit

```
linux01# mount
/dev/hda2 on / type ext2 (rw)
/dev/hda3 on /msdos type msdos (rw)
/dev/CD-ROM on /CD-ROM type iso9660 (ro)
/proc on /proc type proc (rw,none)
```

wird der aktuelle Zustand ausgegeben. Der Befehl *mount* ist standardmäßig nur dem Systemverwalter *root* erlaubt. Bei Bedarf kann aber auch anderen Benutzern

erlaubt werden, bestimmte *mount*-Aktionen ausführen zu dürfen. Sinnvoll kann dies z.B. für das Disketten- oder das CD-ROM-Laufwerk sein. Gesteuert wird dies über Einträge in der Konfigurationsdatei */etc/fstab*. Mit

```
/dev/fd0 /floppy msdos noauto,user,rw 0 0
/dev/CD-ROM /CD-ROM iso9660 ro,noauto,user 0 0
```

wird jedem Benutzer das Mounten von Diskette (*/dev/fd0*) und CD-ROM (*/dev/CD-ROM*) erlaubt. Allgemein hat der Befehl *mount* das Format

```
mount -t typ gerät  mount-point
```

Dabei ist der *Typ* der Name des Dateisystems (Tabelle 2-8). Das *Gerät* ist das physikalische Gerät, auf dem das Dateisystem existiert (der Gerätename in */dev*) und der *Mount-Point* (Aufsetzpunkt) ist das Verzeichnis, auf das dieses Dateisystem aufgesetzt wird (das Verzeichnis muss vor dem Aufruf von *mount* angelegt werden).

Typ	Beschreibung
ext2	Standard-Linux-Dateisystem
msdos	lokales Dateisystem für MS-DOS Partitionen
hpfs	lokales Dateisystem für HPFS-Partitionen
iso9660	lokales Dateisystem für Datenträger in CD-ROM-Laufwerken
nfs	Dateisystem für den Zugriff auf Partitionen entfernter (Remote) Server

Tabelle 2-9 Linux-Dateisysteme (Auszug)

Mit

```
mount -t ext2 /dev/hda2 /mnt
```

wird das Second-Extended-Dateisystem auf */dev/hda2* im Verzeichnis */mnt* aufgesetzt.

Neue Befehle erstellen – alias

Mit *alias* können Befehle oder vollständige Kommandozeilen einen neuen Namen erhalten oder abgekürzt werden. Mit

```
alias del='rm -i'
```

wird z.B. der aus MSDOS bekante Befehl *del* »nachgebildet«.

Nachrichten an alle Benutzer senden – wall

Wenn ein Linux-Server heruntergefahren wird, gibt das System automatisch eine Warnmeldung an alle angemeldeten Benutzer. Mit dem Befehl *wall* kann dies auch manuell erfolgen, um z.B. Warnungen oder wichtige Hinweise schnell an alle Benutzer zu senden. Nach

```
/usr/bin/wall
```

kann der Nachrichtentext geschrieben werden, die Eingabe wird mit Strg D beendet. Wird als Parameter eine Textdatei übergeben, so sendet *wall* deren Inhalt.

Umgebungsvariable anzeigen – env

Auch unter Linux spielen Umgebungsvariable der Shell eine große Rolle. Mit dem Befehl env können die aktuellen Zuordnungen angezeigt werden:

```
# env
PWD=/
PAGER=less
HOSTNAME=linux01
LD_LIBRARY_PATH=/opt/kde/lib
RC_LANG=german
LS_OPTIONS=-a -N --color=tty -T 0
ignoreeof=0
POVRAYOPT=-I/usr/lib/povray/include
QTDIR=/usr/lib/qt
OPENWINHOME=/usr/openwin
LESSKEY=/etc/lesskey.bin
LESSOPEN=|lesspipe.sh %s
MANPATH=/usr/local/man:/usr/man:/usr/X11R6/man:/usr/openwin/man
NNTPSERVER=news
KDEDIR=/opt/kde
LESS=-M -S -I
(...)
```

Ein einzelner Parameter kann mit dem Befehl *echo* angezeigt werden, dabei wird der Variablen ein Dollarzeichen vorangestellt:

```
# echo $ LESSKEY
/etc/lesskey.bin
```

Umgebungsvariable werden mit dem Befehl *set* eingerichtet und bei Bedarf auch im Inhalt geändert. Startet die aktuelle Shell eine weitere Shell, so werden nur die Variablen übergeben, die mit *export* dafür konfiguriert worden sind.

2.4.4 Benutzerverwaltung

Die Benutzerverwaltung auf dem Linux-Server ist Voraussetzung für die Absicherung der Ressourcen gegen unbefugten Zugriff. Das Einrichten und Bearbeiten von Benutzerkonten kann über Shell-Befehle wie *adduser* und *addgroup* erfolgen, wesentlich einfacher ist dies allerdings über YaST realisierbar.

Jeder Benutzer sollte mit dem Shell-Befehl *passwd* regelmäßig sein eigenes Kennwort ändern. Der Systemverwalter *root* kann bei Bedarf mit

```
# passwd <benutzername>
```

jedem Benutzer ein neues Kennwort zuweisen.

Für die Benutzerverwaltung stehen auch eine Vielzahl grafischer Oberflächen zur Verfügung, z. B. das Programm *kmuser* (Abbildung 2-21).

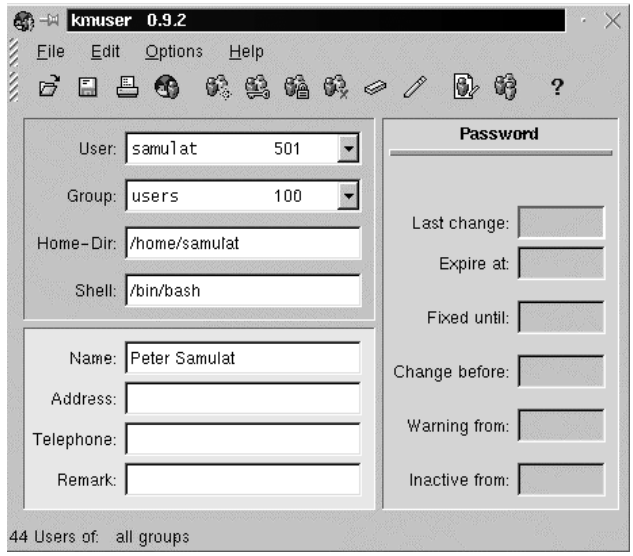


Abbildung 2-21 Benutzerverwaltung mit kmuser

Jeder Benutzer ist Mitglied mindestens einer Gruppe. Meistens erfolgt sogar eine Zuordnung zu mehreren Gruppen, die wiederum bestimmte Projekte oder Organisationsstrukturen repräsentieren. Standardmäßig könnern mindestens 65.536 Benutzer je Linux System eingerichtet werden, damit deutlich mehr als die (theoretisch) möglichen 20.000 Benutzer einer Windows-NT-Domäne.

Jeder Benutzer wird durch eine eindeutige ID (in Abbildung 2-21 die *UID* 501 für den Benutzer *samulat*) repräsentiert. Die systemweite Verwaltung von Benutzern und Gruppen erfolgt über eine Reihe von Dateien im Verzeichnis */etc*. (Tabelle 2-9).

Datei	Inhalt
passwd	Kennwortdatei mit den systembekannten Anwendern
passwd-	Die letzte Version der Kennwortdatei wird mit einem Minuszeichen ergänzt und aufbewahrt. Auch ältere Versionen der Dateien *group* und *shadow* werden so gesichert.
group	Gruppenverwaltung
shadow	enthält die verschlüsselten Informationen des Shadow-Kennwortsystems.

Tabelle 2-10 Konfigurationsdateien zur Benutzerverwaltung

Die Kennwortverwaltung verwendet standardmäßig die systemweit lesbare Datei */etc/passwd*. In diese Datei können nur Benutzer schreiben, die unter der UID des Systemverwalters arbeiten. Die erweiterte Kennwortverwaltung lagert die Kennwortdaten in die für den Benutzer nicht lesbare Datei */etc/shadow* aus. Damit wird vermieden, dass Unbefugte die verschlüsselten Kennwortinformationen stehlen und entschlüsseln können.

2.4.5 Dokumentation

Linux-Distributionen enthalten nicht nur eine beeindruckende Menge von Programmen, sondern auch eine Fülle an Dokumentationen und Hilfetexten. Auch im vorstehenden Text wurde schon mehrfach auf diese Systemdokumentationen hingewiesen, wenn es um weitere Informationen zu Befehlen ging, der Befehl *man* wurde als Befehl zum Abruf von Informationen aus dem *Online-Handbuch* bereits vorgestellt.

Nachfolgend soll ein kurzer Überblick darüber gegeben werden, welche Dokumentationen es gibt, wo diese zu finden sind und wie sie gelesen werden können. Im Vordergrund stehen dabei Dokumentationen, die speziell auf Linux oder auf unter Linux entwickelte Programme eingehen. Einige ausgewählte Dokumentationen, HOWTOs, FAQs und die wichtigsten RFCs (Stand: IV. Quartal 1999) finden Sie auf der CD-ROM im Verzeichnis */usr/doc*.

Dokumentation in elektronischer Form

Die meisten der Linux-spezifischen Dokumentationen sind zu finden in den Verzeichnissen */usr/doc*, */usr/doc/faq*, */usr/doc/howtos*, */usr/src* und */usr/lib*. Dort sind auch die Online-Dokumentationen zu diversen Programmen und Programmiersprachen zu finden, z.B. */usr/doc/perl* für Informationen zur Programmiersprache Perl. Oft werden die Dokumentation, READMEs, FAQs und HOWTOs aber auch im Installationsverzeichnis der jeweiligen Programme installiert.

Die Dokumentationen sind in der Regel in einem der in Tabelle 2-10 dargestellten vier Formate erstellt.

Typ	Beschreibung
ASCII	Diese Texte können direkt mit *less* oder einem beliebigen Editor gelesen werden. Mit Programmen wie *a2ps* oder *mpage* ist die Umwandlung in eine Postscript-Datei und deren Ausdruck möglich.
Postscript	Diese Dateien können z.B. mit dem X-Windows-Programm *ghostview* gelesen werden. Mit *gs* ist der Ausdruck möglich.
DVI	Diese mit LATEX erstellten Dateien können mit xvdi gelesen werden. Mit dvips ist die Umwandlung in eine Postscript-Datei möglich.
HTML	Diese Dateien können mit jedem WWW-Browser (*arena, lynx, netscape*, ...) gelesen werden.

Tabelle 2-11 Dateiformate für Dokumentationen

Dokumentationen werden im Einzelfall als komprimierte Datei abgespeichert, z. B. als *dateiname.gz*. Mit

```
# gunzip dateiname.gz
```

wird diese Datei dekomprimiert und durch die entkomprimierte Version (ohne die Dateierweiterung .gz) ersetzt. Mit *zless* kann der Inhalt der komprimierten Datei direkt angezeigt werden, die direkte Anzeige ist auch möglich mit

```
# zcat German-HowTo.gz | less
```

Archivdateien mit den Endungen *.tar* oder *.tgz* können mit dem später beschriebenen Befehl *tar* entpackt werden.

FAQ – Frequently asked Questiones

Viele Fragen zur Installation und zum Betrieb wurden bereits vielfach gestellt und beantwortet. Die FAQ-Texte stellen eine Sammlung dieser gerade für den Einsteiger sehr wichtigen Fragen und Antworten dar. Es gibt sie zu diversen Programmiersprachen, Netzwerk- und vielen anderen Unix-Themen. Die Texte sind im Regelfall in */usr/doc/faq* oder */usr/doc/FAQ* zu finden (Unter SuSE muss dazu das Paket *manyfaqs* aus der Serie *doc* installiert sein).

HOWTO – kompakte Anleitungen

HOWTO-Texte vermitteln Grundlagenwissen im Bereich Installation, Konfiguration und bei der Behebung von Hardwareproblemen. Unter Linux existieren zwei Formen von HOWTO-Texten: »normale« und Mini-Ausgaben. In der Regel findet man diese Texte in */usr/doc/HOWTO*. Die Texte stammen von unterschiedlichen Autoren und behandeln jeweils einen thematisch eng begrenzten Teilaspekt von Linux.

HOWTOs sind meistens recht knapp gehalten. Sie ähneln den FAQs, bestehen aber nicht aus Frage und Antwort. Allerdings enthalten viele HOWTOs am Ende einen FAQ-Abschnitt.

Aktuelle HOWTO-Texte sollten immer aus dem Internet geladen werden. Bezugsquellen sind z. B.

http://sunsite.unc.edu/LDP/HOWTO
ftp://sunsite.unc.edu/pub/Linux/docs /HOWTO
http://www.vip-com.de/linux/DE-HOWTO.html

Einige der HOWTOs sind bereits ins Deutsche übertragen worden (Paket *howtode* aus der Serie *doc*, installiert in */usr/doc/howto/de*). Die Homepage des *Deutschen Linux HOWTO Projektes*, das sich zum Ziel gesetzt hat, dem deutschsprachigen Linux-Anwender Dokumentation in seiner Muttersprache zur Verfügung zu stellen, ist z. B. unter *www.tu-harburg.de/dlhp/* zu finden.

RFCs

Alle technischen und organisatorischen Aspekte des Internet sind nicht in international gültigen Normen, sondern in *Request for Comments* RFC festgehalten, die in der Reihenfolge ihres Erscheinens durchnummeriert sind. RFC 2200 (früher RFC 1600) enthält eine Liste aller RFCs, die den Status eines anerkannten, offiziellen Internetstandards erlangt haben.

RFCs enthalten Detailinformationen zum IP- oder TCP-Protokoll, zur Struktur des Internet oder auch die Regeln und Besonderheiten für die Vergabe von IP-Adressen. Eine Auswahl der wichtigsten RFCs finden Sie auf der CD-ROM im Verzeichnis */usr/doc/rfc*.

LDP – Das Linux Documentation Project

Eine gute Informationsquelle stellen die sehr umfangreichen Texte dar, die als elektronische Bücher im Rahmen des *Linux Documentation Project* LDP enstanden sind. Verfügbar sind:

install-guide-3.2	Matt Welsh	Linux Installation and Getting Started Guide
khg-0.7	Michael K. Johnso	The Linux Kernel Hackers Guide
lpg-0.4	Sven Goldt, Sven van der Meer u.a.	Linux Programmers Guide
nag-1.0	Olaf Kirch	The Linux Network Administrators Guide
sag-0.5	Lars Wirzenius	The Linux System Administrators Guide
User-beta.1	Larry Greenfield	Linux Users Guide

Alle Bücher sind als *.ps-Dateien erhältlich, einige auch in einer HTML-Version. Informationen über LDP und die aktuellen Bücher können z.B. über *http://metalab.unc.edu/mdw/* bezogen werden.

Linuxforen Newsgroups

Im Internet haben sich eine Vielzahl von Newsgroups und Diskussionsforen gebildet, die spezielle Fragen rund um das Thema diskutieren, Lösungsvorschläge erstellen und konkret auch an der Weiterentwicklung von Linux mitarbeiten. Der Zugang zu diesen Foren ist z.B. über

www.linuxforen.de
www.linuxsearch.de

möglich. Es existieren dort sehr umfangreiche, nach Themen gegliederte Mailinglisten, aus denen wertvolle Informationen bezogen werden können. Auf Anfängerfragen wird in diesen Foren oft etwas barsch reagiert. Die Hinweise, zunächst in die weiteren Unterlagen wie FAQs und HOWTOs zu sehen , sind oft die einzige Antwort.

Linux-Zeitschriften

In den letzten Jahren haben sich eine Reihe von regelmäßig erscheinenden Zeitschriften etabliert, die sich auf das Thema Linux spezialisiert haben. Auch bieten immer mehr Fachzeitschriften qualitativ hochwertige Artikel zu diesem Thema an, so z. B. die Zeitschrift c't.

Über die Homepages dieser Zeitschriften, die teilweise den direkten Zugriff auf die Inhalte bereits erschienener Artikel ermöglichen, können sehr umfangreiche Informationen abgerufen werden, z. B. über:

Linux-Magazin	www.linux-magazin.de
c't	www.heise.de/ct/
iX	www.ix.de/ix/linux/
Linux Journal	www.linuxjournal.com

Online-Support zu Distributionen

Die Anbieter von Linux-Distributionen stellen sehr umfangreiche Sammlungen zur Lösung von Problemen bereit, die oft aus Kundenanfragen resultieren. Der Blick auf die Homepage von Anbietern wie

http://www.debian.de
http://www.redhat.de
http://www.suse.de

ist damit auch dann interessant, wenn diese Distribution nicht verwendet wird. Zu allen Paketen werden z. B. Bug-Reports veröffentlicht, dazu auch gleich die entsprechenden Patches oder Updates.

SuSE bietet eine Support-Datenbank, in der wiederholt auftretende Probleme und deren Lösung beschriebenen werden. RedHat stellt umfangreiche Archive diverser E-Mail-Listen bereit, die nach Stichwörtern durchsucht werden können.

Auch Systemhäuser oder weitere Anbieter von Dienstleistungen im Linux-Bereich veröffentlichen ihre Informationen im Internet, ein Online-Handbuch für Linux ist z. B. unter *http://www.lunetix.de/* zu finden.

man-Pages und Direkthilfen zu Befehlen

Die zum Standardumfang jeder Linux-Distribution gehörenden *man-Pages* sind umfangreiche Sammlungen von Programmbeschreibungen, die mit den später beschriebenen Befehlen wie *man* oder *xman* abgerufen werden können. Den Abschluß einer *man-Pages* bilden oft Tipps & Tricks, sowie die Namen der Konfigurationsdateien. Die *man-Pages* sind thematisch geliedert und können gezielt nach Stichworten durchsucht werden.

Fast alle Programme erlauben die Übergabe eines Parameters, der eine Kurzbeschreibung dieses Befehls auf dem Bildschirm ausgibt. »-help«, »-h«, »-?« sind die am häufigsten dazu verwendeten Optionen, »-v« oder »-V« geben meist die Versionsnummer aus. Viele Programme zeigen die Kurzbeschreibung nach einer Fehlbedienung automatisch an.

2.4.6 Kontrolle des Systemzustandes

Für die laufende Arbeit des Systemverwalters ist es wichtig, über leistungsfähige und einfach handzuhabende Werkzeuge zur Kontrolle und Steuerung des Systemzustandes des laufenden Linux-Systems verfügen zu können. Grundsätzlich können dafür textorientierte Shell-Kommandos verwendet werden, immer mehr sind aber auch grafische Werkzeuge unter X11 verfügbar.

Auswertung der Log-Dateien

Meldungen beim Systemstart

Mit ⇧ Bild↑ und ⇧ Bild↓ können unmittelbar nach dem Linux-Systemstart die dabei erzeugten Meldungen eingesehen werden:

```
LILO boot:
Loading Linux ............
Uncompressing Linux... OK, booting the kernel.
Linux version 2.2.10 (root@Mandelbrot.suse.de) (gcc version 2.7.2.3) #4 Tue Jul
20 17:01:36 MEST 1999
Detected 350801184 Hz processor.
Console: colour VGA+ 80x25
Calibrating delay loop... 349.80 BogoMIPS
Memory: 62864k/65472k available (1260k kernel code, 408k reserved (endbase
0x9f000), 896k data, 44k init)
VFS: Diskquotas version dquot_6.4.0 initialized
CPU: Intel Pentium II (Deschutes) stepping 02
(...)
```

Hier kann z.B. einfach überprüft werden, ob die für das Quoting notwendige Unterstützung im Kernel bereits aktiviert wurde (*Diskquotas*). Des Weiteren werden die Zuordnungen der Rechner-Hardware an die Gerätedateien in */dev* angezeigt.

Die Systemmeldungen werden auch in der Datei */var/log/boot.msg* gespeichert und können so auch zu einem späteren Zeitpunkt ausgewertet werden:

```
(...)
<6>(scsi0) <Adaptec AIC-7890/1 Ultra2 SCSI host adapter> found at PCI 6/0
<6>(scsi0) Wide Channel, SCSI ID=7, 32/255 SCBs
<6>(scsi0) Downloading sequencer code... 374 instructions downloaded
<4>scsi0 : Adaptec AHA274x/284x/294x (EISA/VLB/PCI-Fast SCSI) 5.1.19/3.2.4
<4>       <Adaptec AIC-7890/1 Ultra2 SCSI host adapter>
<4>scsi : 1 host.
```

```
<6>(scsi0:0:0:0) Synchronous at 80.0 Mbyte/sec, offset 15.
<4>  Vendor: IBM       Model: DDRS-34560D       Rev: DC1B
<4>  Type:   Direct-Access                      ANSI SCSI revision: 02
<4>Detected scsi disk sda at scsi0, channel 0, id 0, lun 0
<6>(scsi0:0:2:0) Synchronous at 20.0 Mbyte/sec, offset 16.
<4>  Vendor: PIONEER   Model: CD-ROM DR-U16S    Rev: 1.01
<4>  Type:   CD-ROM                             ANSI SCSI revision: 02
(...)
```

Die Datei /var/log/messages

In */var/log/messages* werden Systemmeldungen gespeichert, die für die Fehlersuche sehr hilfreich sein könen. Mit

```
tail -f /var/log/messages
```

werden über den Befehl *tail* die jeweils letzten 10 Zeilen der Log-Datei auf dem Bildschirm ausgegeben. Die Anzeige wird automatisch aktualisiert. Die Steuerung, welche Meldungen wo und in welchem Umfang ausgegeben werden, erfolgt über die Datei */etc/syslog.conf*. Standardmäßig hat diese folgenden Inhalt (Auszug):

```
# /etc/syslog.conf - Configuration file for syslogd(8)
#
# print most on tty10
kern.warn;*.err;authpriv.none   /dev/tty10
*.emerg                         *

(...)

#
# Warnings in one file
#
*.warn                    /var/log/warn
#
# save the rest in one file
#
*.*;mail.none;news.none     /var/log/messages
```

Die vorstehenden Einträge in */etc/syslog.conf* sorgen dafür, dass eine Reihe von Fehlermeldungen auf Konsole 10 (tty10) ausgegeben wird und dass zusätzlich */var/log/messages* den Großteil der Meldungen aufnimmt. Während der Systemeinrichtung kann dies sehr hilfreich sein, während des Systembetriebes ist es aber eher umständlich, sich jedesmal erst auf dem Server einloggen zu müssen, um dann mit dem Befehl *tail* die letzten aktuellen Meldungen einsehen zu können.

Empfehlenswert ist es, im laufenden System eine der Textkonsolen für die Ausgabe der Systemmeldungen zu reservieren. Um z.B. Die Konsole 5 dazu zu verwenden, ist es lediglich notwendig, in *etc/syslog.conf* die Zeile

```
*.* /dev/tty5
```

einzufügen. Danach werden alle Systemmeldungen (*.*) auf der Konsole 5 angezeigt. Bei Bedarf können die anderen Zeilen auskommentiert werden. Soll der Umfang der Meldungen noch weiter reduziert werden, so ist der Medungstyp nicht mehr »*.*«, sondern wird entsprechend Tabelle 2-11 eingeschränkt:

Syslog-Dienst	Inhalt der Meldungen
authpriv	sicherheitsrelevante Meldungen
cron	Meldungen des cron- und at-Daemons
daemon	Meldungen aller anderen Daemonen
kern	Kernel-Meldungen
local0 – local7	lokale Meldungen für den eigenen Gebrauch
lpr	Meldungen des Druckerdienstes
mail	Meldungen des mail-Dienstes
news	Meldungen des news-Dienstes
syslog	Meldungen, die syslog selbst generiert
user	allgemeine Usermeldungen
uucp	Meldungen des uucp-Dienstes
Die Dringlichkeitsstufen mit abnehmender Priorität	
emerg	Das System oder Teile davon funktionieren nicht mehr.
alert	Sofortiges Eingreifen ist erforderlich.
crit	kritische Fehlerbedingung
error	Fehlerbedingung
warning	Warnungen
notice	normale, aber wichtige Hinweise
info	allgemeine Informationen
debug	Meldungen, die von Programmen zur Fehlersuche ausgegeben werden

Tabelle 2-12 Protokolle von syslog [Wöjü99]

Über *man syslog.conf* können weitere wichtige Konfigrationsmöglichkeiten abgefragt werden. So ist es z.B. auch möglich, Meldungen bestimmter Dringlichkeitsstufen an bestimmte Benutzer (also nicht nur an den *root*) zu senden oder direkt an andere Server weiterzuleiten.

Logdateien anzeigen – ktail

Das KDE-Programm *ktail* ermöglicht das Überwachen der diversen Logdateien in einer einzigen grafischen Oberfläche (Abbildung 2-22). Die Konfiguration erlaubt die Auswahl verschiedener Log-Dateien, deren Änderungen erkannt und angezeigt werden.

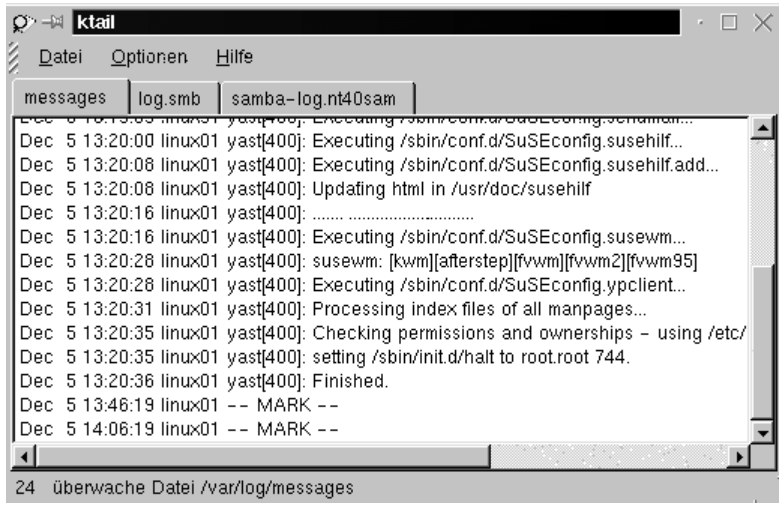


Abbildung 2-22 Log-Dateiüberwachung mit ktail

Auswertung und Steuerung der aktuellen Systemaktivität

Aktive Programme werden unter Linux als Prozesse bezeichnet. Prozesse enstehen durch den als *fork&exec* bezeichneten Mechanismus.

Ein Programm (der Parent-Process) fertigt durch den Aufruf der *fork*-Funktion einen Klon (den Child-Process) als exakte Kopie seiner selbst. Diese Kopie läuft in einer Kopie der Umgebung (*Environment*) des Parent-Prozesses. Schon zu diesem Zeitpunkt haben beide Prozesse unterschiedliche PID (*Process Identifier*). Nun wird der Klon des aufrufenden Prozesses mittels *exec* durch das gewünschte Programm ersetzt.

Die PID identifiziert jeden Prozess eindeutig, ist aber über die Laufzeit des Systems nicht eindeutig an einen Prozess gebunden:

- Jeder Prozess verfügt über eine eigene PID.
- Jeder PID ist zu jedem Zeitpunkt nur einmal vorhanden.
- Terminierte Prozesse geben ihre PID wieder frei.
- Ein neuer Prozess kann zu einem späteren Zeitpunkt die PID eines bereits beendeten Prozesse wieder erhalten.

Für den Systemverwalter ist es wichtig, den aktuellen Prozesszustand darstellen und bei Bedarf beeinflussen zu können. Die dazu zur Verfügung stehenden Werkzeuge sollen jetzt vorgestellt werden.

Anzeige der aktiven Prozesse – ps, top und pstree

Der Befehl *ps* informiert über den aktuellen Zustand von Prozessen. Mit

```
ps ax
```

erhält der Systemverwalter eine Prozessliste für alle Anwender (a), erweitert um alle Prozesse, die an kein Terminal gebunden sind (x). Ein Beispiel dazu zeigt Listing 2-6.

```
  PID TTY STAT TIME COMMAND
    1  ?  S    0:21 init
    2  ?  SW   0:00 (kflushd)
    3  ?  SW   0:00 (kupdate)
    4  ?  SW   0:00 (kpiod)
    5  ?  SW   0:00 (kswapd)
    6  ?  SW   0:00 (md_thread)
   74  ?  S    0:00 /usr/sbin/rpc.ugidd
   82  ?  S    0:00 /usr/sbin/syslogd
   86  ?  S    0:00 /usr/sbin/klogd -c 1
  114  ?  S    0:00 /usr/sbin/rpc.mountd
  117  ?  S    0:00 /usr/sbin/rpc.nfsd
  138  ?  S    0:00 /usr/sbin/inetd
  154  ?  S    0:00 /usr/sbin/lpd
  156  ?  S    0:00 /usr/sbin/httpd -f /etc/httpd/httpd.conf
  169  ?  S    0:00 sendmail: accepting connections on port 25
  182  ?  S    0:00 /usr/sbin/cron
  189  ?  S    0:00 /usr/sbin/nscd
  190  ?  S    0:00 /usr/sbin/nscd
  191  ?  S    0:00 /usr/sbin/nscd
  192  ?  S    0:00 /usr/sbin/nscd
  193  ?  S    0:00 /usr/sbin/nscd
  194  ?  S    0:00 /usr/sbin/nscd
  195  ?  S    0:00 /usr/sbin/nscd
  196  ?  S    0:01 /usr/sbin/nmbd -D
  198  ?  S    0:00 /usr/sbin/nmbd -D
  199  ?  S    0:00 /usr/sbin/smbd -D
  203  ?  S    0:00 /usr/sbin/dhcpd -q eth0
  207   1 S    0:00 login -- root
  208   2 S    0:00 login -- root
  209   3 S    0:00 /sbin/mingetty tty3
  210   4 S    0:00 /sbin/mingetty tty4
  211   5 S    0:00 /sbin/mingetty tty5
  212   6 S    0:00 /sbin/mingetty tty6
  213   1 S    0:00 -bash
 1671   2 S    0:00 -bash
 1771  S1 S    0:00 gpm -t ms -m /dev/mouse
 4333  ?  S    0:00 minicom -s
 7581  ?  S    0:00 twm
 7582  ?  S    0:00 -bash
 7838  ?  S    0:00 xterm
```

```
 7839  ?  S    0:00 bash
 7873  ?  S    0:00 mc
 7875  ?  S    0:00 bash -rcfile .bashrc
26709  ?  S    0:03 /usr/sbin/smbd -D
26876   1 R    0:00 ps ax
   70  ?  S    0:00 /sbin/portmap
  130  ?  S    0:00 /usr/sbin/atd
```

Listing 2-6 Prozessliste (Beispiel) mit dem Befehl ps

Die Spalte PID zeigt die aktuelle Prozessidentifikationsnummer, die z.B. für das später beschriebene manuelle Abbrechen eines laufenden Prozesses wichtig ist. TTY zeigt, zu welchem Terminal dieser Prozess gehört. Ein »?« in dieser Spalte steht für einen Prozess ohne Bindung an ein Terminal. Die Spalte STAT kennzeichnet den aktuellen Status des Prozesses:

R	laufend
S	schlafend
D	nicht störend schlafend
T	angehalten
Z	Zombie
W	Prozess belegt keine Speicherseiten

Der Befehl *ps* ohne weitere Parameter zeigt dem Anwender nur die »eigenen« Prozesse. Wie schon dargestellt, kann bei Programmaufruf eine Reihe von unterschiedlichen Optionen übergeben werden. Viele spezielle Ausgabeformate lassen sich durch Kombinationen von Optionen (z.B. »ax«) erzielen. Auch an dieser Stelle kann für weitere Informationen nur auf die man-Pages zum Befehl *ps* verwiesen werden.

Eine kontinuierliche Ausgabe der Prozessliste (eine Art Systemmonitor) erhält der Anwender durch *top*. Dieser Befehl listet die ressourcenintensivsten Prozesse auf. Periodisch wird auf dem Bildschirm eine zweiteilige Tabelle ausgegeben (Listing 2-7). Deren erster, oberer Teil enthält eine grobe Statistik der Prozessinformationen: wie die mittlere Systemlast, Anzahl der User, die CUP-States und Speicherinformationen. Der zweite Teil zeigt die einzelnen Prozesse und beschreibt deren »Verbrauch« detailliert in mehreren Spalten. Die Anzahl der dargestellten Prozesse hängt von der Größe des verwendeten Terminals (Zeilenanzahl) ab.

```
2:19pm  up 9 days, 18:30,  4 users,  load average: 1.00, 1.00, 1.00_
56 processes: 54 sleeping, 2 running, 0 zombie, 0 stopped
CPU states: 87.2% user, 12.6% system,  0.0% nice,  0.3% idle
Mem:   62912K av,  60948K used,   1964K free,  37300K shrd,  18540K buff
Swap: 128516K av,      0K used, 128516K free                 20632K cached
```

```
  PID USER     PRI  NI  SIZE  RSS SHARE STAT  LIB %CPU %MEM   TIME COMMAND
  295 root      12   0  1052 1052   816 R       0 97.8  1.6 13924m StartNewX
26922 root       1   0   772  772   604 R       0  1.9  1.2   0:00 top
    1 root       0   0   196  196   168 S       0  0.0  0.3   0:21 init
    2 root       0   0     0    0     0 SW      0  0.0  0.0   0:00 kflushd
    3 root       0   0     0    0     0 SW      0  0.0  0.0   0:00 kupdate
    4 root       0   0     0    0     0 SW      0  0.0  0.0   0:00 kpiod
    5 root       0   0     0    0     0 SW      0  0.0  0.0   0:00 kswapK
    6 root       0   0     0    0     0 SW      0  0.0  0.0   0:00 md_thread
   70 bin        0   0   404  404   320 S       0  0.0  0.6   0:00 portmap
   74 root       0   0   484  484   396 S       0  0.0  0.7   0:00 rpc.ugidd
   82 root       0   0   648  648   536 S       0  0.0  1.0   0:00 syslogd
   86 root       0   0   796  796   392 S       0  0.0  1.2   0:00 klogd
  114 root       0   0   756  756   620 S       0  0.0  1.2   0:00 rpc.mountd
  117 root       0   0   748  748   616 S       0  0.0  1.1   0:00 rpc.nfsd
  130 at         0   0   552  552   456 S       0  0.0  0.8   0:00 atd
  138 root       0   0   572  572   480 S       0  0.0  0.9   0:00 inetd
  154 root       0   0   624  624   528 S       0  0.0  0.9   0:00 lpd
  156 root       0   0  1592 1592  1488 S       0  0.0  2.5   0:00 httpd
```

Listing 2-7 Bildschimausgabe des Shellbefehles top

Mit der Taste [H] erhält man die Hilfefunktion zu *top* [Q] beendet das Programm. Mit weiteren Steuertasten können z.B. auch Prozesse in ihrer Priorität verändert (*nice*) oder beendet werden (*kill*).

Eine Ausgabe der Prozessinformationen, die auch die Abhängigkeiten in einer Baustruktur aufgelöst darstellt, liefert der Befehl *pstree*. Normalerweise stellt *pstree* alle Prozesse dar, durch Angabe einer Prozess-ID (PID) kann die Anzeige auf alle nur von diesem Prozess abstammenden Prozesse eingeschränkt werden. Mit der Option »-G« (*graphic*) werden VT100-kompatible Grafikzeichen zur Darstellung der Baumstruktur verwendet, was eine erheblich verbesserte grafische Darstellung bewirkt. Mit

```
watch pstree
```

wird die Anzeige von *pstree* alle zwei Sekunden (Standardwert von *watch*) aktualisiert.

Grafischer ps Befehl – xzap, kpstree

Das Programm *xzap* aus dem Paket *xap* ist die X-Windows-basierte Version des *ps*-Befehls, erweitert um Steuerfunktionen, wie sie *kill* zur Verfügung stellt (Abbildung 2-23).

Eine Besonderheit ist, dass der *ps*-Befehl direkt verwendet wird. Dazu können auch Befehlsoptionen angegeben werden, ihr Eingabefeld ist rechts oben in der Zeile mit den Buttons. Abbildung 2-23 zeigt als Beispiel die Option »ax«. Das Programm *xzap* verfügt über eine Online-Hilfefunktion (*xzap help*).

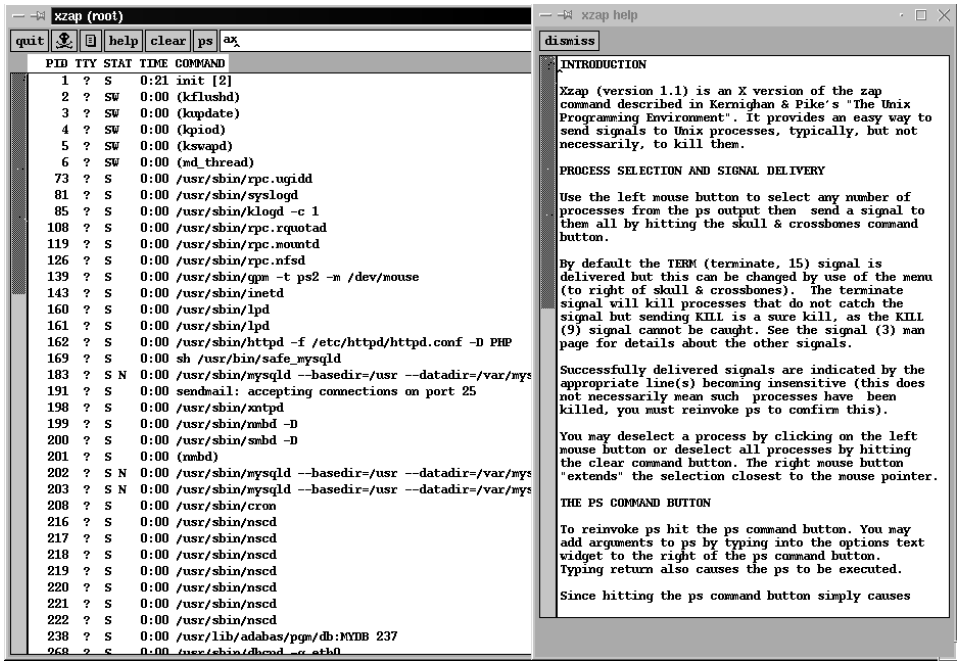


Abbildung 2-23 Das Programm xzap

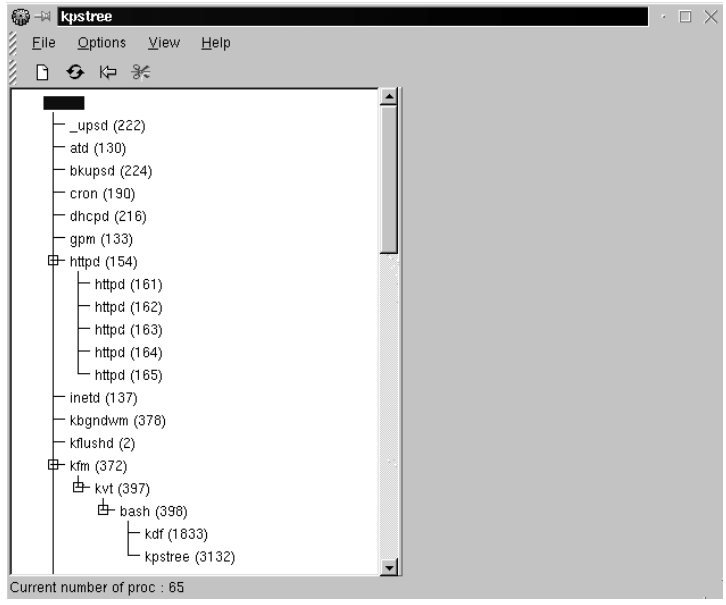


Abbildung 2-24 kpstree

Prozess beenden – kill

Der Befehl *kill* sendet ein Signal an einen bestimmten Prozess. Dieses Signal kann aus der Liste der verfügbaren Signale

```
# kill -l
 1) SIGHUP       2) SIGINT       3) SIGQUIT      4) SIGILL
 5) SIGTRAP      6) SIGABRT      7) SIGBUS       8) SIGFPE
 9) SIGKILL     10) SIGUSR1     11) SIGSEGV     12) SIGUSR2
13) SIGPIPE     14) SIGALRM     15) SIGTERM     17) SIGCHLD
18) SIGCONT     19) SIGSTOP     20) SIGTSTP     21) SIGTTIN
22) SIGTTOU     23) SIGURG      24) SIGXCPU     25) SIGXFSZ
26) SIGVTALRM   27) SIGPROF     28) SIGWINCH    29) SIGIO
30) SIGPWR
```

ausgewählt werden. Normalerweise wird *kill* eingesetzt, um Prozesse zu beenden, die sich nicht mehr selbständig beenden. Mit

```
kill -9 7839
```

wird dem Prozess mit der *PID = 7839* das Signal *SIGKILL* gesendet, der entsprechende Prozess wird beendet. Standardmäßig sendet *kill* das Signal *SIGTERM*, mit der Option »-9« werden auch Prozesse beendet, die dieses Signal ignorieren.

Der Befehl *kill* beendet also im Gegensatz zu anderen Betriebssystemen nicht aktiv Prozesse. Unter Linux übernimmt nur das Betriebssystem diese Aufgabe, indem es signalgesteuert bestimmte Aktionen auslöst, die u.a. auch zur geordneten Beendigung eines Prozesses führen (*TERM*) oder auch zum bedingungslosen Abbruch (*KILL*). In diesem Szenario übernimmt das Programm *kill* nur die Rolle einer Anwenderschnittstelle, die entsprechende Signale an das Betriebssystem schickt.

Die X-Windows-Variante des Befehls *kill* ist *xkill*: Mit

```
# xkill
Select the window whose client you wish to kill with button 1..
```

kann mit dem Mauszeiger ein beliebiges Objekt auf der X-Windows-Oberfläche durch einfaches Anklicken mit der linken Maustaste beendet werden.

Prozesspriorität setzen – nice

Mit dem Shell-Befehl *nice* wird Priorität von Prozessen innerhalb des Systems festgelegt. Die Priorität bestimmt, wieviel CPU-Zeit einem Prozess anteilmäßig zugeteilt wird. Eine Verringerung ist z.B. sinnvoll, um mit rechenintensiven Prozessen nicht das Gesamtsystem unnötig zu verlangsamen; eine Erhöhung könnte bei Dämonenprozessen erforderlich sein und darf nur von Systemverwalter vorgenommen werden.

Die tatsächliche Priorität eines Prozesses berechnet sich aus der Summe der voreingestellten Prioritäten und dem angegebenen *nice*-Wert.

Standardmäßig ordnet *nice* einem Befehl die Priorität 10 zu, positive *nice*-Werte verringern die Bearbeitungspriorität (das Maximum ist 19), negative Werte erhöhen sie. Nur der Systemverwalter kann Prozesse mit einem negativen *nice*-Wert starten, das Maximum liegt bei -20. Ein *nice*-Wert von +20 bewirkt, dass der entsprechende Prozess nur dann ausgeführt wird, wenn das System ansonsten unbeschäftigt ist.

Der Befehl *nice* ohne weitere Parameter gibt die aktuelle Priorität des angegebenen Prozesses aus. Mit

```
user@linux01 > nice  -n 19 gcc bigprogram.c
root@linux01 # nice -n -10 inetd
```

werden das Programms *gcc* und der Daemon *inetd* von einem Benutzer bzw. vom Systemverwalter mit der angegebenen Priorität gestartet.

Mit dem Befehl *renice* kann die Bearbeitungspriorität von bereits laufenden Prozessen verändert werden. Der Benutzer kann *renice* nur auf seine eigenen Prozesse anwenden, der Systemverwalter kann alle Prozesse bremsen oder beschleunigen. Die einfachste Anwendung erfolgt durch die Spezifikation einer PID (Option *-p*):

```
Root@linux01 # renice +10 -p 22445
22445: old priority 0, new priority 10
```

Der KDE Taskmanager – ktop

Das KDE-Programm *ktop* stellt Funktionen zu Verfügung, die an den Taskmanager von Windows NT angelehnt sind. Im Performance-Meter können z. B. die Prozessorauslastung und der Speicherverbrauch der letzten Minuten angezeigt werden (Abbildung 2-25).

Ähnlich wie beim Linux-Shell-Befehl *top* kann eine Prozessansicht dargestellt werden, in der alle aktiven Prozesse dieses Rechners enthalten sind. Es gibt auch die Möglichkeit, die Prozesse ähnlich wie bei *pstree* anzuzeigen, um zu erkennen, welcher Prozess wie aufgerufen und gestartet wurde.

Kpm

Diese Werkzeug zum Prozessmanagement fasst die Funktionen von *top*, *Free* und einen Systemmonitor in einer grafischen Oberfläche zusammen.

Systemüberwachung – xosview, procview

Das grafische Tool *xosview* von Mike Romberg und Brian Grayson zeigt die aktuellen Systemaktivitäten als Balkendiagramme. Die Anzeige kann bei Bedarf angepasst werden.

Tabelle 2-13 zeigt die Bedeutung der standardmäßig angezeigten Werte im Einzelnen.

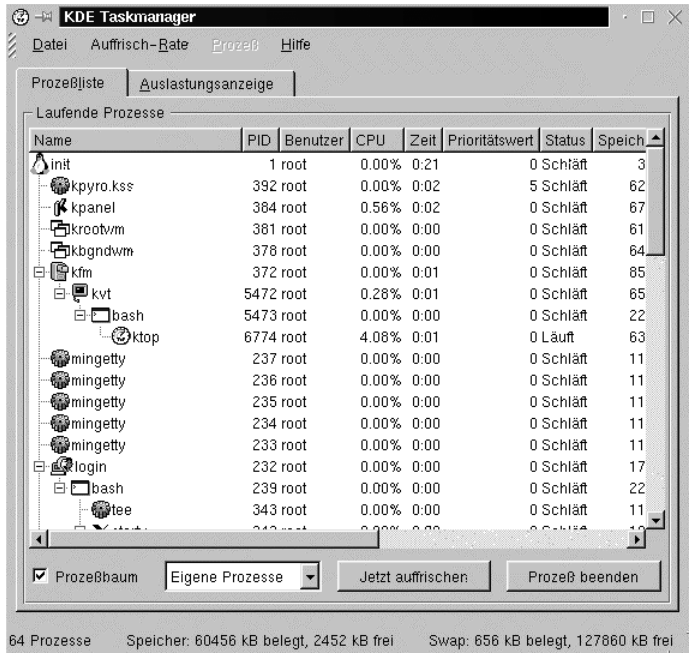


Abbildung 2-25 ktop

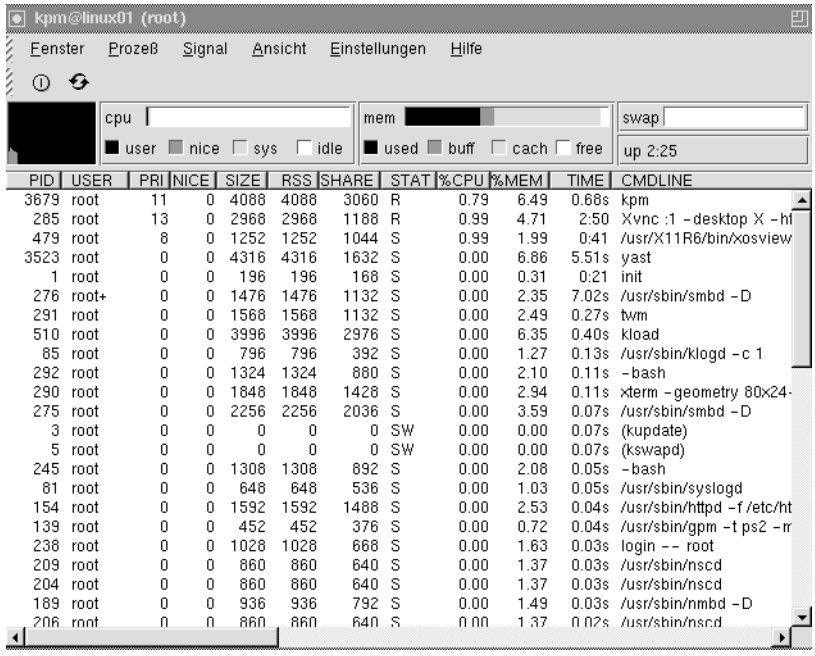


Abbildung 2-26 Das Programm kpm

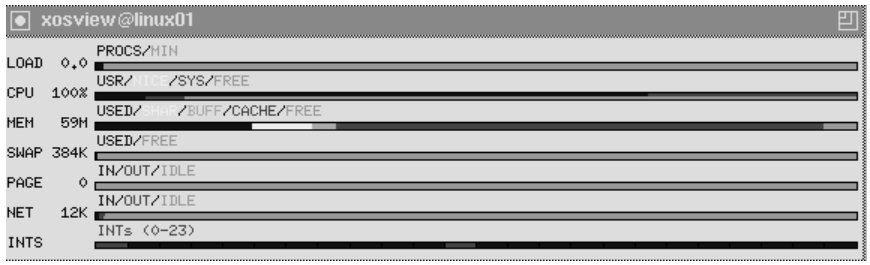


Abbildung 2-27 Das Programm xosview

Wert	Beschreibung
LOAD	Auslastung des gesamten Systems als Zahlenwert. Die aktuellen Aktivitäten von CPU, der Netzwerkkarte und der I/O-Schnittstellen (Festplatte, ...) werden summiert und als ein Gesamtwert dargestellt.
CPU	Anzeige der CPU-Zeit nach Benutzerprozessen/Systemaktivitäten
MEM	Belegung des Systemspeichers. Dabei ist wichtig, dass Linux den Speicher komplett als Cache und Puffer belegt. Interessanter Teil dieser Anzeige ist damit USED+SHARE. Dieser Teilbalken zeigt die wirkliche RAM-Auslastung. Der Balken SWAP gibt die Belegung des Swap-Speichers wieder, zeigt also an, wieviel tatsächlich ausgelagert wurde.
PAGE	zeigt, ob und wie viele Informationen aus dem RAM in den Swap-Speicher gelegt oder herausgenommen wurden.
NET	aktuelle Netzlast
INTS	Anzeige der aktuellen Systemunterbrechungen

Tabelle 2-13 Angezeigte Werte in xosview

Zum Programm *xosview* existiert eine sehr gute Dokumentation, die über

```
# man xosview
```

abgerufen werden kann und Details zu den vielfältigen Konfigurationsmöglichkeiten zeigt. Ein ähnliches Programm, wenn auch etwas älter und mit deutlich weniger Funktionen ist *xcpustat*.

Ein kleiner Systemmonitor, der die aktuellen Meßwerte für CPU-Auslastung und andere Systemparameter über die Zeitachse grafisch dargestellt, ist das Programm *procmeter*:

Über eine einfache Konfiguration wird vorgegeben, welche Meßwerte angezeigt werden sollen, die Auswahl erfolgt über die in Abbildung 2-28 am linken Rand gezeigte Liste. Ähnliche Leistungen zeigt das Programm *kload*, das über unterschiedliche Farben auch die Einhaltung von vorgegebenen Grenzwerten anzeigt.

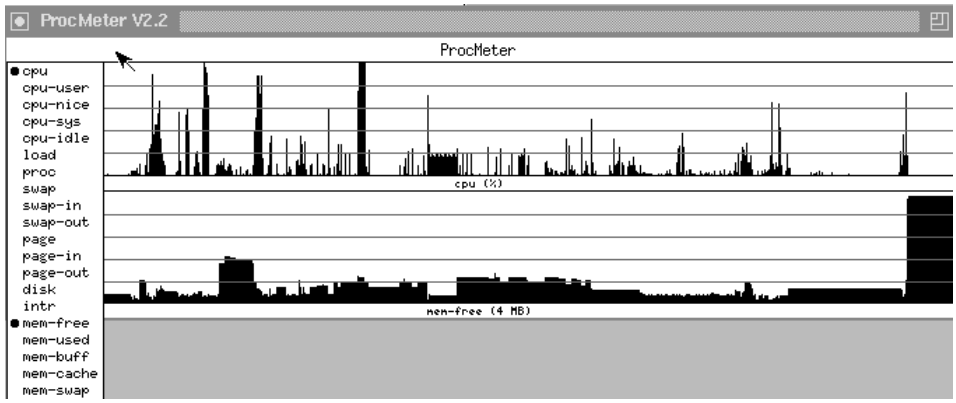


Abbildung 2-28 Das Programm procmeter

Festplatten-Speicherplatz überwachen

Kdf und Kdu

Kdf und Kdu sind die grafischen Oberflächen der Shell-Befehle *df* (*disk free*) und *du* (*disk usage*). Sie zeigen die aktuelle Belegung von Festplatten, den Speicherplatzbedarf von Dateien und Verzeichnissen und geben Auskunft über freien Speicherplatz.

Kdf stellt auf Basis der Einträge in */etc/fstab* alle verfügbaren Medien in einem Fenster dar: (Abbildung 2-29).

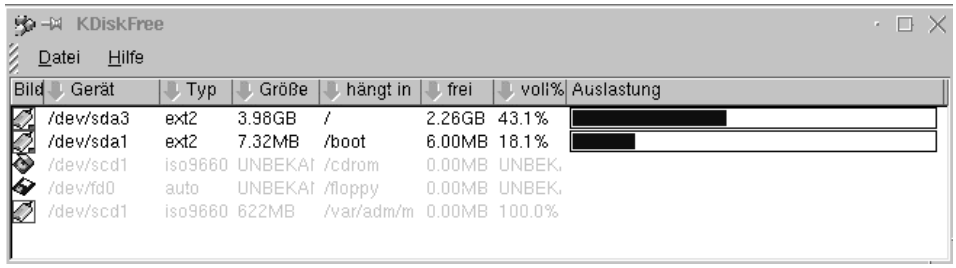


Abbildung 2-29 Kdf

Gemonuntete Dateisysteme werden in normaler Schrift angezeigt, sofern die Belegung unter 95 Prozent liegt. In diesem Fall erscheinen die Einträge in roter Schrift; zusätzlich wird eine Warnung ausgegeben. Nicht gemountete Dateisysteme werden grau angezeigt; die Angabe der Belegung wird nicht durchgeführt.

Per Doppelklick auf eines der angezeigten Dateisysteme wird der Mount-Status geändert.

kdu berechnet den gesamten Speicherplatzbedarf eines Verzeichnis (einschließlich der Unterverzeichnisse).

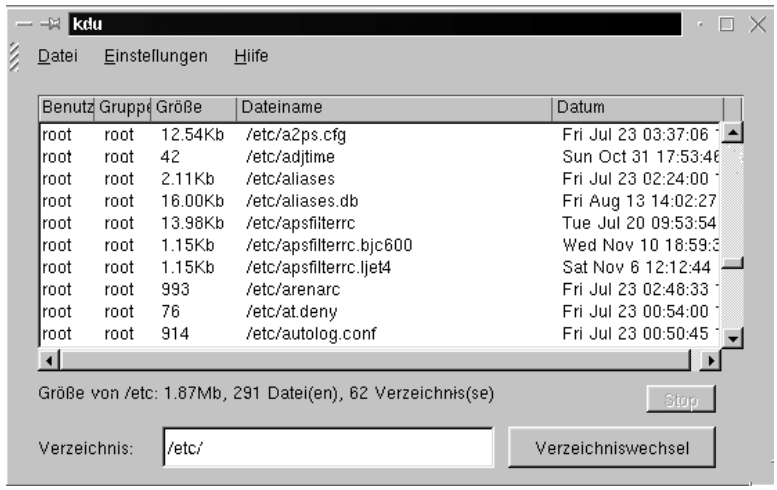


Abbildung 2-30 Die Speicherplatzbelegung anzeigen mit kdu

Die ausreichend große verbleibende freie Speicherkapazität aller Festplatten ist für den Systembetrieb sehr wichtig sollte daher regelmäßig geprüft werden. Dies ist z.B. mit dem Shell-Befehl *df* möglich:

```
# df
Filesystem      1024-blocks  Used Available Capacity Mounted on
/dev/sda3         4175756 1519273  2440409     38%   /
/dev/sda1            7496     957     6139     13%   /boot
```

Steigt die Belegung innerhalb kurzer Zeit stark an, so könnte dies z.B. eine der nachstehenden Ursachen haben:

Benutzer haben sehr große Dateien gespeichert, typischerweiser auch als Anhang an E-Mail (Verzeichnids */var/spool/mail*). Um gezielt nach großen Dateien suchen zu können, kann z.B. der Befehl *find* mit einer Option verwendet werden, die nur Dateien anzeigt, die die angegebene Größe überschreiten. Mit

```
# find /var/spool/mail -size +1000k
```

werden alle Dateien in */var/spool/mail* angezeigt, die über ein 1 Mbyte groß sind. Bei Bedarf und Abklärung mit dem Benutzer können diese Dateien dann gezielt gelöscht werden. Treten solche Probleme häufiger auf, sollte eine auf den Benutzer oder auf Gruppen bezogene Begrenzung der Speicherkapazität auf dem Datenträger (Quoting) eingerichtet werden. Speziell zur Begrenzung der Größe von E-Mail-Dateien können die Maximalwerte auch bei der Konfiguration der entsprechenden Mail-Serverdienste vorgenommen werden.

Problematisch können auch alte Dateien in den Verzeichnissen */tmp*, */temp* und */usr/tmp* werden, wenn hier nicht regelmäßig gelöscht wird. Zweckmäßigerweise werden diese und vergleichbare Verzeichnisse regelmäßig geleert. Die Programierung solcher zeitgesteuerten Aktionen wird später ausführlich beschrieben.

Wird auf diesem Server ein Internetzugang bereitgestellt, belegen die vom Proxy-Server angelegten Cache- und Log-Dateien, z.B. in */usr/local/etc/httpd/logs* und */usr/local/squid/logs/* oft sehr viel Speicherplatz. Auch bei hoher Auslastung dieser Serverdienste sollte es allerdings nicht zu einer Überlastung kommen, bei Bedarf muss das automatisch erfolgende Löschen dieser Dateien in kürzeren Zeitintervallen erfolgen. Der Proxy-Server muss dazu entsprechend konfiguriert werden.

Auch in den privaten Benuzterverzeichnissen führen große Datenmengen schnell zu einer unnötig starken Belegung des Festplatte. Auch hier ist neben der regelmäßigen Kontrolle, insbesondere auf sehr große Dateien, das Einrichten des Quotings ein sehr wichtiger Arbeitsschritt.

KFile System Control

Stark an Aussehen und Funktion der entsprechenden Bedienungsoberflächen von Windows 9x/NT angelehnt ist das *Kfile System Control*, mit dem die aktuelle Festplattenbelegung als Tortendiagramm dargestellt wird (siehe Abbildung 2-31).

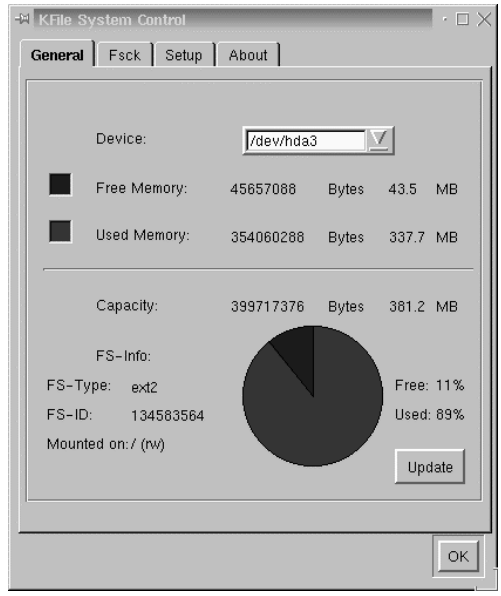


Abbildung 2-31 Das Programm kfile

Weitere Informationen zum Programm *kfile* können über die WWW-Seiten des kde-Projektes *www.kde.org* abgerufen werden.

2.4.7 Arbeiten im Dateisystem

Mit Befehlen wie *mdkir, rmdir, cp* und *mv* kann der Systemverwalter des Linux-Servers Unterverzeichnisse anlegen, Dateien innerhalb des Server-Filesystems erstellen, kopieren und löschen. Die Syntax und Funktion der dafür notwendigen Befehle ist in der Regel unproblematisch, DOS- und Windows-9x/NT-Systeme verwenden nahezu die gleichen Befehle.

Auf diese Art von Arbeiten im Dateisystem soll hier nicht weiter eingegangen werden, kurze Beschreibungen dieser Befehle finden sich an anderer Stelle bzw. können den entsprechenden Literaturstellen entnommen werden. Wichtiger erscheint es, einige unter Windows 9x/NT nicht bekannte Arbeitsgänge darzustellen, die nach eigener Erfahrung tatsächlich eine kleine Hürde darstellen: Unix/Linux kennt keine Laufwerksbezeichner für Disketten-, Festplatten- oder CD-ROM-Laufwerke, der Zugriff auf diese Server-Hardware erfolgt über die Gerätedateien, die im Verzeichnis */dev* gespeichert sind. Wie werden jetzt Disketten verwendet, wie werden sie unter Linux eingerichtet und Daten dort gelesen bzw. geschrieben? Wie erfolgt der Zugriff auf ein CD-ROM-Laufwerk oder wie wird eine weitere Festplatte in das Serversystem integriert?

Datenaustausch mit Disketten

Als erstes Beispiel wird jetzt ein Arbeitsgang beschrieben, der auch mit fertigen Netzwerkverbindungen immer wieder auf den Systemverwalter zukommt: Es besteht immer wieder die Notwendigkeit, kleinere Datenbestände auf Diskette zu speichern, entweder zum Datenaustausch zwischen einzelnen Rechnern oder um kleinere Datensicherungen durchzuführen.

Um eine Diskette im Linux-Dateisystem verwenden zu können, muss diese zunächst vom Systemverwalter in ein vorhandenes Verzeichnis gemountet werden. Standardmäßig ist dies das Verzeichnis */floppy*.

Unter DOS oder Windows 9x/NT formatierte Disketten (*FAT-Dateisystem*) können direkt gelesen und geschrieben werden, die dort gespeicherten Daten unterliegen aber weiterhin der *8.3-Namenkonvention*: Lange Dateinamen und von Linux verwendete Dateiattribute werden nicht gespeichert. Damit kann es empfehlenswert sein, speziell für Linux formatierte Disketten zu verwenden, die diese Nachteile vermeiden:

Schritt 1: Formatieren

Eine 3,5"-HD-Diskette im Laufwerk A: wird mit dem Befehl

```
fdformat /dev/fd0H1440
```

im Format High Density, 1440 kByte eingerichtet. Nach Abschluß der Formatierung erfolgt automatisch eine Prüfung auf fehlerhafte Blöcke.

Schritt 2: Dateisystem anlegen

Disketten, die ausschließlich zum Datenaustausch zwischen Linux-Systemen verwendet werden, sollten zweckmäßigerweise im Format ext2fs (*Extended Two Filesystem*) formatiert werden. Dieses Dateisystem unterstützt lange Dateinamen, alle Dateiattribute bleiben beim Kopieren erhalten.

Mit dem Befehl

```
mkfs.ext2 -c -m0 /dev/fd0H1440
```

wird ein *ext2fs*-Filesystem auf der Diskette im Laufwerk A: angelegt, fehlerhafte Blöcke werden erkannt und automatisch gesperrt, die Speicherplatzreservierung für den Master-user (Standardwert: 5 % der Laufwerkskapazität) wird unterdrückt.

Schritt 3: Diskette mounten

Eine Diskette wird standardmäßig im Verzeichnis */floppy* gemountet (in das Linux-Dateisystem eingebunden). Der Befehl

```
mount -t msdos /dev/fd0 /floppy
```

mountet die FAT-Diskette im Laufwerk A: in das Verzeichnis */floppy*.

Der Befehl

```
umount /floppy
```

beendet den Mount-Vorgang im Verzeichnis */floppy*. Die Diskette kann jetzt entnommen werden.

Vor der Entnahme der Diskette aus dem Laufwerk muss dieses Gerät in jedem »dismounted« werden, um Inkonsistenzen zu vermeiden. Erst durch diese Aktion werden die Daten aus dem Arbeitsspeicher des Linux-Servers auf die Diskette geschrieben. Da Linux das Diskettenlaufwerk während des Mount-Zustandes leider nicht mechanisch verriegeln kann (wie es z. B. bei einem CD-ROM-Laufwerk der Fall ist), kann es passieren, dass eine Diskette zu früh entnommen wird.

Daten von CD-ROM lesen

Auch ein CD-ROM-Laufwerk kann mit dem Befehl *Mount* in das Dateisystem des Linux-Servers aufgenommen werden. Mit

```
mount -t iso9660 -o ro /dev/CD-ROM CD-ROM
```

wird das Gerät */dev/CD-ROM* im Verzeichnis */CD-ROM* gemountet, das Dateisystem *iso9660* ist Standard bei diesem Datenträgertyp. Der Datenträger wird als read only (*ro*) gemountet.

Während die CD-ROM gemountet ist, sperrt Linux dieses Laufwerk, der Datenträger kann nicht entnommen werden. Erst mit

```
umount /dev/CD-ROM
```

wird diese Sperrung aufgehoben. Beachtet werden sollte, dass der Befehl *umount* nicht angewendet werden kann, wenn das angegebene Verzeichnis noch anderweitig verwendet wird, z. B. in dieses Verzeichnis hingewechselt wurde.

2.4.8 Server-Uhrzeit setzen

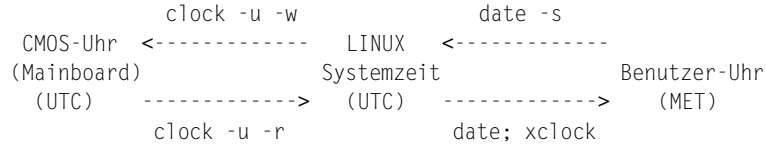
Das interne Zeitmanagement von LINUX arbeitet mit »drei verschiedene Uhrzeiten«:

- die Zeit der auf der PC-Hauptplatine (Mainboard) installierten CMOS-Uhr (RTC). Dies kann im BIOS-Setup, mit dem DOS-Befehl *time* oder mit dem Unix-Shell-Befehlen *clock* oder *hwclock* eingestellt werden.
- die LINUX-interne Uhrzeit (Systemzeit, Kernelzeit). Sie wird grundsätzlich als UTC (*Universal Time, Coordinated*), häufig auch noch als GMT (*Greenwich Mean Time*) bezeichnet. UTC hat den Vorteil, dass die Umstellung auf Sommerzeit (MEST, MET DST) automatisch erfolgen kann. Uhren von Linux-Rechnern sollten daher grundsätzlich so eingegestellt werden. Die Systemzeit wird bei jedem Systemstart über die CMOS-Uhr eingestellt und neu berechnet: Sie ist realisiert als ein Zähler, der die Anzahl der Sekunden seit dem 1. Januar 1970, 00:00:00 Uhr (also seit 1969) zählt.
- die für Benutzerprozesse geltende Zeit. Diese wird unter Berücksichtigung der eingestellten Zeitzone aus der internen Uhrzeit berechnet. Unter Verwendung der Regeln aus der Datei */usr/lib/zoneinfo/localtime* berechnen Programme wie *date* und *xclock* aus der Systemzeit die Uhrzeit der gewählten Zeitzone. Zu diesen Regeln gehört auch die automatische Umstellung von Sommer- auf Winterzeit.

Während jeder Linux-Installation muss angegeben werden, in welcher Zeitzone dieser Rechner betrieben werden soll. Die Zeitzone für Deutschland ist immer MET (*Middle European Time*). In der Konfigurationsdatei */etc/rc.config* stehen damit die Einträge

```
GMT = "-u"
TIMEZONE="MET"
```

Die nachfolgende Skizze stellt das Linux-Zeitmanagement noch einmal in einer Übersicht dar:



Der Befehl *date* ändert das Systemdatum oder gibt es aus. Die Ausgabe von *date* kann, gesteuert durch Optionen, in den unterschiedlichsten Formaten dargestellt werden. Mit

```
date
Sun Oct 31 15:22:12 MET 1999
```

wird die aktuelle Systemzeit im Standardformat ausgegeben, MET kennzeichnet die *Timezone*. Die Formatsteuerung der Zeitangabe erfolgt über eine als Option übergebene Zeichenkette, die spezielle Bezeichner enthält, die von *date* zu den aktuellen Zeitkomponenten expandiert werden. Tabelle 2-14 zeigt eine Auswahl der zugelassenen Bezeichner.

Bezeichner	wird expandiert zu
%H	Stunden (00 ... 23)
%M	Minuten (00 ... 59)
%S	Sekunden (00 ... 59)
%s	Sekunden seit dem 01.01.1970, 00:00:00
%d	Tag (01 ... 31)
%j	Jahrestag (01 ... 366)
%a	Wochentag, abgekürzt
%A	Wochentag, ausgeschrieben
%m	Monat (01 ... 12)
%b	Monatsname, abgekürzt
%B	Monatsname, ausgeschrieben
%y	zweistellige Jahreszahl
%Y	vierstellige Jahreszahl

Tabelle 2-14 Formatkennzeichner zur Ausgabe der Systemzeit

Dazu ein Beispiel: Mit dem Befehl

```
date +'%A, den %d. %B %Y, aktuelle Zeit %k:%M'
```

wird die aktuelle Systemzeit in der Form

```
Sunday, den 31. October 1999, akuelle Zeit: 15:33
```

ausgegeben. Der Systemverwalter *root* kann mit dem Befehl

```
date -s ' 1999-10-31 16:35:00'
```

die Systemzeit auf den angegebenen Wert setzen. Jede Veränderung, die mit dem Befehl *date* vorgenommen wurde, gilt aber immer nur bis zu nächsten System-

start. An der Einstellung der CMOS-Uhr ändert *date* nichts. Dieses geschieht mit dem Befehl *hwclock* (*clock*), der ebenfalls nur vom Systemverwalter *root* ausgeführt werden darf. Mit

```
hwclock -ur
```

wird die aktuelle Zeit der mit UTC laufenden CMOS-Uhr angezeigt. Soll eine mit *date* vorgenommene Änderung in Datum oder Uhrzeit dauerhaft gespeichert werden und läuft die CMOS-Uhr mit *lokaler Zeit*, so wird mit

```
hwclock -w
```

die Systemzeit gespeichert. Läuft die CMOS-Uhr auf UTC (GMT), so muss die Systemzeit mit

```
hwclock -uw
```

gespeichert werden. Mit dem Befehl

```
hwclock --set --date="10/31/99 15:21:30"
```

wird die CMOS-Uhr auf den 31. 10. 1999, 15:21:30 Uhr eingestellt. Weitere Informationen zu den Befehlen *hwclock* oder *clock* können mit

```
man 8 clock
```

abgerufen werden.

Sollte es auch einem Linux-Rechner Schwierigkeiten mit der Zeiteinstellung geben, so sollte die Umgebungsvaribale *TZ* der Link *locatime* im Verzeichnis */usr/lib/zoneinfo* überprüft werden. Dieser verweist auf */etc/localtime*, in dieser Datei sollte der Eintrag UTC enthalten sein.

Alle Änderungen an der Zeitzone und größere Veränderungen von Datum oder Uhrzeit sollten nur dann vorgenommen werden, wenn alle anderen Arbeiten im System beendet sind, d.h. Dateien abgeschlossen und Programme beendet sind, da es sonst zu erheblichen Inkonsistenzen durch fehlerhafte Zeitstempel kommen kann. Diese Arbeiten sollten grundsätzlich nur im Single-User-Modus durchgeführt werden.

2.4.9 Konfiguration von zeitgesteuerten Ereignissen

Regelmäßig wiederkehrende, zeitgesteuerte Ereignisse und Aktioen werden über den *cron*-Daemon gesteuert. Die Programmierung erfolgt über Tabellen im *cron*-spezifischen Format.

In der Linux-Shell wird dazu der Befehl *crontab* verwendet. Die cron-Tabellen könnten auch mit einem einfachen Editor bearbeitet werden, die Verwendung spezieller Programme ist aber empfehlenswert.

cron-Tabellen

cron-Tabellen bestehen aus zwei Teilen: Zunächst wird der Zeitpunkt der Befehlsausführung einschließlich des Benutzernamens definiert, dann folgt der auszuführende Befehl bzw. das Skript. Mit dem Shell-Befehl kann die aktuelle Konfiguration abgerufen werden. Zum Beispiel:

```
SHELL=/bin/sh
PATH=/usr/bin:/usr/sbin:/sbin:/bin
MAILTO=root
43 6 * * *  $HOME/bin/cron.daily
* * * * *  test -x /usr/sbin/atrun && /usr/sbin/atrun
* * * * * date>/dev/ttyp0
```

Die Angabe des Ausführungszeitpunktes erfolgt über fünf Werte: Stunde, Minute, Tag, Monat und gegebenenfalls der Wochentag. Alle Werte können aus konkreten Angaben oder Platzhaltern (Mustern) bestehen, ein Sternchen steht für einen beliebigen Wert:

```
minute stunde tag monat wochentag kommando
```

Die einzelnen Einträge werden durch Leer- oder Tabulatorzeichen getrennt. Die Zeit-angaben sind Zahlen (Minute: 0-59, Stunde: 0-23, Tag: 1-31, Monat: 1-12, Wochentag: 0-6 (0=Sonntag)), durch Komma getrennte Zahlenfolgen (z.B. 1, 3, 5) oder Zahlenbereiche (z.B.: 3-8). Steht anstelle des Zeiteintrags ein »*«, so ist damit »*zu jeder vollen...*« gemeint.

Im o.a. Beispiel wird jede Minute der Befehl *date* ausgelöst. Die Ausgabe dieses Befehls wird umgeleitet auf */dev/ttyp0*. Soll das Programm */usr/lib/atrun* alle 10 Minuten ausgeführt werden, so enthält *crontab* die Zeile:

```
0,10,20,30,40,50 * * * * /usr/lib/atrun
```

Mit dem Shell-Befehl *crontab -e* kann die aktuelle Konfiguration bearbeitet werden.

Kcrontab

Das KDE-Programm *kcrontab* ist die grafische Oberfläche für den Shell-Befehl *crontab*.

In Abbildung 2-32 ist lediglich eine zeitgesteuerte Aktion programmiert: Jeden Freitag Morgen um 02.00 Uhr wird das Skript */etc/sicherung* ausgeführt. Soll eine neue Aktion programmiert werden, so erfolgt dies über den Menüpunkt *Bearbeiten -> Neuer Eintrag* (oder direkt ausgelöst über die Taste Einfg).

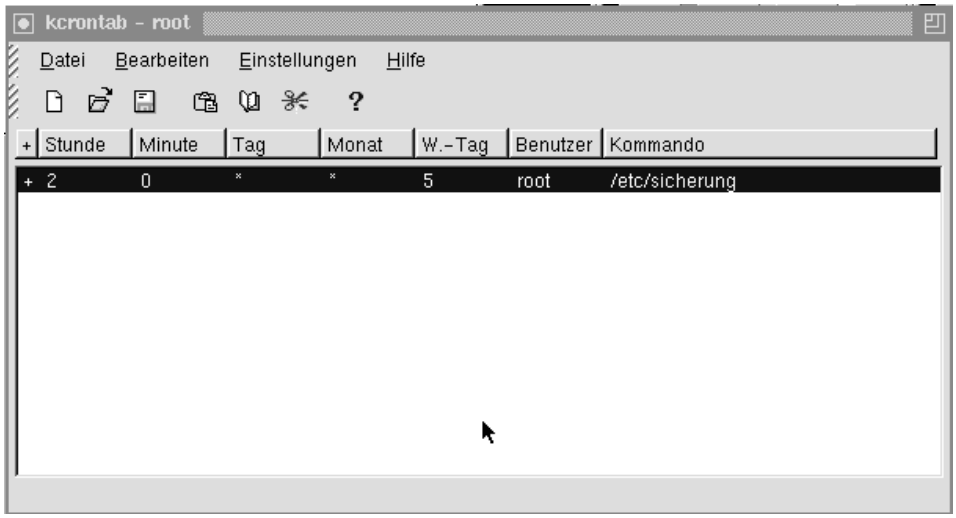


Abbildung 2-32 Das Programm kcrontab

Hier soll an jedem Werktag um 22.00 Uhr der Inhalt des Verzeichnisses */usr/tmp* gelöscht werden, diese Aktion soll im gesamten Jahr ausgeführt werden.

Gerade für die Definition zeitgesteuerter Ereignisse kann es zusätzlich wichtig sein, bestimmte Aktionen nur dann auszuführen, wenn ein direkt davor durchgeführter Test dies auch notwendig macht. Unter Unix/Linux wird dies einfach über eine »UND«-Verknüpfung von zwei Befehlen erreicht: Das Shell-Steuerzeichen »&&« verknüpft die beiden rechts und links davon stehenden Befehle so miteinander, dass der rechts stehende Befehl nur dann ausgeführt wird, wenn der links stehende erfolgreich beendet werden konnte. Dazu ein einfaches Beispiel. Mit der Befehlszeile

```
who | grep root 2> /dev/null && echo root_ist_da!
```

wird der Text »root ist da!« nur dann auf dem Bildschirm ausgegeben, wenn ein Eintrag für diesen Benutzer in der mit dem Befehl *who* erstellten Usertabelle gefunden werden konnte.

webmin

Auch das an späterer Stelle ausführlich vorgestellte zentrale Administrationstool *webmin* bietet eine sehr komfortable Schnittstelle zur Programmierung von zeitgesteuerten Ereignisssen. Wird im Modul *Scheduled Cron Jobs* beispielsweise aus der *cron*-Liste der Eintrag für */etc/sicherung* ausgewählt, so können alle Details in einer grafischen Oberfläche eingestellt werden.

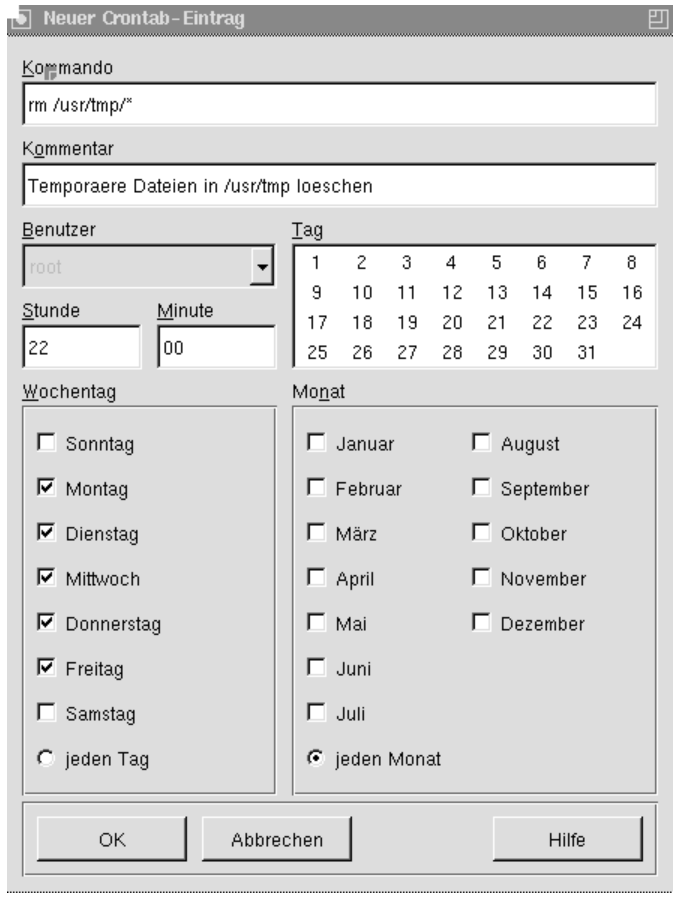


Abbildung 2-33 Neuen Eintrag erstellen mit kcrontab

Nach Klick auf die Taste *Run Now* kann das hier verwendete Kommando zu Testzwecken sofort gestartet werden, ebenfalls ist es sehr leicht möglich, *cron*-Einträge vorübergehend zu deaktivieren.

2.4.10 Betreuerruf durch Pager

Treten schwerwiegende Fehler an einem Linux-Server auf, soll dieser Server automatisch entsprechende Warnmeldungen an den Systemverwalter schicken. So können Speicherplatzmangel im Festplattensystem, der Ausfal der Netzspannung oder typische Überlastungszustände vielleicht noch rechtzeitig erkannt und behoben werden. Die Benachrichtigung über E-Mail kann in diesem Fall zu langsam sein, technisch ist die Erreichbarkeit per Rundsendung oder Meldung auf dem Bildschirm des Netzwerkarbeitsplatzes nicht immer gegeben.

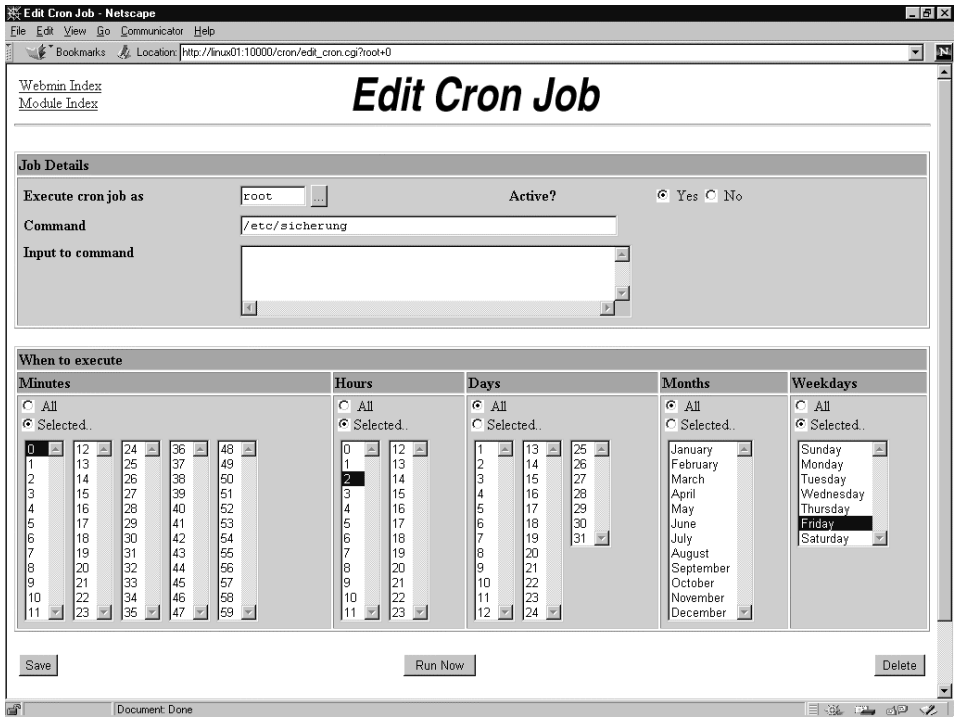


Abbildung 2-34 cron-Administration mit webmin

In solchen Fällen kann es sinnvoll sein, die Nachricht an einen Pager (Quix, Scall, Cityruf oder per SMS an ein Handy) zu senden. Diese Nachricht wir sofort zugestellt, unabhängig davon, wo der Benutzer gerade ist.

Mit dem zum Lieferumfang der SuSE-Distribution gehörenden Programmpaket *yaps* von Ulrich Dessauer (yet another pager software, *ftp://ftp.sta.com/pub/fk/yaps*) kann ein automatischer Betreuerruf realisiert werden, der sowohl mit Modem oder auch über ISDN funktioniert (*yaps* gehört zu der Serie *n).*

Die Konfiguration erfolgt über die Datei */etc/yaps.rc*, die ähnlich aufgebaut ist wie eine Windows-Konfigurationsdatei. Im globalen Abschnitt werden die verfügbaren Dienste angegeben (z.B. *D1*, *D2privat* oder *Quix*), hier werden auch die nutzbaren Modems smit ihren Einstelldaten eingetragen. Danach folgt für jedes Modem ein eigener Abschnitt, u.a. wird hier eingetragen, für welche Zielrufnummern dieser Dienst zuständig ist.

Für den Testlauf sollte in der Konfigurationsdatei */etc/yaps.rc* der Parameter *verbose* auf den Wert *4* gesetzt werden, um den Verbindungsaufbau mit der jeweiligen Pager-Zentrale besser verfolgen zu können.

Ein typisches Problem entsteht immer dann, wenn ein analoges Modem an einer Nebenstellenanlage betrieben wird. Das Modem erwartet standardmäßig ein Freizeichen vor dem Wählen. Dies wird aber von der Nebenstellenanlage nicht erzeugt, der Wählvorgang findet nicht statt. Damit *yaps* trotzdem richtig funktionieren kann, muss in diesem Fall in der modemspezifischen Konfiguration die Zeile

```
init   \\r 1200D ATZ\r <OK ATE000V1X3\r <OK
```

eingetragen werden. Das vorher vorhande *X4* wird durch *X3* ersetzt.

Das Versenden einer SMS- oder Pager-Nachricht erfolgt nach Abschluß der Konfiguration aus der Kommondzeile. Mit

```
# yaps 01725337212 "Dies ist eine Nachricht"
```

wird eine SMS-Nachricht mit dem angegebenen Text an das Handy 0172 5337212 gesendet.

Pearson Education

ADDISON-WESLEY

3 Konfiguration der Netzwerkdienste

Moderne PC-Netzwerke verwenden das Protokoll TCP/IP, auch wenn es hohe Anforderungen an die Kenntnisse des Systemverwalters im Bereich Planung, Konfigurierung und Betrieb stellt. Die nachfolgende Anleitung folgt den im ersten Kapitel festgestellten Anforderungen an Netzwerkressourcen, vorgestellt wird eine stabile und praxisorientierte Lösung für kleine und mittlere Netzwerke.

3.1 Standard-Netzwerkfunktionen

3.1.1 TCP/IP-Basiskonfiguration

Die Planung der Netzwerkorganisation und der Adressräume unter TCP/IP erfordern vom Systemverwalter detailierte Kenntnisse. Alle Planungsschritte, insbesondere die Lastverteilung auf einzelne Netzwerksegmente (*Subnets, Stränge*) ist sehr zeitaufwändig und fehlerträchtig. Die grundlegenden Planungsarbeiten in größeren TCP/IP-Netzwerken, insbesondere, wenn WAN-Anbindungen oder Netzwerkkopplungen über Router erfolgen müssen, sollten nur von qualifizierten Personen geplant und realisiert werden.

Die Adressierung im TCP/IP-Netzwerk erfolgt über IP-Adressen und über Hostnamen, die fest an eine Adresse gebunden sind. Die Hostnamen entsprechen dabei den Rechnernamen, die in der Regel um die Bezeichnung der Domäne, zu der dieser Host gehört, erweitert werden. Es enstehen so die aus dem Internet bekannten *full qualified host names* wie z.B.

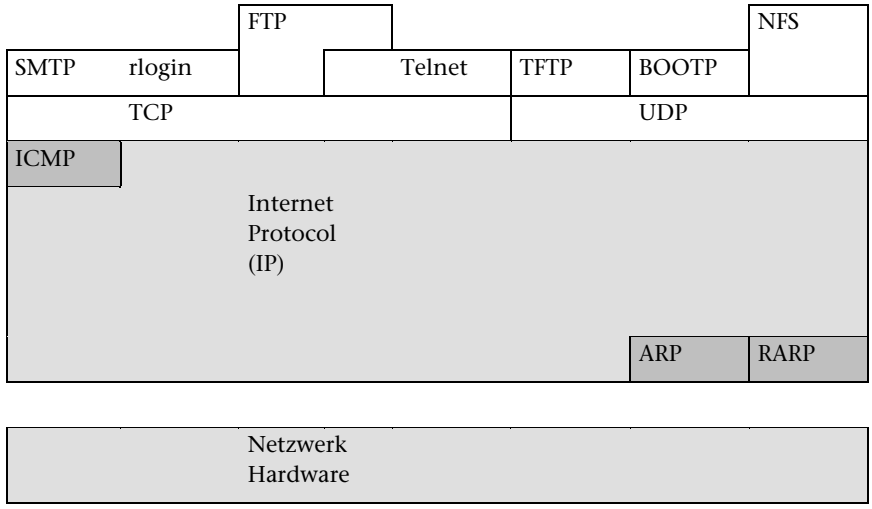
```
linux01.samulat.de
```

In jedem TCP/IP-Netzwerk ist damit die Namensauflösung von besonderer Wichtigkeit. Damit diese funktionieren kann, muss der Systemverwalter dafür sorgen, dass ein Host immer die gleiche IP-Adresse erhält. Jeder Host im Netzwerk muss diese Namensauflösung durchführen können, realisiert wird dies entweder über einen zentralen Serverdienst (*DNS, Domain Name Server*) oder über die auf jedem Rechner notwendige Steuerdatei *hosts*.

TCP/IP

TCP/IP ist eine Abkürzung für verschiedene Standards mit unterschiedlichen Merkmalen und Funktionen. Die Standards werden in RFCs (*Requests for Comments*) veröffentlicht. Die RFCs sind Arbeitspapiere, Protokollspezifikationen oder auch nur Kommentare zu aktuellen Themen der *Internet Community*, also keine anerkannten Normen.

Die Buchstaben TCP/IP stehen für zwei Kommunikationsprotokolle, TCP (*Transmission Control Protocol*) und IP (*Internet Protocol*). TCP/IP wird in der Regel aber auch verstanden als Abkürzung für die gesamte Kommunikationsarchitektur. Diese weitaus größere Sammlung von Standards hat offiziell die Bezeichnung IPS (*Internet Protocol Suite*).



Legende:

ARP	Address Resolution Protocol	telnet	Remote Terminal Loging
RARP	Reverse Address Resolution Protocol	TFTP	Trivial File Transfer Protocol
ICMP	Internet Control Message Protocol	BOOTP	Boot Protocol
SMTP	Simple Mail Transfer Protocol	NFS	Network File System
Rlogin	Remote Login	UDP	User Datagram Protocol
FTP	File Transfer Protocol	TCP	Transmission Control Protocol

Abbildung 3-1 Die Architektur von TCP/IP

IP-Protokoll

Das IP-Protokoll ist der Netzwerkschicht (Ebene 3) des OSI-Modells zuzuordnen (Abbildung 3-2). In dieser Schicht wird der Übertragungsweg zwischen Sender und Empfänger festgelegt, logische Adressen und Namen werden in physikalische Adressen übersetzt und die logische Verbindung wird auf- bzw. abgebaut. Auf diese Weise transportiert IP Datenpakete von einem Sender über mehrere Netze hinweg zum Empfänger.

Ebene	Schicht im OSI-Modell	Sicherer Übertragungsweg	Schneller Übertragungsweg	
7	Anwendung	FTP	NFS	
6	Darstellung	Telnet SMTP		
5	Sitzung	BSD Sockets	AT&T – TLI	
4	Transport	TCP	UDP	
3	Netzwerk	IP	ARP RARP ICMP	
2	Verbindung	802.3/Ethernet	802.4	802.5
1	Bit-Übertragung			

Abbildung 3-2 TCP/IP im ISO-/OSI-Schichtenmodell

IP-Pakete werden als *Datagramme* bezeichnet und sind voneinander unabhängig. IP gehört zu den informellen Protokollen; Empfangsquittungen werden nicht ausgegeben:

Das Protokoll garantiert weder den Erhalt eines Paketes noch die Einhaltung der Reihenfolge!

IP-Adresse und Quadrupelformat

Im Gegensatz zur Ethernet-Adresse (MAC-Adresse, Länge: 48 Bit) umfasst eine IP-Adresse nur 32 Bit, geschrieben als *w.x.y.z.* (gepunktetes Quadrupel-Format), wobei x, y, z und w Dezimalwerte zwischen 0 und 255 darstellen, z.B.

```
199.34.57.10
```

Im Internet verwendete IP-Adressen werden zentral durch das *Network Information Center* (NIC) vergeben. Es sind A-, B- und C-Klasse-Netzwerke zu unterscheiden (Tabelle 3-1).

A-Klasse	Die ersten 8 Bit (Wertebereich 0..126) werden vom NIC vergeben. Somit können max. 127 Klasse-A-Netzwerke existieren. Da von den 32 Bit 24 Bit frei verfügbar bleiben, können bis zu 16 Millionen Hosts adressiert werden.
B-Klasse	Für mittelgroße Netzwerke werden die ersten 16 Bit vom NIC zugewiesen (Wertebereich der ersten 8 Bit: 128..191). Max 16.384 Klasse-B-Netzwerke mit jeweils 64.535 Hosts können existieren.
C-Klasse	Für kleine Netzwerke bleiben nur 8 Bit frei verfügbar. Das NIC vergibt die ersten 24 Bit der IP-Adresse (Wertebereich der ersten 8 Bit: 192..223).

Tabelle 3-1 Netzwerk-Klassen

1. Quadrupel	Verbundnetzklasse
0..126	A-Klasse-Netzwerke
127	reserviert: Loopback-Adresse
128..191	B-Klasse-Netzwerke
192..223	C-Klasse-Netzwerke
224..239	reservierte Multicast-Adressen
240..255	reservierte Versuchsadressen

Tabelle 3-2 Verbundnetzklassen und reservierte Adressen

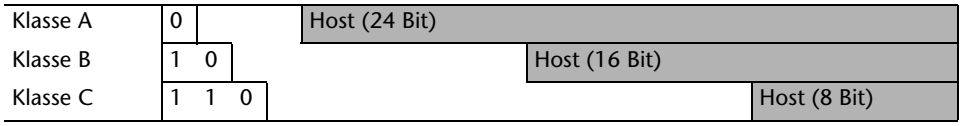


Abbildung 3-3 Kennzeichnung der Verbundnetzklassen

Regeln für IP-Adressen

Für die Bildung der IP-Adressräume gelten eine Vielzahl von Regeln, in jedem Fall sollten die nachstehenden berücksichtigt werden:

- Die »letzten« Quadrupel dürfen nie 0 (= Netzwerkadresse) oder 255 (= Broadcast-Adresse) sein.
- Als Hostadresse sind »nur Nullen« und »nur Einsen« nicht zugelassen.
- Die Adresse 127.0.0.1 ist reserviert für Rückkopplungen.

Mit Hilfe der *Subnet-Mask* entscheidet die IP-Software eines Hosts, ob ein anderer Host zum selben Teilnetz gehört oder nicht.

Classless Internetwork Domain Routing (CDIR)

CDIR-Netzwerke werden auch als »*slash-x*-Netzwerke« bezeichnet, wobei *x* die Anzahl der Bits angibt (*Subnet Mask*), die vom InterNIC fest vorgegeben wird. Ein Klasse C Netzwerk alter Art entspricht z. B. einem *slash-24*-Netzwerk. Die Flexibilität von CDIR ermöglicht die effiziente Zuweisung von IP-Adressen (Abbildung 3-4).

INTERNIC Netzwerk-Typ	Subnet Mask				Anzahl Hosts (ungefähr)
Slash 0	0	0	0	0	4.000.000.000
Slash 1	128	0	0	0	2.000.000.000
Slash 2	192	0	0	0	1.000.000.000
Slash 3	224	0	0	0	500.000.000

Abbildung 3-4 CDIR-Netzwerktypen (Auszug)

INTERNIC Netzwerk-Typ	Subnet Mask				Anzahl Hosts (ungefähr)
Slash 4	240	0	0	0	250.000.000
Slash 5	248	0	0	0	128.000.000
Slash 6	252	0	0	0	64.000.000
Slash 7	254	0	0	0	32.000.000
Slash 8	255	0	0	0	16.000.000
...	...	...	...	...	...
Slash 22	255	255	252	0	1.024
Slash 23	255	255	254	0	512
Slash 24	255	255	255	0	256
Slash 25	255	255	255	128	128
Slash 26	255	255	255	192	64
Slash 27	255	255	255	224	32
Slash 28	255	255	255	240	16
Slash 29	255	255	255	248	8
Slash 30	255	255	255	252	4
Slash 31	255	255	255	254	2
Slash 32	255	255	255	255	1

Abbildung 3-4 CDIR-Netzwerktypen (Auszug)

In der Praxis sind nicht alle »slash-x-Netzwerke« sinnvoll anzuwenden, so z.B. slash 31, slash 32 und auch slash 0. Zur Verwendung von CDIR-Netzwerken folgt jetzt ein Beispiel:

Die einer Firma zugewiesene Netzwerkadresse ist 198.8.124.0 (Klasse-C-Netzwerk). Die Firma benötigt ein IP-Nummernschema für mindestens 9 lastgetrennte Netzwerksegmente mit jeweils maximal 12 Hosts je Segment.

Realisiert wird ein Netzwerk vom Typ *Slash 28*: Die resultierende Subnet-Mask umfaßt so die vorgegebenen ersten drei Quadrupel und die vier höchsten Bit im letzten Quadrupel. Somit sind insgesamt 16 Teilnetz-werke mit jeweils **14** Hosts möglich (siehe Tabelle 3-3).

Netzwerksegment	**Adressen (mathematisch)**	**Adressen (tatsächlich)**
A	198.8.124.0 bis .15	198.8.124.1 bis .14
B	198.8.124.16 bis .31	198.8.124.17 bis .30
C	198.8.124.32 bis .47	198.8.124.33 bis .46
D	198.8.124.48 bis .63	198.8.124.49 bis .62

Tabelle 3-3 Planung von Teilnetzwerken (Beispiel)

Auch in einem Netzwerksegment darf die Hostadresse nicht ausschließlich aus Nullen oder Einsen bestehen. Je Segement sind damit tatsächlich immer genau zwei Hostadressen weniger als mathematisch möglich erlaubt!

Die Auswertung auf »Netzwerk-« oder »Broadcast-Adresse« erfolgt über die Subnet-Mask des jeweiligen Host. Bei diesem Beispiel mit insgesamt 16 möglichen Netzwerksegmenten gehen dabei bereits 16 * 2 = 32 Hostadressen »verloren«.

TCP-Protokoll

Das Transmission Control Protocol (TCP) ist ein verbindungsorientiertes End-to-End-Protokoll, das der Transportschicht (Ebene 4) des OSI-Modells zuzuordnen ist. Die zu übertragenden Dateien werden hierbei in nummerierte Datenblöcke zerlegt. Vor Beginn der Übertragung vereinbart TCP die maximale Blockgröße zwischen Sender und Empfänger.

TCP bietet den höheren Schichten folgende Dienste an:

- Durch einen Quittungsmechanismus wird eine korrekte Datenübertragung gewährleistet. Treten Fehler bei der Datenübertragung auf, sendet TCP das Paket erneut. Die Datenpakete sind fortlaumfend numeriert, Prüfsummen gewährleisten den korrekten Inhalt des Datenpaketes. Sicherheits- und Vorrangklassen können gewählt werden.
- Verbindungshandling (Aufbau, Transferphase, Abbau)
- Flusskontrolle durch Nummerierung des Datenbytes
- Multiplexen der Verbindungen: Mehrere Tasks können über die gleiche Netzwerkadresse mittels unterschiedlicher TCP-Ports kommunizieren.

TCP Ports

Server-Anwendungen oder Prozesse, die TCP zum Transport verwenden, haben mindestens eine voreingestellte Anschlussnummer (TCP Port Address). Die vorein-gestellten Anschlussnummern für FTP-Serverdienste sind z.B. 20 (Daten) und 21 (Kontrolle). Diese Anschlusszuweisungen werden *bekannte Anschlussnummern* genannt und sind in RFC 1700 dokumentiert.

Tabelle 3-4 zeigt eine Auswahl bekannter Anschlussnummern.

Anschlussnummer	Prozessbezeichnung	Beschreibung
1	TCPMUX	TCP Port Service Multiplexer
2	RJE	Remote Job Entry
20	FTP-DATA	File Transfer Protocol – Daten

Tabelle 3-4 TCP- Anschlussnummern (Auswahl)

Anschlussnummer	Prozessbezeichnung	Beschreibung
21	FTP-	File Transfer Protocol – Kontrolle
23	TELNET	Telnet
25	SMTP	Simple Mail Transfer Protocol
42	NAMESERV	Host-Namen-Server
49	LOGIN	Anmelde-Host-Protokoll
53	DOMAIN	Domänen-Namensauflösung
69	TFTP	Trivial File Transfer Protocol
80	HTTP	HTTP
103	X400	X.400
156	SQLSERV	SQL Server

Tabelle 3-4 TCP- Anschlussnummern (Auswahl)

Anschlussnummern werden zusammen mit einer IP-Adresse verwendet, um einen *Socket* zu bilden. Sockets haben immer eine ihnen zugeordnete Nummer (oder Adresse) und bezeichnen einen Endpunkt.

ARP-Protokoll

Das in RFC826 definierte *Address Resolution Protocol* (ARP) ist ein Protokoll zur Adressauflösung. Es ordnet den logischen IP-Adressen (Verbundnetzadressen) physische Adressen (Ethernet-Adressen, MAC-Adressen) zu und umgekehrt.

Eine Station, die Verbindung mit einer anderen Station aufnehmen will, sendet zu diesem Zweck ein Broadcast-Paket (ARP-Request), das die eigenen Adressen sowie die gesuchte IP-Adresse beinhaltet. Die eigene IP-Adresse wird aus der Database oder speziellen Konfigurationsdateien entnommen, die eigene Ethernet-Adresse wird von den Treibern der Netzwerkkarte ermittelt. Die gesuchte IP-Adresse wird entweder nummerisch eingegeben oder über den Hostnamen aus der Datei *hosts* ermittelt.

Die aufgerufene Station erkennt ihre IP-Adresse und übersendet daraufhin ihre Ethernet-Adresse (physikalische Adresse, MAC-Adresse) an die rufende Station. Die rufende Station trägt diese Adressen in die ARP-Tabelle ein, damit sie für weitere Pakete direkt zur Verfügung steht.

Der Shell-Befehl *arp* dient zur Anzeige oder Änderung der Übersetzungstabellen, die von *Address Resolution Protocol* für die Umsetzung von IP-Adressen in physische Ethernet- oder Token-Ring-Adressen verwendet werden. Mit

```
arp -a
```

werden anhand einer TCP/IP-Abfrage alle aktuellen ARP-Einträge angezeigt. Wird zusätzlich eine IP-Adresse angegeben, so werden nur die IP-Adresse und die physische Adresse des betreffenden Computers angezeigt:

```
# arp -a
portsam.samulat.de (192.168.100.12) at 00:60:08:92:A2:36 [ether] on eth0
nt40ws.samulat.de (192.168.100.10) at 00:80:48:D7:ED:11 [ether] on eth0
w98client.samulat.de (192.168.100.11) at 00:90:27:22:CD:52 [ether] on eth0
a002.samulat.de (192.168.100.21) at 00:80:C8:77:FC:C1 [ether] on eth0
```

TCP/IP-Konfiguration prüfen und bearbeiten – ifconfig

Der Befehl *ifconfig* zeigt die Konfigurationsdetails der fertig konfigurierten Schnittstellen. Wird *ifconfig* ohne weitere Parameter aufgerufen, so werden alle Netzwerkschnittstellen aufgelistet:

```
# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:80:48:C5:D2:92
          inet addr:192.168.100.1  Bcast:192.168.100.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:57574 errors:0 dropped:0 overruns:0 frame:2
          TX packets:56650 errors:0 dropped:0 overruns:0 carrier:0
          collisions:275 txqueuelen:100
          Interrupt:11 Base address:0xb800

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:3924  Metric:1
          RX packets:8862 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8862 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
```

Mit *ifconfig* kann aber auch eine neue Schnittstelle für die Netzwerkschicht des Kernel sichtbar gemacht werden. Möglich ist die Zuweisung einer IP-Adresse und verschiedener anderer Parameter sowie die Aktivierung der Schnittstelle. Mit

```
# ifconfig <interface> <ip-adresse>
```

wird dem *<interface>* die Adresse *<ip-adresse>* zugewiesen und es wird aktiviert. Alle anderen Parameter werden auf Standardwerte gesetzt, so wird die Subnet-Mask z. B. aus der angegebenen IP-Adresse abgeleitet.

TCP/IP Verbindungen prüfen – ping

Der Shell-Befehl *ping* wird verwendet, um Verbindungen zu einem oder mehreren Remote-Hosts zu überprüfen, es verwendet ICMP-Echoabfrage- und Echoantwortpakete, um festzustellen, ob ein bestimmtes IP-System in einem Netzwerk funktionsfähig ist.

Dieses Dienstprogramm eignet sich zur Diagnose von IP-Netzwerk- oder -Router-Fehlern (siehe Abbildung 3-5).

```
# ping nt40ws
PING nt40ws.samulat.de (192.168.100.10): 56 data bytes
64 bytes from 192.168.100.10: icmp_seq=0 ttl=128 time=0.940 ms
64 bytes from 192.168.100.10: icmp_seq=1 ttl=128 time=0.601 ms
64 bytes from 192.168.100.10: icmp_seq=2 ttl=128 time=0.595 ms
64 bytes from 192.168.100.10: icmp_seq=3 ttl=128 time=0.604 ms
64 bytes from 192.168.100.10: icmp_seq=4 ttl=128 time=0.605 ms
64 bytes from 192.168.100.10: icmp_seq=5 ttl=128 time=0.588 ms
64 bytes from 192.168.100.10: icmp_seq=6 ttl=128 time=0.542 ms
-- nt40ws.samulat.de ping statistics ---
7 packets transmitted, 7 packets received, 0% packet loss
round-trip min/avg/max = 0.542/0.639/0.940 ms

Syntax: PING [-t] [-a] [-n Anzahl] [-l Größe] [-f] [-i TTL]
             [-v TOS] [-r Anzahl] [-s Anzahl] [[-j Host-Liste]
             [-k Host-Liste]] [-w Timeout] Zielliste

Optionen:
    -t             Sendet fortlaufend Ping-Signale zum
                   angegebenen Host
    -a             Adressen zu Host-Namen auswerten
    -n Anzahl      Anzahl zu sendender Echo-Anforderungen
    -l L"nge       Pufferlänge senden
    -f             Flag für "Don't Fragment" setzen
    -i TTL         Time To Live.
    -v TOS         Type Of Service.
    -r Anzahl      Route für Anzahl Hops aufzeichnen
    -s Anzahl      Zeiteintrag für Anzahl Abschnitte (Hops)
    -j Host-Liste  "Loose Source Route" gemäß Host-Liste
    -k Host-Liste  "Strict Source Route" gemäß Host-Liste
    -w Timeout     Timeout in Millisekunden für eine Antwort
```

Abbildung 3-5 Syntax des Befehls ping unter Windows NT

Protokollstatistik – netstat

Der Shell-Befehl *netstat* zeigt Protokollstatistiken und aktuelle TCP/IP-Netzwerkverbindungen an. Der Befehl hat die allgemeine Form

```
netstat [-a][-e] [-n] [-s] [-p Protokoll] [-r] [Intervall]
```

Die möglichen Parameter (Auswahl) zeigt Tabelle 3-5.

-a	zeigt alle Verbindungen und abhörende Anschlüsse an. Server-Verbindungen werden normalerweise nicht angezeigt.
-e	zeigt die Ethernet-Statistik an. Kann zusammen mit dem Parameter -s kombiniert werden.
-n	zeigt Adressen und Anschlussnummern in nummerischer Form an (es wird nicht versucht, die entsprechenden Namen abzufragen).
-s	zeigt Statistik protokollweise an. Standardmäßig wird die Statistik für TCP, UDP, ICMP und IP angezeigt. Mit dem Parameter -p können Sie eine Teilmenge der Standardanzeige angeben.
-p Protokoll	zeigt die Verbindungen für das mit Protokoll angegebene Protokoll an. Mögliche Werte für Protokoll sind tcp oder udp. Wird dieser Parameter zusammen mit dem Parameter -s zur protokollweisen Statistikanzeige verwendet, kann für Protokoll tcp, udp, icmp oder ip angegeben werden.
-r	zeigt den Inhalt der Routing-Tabelle an.
Intervall	zeigt die gewählte Statistik nach der mit Intervall angegebenen Anzahl Sekunden erneut an. Drücken Sie STRG+C zum Beenden der Intervallanzeige. Ohne Angabe dieses Parameters gibt netstat die aktuellen Konfigurationsinformationen nur einmal aus.

Tabelle 3-5 Parameter für den Befehl netstat

TCP/IP-Konfigurationsdateien

/etc/hosts

In der *hosts*-Datei wird die Zuordnung von TCP/IP-Hostname und IP-Adresse vorgenommen. Die Datei *hosts* bildet die Namen statisch auf die Adressen ab.

Die Zuordnungen erfolgen zeilenweise, beginnend jeweils mit der IP-Adresse. Getrennt durch jeweils mindestens ein Leerzeichen folgen dann ein oder auch mehrere Hostnamen, die dieser Adresse zugeordnet sind. Das Zeichen »#« kennzeichnet eine Kommentarzeile.

```
# hosts     This file describes a number of hostname-to-address
#           mappings for the TCP/IP subsystem.  It is mostly
#           used at boot time, when no name servers are running.
#           On small systems, this file can be used instead of a
#           "named" name server.
# Syntax:
#
# IP-Address  Full-Qualified-Hostname  Short-Hostname

127.0.0.1 localhost
192.168.100.1   linux01.samulat.de      linux01
192.168.100.10  nt40ws.samulat.de       nt40ws
192.168.100.5   linux05.samulat.de      linux05
```

Im vorstehenden Beispiel ist der Rechner mit dem Namen *nt40ws* unter der IP-Adresse *192.168.100.10* erreichbar, als Name kann aber auch *nt40ws.samulat.de* angegeben werden. In jeder Hostdatei ist standardmäßig die Adresse *127.0.0.1* als *localhost* oder *loopback* eingetragen.

Wird im Netzwerk kein Server zur Auflösung von Hostnamen verwendet (DNS, Domain Name Server), so muss jeder im Netzwerk vorhandene Rechner die vollständige datei *hosts* erhalten, dies gilt auch für Client-PCs unter Windows 95/98 oder Windows NT. Die Datei *hosts* stellt damit das Minimum dar, das in Sachen Namensauflösung zu leisten ist. Fehlt diese Datei, werden selbst einfache Netzwerkarbeiten durch Timeouts von bis zu einer Minute (!) sehr langwierig. Die Namensauflösung muss über Broadcasts erfolgen.

Wird in einem kleinen Netzwerk die Namensauflösung durch die Datei *hosts* durchgeführt, so sollte diese zentral auf einem Server gepflegt und bei jedem Start eines Clients automatisch geladen werden.

/etc/hosts.allow

Diese Datei dient der Zugangskontrolle von Nutzern/Diensten anderer Rechner. Für bestimmte Hosts/Netzwerke kann hier der Zugriff auf bestimmte lokale Dienste reguliert werden:

```
# <service list> : <host list> [: command]
#
# Mail ist jedem gestattet
#
in.smtpd: ALL

# Telnet und FTP wird nur Hosts derselben Domain und dem
# Rechner lxws01 erlaubt.
#
telnetd, ftpd: LOCAL, lxws01.domain.de

# Finger ist jedem erlaubt, aber root wird per Mail dar-
# ueber informiert
#
fingerd: ALL: (finger @%h | mail -s "finger from %h" root)
```

/etc/hosts.deny

Hier wird der Zugang zu bestimmten Diensten des Rechners für bestimmte Hosts/Netzwerke explizit untersagt.

```
# Aufbau von /etc/hosts.deny analog zu /etc/hosts.allow
#
ALL: edu.saxedu.de

# Die remote--Shell ist eine bekannte Sicherheitsluecke
#
rsh: ALL
```

Beide Dateien werden u.a. vom TCP-Wrapper ausgewertet. Dabei wird zuerst die *hosts.allow* und dann die *hosts.deny* betrachtet. Es gilt: Wurde etwas in der *hosts.allow* explizit gestattet, darf es nicht mehr verboten werden. Selbst wenn ein entsprechender Eintrag in der *hosts.deny* steht.

/etc/hosts.equiv

Alle von diesen Rechnern aus zugreifende Nutzer haben dieselben Rechte wie lokale Nutzer (*Trusted Hosts*) und können ohne explizite Angabe eines Paßwortes über entfernte Dienste auf den Rechner zugreifen. Aus Sicherheitsgründen sollte auf die Möglichkeiten dieser Konfiguration verzichtet werden.

```
# Aufbau einer /etc/hosts.equiv
#
tubbi.galaxis.de tubbi
#
```

/etc/networks

Netzwerknamen werden hier in Netzwerkadressen umgesetzt:

```
# Typischer Aufbau einer /etc/networks
# Netzwerkname  Netzwerkadresse
#
oopback  127.0.0.0
localnet  192.168.10.0
edu-net   192.168.85.0
```

/etc/host.conf

Die Vorgehensweise beim Auflösen von Rechner- und Netzwerknamen in entsprechende Adressen wird hier bestimmt. Die Datei ist für den Resolver wichtig.

```
# Aufbau der /etc/host.conf
#
order hosts bind
multi on
```

In der Zeile *order* wird die Reihenfolge bei der Auflösung von Rechnernamen in entsprechende Adressen festgelegt: Um keine unnötigen Netzwerkbelastungen entstehen zu lassen, wird zuerst die lokale Datei */etc/hosts* (*hosts*) ausgewertet, erst danach erfolgt dann die Anfrage bei einem Nameserver *(bind)*. Auch ein NIS-Server könnte abgefragt werden *(nis)*. Diese Reihenfolge, in der die tatsächliche Namensauflösung durchgeführt wird, setzt voraus, dass die eigenen Hosts in */etc/hosts* geführt werden, da sonst immer der Nameserver diese Auflösung übernehmen muss!

multi on / multi off erlaubt bzw. verbietet die Vergabe von mehreren IP-Adressen an einen in */etc/hosts* eingetragenen Rechner.

etc/resolv.conf

Neben */etc/host.conf* spielt auch diese Datei bei der Namensauflösung eine Rolle. Angegeben werden in dieser Datei:

- die Domain des Rechners
- eine Liste von Domains zur Namensauflösung (Unvollständige Hostnamen werden durch Anhängen der Domains vervollständigt.)
- die Adressen von Nameservern

```
# Aufbau einer /etc/resolv.conf
#
domain galaxis.de
search galaxis.de edu.saxedu.de de
nameserver 192.168.85.1
```

Die Angabe für den *nameserver* wird wichtig, wenn das Netzwerk über einen Internetzugang verfügt. Hier wird dann die Adresse des Nameservers des Internet Providers angegeben, um die Auflösung von Internet-Adressen zu ermöglichen. Grundsätzlich können mehrere Einträge für *nameserver* durchgeführt werden, die Server werden dann bei Bedarf in der vorgegebenen Reihenfolge abgefragt.

/etc/protocols

Die Protokollnummern der Transportprotokolle sind hier so eingetragen, wie sie im IP-Protokollkopf erscheinen.

```
# protocols This file describes the various protocols that are
#           available from the TCP/IP subsystem.  It should be
#           consulted instead of using the numbers in the ARPA
#           include files, or, worse, just guessing them.
#
ip      0      IP      # internet protocol,pseudo protocol number
icmp    1      ICMP    # internet control message protocol
igmp    2      IGMP    # internet group multicast protocol
ggp     3      GGP     # gateway-gateway protocol
tcp     6      TCP     # transmission control protocol
pup     12     PUP     # PARC universal packet protocol
udp     17     UDP     # user datagram protocol
idp     22     IDP     # WhatsThis?
raw     255    RAW     # RAW IP interface
(...)
```

etc/services

Die Portnummern und Bezeichnungen der vorhandenen Netzwerkdienste stehen hier.

```
# Ausschnitte aus /etc/services
#
# Aufbau eines Eintrages:
# <Dienstname> <Port/Protokoll> [<Alias>]
#
tcpmux          1/tcp         # TCP port service multiplexer
echo            7/tcp
echo            7/udp
systat          11/tcp        users
netstat         15/tcp
ftp             21/tcp
ssh             22/tcp
ssh             22/udp
telnet          23/tcp
# 24 - private
smtp            25/tcp        mail
# 26 - unassigned
time            37/tcp        timserver
time            37/udp        timserver
rlp             39/udp        resource   # resource location
whois           43/tcp        nicname
domain          53/tcp        nameserver # name-domain server
domain          53/udp        nameserver
finger          79/tcp
kerberos        88/tcp        krb5       # Kerberos v5
kerberos        88/udp
(...)
```

Dynamische Zuweisung von IP-Adressen: DHCP

Ein Host kann seine IP-Adresszuweisung dynamisch beim Booten durch eine *bootp*-Abfrage (RFC 0952) erhalten. Dazu muss im Netzwerk ein *bootp*-Server oder ein entsprechendes Gateway konfiguriert sein, der zur Hardware-Adresse die entsprechende IP-Adresse bestimmt und an den Client übergibt. Das in RFC 2132 definierte *Dynamic Host Configuration Protocol* (DHCP) stellt eine Erweiterung des *bootp*-Protokolls dar.

Die Verwendung von DHCP kann den Netzwerkadministrator von der Arbeit entlasten, jedem einzelnen Client-PC in seinem Netzwerk manuell die gesamte IP-Konfiguration zu programmieren. Insbesondere dann, wenn Clients in einen anderen Netzwerkstrang wechseln oder neue Geräte installiert werden müssen, ist dies ohne DHCP immer wieder mit einem erheblichen Arbeitsaufwand verbunden.

Ein DHCP-Server weist Client-PCs auf Anforderung dynamisch eine vollständige IP-Konfiguration zu, so z. B.

- IP-Adresse
- Subnet-Mask und Broadcast-Adresse

- statische Routen
- verwendete Nameserver

Die Zuweisung basiert auf der eindeutigen 32-Bit-langen MAC-Adresse der Netzwerkkarte im Client-PC, einer »Seriennummer«, die weltweit eindeutig sein soll.

Die IP-Adresszuweisung kann entweder dynamisch oder statisch (entsprechend *bootp*) erfolgen. Bei DHCP-Anfragen übermittelt der DHCP-Server sowohl die Informationen aus der *host*-Klammer (statische Zuweisung) als auch die Informationen aus der passenden *subnet*-Klammer (dynamsiche Zuweisung), auch wenn der Client sich nicht in einer solchen Klammer befindet.

Statische Adresszuweisungen sind z. B. notwendig für Printserver, sie sollten aber, auch wenn dies mit zunächst sehr groß scheinendem Arbeitsaufwand verbunden ist, auch für alle Client-PCs erfolgen. Der Grund dafür liegt in der Namensauflösung, die in der Regel mit einer dynamischen IP-Adresszuweisung nicht vereinbar ist.

Bei der Planung und Realisierung der DHCP-Konfiguration sind einige wichtige Grundsätze zu beachten:

- Für einen Adressbereich darf nur ein DHCP-Server existieren. Die DHCP-Server können sich untereinander nicht abstimmen, da es sonst zu einer doppelten Vergabe von IP-Adressen kommt.
- Der Zeitraum, für den eine dynamische IP-Adresszuweisung gültig ist, darf nicht unnötig kurz sein. Die *default lease time* sollte typischerweise im Bereich von 6 bis 24 Stunden liegen, so dass eine einmal zugewiesene Adresse auch für diesen Zeitraum der Client »gehört«, auch wenn nach der Zuweisung der DHCP-Server vorübergehend nicht zur Verfügung steht. Für den gleichen Zeitraum ist diese Adresse aber auch im Adressraum des DHCP-Servers nicht mehr für einen weiteren Client verfügbar.
- Der DHCP-Client kann nur mit dem Protokoll TCP/IP arbeiten, wenn er eine Adresse zugeteilt bekommt. Aus Zuverlässigkeitsüberlegungen sollten damit immer zwei DHCP-Server für die Adressvergabe genutzt werden. Dabei muss der für einen Strang zur Verfügung stehende Adressbereich auf die beiden Server aufgeteilt werden. Der erste Server erhält optimalerweise 2/3, der zweite (Reserve-) Server erhält 1/3 der möglichen Adressen.
- Router, Server und Printserver müssen in jedem Fall über feste Adressen verfügen. Diese werden entweder fest am Gerät programmiert, oder es werden über DHCP statische Adresszuweisungen vorgenommen.

Installation und Start

Zum Betrieb eines DHCP-Servers muss das Paket *dhcpd* aus der Serie *n* installiert sein. Mit dem Eintrag

```
START DHCPD=YES
```

in */etc/rc.config* wird der *dhcp*-Daemon beim nächsten Booten automatisch gestartet. (dazu gehören die Variablen *DHCPD_INTERFACE, START_DHCRELAY und DHCRELAY_SERVERS*).

Mit

```
man dhcpd
```

können viele weitergehende Informationen über Installation und Konfoguration von *dhcp* abgerufen werden, z. B. für Server mit mehreren Netzwerkkarten, Multihoming- oder *bootp*-Konfigurationen.

Konfiguration in /etc/dhcpd.conf

Jede Client-Information besteht aus einem Schlüsselwort und den dazu gehörenden Parameter(n), getrennt durch ein Leerzeichen. Jede Zeile wird mit einem Semikolon abgeschlossen.

Mehrere Informationen werden durch geschweifte Klammern »{« und »}« in logischen Blöcken zusammengefasst. Alle so geklammerten Informationen beziehen sich ausschließlich auf den angegebenen Rechner.

Angaben, die für ein gesamtes Subnet gelten, werden in einem Subnet-Block zusammengefasst.Hier kann z. B. festgelegt werden, welcher IP-Adressvorat im Subnet verwendet werden darf.

```
# dcpd.conf
# Configuration file for ISC dhcpd

# Identifikation
server-identifier linux01.samulat.de;

# Globale Einstellungen
option subnet-mask 255.255.255.0;
option broadcast-address 192.168.100.255;
option routers 192.168.100.5;
option domain-name-servers 194.25.2.129;
option domain-name "samulat.de";
option netbios-name-servers 192.168.100.1;

# Dynamische IP Adressvergabe
default-lease-time 86400;
max-lease-time 604800;
```

```
subnet 192.168.100.0 netmask 255.255.255.0 {
  range 192.168.100.100 192.168.100.254;}

# Statische IP Vergabe (z.B. fuer Printserver)
group {
  use-host-decl-names on;
  host ntws40 {
    hardware ethernet 00:80:48:d7:ed:11;
    fixed-address 192.168.100.10;
  }
  host a002 {
    hardware ethernet 00:80:c8:77:fc:c1;
    fixed-address 192.168.100.21;
  }
}
```

Die oben gezeigte */etc/dhcpd.conf* ist eine einfache Konfigurationsdatei für einen DHCP-Server mit einer Netzwerkkarte, alle Clients sind am gleichen Netzwerkstrang angeschlossen.

Der DHCP-Server hat den Namen (*Server Identifier*) *linux01.samulat.de*, der Abschnitt *Globale Einstellungen* enthält die für alle Zuweisungen gültigen Parameter, wie z.B. die Subnet-Mask, Broadcast-Adresse und den Domänen-Namen.

Für die dynamische IP-Adressvergabe ist die *default lease time* mit 86.400 s (= 24 h), die *max lease time* mit 604.800 s (= 7 Tage) eingestellt. Der verfügbare Adressraum *range* ist 192.168.100.100 bis 192.168.100.254.

Es folgen zwei Einträge für statische Adresszuweisung. Die Rechner *ntws40* und *a002* erhalten die hier angegebenen festen Adressen. Die hier verwendeten Adressen dürfen natürlich nicht im zugelassenen Adressraum für die dynamische Vergabe liegen!

Betrieb und Test

Die Anforderung einer IP-Adresszuweisung und die Zuteilung durch den DHCP-Server wird in */var/log/messages* protokolliert:

```
(...)
00:90:27:22:cd:52 via eth0
Oct 26 22:02:49 linux01 dhcpd: DHCPDISCOVER from 00:90:27:22:cd:52 via eth0
Oct 26 22:02:49 linux01 dhcpd: DHCPOFFER on 192.168.100.11 to 00:90:27:22:cd:52
via eth0
Oct 26 22:02:49 linux01 dhcpd: DHCPREQUEST for 192.168.100.11 from
00:90:27:22:cd:52 via eth0
Oct 26 22:02:49 linux01 dhcpd: DHCPACK on 192.168.100.11 to 00:90:27:22:cd:52
via eth0
(...)
```

Im Fehlerfall sollte in */var/log/messages* vor allem kontrolliert *werden, ob DHCPREQUEST*-Pakete des Clients überhaupt am DHCP-Server ankommen.

Die aktuellen Leases werden in */var/state/dhcp/dhcpd.leases* protokolliert:

```
# All times in this file are in UTC (GMT), not your local timezone.   This is
# not a bug, so please don't ask about it.   There is no portable way to
# store leases in the local timezone, so please don't request this as a
# feature.   If this is inconvenient or confusing to you, we sincerely
# apologize.   Seriously, though - don't ask.
# The format of this file is documented in the dhcpd.leases(5) manual page.

lease 192.168.100.101 {
   starts 2 1999/10/26 19:22:18;
   ends 3 1999/10/27 19:22:18;
   hardware ethernet 00:90:27:22:cd:52;
   uid 01:00:90:27:22:cd:52;
   client-hostname "W98Client";
}
lease 192.168.100.102 {
   starts 2 1999/10/26 18:04:06;
   ends 3 1999/10/27 18:04:06;
   hardware ethernet 00:60:08:92:a2:36;
   uid 01:00:60:08:92:a2:36;
   client-hostname "PORTSAM";
}
(...)
```

Verwaltung mit kcmdhcpd

Das KDE-Programm *kcmdhcpd* ist die grafische Oberfläche zur Konfiguration und Verwaltung des *dhcp*-Servers (Abbildungen 3-6 und 3-7).

Konfiguration und Test eines dhcp-Clients unter Linux

In einem Netzwerk mit mehreren Linux-Servern kann eine Maschine die Aufgaben eines DHCP-Servers übernehmen, die anderen Server beziehen als DHCP-Client ihre IP-Adresse von dort. Auf dem DHCP-Client muss dazu das Paket *dhclient* aus der Serie *n* installiert werden.

Kann ein Client keine DHCP-Zuweisung erhalten, so sollte mit *traceroute* überprüft werden, ob der Server im Netzwerk direkt erreichbar ist, oder ob noch ein Host (*Router*) dazwischen liegt. In einem solchen Fall muss entweder die Netztopologie angepasst werden, oder es muss auf jedem dazwischen liegenden Host ein *DHCP relay agent* installiert werden.

Mit dem Befehl *ifconfig*. wird überprüft, ob die IP-Adressdaten richtig an den DHCP- Client zugewiesen wurden.

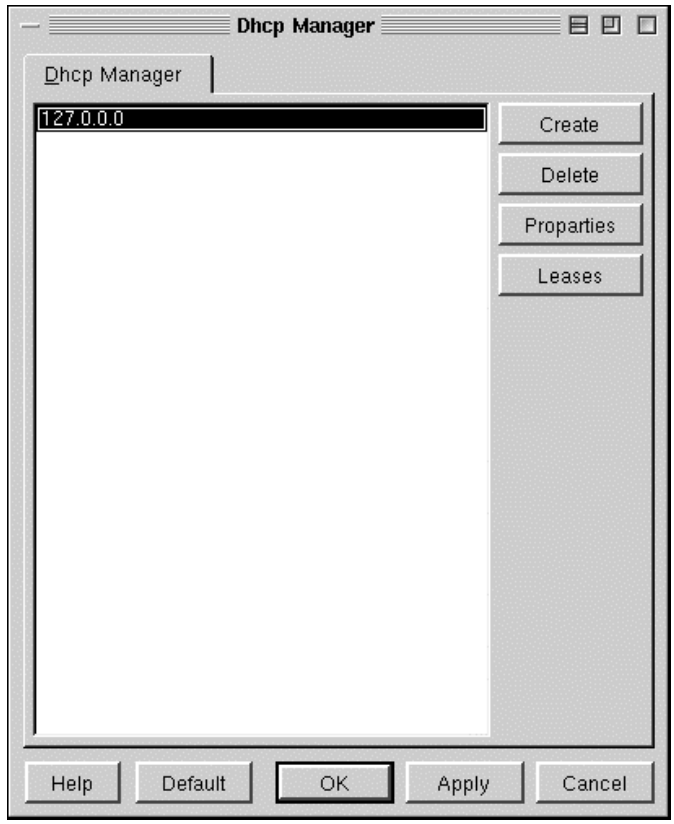


Abbildung 3-6 Das Programm kcmdhcpd

Konfiguration und Test eines DHCP-Clients unter Windows

Jeder Windows-Client kann so konfiguriert werden, dass er nach jedem Neustart des Rechners automatisch vom nächstern DHCP-Server eine IP-Adresszuweisung anfordert. Dazu muss lediglich in der Konfigurationsform des Protokolls TCP/IP die Option *IP-Adresse von einem DHCP-Server beziehen* aktiviert werden (siehe Abbildung 3-8):

Zusätzlich dürfen keine weiteren TCP/IP-Optionen an dieser Stelle angegeben werden, denn diese überschreiben, die über DHCP zugewiesenen Parameter!

Ob die IP-Adresszuweisung erfolgreich ausgeführt werden konnte, wird mit den Befehlen *ipconfig /all* (Windows NT 4.0) oder *winipcfg* (Windows 95/98) geprüft (siehe Abbildung 3-9):

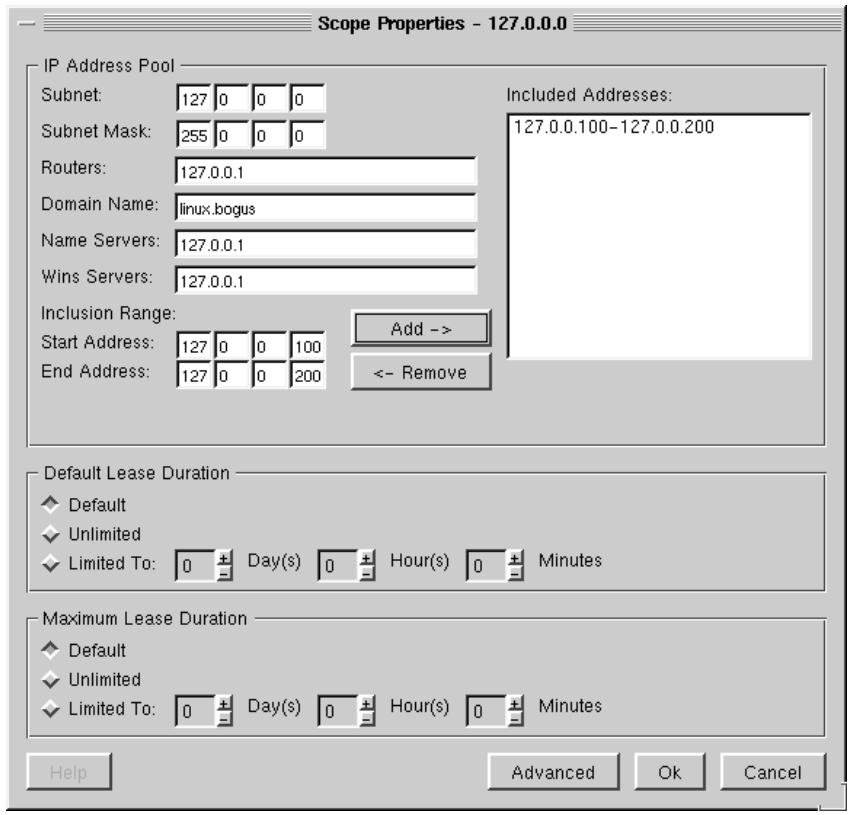


Abbildung 3-7 Definition eine IP-Adressraumes mit kcmdhcpd

Netzwerktools

knu

Knu steht für KDE Network Utilities und bezeichnet ein universelles Testprogramm für Netzwerkverbindungen. Die Programme *ping*, *traceroute* und *finger* werden über diese grafische Oberfläche aufgerufen (siehe Abbildung 3-10):

kfinger

kfinger ist die grafische Oberfläche für den Shell-Befehl *finger,* mit dem getestet werden kann, welche Anwender sich momentan aktiv im System befinden (angemeldet sind). Die Prüfung kann über *kfinger* in periodischen Abständen erfolgen, zusätzlich kann eine *talk*-Verbindung aufgebaut werden.

ktalk

ktalk ist die KDE-Variante des *talk*-Befehls. Mit *ktalk* wird eine bidirektionale Chat-Verbindung zwischen zwei oder mehreren Benutzern aufgebaut. Diese können Eingaben in ein Fenster schreiben und erhalten Antworten in einem zweiten Fenster.

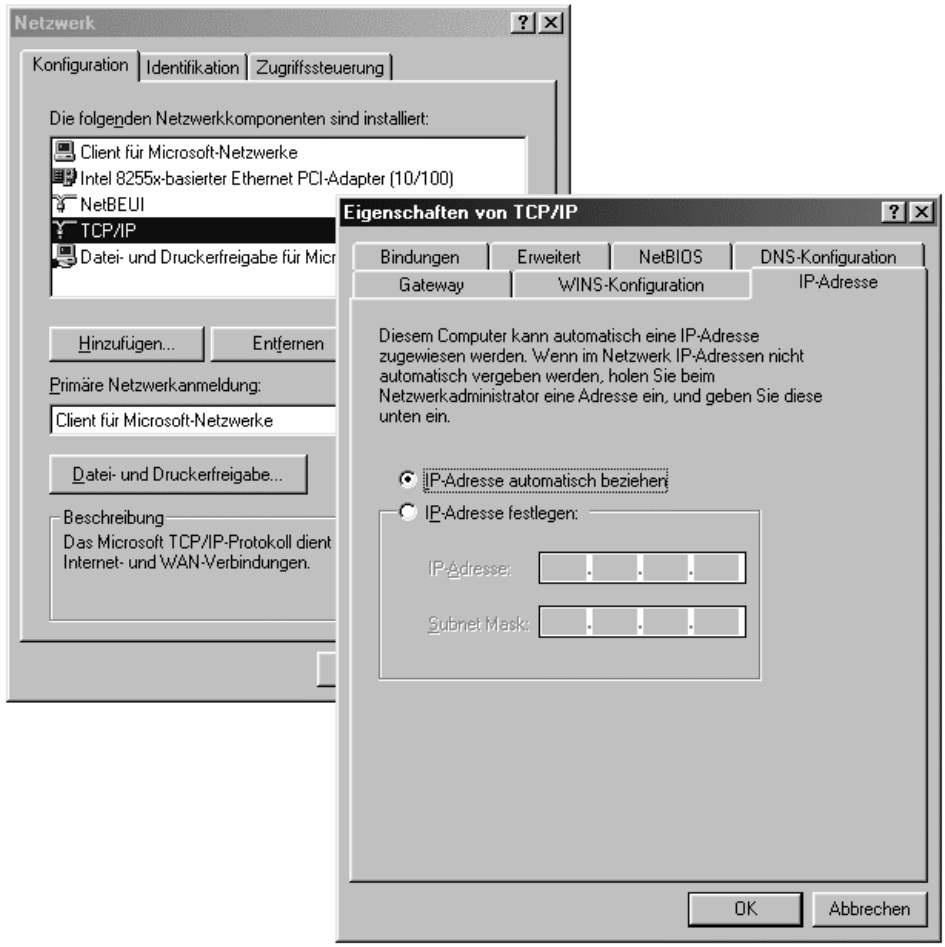


Abbildung 3-8 Konfiguration von TCP/IP unter Windows 98

ksniffer

Das Erstellen von Netzwerkstastiken ist mit dem Programm *ksniffer* möglich. Auf grafischen Oberflächen können die absoluten Zahlenwerte dargestellt werden, entweder für alle oder auch nur für ausgewählte Netzwerk-Interfaces (Abbildung 3-11).

Server-Fernsteuerung

Der direkte Zugriff auf einen Server per Tastaur und Maus ist eher die Ausnahme. Die Server stehen nicht mehr ungeschützt am Schreibtisch des Systemverwalters, sondern sie unterliegen baulichen Absicherungsmaßnahmen und werden in Räumen zusammengefasst, in der die empfindliche Technik geschützt ist.

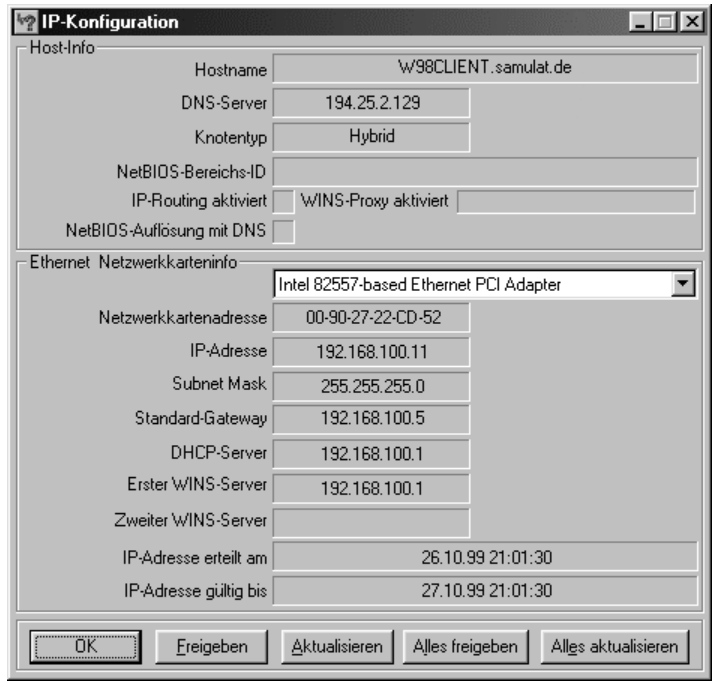


Abbildung 3-9 Der Ausgabe des Befehl winipcfg unter Windows 98

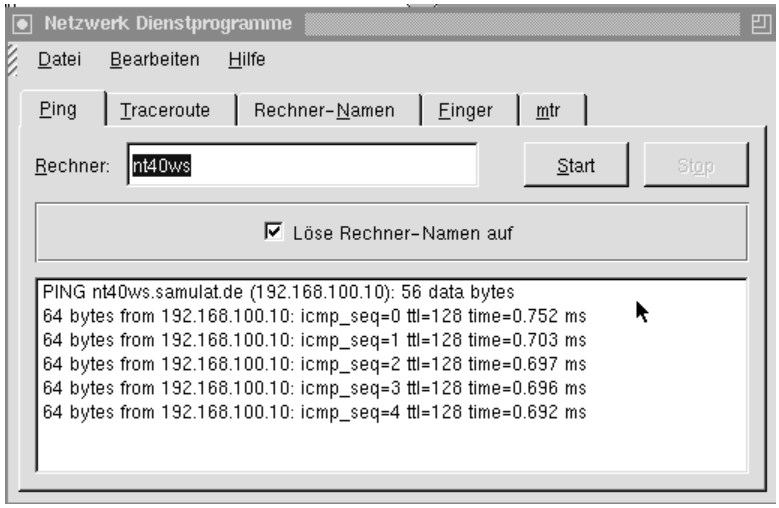


Abbildung 3-10 Programm knu

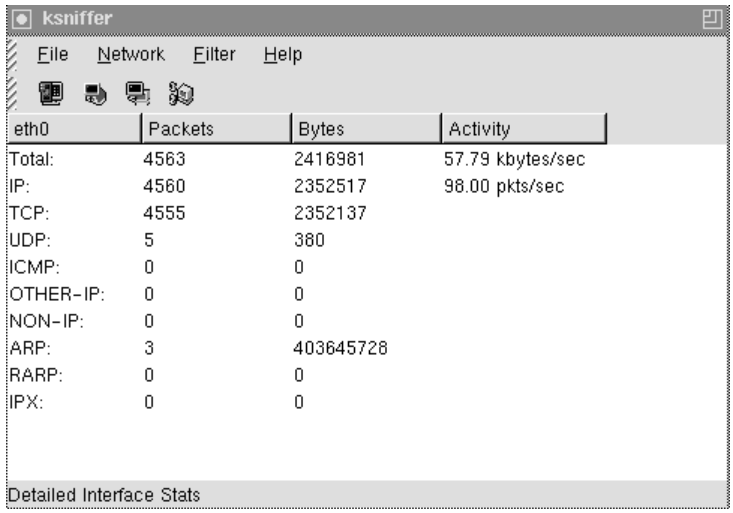


Abbildung 3-11 Programm ksniffer

Für den Systemverwalter ist es damit unumgänglich, bestehende Netzwerkverbindungen zu nutzen und von seinem PC-Arbeitsplatz auf möglichst alle Serverfunktionen (fern-)zugreifen zu können, so dass ihm das Schicksal des vereinsamten Kellerkindes erspart bleibt. Die dazu benötigten Programme lassen sich grob in drei Kategorien einteilen:

- *Terminalprogramm*
  Es stellt dem Bediener eine virtuelle, textorientierte Konsole zur Bedienung des Servers zur Verfügung. Die Programmierung erfolgt über Shell-Kommandos.
- *Virtueller_Desktop*
  Dies ist ein Programm, dass dem Bediener den echten Fernzugriff auf die grafische Bedieneroberfläche des Servers ermöglicht (virtueller Desktop). Alle am Server direkt verfügbaren Werkzeuge zur Systemsteuerung stehen uneingeschränkt zur Verfügung.
- *Administrationsprogramm*
  Das Programm gibt dem Systemverwalter über eine eigene Oberfläche den Zugriff auf Funktionen zur Systemverwaltung. Ein Administrationsprogramm kann eine eigenständige Anwendung darstellen, die auf dem Client-PC installiert werden muss oder sie ist in Perl oder Java programmiertes Client/Server-Modell, das über einen WWW-Browser bedienbar ist..

Voraussetzung für jede Server-Fersteuerung ist es, dass der Bediener die Anmeldung am Zielsystem so ausführen kann, dass die für die Systemverwaltung erforderlichen Berechtigungen gegeben sind. Unter Linux ist es dem Systemverwalter *root* grundsätzlich nicht erlaubt, einen Fernzugriff auf den Server auszuführen.

Diese Anmeldung könnte zwar erlaubt werden (z.B. über das Konfigurationsprogramm YAST); es ist aber aus Sicherheitsgründen nicht sinnvoll und auch nicht notwendig, diese Einschränkung aufzuheben: Wenn der Systemverwalter sich zunächst mit einem Standard-Benutzeraccount anmeldet, kann dieser mit dem Shell-Befeh *su* und dem richtigen Kennwort jederzeit zum echten *root* werden.

Terminalprogramm Telnet

Das zum Standardumfang von TCP/IP gehörende Programm *telnet* ist die Benutzerschnittstelle zum Kommunikationsprotokoll *telnet*, das die Erstellung einer Kommunikationsbeziehung zu jedem TCP/IP-Host ermöglicht, der den entsprechenden Server-Dienst zur Verfügung stellt. Linux-Systeme führen diesen Server-Dienst, im Gegensatz zu Windows NT- oder Novell-Servern, standardmäßig aus. *Telnet* ist ein relativ einfaches, aber leistungsfähiges Protokoll, das dem Bediener eine virtuelle Textkonsole zur Verfügung stellt.

Das Programm ist unter Win9x/NT-Systemen verfügbar, sobald TCP/IP installiert ist. Das Programm wird in der Regel von der Kommandozeile gestartet (Abbildung 3-12). Als Parameter reicht es in der Regel aus, den Hostnamen oder die IP-Adresse des Host anzugeben.

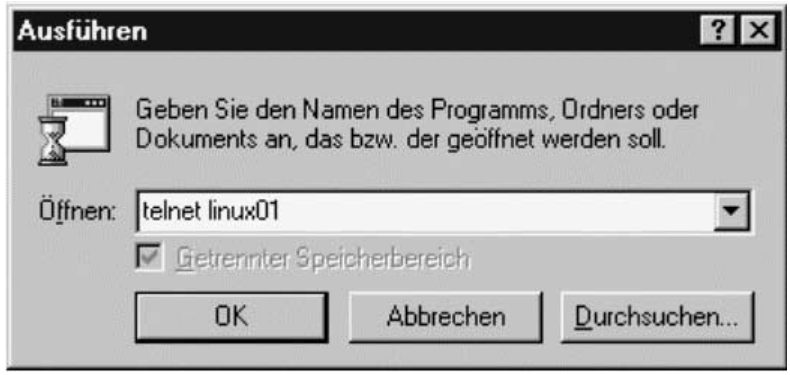


Abbildung 3-12 Start von telnet unter Windows 9x/NT (Beispiel)

Abbildung 3-13 zeigt die Oberfläche des Programmes *telnet* unter Windows NT und die Anmeldung als Superuser mit dem Befehl *su*. Der Verbindungsaufbau erfolgt vom Rechner *nt40ws.samulat.de*.

In den meisten Fällen wird der eben beschriebene einfache Telnet-Aufruf genügen. Es gibt aber eine Reihe von weiteren möglichen Befehlszeilenoptionen. Tabelle 3-6 zeigt eine Auswahl, weitere Informationen können den man-Pages zu *telnet* entnommen werden.

Abbildung 3-13 Telnet: Programmoberfläche

Option	Beschreibung
Portnr	Standardmäßig wird für eine Telnet-Verbindung immer der TCP-Port 23 verwendet. Mit dieser Option kann die Portnummer bei Bedarf geändert werden.
-a	automatisches Login ausführen
-e *zeichen*	Das Zeichen *zeichen* ist Fluchtzeichen für diese Telnet-Sitzung.
-l *username*	Der Name *username* wird (zusammen mit der Option -a) zum automatischen Login verwendet.
-n *datei*	Einzelschrittmodus, die Daten werden in *datei* gespeichert.

Tabelle 3-6 Telnet-Optionen (Auswahl)

Telnet-Verbindungen bringen immer wieder Probleme mit der Funktion von Steuer- und Sonderzeichen sowie den Funktionstasten. Bietet das Telnet-Programm unterschiedliche Terminal-Emulationen an (in der Regel mindestens VT100 und VT220), so sollten zunächst unterschiedliche Einstellungen versucht werden. Auf der Serverseite wird die Terminalemulation mit der Umgebungsvariablen *TERM* beeinflußt. Diese kann z.B. auch während einer Telnet-Sitzung verändert werden. Sie muss dann aber mit dem Shell-Kommando *export* zusätzlich mit den anderen Shells bekanntgegeben werden:

```
# TERM = vt100
# export TERM
```

Einige Distributionen erstellen im HOME-Verzeichnis der Benutzer Konfigurationsdateien, durch die weitere Einstellungen der Terminals vorgenommen werden können. Sie beeinflussen das Verhalten der virtuellen Terminals bis zur einzelnen Taste. Bei der SuSE-Distribution tragen diese Dateien Namen wie *uitrc.Terminalname*.

Soll eine Telnet-Sitzung dazu verwendet werden, einen Prozess zu starten, der auch nach Beenden dieser Sitzung weiter aktiv bleibt, muss ein kleiner Trick zu Anwendung kommen: Der Shell-Befehl */usr/bin/nohup* verhindert, dass der Prozess beim Beendenden der aktuellen Sitzung ebenfalls beendet wird. Der Prozess muss zusätzlich mit einem nachgestellten »&« als Hintergrundprozess gestartet werden. Als Beispiel soll der Start ein Kompilation mit *make* gezeigt werden [ROEH97]:

```
# nohup make -d -f myfile &
[1] 1278
# nohup: appending output to `nohup.out´
# exit
Connection closed by foreign host
#
```

Die Ausgaben des Programms *make* werden jetzt in die Datei *nohup.out* umgeleitet und der Prozess bleibt auch nach dem Ende der Sitzung aktiv.

Mit dem Befehl *tee* kann eine Telent-Sitzung vollständig dokumentiert werden, d.h. alle Ein- und Ausgaben werden in einer Datei protokolliert. Mit

```
telenet linux01 | tee protocol.log
```

werden alle Aktionen dieser Sitzung in *protocol.log* aufgezeichnet.

Sollte *telnet* z.B. nach versehentlicher Ausgabe einer Binärdatei mit *cat* oder nach einem fehlgeschlagenen Versuch mit *setterm* nur noch unleserliche Sonderzeichen anzeigen, so kann diese Textkonsole mit dem Befehl *reset* reinitialisiert werden.

Terminalprogramm TeraTerm Pro

Das Programm *TeraTerm* ist ein textorientiertes Terminalprogramm für Windows 9x/NT. Es erlaubt das Herstellen von Verbindungen über eine serielle Schnittstelle oder über TCP/IP (Telnet Connection). *TeraTerm* bietet

- Emulationen für VT100, VT200/300 und TEK4010,
- Kermit, XMODEM, ZMODEM, B-PLUS und Quick-VAN-Protokolle zDm-Dateitransfer,
- eine Skriptsprache »Tera Term Language« und
- japanische und russische Zeichensätze.

Unter *http://www.vector.co.jp/authors/VA002416/teraterm.html*. sind weitere Informationen zum diesem Programm sowie eine Download-Möglichkeit zu finden. In der SuSE-Distribution ist TeraTerm auf der ersten CD im Verzeichnis *dosutils* zu finden.

Nach der Programminstallation und dem Programmstart wird zunächst festgelegt, welche Verbindungsart (TCP/IP oder seriell) verwendet werden soll (Abbildung 3-14). Bei einer TCP/IP-Verbindung ist der Hostname und die IP-Adresse des Zielrechners anzugeben.

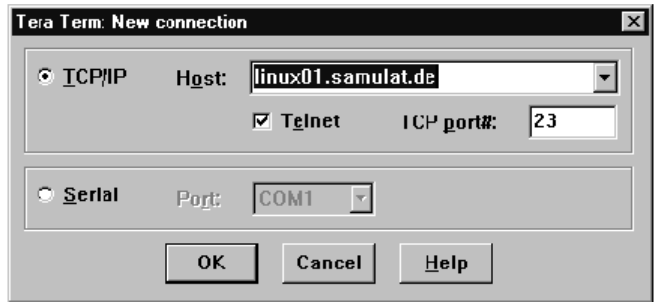


Abbildung 3-14 TeraTerm: Aufbau der Verbindung

Nach dem erfolgreichen Aufbau der gewünschten Verbindung kann die Anmeldung am System erfolgen (Abbildung 3-15). Zu beachten ist, dass viele sonst direkt am Linux-System zu verwendenden Tastaturkommondos nicht richtig funktionieren. Die Funktionstasten F1 bis F12 sind nicht zur Steuerung verwendbar, viele weitere Tastenkombinationen nur teilweise.

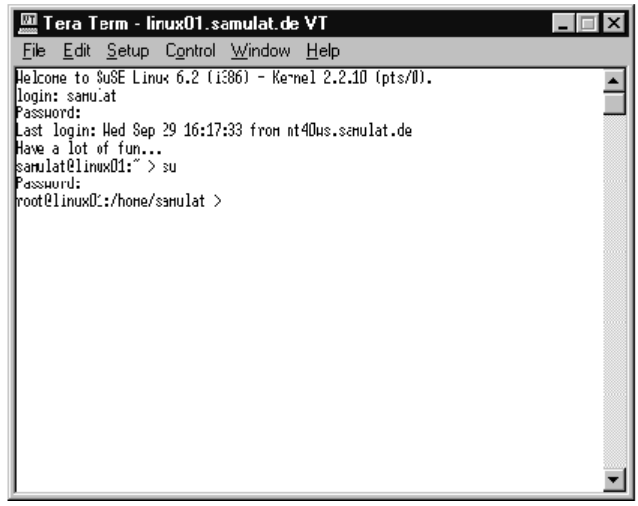


Abbildung 3-15 TeraTerm: Programmoberfläche

Empfehlenwert ist die gut gegliederte und sehr umfangreiche Hilfe-Funktion, in der auch viele Kommandos zur Steuerung und Automatisierung der Verbindung ausführlich dokumentiert sind.

Virtuelle Konsole mit VNC

Für die Systemverwalter von Windows- oder Novell-Netzwerken ist die Arbeit mit den dazu gehörenden grafisch orientierten Betriebssystemoberflächen selbstverständlicher Standard. Natürlich bietet auch Linux mit XWindows eine solche grafische Oberfläche, wobei eine Umstellung auf textorientierte Werkzeuge für den Server-Fernzugriff wie die vorstehend beschriebene *telnet* oder *TeraTerm* schwer fallen. Auch hier soll natürlich der Fernzugriff mit den grafischen Oberflächen möglich sein.

Das VNC-Protokoll

Das Programmpaket *VNC* (*Virtual Network Computing*) ermöglicht die Fernsteuerung eines Linux-Servers über ein grafisches Interface. Müssen dazu unter Windows Zusatzprogramme wie z.B. *PC Anywhere* oder *Carbon Copy* eingesetzt werden, bietet das auf dem Prinzip der *Remote Framebuffer* (RFB) basierende VNC-Protokoll einen völlig anderen Ansatz: Es erlaubt den nahezu betriebssystemunabhängigen Fernzugriff. Voraussetzung ist lediglich eine bestehende Netzwerkverbindung unter TCP/IP.

VNC realisiert eine Client-Server-Verbindung, bei der die auf dem fernzusteuernden Rechner installierte Serversoftware die Bildinformationen per RFB an die Client-Anwendung (Viewer) sendet: (Abbildung 3-16).

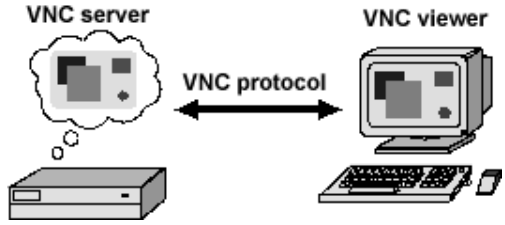


Abbildung 3-16 VNC Konfiguration

Die Software ist unter anderem lauffähig unter X/Unix, (Linux), Windows 3.1/9x/NT und Macintosh, ein Java-basierter Client ermöglicht sogar den Fernzugriff auf einen VNC-Server über einen Web-Browser.

VNC ist somit nicht nur für die Fernadministration von Linux-Servern durch Windows-Clients einsetzbar, auch alle im Netzwerk laufenden Windows- und Mac-Rechner können mit der gleichen Software »ferngesteuert« werden, soweit diese per TCP/IP erreichbar sind. Die Konfigurationsmöglichkeiten des VNC Viewers erlauben es dabei sogar, mehrere Clients gleichzeitig auf einen VNC-Server zugreifen zu lassen, sodass ein gemeinsames Arbeiten möglich wird.

Interessantes Detail ist, dass die Speicherung der Bildinformation ausschließlich durch den VNC-Server erfolgt, so dass eine Verbindung jederzeit unterbrochen und ohne Datenverlust wiederhergestellt werden kann, auch von wechselnden Rechnern.

Aus Sicht einer Linux-Anwendung verhält sich der VNC-Server wie ein Standard X-Windows-Display, nur dass es nicht direkt auf den gleichen Rechner zur Anzeige kommt

Weitere Informationen zum Programmpaket VNC stehen im Internet unter *www.uk.research.att.com/vnc* zur Verfügung.

VNC-Server für Linux

Um einen VNC-Server unter Linux verwenden zu können, muss das entsprechende Programmpaket zunächst installiert werden.

Die aktuelle VNC-Software für Linux kann über *www.uk.research.att.com/vnc* aus dem Internet geladen werden oder man installiert aus der SuSE-Distribution das Paket *vnc* aus der Serie *xap* (2,8 MByte).

Der VNC-Server wird durch den Befehl

```
vncserver
```

auf dem Linux-Server gestrartet. Ist dies der erste Start auf diesem Rechner, so wird zunächst ein Kennwort abgefragt, dass später immer dann vom VNC Client abgefragt wird, wenn die Verbindung zu diesem Server hergestellt werden soll:

```
# vncserver
You will require a pasword to access your desktops.
Password:
Verify:

New ‚X' desktop is linux01:1

Creating default startup script /root/.vnc/xstartup
Starting applications specified in /root/.vnc/xstartup
Log file is /root/.vnc/linux01:1.log
# _
```

Grundsätzlich können auf einem Linux-Server mehrere VNC-Server gestartet werden, um z.B. unterschiedliche Informationen abrufen zu können. Alle VNC-Server verwenden dann das gleiche Kennwort. Mit dem Befehl

```
vncpasswd
```

kann das VNC-Server-Kennwort jederzeit geändert werden.

Der oder die VNC-Server werden vom Client über den Rechnernamen bzw. Rechner-IP-Adresse und eine laufende Nummer (Displaynummer) angesprochen, dabei hat der zuerst gestartete VNC-Server immer die Nummer 1. Heißt der Linux Server *linux01*, so sind die Namen der beiden ersten VNC-Server *linux01:1* und *linux01:2*.

VNC-Viewer für Win32

Der VNC-Viewer für Win9x/NT ist das Programm *vncviewer.exe* in der Version 3.3.3r1. Eine Installation muss nicht erfolgen; für den Start des Clients reicht der Zugriff auf das Programm *vncviewer*. Nach dem Programmstart ist zunächst der Host-Name und die Displaynummer anzugeben (Abbildung 3-17).

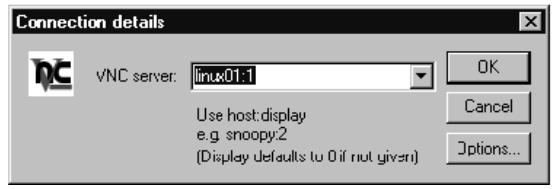


Abbildung 3-17 Startform des Programms vncviewer.exe

Ist der VNC-Server unter dem angegebenen Namen im Netzwerk erreichbar, so wird die Verbindung nur dann hergestellt, wenn in der nächsten Form das aktuelle Server-Kennwort eingegeben wird (Abbildung 3-18).

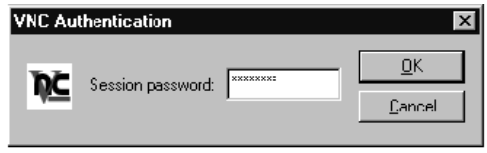


Abbildung 3-18 Abfrage des VNC-Server Kennwortes

Der VNC-Viewer stellt jetzt den aktuellen Bildinhalt des VNC-Servers dar (Abbildung 3-19).

In der Standardkonfiguration ist zunächst nur ein Terminalfenster auf dem Desktop geöffnet. Weitere textorientierte Terminals können z.B. mit dem Shell-Befehl »*xterm* &« geöffnet werden.

Im Gegensatz zu den textorientierten Programmen zur Serverfernsteuerung, wie z.B. *telnet* und *TeraTerm* gibt es mit dem VNC-Viewer keine Einschränkungen in der Programmsteuerung via Funktions- oder Steuertasten. Es steht der vollständige Bedienungsumfang zur Verfügung, so läßt sich z.B. auch das Programm *mc* (*midnight commander*) ohne Einschränkungen verwenden.

Interessant ist, wie unter VNC Standardfunktionen zur Steuerung des Desktops, z.B. Fenster in der Größe zu verändern oder zu schließen, angesteuert werden: Wird der Mauszeiger auf dem freien Desktop positioniert und die linke Maustaste

für längere Zeit gedrückt, so erscheint das Menü *Tvm*, in dem diese und weitere Funktionen ausgewählt werden können. Um ein Fenster zu beenden, klickt man mit der linken Maustaste auf *Kill*, danach ist der Mauszeiger ein Käfer, der dann auf dem zu beendenden Fenster positioniert wird. Ein weiterer Mausklick schließt dann das Fenster.

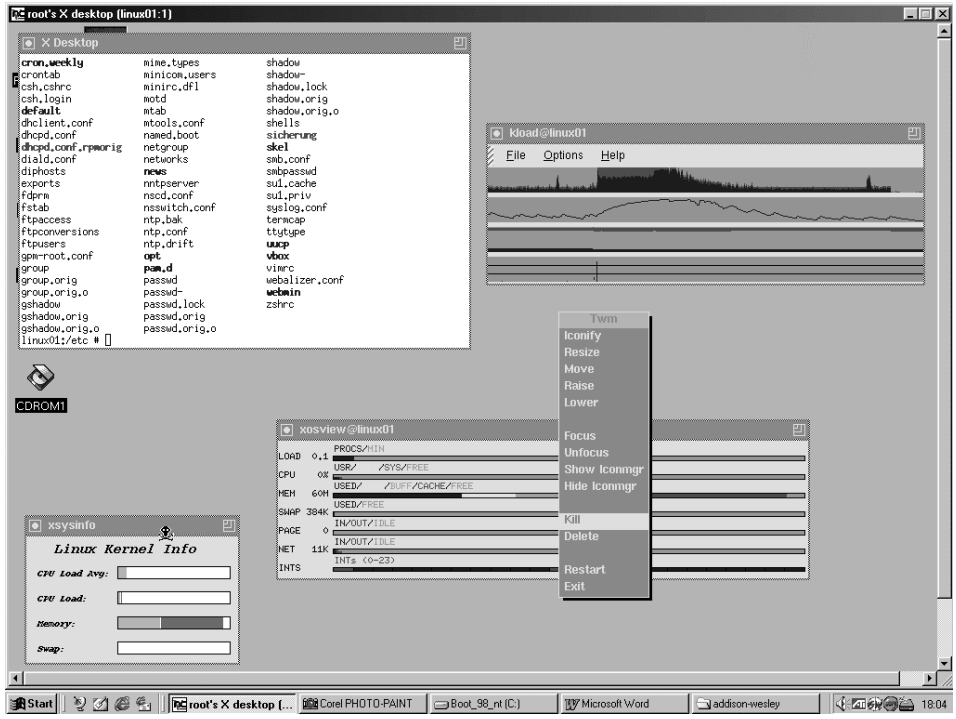


Abbildung 3-19 Der VNC-Viewer unter Windows 98

Das Programm *vncviewer* kann mit einer Vielzahl von Optionen gestartet werden, mit *vncviewer -h* erhält man eine vollständige Liste. Hier eine kurze Auswahl:

- *-shared*
  Mit dieser Option kann von mehreren Clients gemeinsam auf einen VNC-Server zugegriffen werden. Jeder Viewer stellt, da ja nur der VNC-Server die Bildinformationen speichert, jede von einem beliebigen Client aus vorgenommene Änderung sofort dar. Ohne diese Option werden bestehende Verbindungen getrennt, es erfolgt ein Exklusiv-Zugriff.
- *-viewonly*
  Am Viewer vorgenommene Bedienungen per Maus oder Tastatur werden nicht an den VNC-Server weitergegeben. Damit kann ein Display dargestellt werden, das von diesem Platz nicht verändert werden kann, z.B. für einen Messe- oder Austellungsbetrieb ist diese Option sehr hilfreich.

- *-periodms*
  Diese Option kann bei Bedarf die Netzwerklast verringern. Der Bildaufbau erfolgt in dem in Millisekunden angegebenen Zeitraster.

Internet-Browser als VNC-Viewer

Ganz ohne Installation eines Viewer-Programmes kann auf einem VNC-Server auch dann zugegriffen werden, wenn der Client über einen Java-fähigen WWW-Browser überfügt.

Der VNC-Server arbeitet immer auch als einfacher WWW-Server. Stellt man über den WWW-Browser die Verbindung her, so wird der Bildinhalt über den dann vom Server geladenen JAVA-Viewer angezeigt. Der VNC-WWW-Server antwortet auf HTTP Verbindungen mit der Portnummer 5800 + Displaynummer:

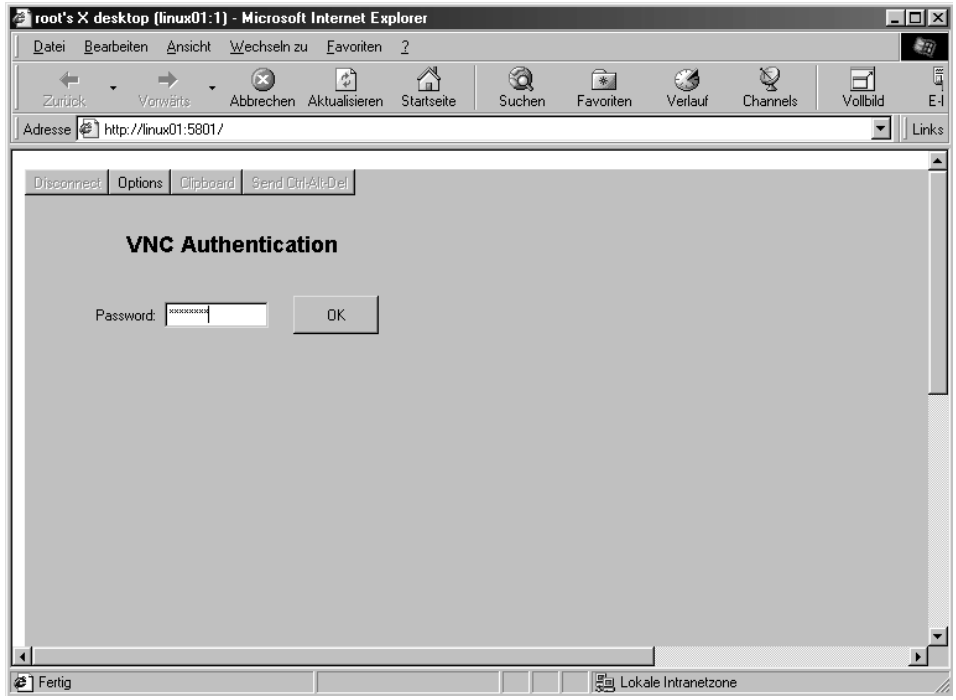


Abbildung 3-20 VNC-Client mit WWW-Browser: Herstellen der Verbindung

Abbildung 3-20 zeigt den Zugriff auf den Server *linux01* mit der Displaynummer 1. Im Vergleich zum vorstehend beschriebenen Viewer für Win32 ist der Java-orientierte VNC- Client zwar etwas langsamer, es entfällt aber in jedem Fall die Notwendigkeit, spezielle Software für den Fernzugriff verfügbar zu halten.

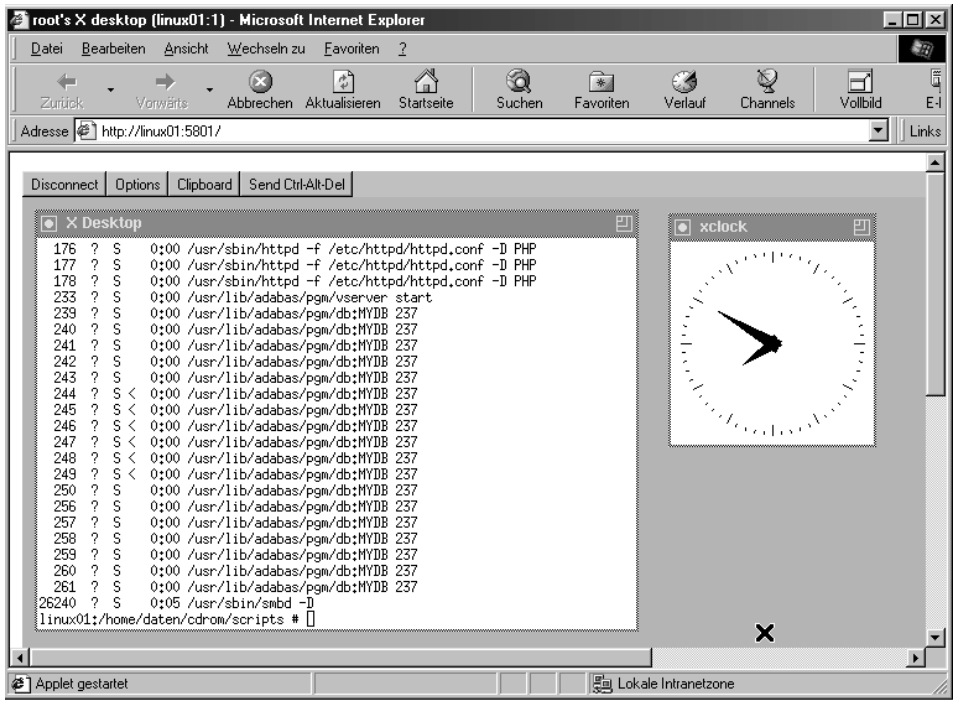


Abbildung 3-21 Java VNC Client

Adminstrationsprogramm webmin

webmin ist ein WWW-basiertes Programm zur Systemverwaltung von UNIX-Rechnern. Mit jedem WWW-Browser, der Tabellen und Formulare unterstützt, können die Benutzerverwaltrung und Dienste wie DNS, DHCP, Server Apache und vieles mehr einfach verwaltet werden. Unterstützt der Browser JAVA, so können auch Arbeiten zur Dateiverwaltung ausgeführt werden.

webmin ist geschrieben in Perl 5, das Programm besteht aus einem einfachen WWW-Server und einer Reihe von cgi-Skripten, die die UNIX-Systemdateien (in erster Linie die Dateien in */etc*) direkt modifizieren. Voraussetzung für die Installation und den Betrieb von *webmin* auf einem Server ist lediglich eine Perl-Laufzeitumgebung, wie sie heute zum Standardumfang fast aller Linux-Distributionen gehört.

Das Programm *webmin* ist unterhalb der Version 1.0 frei verfügbar, die nachfolgenden Beispiele beziehen sich auf die Version 0.74 (10.10.1999) (auf der CD-ROM im Verzeichnis */utils/webmin*).

Für erhöhte Sicherheitsanforderungen kann *webmin* auch SSL-Verbindungen nutzen: Die dafür benötigte Erweiterung muss jedoch zusätzlich geladen werden.

Weitere Informationen und Download-Möglichkeiten für *webmin* und dazu gehörende Erweiterungen sind zu finden unter *http://www.webmin.com/webmin*.

Zur Installation wird die Datei *ftp://ftp.webmin.com-0_74.tar.gz* benötigt. Das nachfolgende Listing zeigt exemplarisch den weiteren Installationsablauf:

```
root@linux01 # cp webmin-0.74.tar.gz /usr/local
cd /usr/local
gunzip webmin-0_74.tar.gz
tar xf webmin-0_74.tar
cd webmin-0_74
#./setup.sh
```

Das Skript *setup.sh* fragt die weiteren Konfigurationsdetails ab, wobei in der Regel die Standardvorgaben mit [Return] unverändert übernommen werden können. Das Web-*Server Login* sollte den Namen *admin* behalten; das Kennwort ist frei wählbar. Zu beachten ist lediglich, dass die Option *Start Webmin at boot time* mit »*Y*« beantwortet wird, damit nach dem nächsten Systemstart des Linux-Servers *webmin* automatisch gestartet wird.

Das Skript *setup.sh* startet anschließend sofort das Programm.

Webmin verwendet standardmäßig den HTTP-Port 10000, sodass der Verbindungsaufbau von einem WWW-Browser mit der Adresse *Hostname:10000* erfolgt. Nach der Abfrage des bei der Installation festgelegten Benutzernamens und des Kennworts wird die Arbeitsoberfläche von *webmin* im WWW-Browser angezeigt:

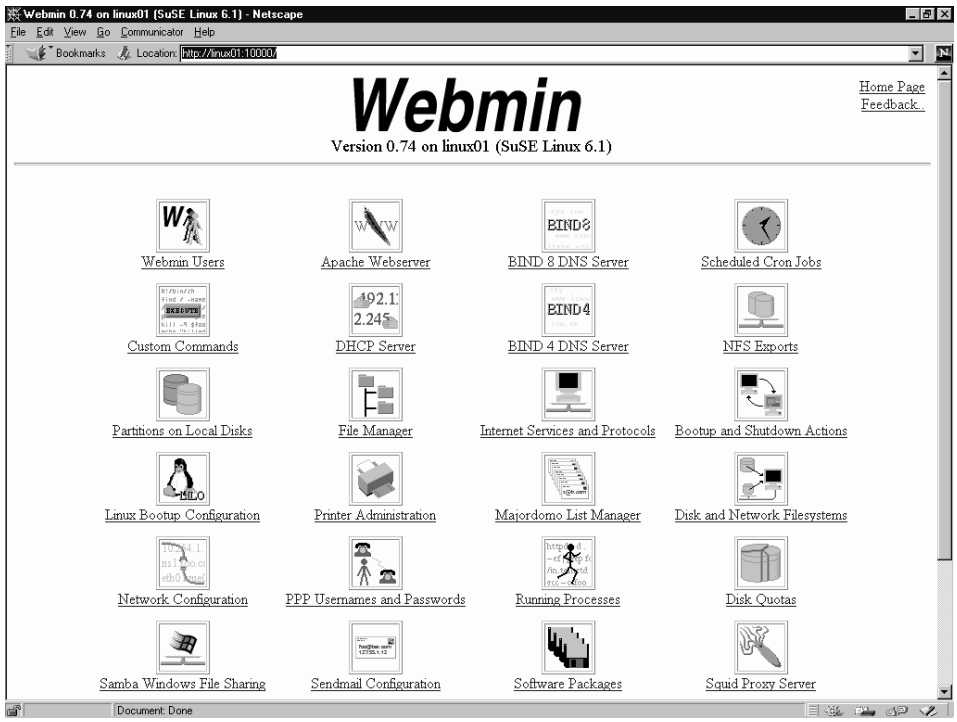


Abbildung 3-22 Bedieneroberfläche von webmin 0.74 (Ausschnitt)

Die in *webmin* zur Verfügung stehenden Funktionen werden über Module realisiert. Eigene Erweiterungen können jederzeit vorgenommen werden. Tabelle 3-7 zeigt die in der Version 0.74 standardmäßig enthaltenen Module.

Modul Name	Beschreibung
Linux Boot Loader	Edit kernels and partitions selectable at boot time with LILO.
Telnet Login	Login to your system with telnet.
Custom Commands	Create buttons to execute commonly used commands on your system.
Network Configuration	Configure interfaces, DNS, routing and /etc/hosts for Solaris and Redhat Linux.
DHCP Server	Manage subnets, hosts and groups for ISC DHCPD version 2.
Majordomo List Manager	Create and configure mailing lists for Majordomo version 1.94.
File Manager	View, edit and change permissions on files and directories on your system with a Windows-like file manager.
FreeBSD NFS Exports	Edit file shares from the FreeBSD `/etc/exports` file.
HPUX NFS Exports	Edit file shares as defined in the HPUX `/etc/exports` file.
PPP Usernames and Passwords	Manage dialout and dialin PPP users under Linux.
Squid Proxy Server	Configure Squid options, ACLs, caching parameters and proxy users.
Sendmail Configuration	Manage sendmail aliases, masquerading, address rewriting and other features.
Printer Administration	Create and edit local and remote printers. Supports Windows print servers and Ghostscript print drivers.
BIND 4 DNS Server	Create and edit domains and DNS records.
BIND 8 DNS Server	Create and edit domains, DNS records and BIND 8 options.
Apache Webserver	Configure almost all Apache directives in versions 1.1, 1.2 and 1.3.
Schedule Cron Jobs	Create, edit and delete cron jobs.
Solaris NFS Shares	Edit file shares as defined in the `/etc/dfs/dfstab` file.
Linux NFS Exports	Edit NFS file shares defined in `/etc/exports`.
Internet Services and Protocols	Edit services in `/etc/inetd.conf`, `/etc/services` and `/etc/rpc`.
Bootup and Shutdown Actions	Setup scripts to be run at boot time from `/etc/init.d` or `/etc/rc.local`.
Disk and Network Filesystems	Mount filesystems and swap files usually configured in `/etc/fstab` or `/etc/vfstab`.

Tabelle 3-7 Standard-Module in webmin 0.74

Modul Name	Beschreibung
Samba Windows File Sharing	Create and edit samba file and print shares.
Users, Groups and Passwords	Create and edit Unix users and groups from the `/etc/passwd` and `/etc/group` files.
Linux Partitions on Local Disks	Create and edit paritions on local SCSI and IDE disks on Linux
Solaris Partitions on Local Disks	Create and edit paritions on local disks on Solaris.
Running Processes	List, kill and renice running processes on your system.
Software Packages	Manage Solaris, Redhat RPM and HPUX packages on your system, and install new packages.
Disk Quotas	Setup and edit user or group disk quotas for local filesystems.
Webmin Configuration	Change the admin username/password, web server port and list of allowed hosts for the webmin server.
Webmin Users	Create Webmin users and configure which modules and features they are allowed to access.

Tabelle 3-7 Standard-Module in webmin 0.74

Das nachstehende Beispiel verdeutlicht die Arbeit mit *webmin*. Angenommen wird, dass in der DHCP-Konfiguration des Servers *linux01* eine Änderung vorgenommen werden soll. Dazu muss die Datei */etc/dhcpd.conf* editiert und der DHCP-Dienst neu gestartet werden.

Der erste Arbeitschritt erfolgt über den *File Manager* (Abbildung 3-23).

Der Filemanager stellt im rechten Fenster die Struktur des Dateisystems grafisch dar, per Mausklick wird in das Verzeichnis */etc* gewechselt. Das linke Fenster zeigt jetzt den aktuellen Inhalt, aus dem wiederum per Mausklick die Datei */etc/dhcpd.conf* ausgewählt wird.

Ein Klick auf die Taste *Edit* startet den Editor, mit dem die Bearbeitung dieser Konfigurationsdatei erfolgen kann. Abschließend wird die Datei */etc/dhcpd.conf* über Save gespeichert.

Im zweiten Arbeitsschritt muss jetzt der DHCP-Dienst neu gestartet werden. Das kann über das Modul *Bootup and Shutdown Actions* erfolgen, über das Dienste nicht nur konfiguriert werden können, sondern auch jederzeit manuell gestartet und gestoppt werden können (Abbildung 3-24).

Aus diesem Menü wird per Mausklick der Dienst *dhcp* ausgewählt. Angezeigt wird dann die aktuelle Konfiguration des Dienstes (Abbildung 3-25) .Hier kann bestimmt werden, zu welchem Runtime-Level dieser Dienst gestartet und gestoppt wird und wie das dazu verwendete Skript aussieht.

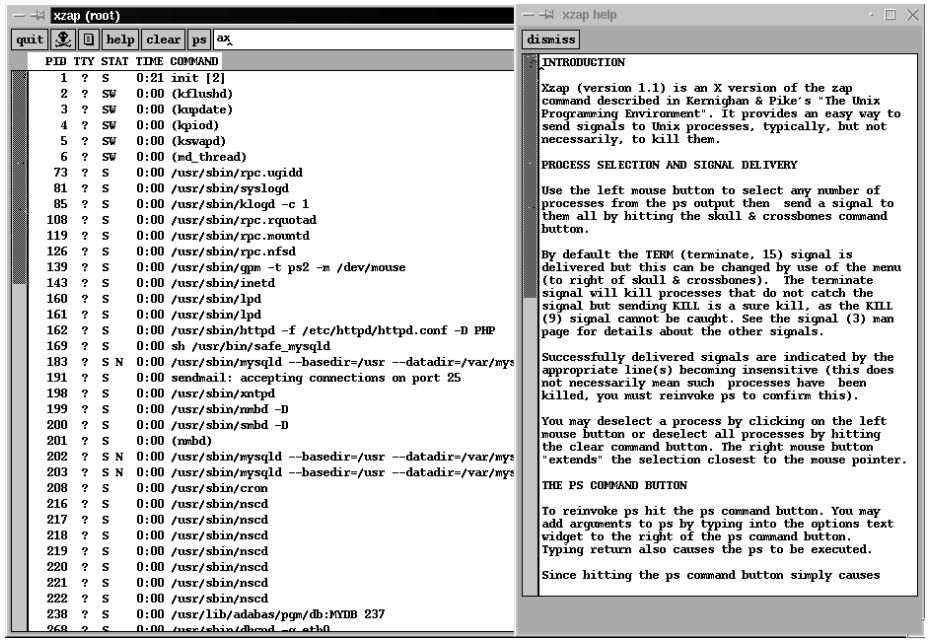


Abbildung 3-23 Arbeit mit dem webmin Filemanager

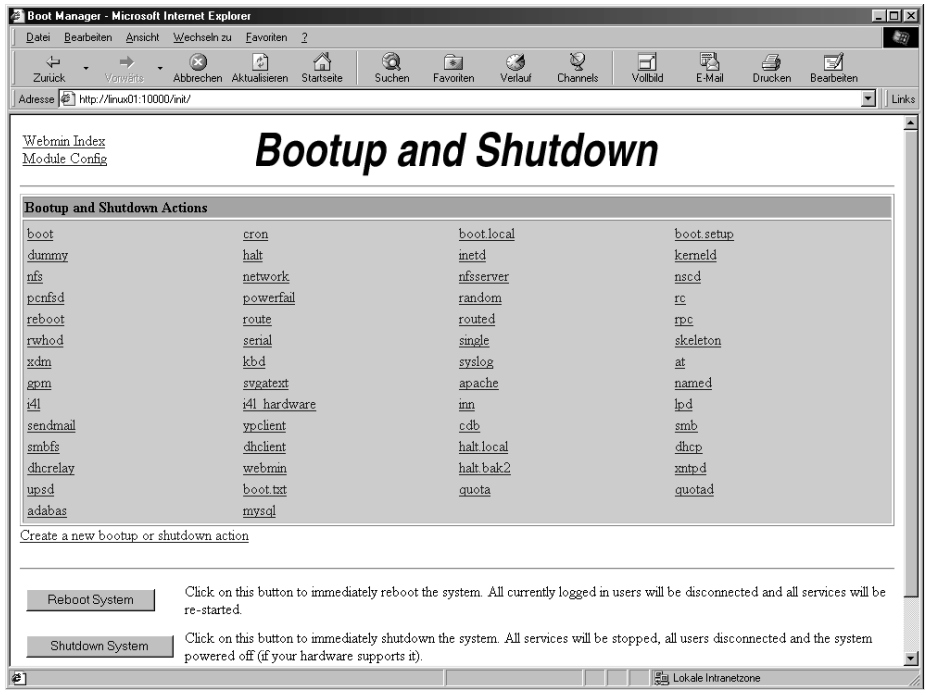


Abbildung 3-24 webmin Bootup and Shutdown Actions

Über die Tasten *Stop Now* und *Start Now* kann der Dienst *dhcp* neu gestartet werden.

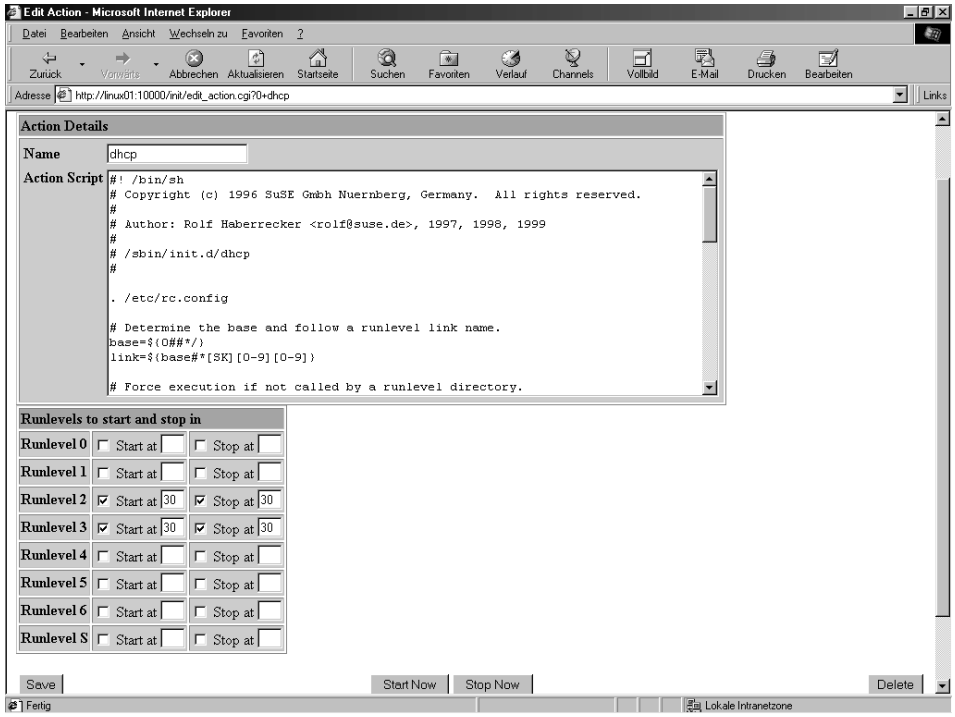


Abbildung 3-25 Informationen zum Dienst dhcp

Das Programm *webmin* wird bei der weiteren Konfiguration des Linux-Servers immer wieder eine wichtige Rolle spielen. Es wird dann als leistungsfähige grafische Oberfläche für die zur Konfiguration der Netzwerkdienste notwendigen Shell-Befehle verwendet. Der zur Grundkonfiguration notwendige Aufwand sinkt erheblich.

Aus diesem Grund sollte *webmin* grundsätzlich auf dem Linux-Server eingerichtet werden.

3.2 Druckdienste

3.2.1 Der Linux-Server als Druckserver

Die Aufgaben eines Druckservers im Netzwerk wahrzunehmen, gehört zu den Standardaufgaben jedes Servers. Direkt am Server oder auch an anderer Stelle im Netzwerk angeschlossene Druckgeräte sollen zentral für alle Benutzer bereitgestellt und mit minimalem Aufwand verwaltet werden.

Damit der Linux-Server zum Druckserver wird, sind grundsätzlich mindestens zwei Ansätze möglich, die auch von den Leistungsmöglichkeiten der eingesetzten Clients abhängig sind:

- Druckserver mit TCP/IP-Standardfunktionen oder
- Bereitstellung der Drucker über den Dienst *SAMBA*

Auf das später ausführlich behandelte Programmpaket *SAMBA* soll an dieser Stelle noch nicht eingegangen werden. Dargestellt werden soll zunächst, Planung, Betrieb und Verwaltung eines Standard-Druckservers unter TCP/IP.

Drucken unter Linux ist, auch wenn im ersten Schritt nur die Installation und der Betrieb eines direkt am Linux-Server angeschlossenen »lokalen« Druckers dargestellt werden soll, immer ein Netzwerk-Vorgang. Das konfigurierte TCP/IP-Interface, mindestens aber das lokale *Loopback*-Interface wird bereits in dieser Phase benötigt.

Druckaufträge werden unter Linux nicht direkt zum Druckgerät gesendet, sondern über Filterprogramme aufgearbeitet und dann in Warteschlangen (*Queues*) in Verzeichnissen ab */var/spool* abgelegt. Jede Warteschlange wird genau von einem Drucker abgearbeitet, es können aber mehrere Warteschlangen für einen Drucker eingrichtet werden.

Das eigentliche Drucken erfolgt über einen Hintergrundprozess, den *lpd*-Daemon. SuSE Linux enthält das *BSD Spooling System der University of California at Berkeley*.

Installation eines lokalen Druckers am Linux-Server

Wie auch bei anderen Betriebssystemen werden Drucker in der Regel an einer parallelen Schnittstelle angeschlossen, maximal sind drei Schnittstellen je Rechner möglich. Die zur Ansteuerung genutzten Gerätedateien sind */dev/lpx*, wobei *x* ab 0 zählt:

Schnittstelle	Gerätedatei
LPT1	/dev/lp0
LPT2	/dev/lp1
LPT3	/dev/lp2

Standardmäßig verfügt jeder PC nur über die parallele Schnittestelle *LPT1*. Ein einfacher Test für Drucker und Schnittstelle kann vom Systemverwalter *root* z. B. mit dem Befehl

```
cat textdatei > /dev/lp0
```

durchgeführt werden. Der Inhalt von *textdatei* wird direkt an die Schnittstelle ausgegeben.

Mehrere parallele Schnittstellen einrichten

Soll der Linux-Server als zentraler Druckserver eingesetzt werden, kann es ohne weiteres sinnvoll sein, mehr als eine parallele Druckerschnittstelle einzurichten. Frei konfigurierbare Zusatzkarten, die neben der schon vorhandenen Schnittstelle LPT1 (*/dev/lp0*) auch LPT2 und LPT3 bereitstellen, sind schon für wenig Geld zu haben. Ist eine solche Zusatzkarte eingebaut und meldet das BIOS beim Rechnerstart zwei oder drei parallele Schnittstellen, so wird diese Konfigurationsänderung von Linux nicht automatisch erkannt. Wird versucht, den oben für *LPT1* beschriebenen einfachen Test auf die Schnittstellen */dev/lp1* bzw. */dev/lp2* auszuführen, erfolgt eine Fehlermeldung.

Linux verwaltet die parallelen Schnittstellen über das Subsystem (Modul) *parport*. Dabei ist *parport* das eigentliche Subsystem des Kernels, während *parport_pc* für die hardwareseitige Einbindung der Schnittstelle zuständig ist. Ob diese Module richtig geladen wurden, zeigt der Befehl

```
#lsmod
Module        Size  Used by
parport_probe 2916  0 (autoclean)
parport_pc    5504  2 (autoclean)
lp            5184  0 (autoclean)
parport       6476  2 (autoclean) [parport_probe parport_pc lp]
(...)
```

In der mit *lsmod* erstellten Liste der aktuell geladenen Module sollten zumindest Zeilen für *parport* und für *parport_pc* enthalten sein. Damit *parport* beim nächsten Systemstart mehr als nur eine parallele Schnittstelle erkennen kann, muss die Konfigurationsdatei */etc/conf.modules* manuell bearbeitet werden. In der Zeile *options parport_pc* sind die tatsächlichen I/O-Adressen aller parallelen Schnittstellen anzugeben:

```
(...)
alias parport_lowlevel    parport_pc
options parport_pc io=0x378,0x3BC irq=none,none
(...)
```

In dem oben gezeigten Auszug aus */etc/conf.modules* ist eine zusätzliche parallele Schnittstelle mit der Adresse *0x3BC* eingetragen worden (die Adressangaben werden jeweils durch ein Komma getrennt).

Beim nächsten Systemstart sollte *parport* alle parallelen Schnittstellen richtig erkennen und in */var/log/messages* die aktuelle Konfiguration ausgeben:

```
(...)
Nov  9 18:51:16 linux01 kernel: parport0: PC-style at 0x378 [SPP,PS2]
Nov  9 18:51:16 linux01 kernel: parport1: PC-style at 0x3bc [SPP,PS2]
Nov  9 18:51:16 linux01 kernel: parport0: Printer, Hewlett-Packard HP 6L
Nov  9 18:51:16 linux01 kernel: parport1: Printer, Hewlett-Packard HP 6L
```

```
Nov  9 18:51:16 linux01 kernel: lp0: using parport0 (polling).
Nov  9 18:51:16 linux01 kernel: lp1: using parport1 (polling).
(...)
```

Nach dem obigen Auszug aus */var/log/messages* verfügt der Linux-Server über zwei parallele Schnittstellen, an denen jeweils ein Drucker HP 6L angeschlossen ist. Die Schnittstellen können über die Gerätedateien */dev/lp0* und */dev/lp1* angesprochen werden.

Die aktuelle Anzahl der parallelen Schnittstellen kann auch mit

```
#ls /proc/parport
.   ..   0   1
```

abgefragt werden. *parport* hat auch in diesem Beispiel zwei parallele Schnittstellen erkannt. Die Unterverzeichnisse *0* und *1* enthalten ASCII-Dateien, die die dazu gehörenden Konfgurationsdetails zeigen.

Manuelle Konfiguration

Die aktuelle Druckerkonfiguration ist in der Datei */etc/printcap* gespeichert, wobei die installierten logischen Druckgeräte über die dort definierten Namen identifiziert werden. Jedes Druckgerät kann einen odere mehrere Namen erhalten.

Auch Linux-Systeme kennen den Begriff des *Standarddruckers*. Die Umgebungsvariable *$PRINTER* legt fest, welcher der in */etc/printcap* definierten Druckgeräte der aktuelle Standarddrucker ist:

```
# echo $PRINTER
lp
```

Ist diese Variable nicht gesetzt, so wird der erste Drucker in */etc/printcap* der Standarddrucker.

Im folgenden Abschnitt soll zunächst eine einfache Druckerkonfiguration Abbildung 3-26 erstellt werden, bei der ein Drucker vom Typ *Hewlett-Packard Laserjet 6L* an der Schnittstelle *LPT1* des Linux-Servers angeschlossen wird. Die hier vorgestellten Methoden und Befehle zum Drucken und zur Druckersteuerung beziehen sich zunächst alle auf diese Grundkonfiguration.

In der Konfigurationsdatei */etc/printcap* wird jede verfügbare Druckwarteschlange durch einen einzeiligen Eintrag definiert. Jedes Zeilenende (*newline*) schließt einen solchen Eintrag ab. Aus Gründen der besseren Übersichtlichkeit können trotzdem mehrzeilige Darstellungen erstellt werden, wenn – für lange Einträge – das Zeilenende durch einen unmitelbar vorausgehenden Backslash »\« aufgehoben wird.

Jeder Eintrag beginnt mit einem oder mehreren Namen für die Druckwarteschlange, die jeweils mit einem »|« getrennt sind. Es folgt eine Liste von Spezifikationen. Leere Zeilen und Zeilen, die mit einem »#« beginnen (Kommentare) werden ignoriert.

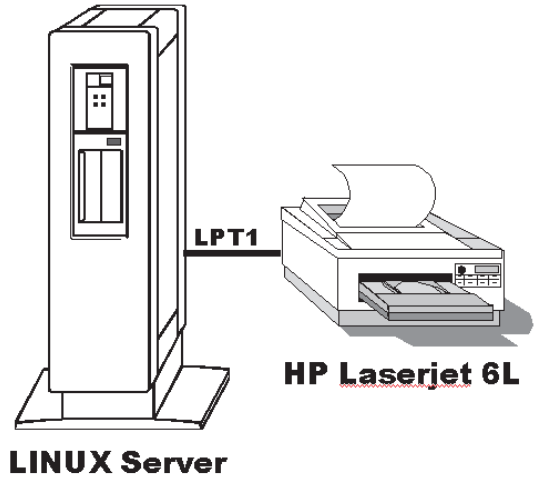


Abbildung 3-26 Lokaler Drucker am Linux-Server

In der vorinstallierten */etc/printcap* sind bereits eine Reihe von Druckwarteschlangen vorkonfiguriert, aber noch auskommentiert. Unter anderem ist dort auch der nachstehende Eintrag für einen HP-Drucker enthalten:

```
(...)
# HP Laser jet plus
#lp|hpj:\
#        :lp=/dev/lp1:\
#        :sd=/usr/spool/lp1:\
#        :mx#0:\
(...)
```

Die hier noch auskommentierte Druckwarteschlange kann unter den Namen *lp* und *hpj* angesprochen werden. Das Spoolverzeichnis ist */usr/spool/lp1*, die Schnittstelle ist */dev/lp1*. Die Größe der akzeptierten Druckjobs ist nicht begrenzt: *mx#0*.

Abschließend muss das Spoolverzeichnis manuell angelegt und mit ausreichenden Berechtigungen ausgestattet werden.

Druckerinstallation mit YaST

Das einfachste und sicherste Verfahren, den direkt am Linux-Server angeschlossenen Drukker einzurichten, ist die Installation mit *YaST*. Im Hauptmenü wird dazu *Administration des Systems*, *Hardware in das System integrieren* und schließlich *Drucker konfigurieren* ausgewählt. In der dann erscheinenden Form können dann unter anderem die Angaben zu Druckertyp und Schnittstelle eingetragen werden (Abbildung 3-27).

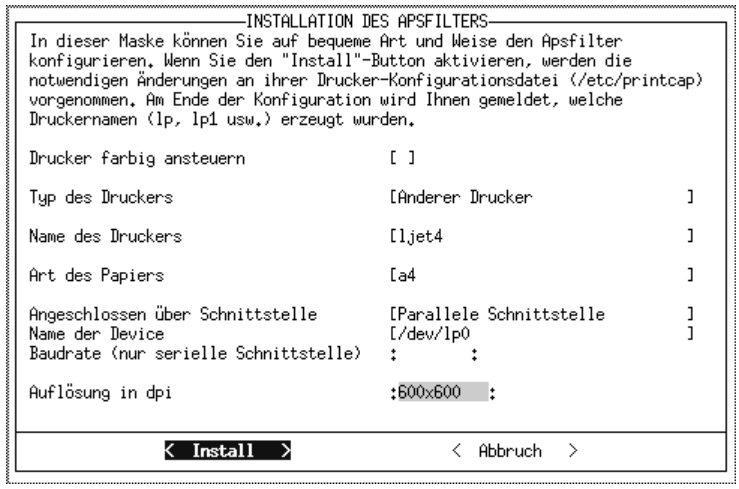


Abbildung 3-27 Druckerkonfiguration mit YaST

Die von *YaST* vorgenommene Konfiguration erfolgt über das später genauer vorgestellte Programm *apsfilter*, das Druckerfilter einrichtet, die die Daten eines Druckjobs in das drukkerspezifische Format überführen. Sie entsprechen damit in ihrer Funktion den *Druckertreibern* auf Systemen unter Windows 9x/NT. Für den in diesem Beispiel verwendeten Drucker *HP Laserjet 6L* ist der Druckerfilter mit dem Namen *ljet4* optimal geeignet, wenn die *Auflösung in dpi* manuell auf *600 x 600* Punkte eingestellt wird.

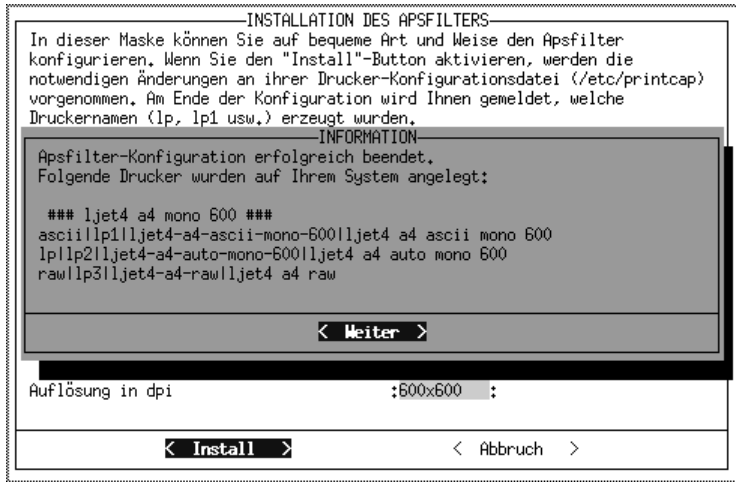


Abbildung 3-28 Einträge in /etc/printcap

Nach dem Abspeichern der Konfiguration in */etc/printcap* werden die Namen der angelegten Druckwarteschlangen zur Kontrolle angezeigt (Abbildung 3-28) und es erfolgt die Installation der Druckerumgebung:

- Anlegen der Spoolverzeichnisse in /var/spool/lpd/
- Anlegen der benötigten Druckerfilter unter /var/lib/apsfilter/bin/ (erstellt werden symbolische Links auf die Datei /var/lib/apsfilter/apsfilter)
- Erstellen der Konfigurationsdatei /etc/apsfilterrc (wenn noch nicht vorhanden) und der druckerspezifischen Konfigurationsdatei /etc/apsfilterrc.druckername (dabei ist druckername der Ghostscript-Druckername)

Druckerinstallation mit apsfilter

Das Programm *apsfilter*, das auch bei der eben beschriebenen Druckerkonfiguration von *YaST* verwendet wird, ist ein leistungsfähiges Werkzeug zur Installation und Konfiguration von Druckern unter Linux. *apsfilter* installiert bei Bedarf Druckerfilter, um Postscript-Dateien (nur: *Ghostscript*) auch auf nicht Postscript-fähigen Druckern ausgeben zu können. Postscript ist in der Unix-Welt das Standardformat für druckbare Daten.

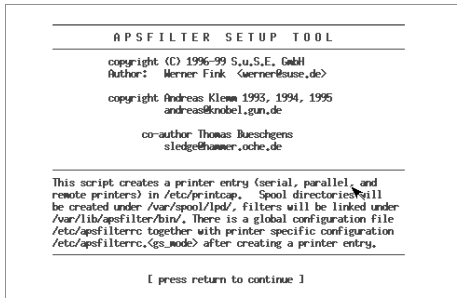
Tabelle 3-8 zeigt die vom Programm *apsfilter* für einen Drucker angebotenen Warteschlangentypen.

lp	Standard-Warteschlange für alle Dateiformate
lp-mono	entspricht lp, wird aber nur bei Farbdruckern angelegt und druckt schwarz-weiß.
asciii	dient dem Ausdrucken von Dateien im ASCII-Format, auch wenn das Spoolingsystem ein anderes Format vermutet (z. B. bei deutschen Umlauten).
raw	dient dem Ausdrucken von Daten, die bereits im druckerspezifischen Format vorliegen. Eine Konvertierung findet nicht statt.

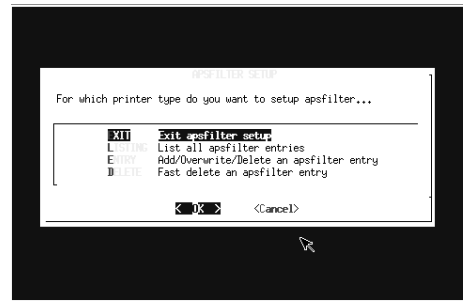
Tabelle 3-8 Warteschlangetypen

/var/lib/apsfilter/SETUP arbeitet menügesteuert, es ermöglicht das Auflisten und das Hinzufügen/Löschen von *apsfilter*-Druckerkonfigurationen. Beim Hinzufügen arbeitet *SETUP* nahezu gleich wie das Programm *YaST*, dazu kann *SETUP* auch Warteschlangen anlegen und löschen, die auf entfernte (*remote*) Hosts verweisen, sowie eine Vorfilterung für Netzwerkdrucker installieren.

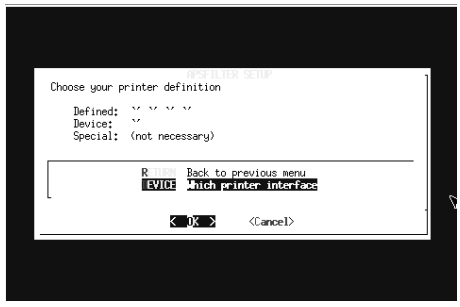
Die Bedienung von SETUP ist gewöhnungsbedürftig, daher soll nachfolgend (Abbildung 3-29) exemplarisch der Konfigurationsablauf für die Beispielkonfiguration dargestellt werden (die in Klammern angegebenen Zahlen bezeichnen die Arbeitsschritte).



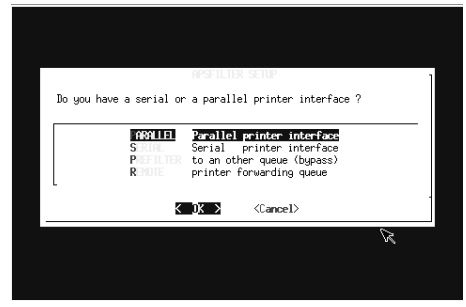
(1) Programmstart von *apsfilter SETUP*



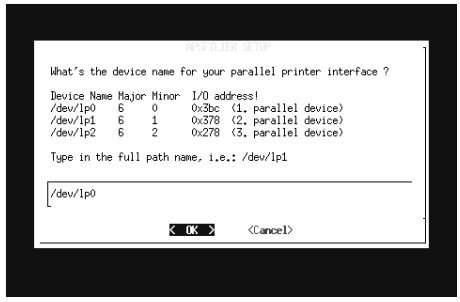
(2) Das Hauptmenü. Auswahl: *Add/Overwrite/ Delete an apsfilter entry*



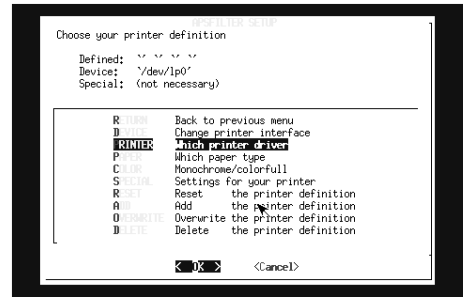
(3) Bisher kein Eintrag in */etc/printcap*. Auswahl: *Which Printer interface*



(4) Drucker ist an einem *Parallel printer interface* angeschlossen.

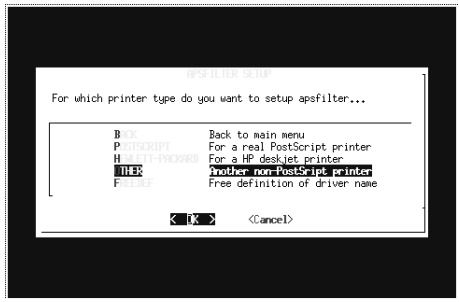


(5) Schnittstelle */dev/lp0 einstellen*

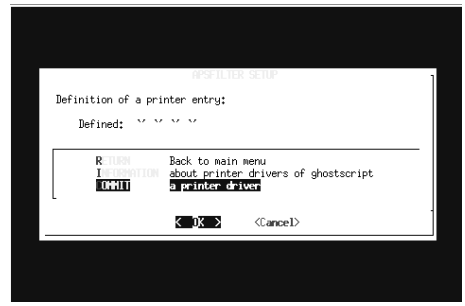


(6) Auswahl: *Which printer driver*

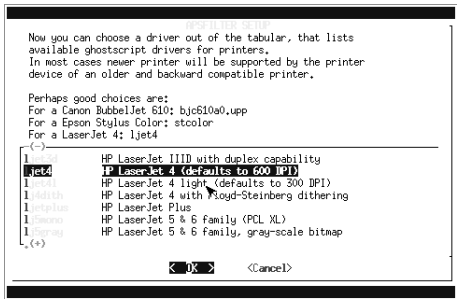
Abbildung 3-29 Druckerkonfiguration mit apsfilter



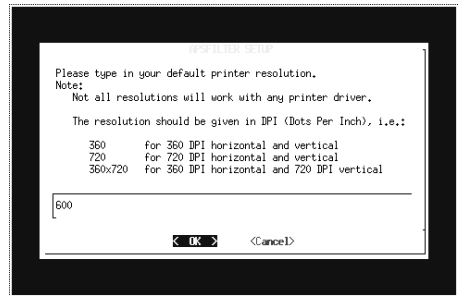
(7) Der Drucker gehört in die Gruppe: *Another non-Postscript printer*



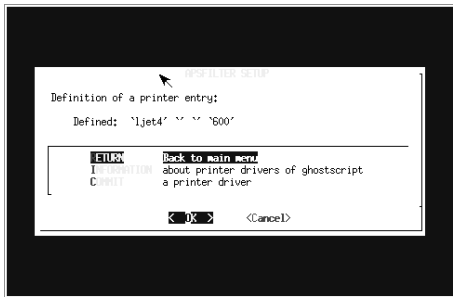
(8) Auswahl: *COMMIT a printer driver*



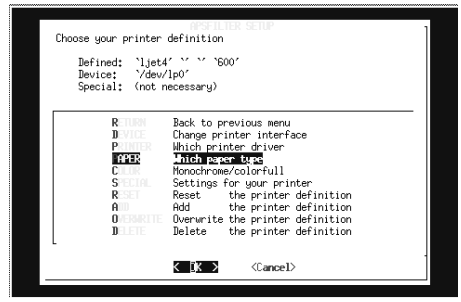
(9) Druckertyp: *ljet4*



(10) Auflösung manuell auf *600 dpi* ändern

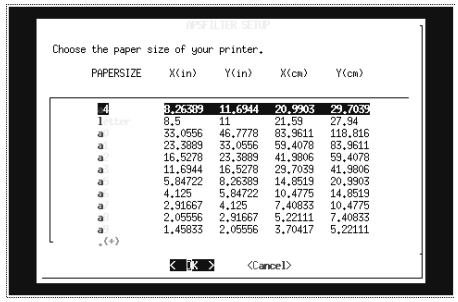


(11) Auswahl: *RETURN to main menu*

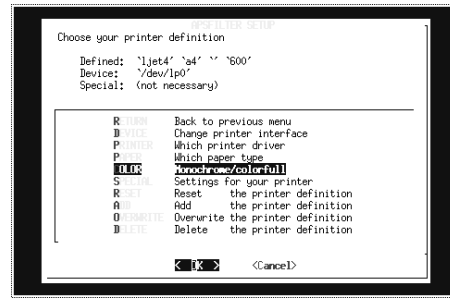


(12) Papiergröße einstellen: *Which Papersize*

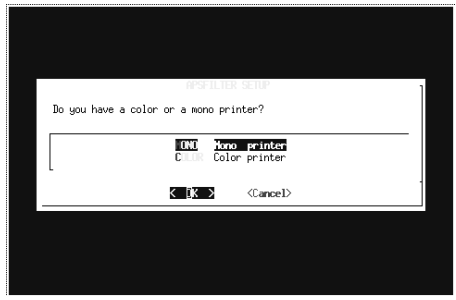
Abbildung 3-29 Druckerkonfiguration mit apsfilter



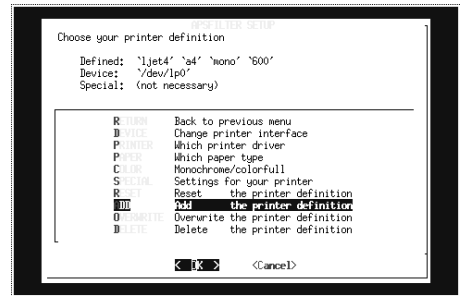
(13) A4-Papierformat: *a4*



(14) Auswahl: *COLOR Monchrome/colorfull*

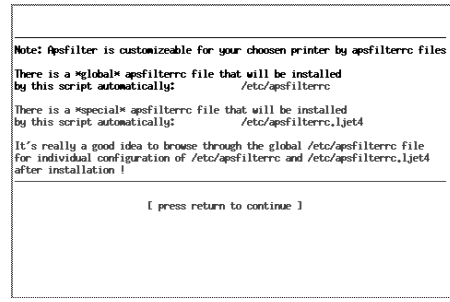


(15) Typ »Schwarzweißdrucker« einstellen: *MONO Mono printer*



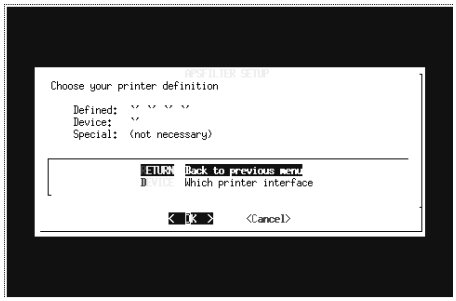
(16) Einstellungen speichern: *Add the printer definition*

(17) Rückmeldung: Namen der erstellten Druckwarteschlangen

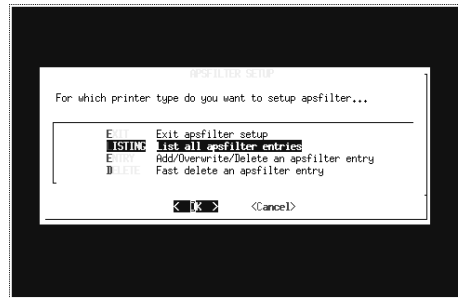


(18) Hinweis zur Druckerinstallation

Abbildung 3-29 Druckerkonfiguration mit apsfilter

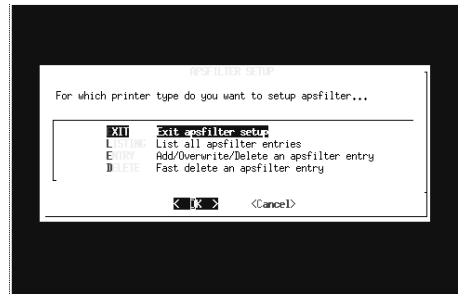


(19) Konfiguration abschließen.



(20) Abschließende Kontrolle

(21) Abfrage der aktuellen Druckerkonfiguration in */etc/printcap*



(22) *SETUP* beenden.

Abbildung 3-29 Druckerkonfiguration mit apsfilter

Der *apsfilter* legt beim Setup eine Reihe von Dateien wie *aps-laserjet-a4-auto-mono* im Verzeichnis */usr/lib/apsfilter/filter* an, die in */etc/printcap* unter *if=* verwendet werden. Die von *apsfilter* erstellten Dateien sind Druckerfilter, die z.B. Postscript-Dateien oder ASCII-Texte in ein für den Drucker verständliches Format umsetzen. Die Druckerfilter können und dürfen je Druckertyp nur einmal angelegt werden.

Sonderfall: Mehrere gleiche Druckgeräte konfigurieren

Mit den jetzt beschriebenen Verfahren lassen sich ein einzelner Drucker oder auch an mehreren Schnittstellen angeschlossene unterschiedliche Drucker auf einfachem Wege konfigurieren. Problematisch wird es, wenn mehrere Drucker gleichen Typs angesteuert werden sollen. Versucht der Systemverwalter über *YaST* oder über das *apsfilter SETUP* z.B. noch einmal einen Drucker einzurichten, der ebenfalls den schon eingerichteten Druckerfilter *ljet4* benötigt, wird eine Fehlermeldung ausgegeben. Die Konfiguration kann nicht durchgeführt werden.

Dasselbe Problem stellt sich, wenn in einem Netzwerk mehrere gleiche Drucker vorhanden sind, die über einen Druckserver erreichbar sein sollen.

Es ist dann notwendig, die Konfiguration für das zweite Druckgerät manuell durchzuführen.

In dem nachfolgenden Beispiel wird angenommen, dass zwei Drucker vom Typ *HP 6L* direkt am Linux-Server angeschlossen sind. Der erste Drucker ist wie vorstehend beschrieben konfiguriert worden; der zweite Drucker ist an */dev/lp1* angeschlossen und soll über den Druckernamen *hp6l_2* angesprochen werden. Der Ducker soll nur Postscript-Dateien ausgeben. Die Arbeitsschritte sind:

Schritt 1: Druckwarteschlange anlegen

Die neue Druckwarteschlange (*Queue*) mit dem Namen *hp6l_q2* soll im Verzeichnis */var/spool/lpd* erstellt werden. In diesem Verzeichnis liegen bereits die drei Queues *ljet4-a4- ...* für den ersten Drucker. Zweckmäßigerweise wird die Auto-Queue des ersten Druckers mit den darin enthaltenen Dateien kopiert und *hp6l_q2* umbenannt:

```
# ls -l /var/spool/lpd
total 7
drwxr-xr-x   2 lp       lp           1024 Nov  2 20:01 ljet4-a4-ascii-mono-600
drwxr-xr-x   2 lp       lp           1024 Nov  9 18:45 ljet4-a4-auto-mono-600
drwxr-xr-x   2 lp       lp           1024 Nov  5 19:54 ljet4-a4-raw
drwxr-xr-x   2 lp       lp           1024 Nov  9 19:34 hp6l_q2
-rw-r--r--   1 root     root            4 Nov  9 19:33 lpd.lock
```

Schritt 2: /etc/printcap bearbeiten

Der zweite Drucker soll nur Postscript-Dateien ausgeben können. Aus diesem Grund ist es ausreichend, nur einen neuen Eintrag in */etc/printcap* zu erstellen. Die Konfiguration für den Drucker *hp6l_2* ist, wie der nachstende Auszug aus der */etc/printcap* zeigt, der Auto-Queue des ersten Druckers ähnlich:

```
(...)
lp|lp2|ljet4-a4-auto-mono-600|ljet4 a4 auto mono 600:\
  :lp=/dev/lp0:\
  :sd=/var/spool/lpd/ljet4-a4-auto-mono-600:\
  :lf=/var/spool/lpd/ljet4-a4-auto-mono-600/log:\
  :af=/var/spool/lpd/ljet4-a4-auto-mono-600/acct:\
  :if=/var/lib/apsfilter/bin/ljet4-a4-auto-mono-600:\
  :la@:mx#0:\
  :tr=:cl:sh:sf:
#
raw|lp3|ljet4-a4-raw|ljet4 a4 raw:\
  :lp=/dev/lp0:\
  :sd=/var/spool/lpd/ljet4-a4-raw:\
  :lf=/var/spool/lpd/ljet4-a4-raw/log:\
  :af=/var/spool/lpd/ljet4-a4-raw/acct:\
  :if=/var/lib/apsfilter/bin/ljet4-a4-raw:\
  :la@:mx#0:\
  :tr=:cl:sh:sf:
```

```
#
# Manuell eingefuegte Autoqueue fuer zweiten HP 6L
#
hp6l_2:\
  :lp=/dev/lp1:\
  :sd=/var/spool/lpd/hp6l_q2:\
  :lf=/var/spool/lpd/hp6l_q2/log:\
  :af=/var/spool/lpd/hp6l_q2/acct:\
  :if=/var/lib/apsfilter/bin/ljet4-a4-auto-mono-600:\
  :la@:mx#0:\
  :tr=:cl:sh:sf:

### END   apsfilter: ### ljet4 a4 mono 600 ###
```

Als Druckerschnittstelle für den Eintrag *hp6l_2* ist */dev/lp1* angegeben, die Parameter für *sd,lf* und *af* verweisen auf die im ersten Schritt erstellte Druckwarteschlange und dort enthaltene Dateien. Der benötigte Druckerfilter ist bereits beim Einrichten des ersten Druckers installiert worden und wird im Eintrag *if* einfach nochmals verwendet.

Schritt 3: Drucker-Daemon neu starten

Mit dem Shell-Befehl

```
rclpd restart
```

wird der Drucker-Daemon abschließend neu gestartet, dazu wird die aktuelle Konfiguration aus */etc/printcap* verwendet.

Drucken und Druckersteuerung unter Linux

ASCII-Druckauftrag senden – lpr

Das Programm *lpr* ist die »Anwenderschnittstelle« zu den eingerichteten Druckwarteschlangen. Mit

```
lpr -Pqueue textdatei
```

wird der Inhalt von *textdatei* in die Druckwarteschlange *queue* gestellt. Entfällt die Angabe der Druckwarteschlange über die Option *-P*, so verwendet *lpr* die Voreinstellung der Umgebungsvariablen *$PRINTER* (Standarddrucker). *lpr* erzeugt für den Job eine Steuerdatei (*cf*-Datei), eine Kopie der Dateidatei und übergibt beide dem Druckmanager *lpd*, der den Druckauftrag im Spool-Verzeichnis der Druckwarteschlange ablegt.

Der Befehl *lpr* verfügt über eine Reihe von Optionen, z. B. um bestimmte Druckerfilter auszuwählen.

Druckauftrag löschen – lprm

Der Befehl *lprm* entfernt Druckaufträge aus der Druckwarteschlange, bevor diese gedruckt werden. Ein schon laufender Ausdruck wird abgebrochen. Ein Benutzer kann nur Aufträge löschen, die auch von ihm erstellt wurden. Wird versucht, einen Druckauftrag ohne ausreichende Berechtigung zu löschen, wird die Fehlermeldung *Permission denied* ausgegeben. Der Systemverwalter *root* kann mit

```
lprm -
```

alle Druckaufträge stoppen. Ein Benutzer kann mit Angabe der Druckwarteschlange (Option *-Pqueue*) alle eigenen dort gespeicherten Druckaufträge löschen oder durch Angabe der mit *lpq* ermittelten Jobnummer gezielt einen einzelnen Druclkauftrag entfernen.

Druckwarteschlangen anzeigen – lpq

Der Befehl *lpq* zeigt dem Benutzer gehörende Jobs in einer Druckwarteschlange. Mit

```
lpq -Pqueue
```

werden alle eigenen Druckjobs in der Warteschlange *queue* in Listenform ausgegeben:

```
# lpq -Plp2
lp2 is ready and printing
Rank    Owner     Job      Files              Total Size
active  samulat   51       text1              355201 bytes
1st     samulat   52       text2              12654 bytes
2nd     samulat   53       text4              248 bytes
```

Die in der Spalte *job* dargestellte Druckjobnummer identifiziert den Druckauftrag im Drucksystem. Insbesondere beim Löschen von Druckjobs oder auch zum Ändern der Druckreihenfolge wird diese Angabe benötigt (vergl. nachstehende Beschreibung zum Befehl *lpc*).

Druckwarteschlangen verwalten – lpc

Das Programm *lpc* kontrolliert das BSD-Drucksystem, es ermöglicht dem Systemverwalter *root* die hardwarenahe Kontrolle von Druckern und den dazu gehörenden Druckwarteschlangen. *lpc* kennt zwei Betriebsarten: Im Befehlsmodus wird genau ein Befehl an einen oder mehrere Drucker gesendet, im interaktiven Modus erfragt *lpc* nacheinander die Befehle, die auszuführen sind. Mit *lpc* werden

- Drucker im Drucksystem ein- und ausgeschaltet,
- Druckwarteschlangen ein- und ausgeschaltet,
- die Reihenfolgen von Druckaufträgen verändert und
- Statusinformationen abgerufen.

Mit dem Befehl

```
lpc status all
```

werden die Statusinformationen aller angeschlossenen Drucker ausgegeben:

```
lp:
      queuing is enabled
      printing is enabled
      2 entries in spool area
      lp is ready and printing

ascii:
      queuing is enabled
      printing is enabled
      no entries
      printer idle
(...)
```

Der Befehl *lpc* verfügt über eine Reihe von weiteren Optionen, Tabelle 3-9 zeigt eine Auswahl.

Befehl	Aktion
lpc abort lp2	bricht den Druckdaemon für den angegebenen Drucker *lp2* sofort ab und verhindert den Neustart durch *lpq*. Der Drucker wird gesperrt.
lpc clean all	beseitigt die Reste von alten Druckaufträgen. Auch alle Dateien, die zu nicht mehr vollständigen Druckaufträgen gehören, werden gelöscht.
lpc disable lp2	sperrt die Queue des angegebenen Druckers *lp2*.
lpc enable lp2	reaktiviert die Queue des angegbenen Druckers *lp2*.
lpc stop lp2	beendet zunächst einen eventuell noch aktiven Druckauftrag. Danach wird der angegebene Drucker *lp2* disabled.
lpc topq lp2 49	stellt für den Drucker *lp2* den Druckjob mit der Jobnummer *49* an die erste Position der Warteschlange.

Tabelle 3-9 Optionen für den Befehl lpc (Auswahl)

Druckwarteschlangen verwalten – klpq

Das Programm *klpq* ist die grafisch orientierte KDE-Variante der Befehle *lpq*, *lpc* und *lprm*. KDE-typisch ist die Möglichkeit, eine *URL* anstelle des Dateinamens anzugeben sowie Drag & Drop-Funktionen. Wird mit der Maus eine Datei auf das *klpq*-Fenster gelegt, so wird diese in die Queue aufgenommen.

klpq ist universell anwendbar, bei Bedarf werden neben dem *BSD*-Drucksystem auch *PTR* und *LPRng* unterstützt (Menü *Config*).

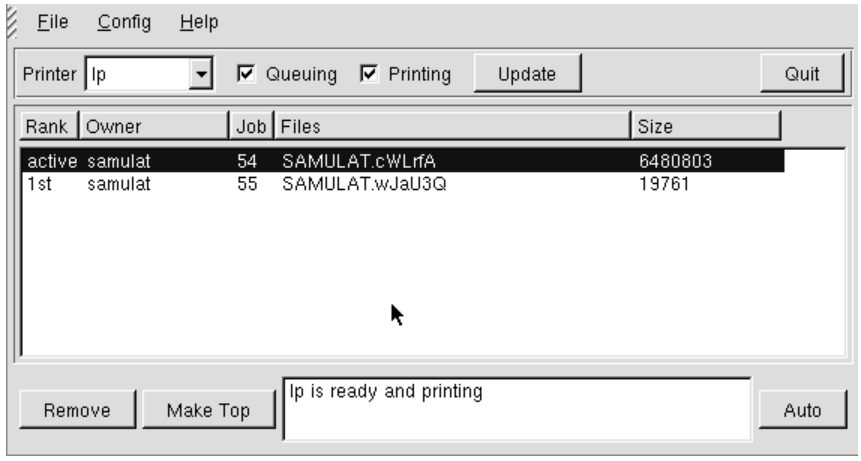


Abbildung 3-30 Druckwarteschlangenverwaltung mit kplq

Die Funktion der Schaltelemente sollte, ausgehend von der Kenntnis der Programme *lpq*, *lpc* und *lprm*, selbsterklärend sein. Wichtig ist, dass die Anzeige der aktuellen Druckaufträge standardmäßig nicht dynamisch aktualisiert wird, das gewünschte Zeitintervall muss zunächst nach Klick auf die Taste *Auto* eingestellt werden. Ein Klick auf die Taste *Update* löst die sofortige Aktualisierung der dargestellten Informationen aus.

Postscript Druckauftrag senden – a2ps

Mit dem Programm *a2ps* können ASCII-Dateien konvertiert und auf einem Postscript-Drukker ausgegeben werden. Mit

```
a2ps /etc/printcap -Plp2
```

wird der Inhalt der Datei */etc/printcap* auf dem Postscript-Drucker *lp2* ausgegeben (Abbildung 3-31). Mit der Option *-P* wird das Druckgerät angegeben. Entfällt diese Option, wird der Standarddrucker verwendet.

In der Standardkonfiguration gibt *a2ps* jeweils 2 Druckseiten je A4-Seite aus. Der Ausdruck erhält Rahmen. Zusätzlich werden der Name der ausgedruckten Datei, Druckdatum und Seitenzahlen ausgegeben. Diese und viele andere Parameter sind in der Konfigurationsdatei */etc/apsfilterrc* enthalten, die bei Bedarf angepasst werden kann.

Mit *a2ps* erstellte Ausdrucke sind sehr gut zur schnellen papiergebundenen Dokumentation eines Linux-Servers geeignet.

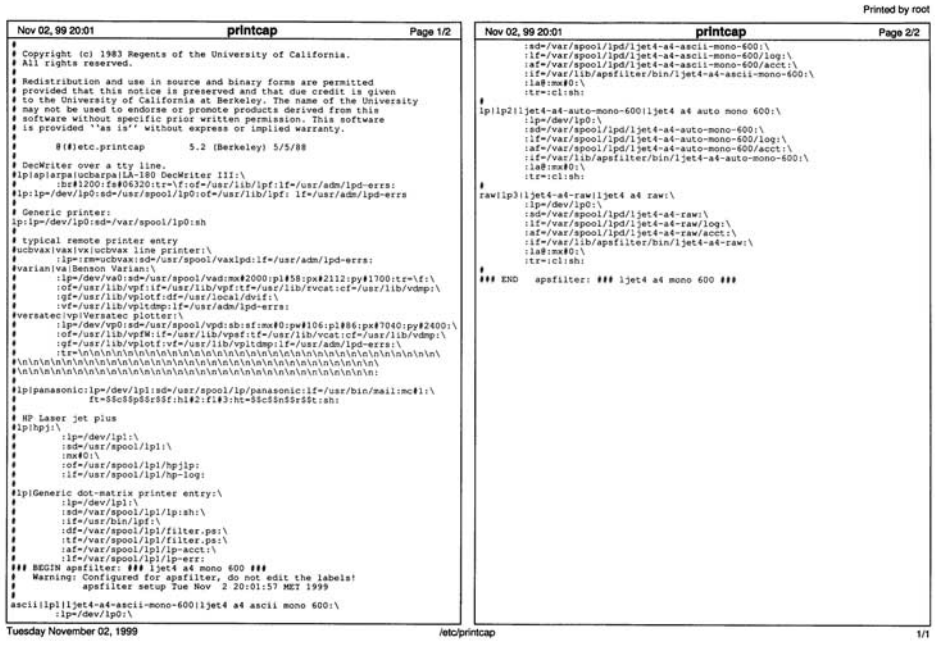


Abbildung 3-31 Druckausgabe mit dem Programm a2ps

Druckserver unter Linux

Der *lpr* (Line-Printer, definiert in RFC 1179) gehört zu den Netzwerkprotokollen und Hilfsprogrammen des TCP/IP-Protokoll-Paketes. *lpr* ist ein Standard für die Übermittlung von Druckaufträgen, mit dem ein Druck-Client seine Druckaufträge in Druckwarteschlangen auf einem Server, auf dem der Spooler Dienst *lpd* läuft, umleitet.

Damit ein *lpr*-Client Druckaufträge versenden kann, benötigt er die Netzwerkadresse des *lpd*-Servers und den Namen, über den der *lpd*-Dienst das Druckgerät identifiziert. Zusammen mit dem Druckauftrag sendet der Druck-Client auch Anweisungen zur Verarbeitung des Druckauftrags.

Sollen andere Rechner als Druck-Clients über das Netzwerk auf einen Druckserver unter Linux zugreifen, so ist es unbedingt erforderlich, auf dem Druckserver zunächst die dafür notwendigen Berechtigungen zu erteilen. Welche Rechner auf den Duckserver zugreifen dürfen, wird in der Konfigurationsdatei */etc/hosts.lpd* festgelegt:

```
# hosts.lpd This file describes the names of the hosts which
# are to be considered "equivalent", i.e. which are to be
# trusted enough for allowing remote lpr(1) commands.
#
```

```
# hostname

192.168.100.10
```

Im obigen Beispiel darf nur der Rechner mit der IP-Adresse 192.168.100.10 auf diesen Druckserver zugreifen.

Netzwerkdrucker vom Linux-Client nutzen

Im nächsten Konfigurationsbeispiel soll jetzt ein Linux-Client über das Netzwerk auf den Drucker am Linux-Server mit dem Hostnamen *linux01.samulat.de* zugreifen können (siehe Abbildung 3-32).

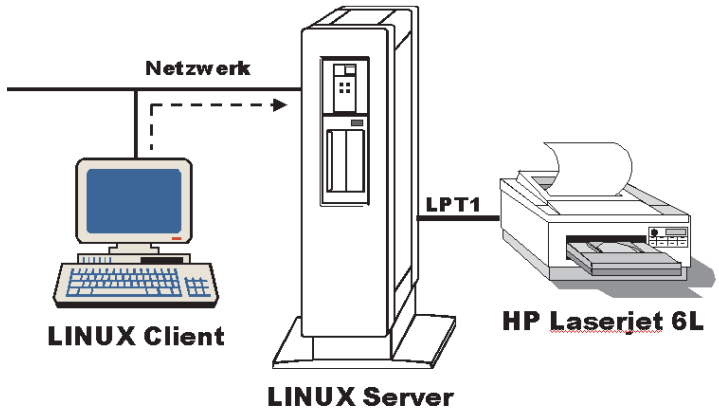


Abbildung 3-32 Lokaler Drucker am Linux-Server

Am Linux-Server sind dazu keine weiteren Arbeiten mehr erforderlich. Der Netzwerkdrucker muss lediglich manuell als zusätzlicher Drucker in */etc/printcap* des Druck-Clients eingetragen werden. An Stelle eines lokalen Druckers ist hier der Verweis auf die Druckwarteschlange des Servers einzutragen.

```
lp:\
:rm=linux01.samulat.de
:rp=lp2:\
:sd=/var/spool/linux01.samulat.de:
:la:mx#0
:sh
```

Der ohne weitere Angaben ausgeführte Eintrag *lp* ist in jedem Fall notwendig, weiter sind *rp* (*remote printer*), *rm* (*remote machine*) und die Druckwarteschlange *sd* mit den entsprechenden Werten einzutragen.

Netzwerkdrucker vom Windows NT Client nutzen

Windows NT bietet sowohl *lpr*- als auch *lpd*-Dienste für das Drucken unter TCP/IP. Um diese Dienste nutzen zu können, ist zunächst der *Mircrosoft TCP/IP-Druckdienst* zu installieren (Abbildung 3-33).

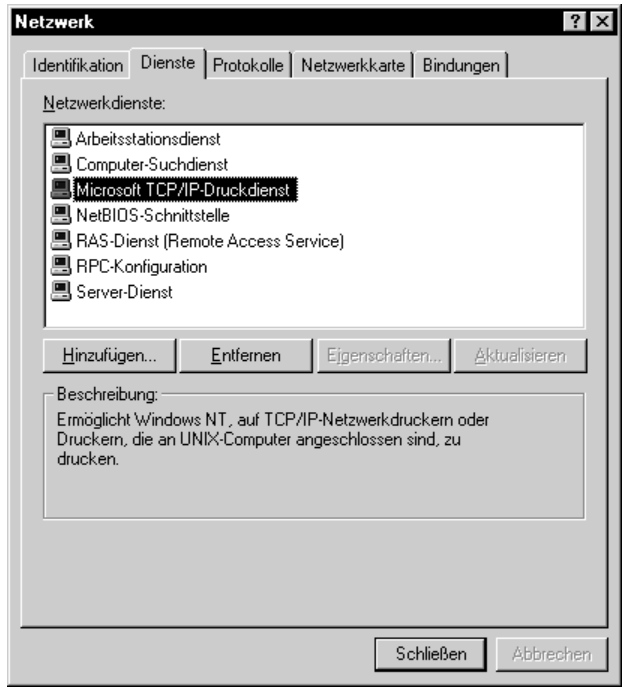


Abbildung 3-33 Einrichten des Microsoft TCP/IP-Druckdienste unter Windows NT

Um einen Drucker des Linux-Druckservers nutzen zu können, ist die Einrichtung eines logischen Druckers unter Windows NT erforderlich.

Die Einrichtung muss als NT-Systemverwalter durchgeführt werden, am einfachsten geschieht dies mit dem Assistenten *Neuer Drucker -> Neuer lokaler Drucker* (Abbildung 3-34).

Um den unter Linux installierten Netzwerkdrucker erreichen zu können, muss ein neuer Anschluss definiert werden. Nach Klick auf die Taste *Hinzufügen* kann zunächst der Typ des Printservers ausgewählt werden, in diesem Fall *LPR Port* (Abbildung 3-35).

Nach Klick auf die Taste *Neuer Anschluss* wird der Name oder die IP-Adresse des Druckservers und der Name des Druckgerätes abgefragt (Abbildung 3-36).

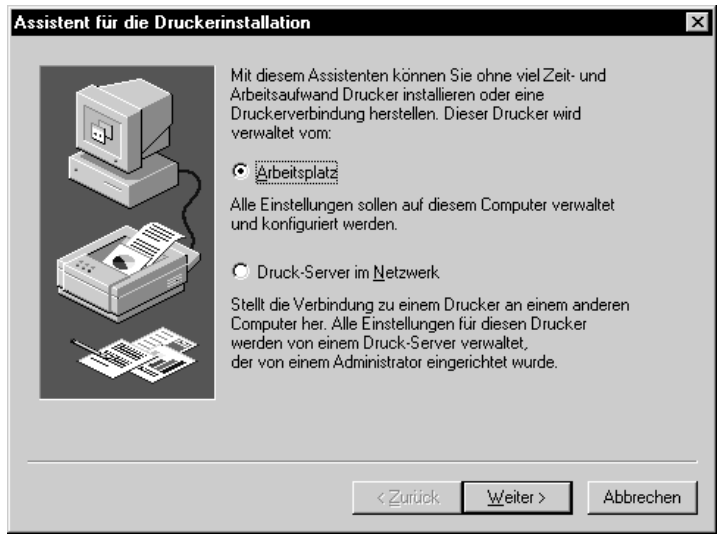


Abbildung 3-34 Neuer lokaler Drucker

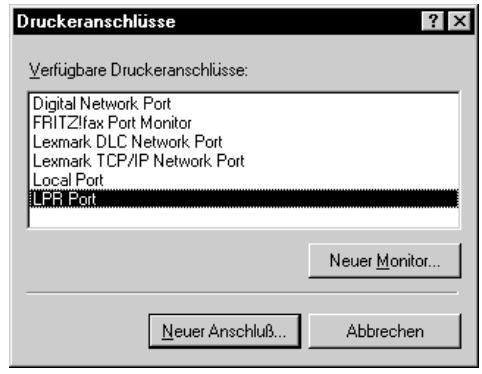


Abbildung 3-35 Auswahl des Druckeranschlusses »LPR Port«

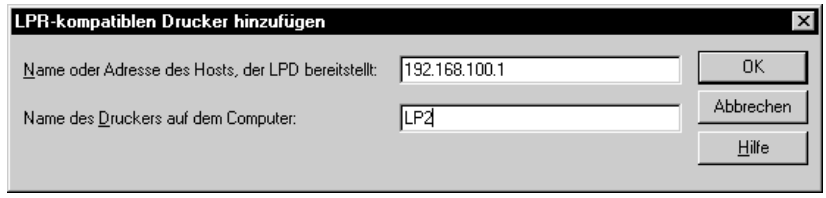


Abbildung 3-36 Auswahl der »Queue«

Nachdem der neue Anschluss ohne Fehler verbunden werden konnte, steht dieser jetzt als »verfügbarer Anschluss« in der Anschlussliste dieses PC-Arbeitsplatzes (Abbildung 3-37).

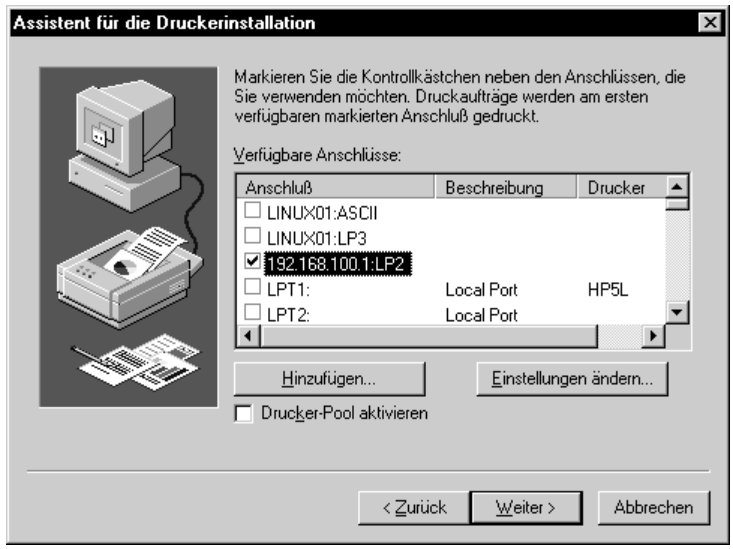


Abbildung 3-37 Neuer Anschluss

Der weitere Installationsgang (Auswahl des Druckertreibers, optionale Treiber, Freigabe, ...) entspricht dann wieder dem Windows-Standardablauf. Darauf soll an dieser Stelle nicht weiter eingegangen werden.

3.2.2 Einbinden von Netzwerkdruckern

Interne oder externe Printserver-Boxen stellen eine inzwischen sehr preiswerte Methode dar, die Ressource-Drucker im Netzwerk zur Verfügung zu stellen. Auf dem Markt sind eine Vielzahl von Geräten, die dann eine bis drei parallele Schnittstellen anbieten. Printserver-Boxen verfügen in der Regel nicht über sehr große Arbeitsspeicher, da heute aber bereits die Drucker häufig über Speicher in der Größenordnung von mehreren Megabytes verfügen, ist dies in der Regel zu vernachlässigen.

Fast alle Printserver-Boxen können auch ohne spezielle Installationssoftware direkt für den Druckbetrieb in einem TCP/IP-Netzwerk konfiguriert werden, was diese gerade auch für den Einsatz in Linux-Systemen sehr interessant macht. Unterstützt der Printserver TCP/IP, wie z. B. fast alle Geräte von AXIS, Intel oder Hewlett-Packard, so ist dafür nur die Zuweisung einer IP-Adresse notwendig, dies kann entweder manuell über *arp* oder über DHCP erfolgen. Bekannt sein muss dafür die zwölfstellige MAC-Adresse des zum Printserver gehörende Netzwerkinterfaces (diese Nummer ist in der Regel direkt am Gehäuse ablesbar).

Ist kein DHCP-Server verfügbar, kann die Adresszuweisung manuell mit

```
arp -s 192.168.8.42 00:40:8c:30:63:0e
```

vorbereitet werden. In dem Beispiel soll dem Printserver mit der MAC-Adresse *00:40:8c:30:63:0e* manuell die IP-Adresse *192.168.8.42* zugewiesen werden. Der Printserver erzeugt unmittelbar nach dem Einschalten des Gerätes einen *arp-Request* und übernimmt dann die voreingestellte Adresse. Der abschließende Test erfolgt mit

```
ping 192.168.8.42
```

Der vom Printserver erzeugte *arp-Request* kann nicht geroutet werden, bei Schwierigkeiten mit dieser Art von Adresszuweisung sollte zunächst geprüft werden, ob nicht ein Router zwischen Rechner und Printserver liegt.

Empfehlenswerter ist aber die statischen Adresszuweisung über einen DHCP-Server. Die Konfigurationsdatei */etc/dhcpd.conf* erhält dann den Eintrag

```
host axis01 {
    hardware ethernet 00:40:8c:30:63:0e;
    fixed-address 192.168.8.42;
  }
```

mit dem die vorher manuell erstellte Adresszuweisung nun über den DHCP-Server ausgeführt wird.

Zur Namensauflösung sollte auch jeder Printserver einen Eintrag in der Datei */etc/hosts* erhalten. Die weitere Konfiguration der Druckumgebung und die Einbindung der Printserver-Box erfolgt dann wie schon beschrieben in */etc/printcap*.

3.3 Datei- und Verzeichnisdienste

Der Linux-Server kann auf einfachem Weg in ganz unterschiedlich strukturierten Netzwerken als File- und Druckserver eingesetzt werden. Dabei kann der Linux-Server seine hohe Leistungsfähigkeit, Stabilität und hohe Verfügbarkeit im direkten Vergleich mit anderen Serverkomponenten unter Novell oder Windows NT beweisen.

Die Integration des Linux-Servers in heterogene Netzwerkumgebungen ist oftmals allein schon durch die Verwendung von TCP/IP und den darauf basierenden Diensten und Programmen möglich, insbesondere dann, wenn Intranet- oder Internet-Umgebungen realisiert werden sollen.

Noch mehr Einsatzmöglichkeiten zeigen sich, wenn man TCP/IP-Dienstleistungen und die zur Verfügung stehenden Emulationen der anderen PC-Netzwerkbetriebssysteme berücksichtigt (siehe Tabelle 3-10).

TCP/IP	In reinen Unix- oder TCP/IP-Umgebungen kommt das Network File System (NFS) zum Einsatz. Entsprechende Server- und Clientsoftware sind in der Distribution enthalten.
Microsoft Windows	Emulation eines Windows-File-Servers über das Softwarepaket samba
Novell Netware	Emulation eines Netware 2.x-/3.x-Fileservers über das Softwarepaket marsnwe. Bereitstellung von File- und Druckdiensten. Die Systemverwaltung erfolgt über Netware-Standardprogramme wie syscon oder pconsole.
Apple MAC	Die Apple Talk Protocol Suite for Unix (netatalk) ermöglicht den Zugriff auf Dateien und Drucker.

Tabelle 3-10 Emulationen von Netzwerkbetriebssystemen unter Linux

Auf die Emulation eines Windows-Datei- und -Druckservers *samba* wird später noch detailliert eingegangen, interessant ist aber auch die Möglichkeit, einen Novell-Netware-3.x- oder -4.x-Server durch eine Emulation so gut nachbilden zu können, dass schon vorhandene Client-Installationen unverändert weiter betrieben werden können.

Mit der Emulation *marsnwe* – sie gehört zum Standardumfang fast aller Distributionen – kann ein veralteter Novell-3.x-Server mit geringem Aufwand, für die Clients unbemerkt, im Netzwerk ersetzt werden. Damit ist oft ein kostengünstiger Weg möglich, überalterte Hardware aus dem Netzwerk zu nehmen, die Funktionen des Servers aber möglichst unverändert beizubehalten.

3.3.1 Quoting

Es wurde bisher schon mehrfach deutlich, dass die Begrenzung des für einen Benutzer oder für Gruppen zur Verfügung stehenden Speicherplatzes auf den Datenträgern für die Gesamtfunktion jedes Servers sehr wichtig ist. Novell bietet seit vielen Jahren dieses Quoting an. Bereits ind der Netware Version 3.x waren dazu leistungsfähige Mechanismen vorhanden, die in den neueren Versionen weiter verbessert wurden. Windows NT 4.0 bietet erstaunlicherweise keine Quoting-Funktion, erst Windows 2000 soll wieder die Limitierung der Speicherkapazitäten ermöglichen.

Bereits bei der Planung der Festplattenpartitionierung war die Idee der Begrenzung der zur Verfügung stehenden Speicherkapazitäten wichtige Vorgabe. Dieses ist zwar schon ein wichtiger Schritt; in der Regel ist aber immer noch ein Quoting-Mechanismus notwendig, der in einer oder mehreren Partitionen verfügbaren Speicherplatz bezogen auf einzelne Benutzer oder auf Gruppen begrenzt. Idealerweise sollte so ein Begrenzungsmechanismus sogar auf einzelne Verzeichnisse anwendbar sein (wie es z. B. Novell ab der Version 4 ermöglicht).

Installation und Grundkonfiguration

Um ein Quoting unter Linux einrichten zu können, muss zunächst eine entsprechende Option im Kernel vorhanden sein. Die Kernel der SuSE-Distribution enthalten diese Option standardmäßig, sodass kein neuer Kernel kompiliert werden muss.

Zunächst muss das Paket *quota* aus der Serie *ap* installiert werden. In der Konfigurationsdatei */etc/rc.config* ist der Eintrag

START_QUOTA="YES«

vorzunehmen, um die Quoting-Dienste beim nächsten Booten automatisch zu starten.

Um das Quoting durchführen zu können, muss vor dem nächsten Systemstart in */etc/fstab* angegeben werden, welche Partitionen grundsätzlich Begrenzungen erlauben sollen.

Das Quoting kann für Benutzer oder Gruppen erfolgen. Der Eintrag in */etc/fstab* legt fest, welche dieser beiden Möglichkeiten genutzt werden oder ob beide aktiviert werden sollen. Da Linux nur Quotierungen auf der Ebene der Partition kennt, nicht bezogen auf einzelne Verzeichnisse, muss dies bereits bei der Planung der Partitionen berücksichtigt werden.

Die nachfolgenden Beispiele mit Auszügen aus einer fiktiven */etc/fstab* sollen die Möglichkeiten zur Definition der Quotierung verdeutlichen (Tabelle 3-11).

(1) Benutzerbezogenes Quoting für die Partition */dev/hda2*					
/dev/hda1	/	ext2	Defaults	1	1
/dev/hda2	/	ext2	defaults,usrquota	1	1
(2) Gruppenbezogenes Quoting für die Partition */dev/hda2*					
/dev/hda1	/	ext2	Defaults	1	1
/dev/hda2	/	ext2	defaults,grpquota	1	1
(3) Benutzer- und gruppenbezogenes Quoting für die Partition */dev/hda2*					
/dev/hda1	/	ext2	Defaults	1	1
/dev/hda2	/	ext2	defaults,usrquota,grpquota	1	1

Tabelle 3-11 Quoting unter Linux

Die für die Quotierung zusätzlich benötigten Informationen werden in den Dateien *quota.user* und *quota.group* gespeichert, die im jeweiligen Startverzeichnis der Partition manuell angelegt werden. Der Besitzer dieser Datei oder Dateien sollte *root* sein. Mit

```
# touch /partition/quota.user
# touch /partition/quota.group
# chmod 600 /partition quota.*
```

werden mit dem Befehl *touch* zwei Dateien der Länge Null mit den vorgegebenen Namen erzeugt. Die Berechtigung wird über *chmod* nur für *root* erteilt, kein anderer Benutzer darf auf diesen Dateien irgendwelche Rechte haben.

Nach dem Abschluss dieser Konfigurationsarbeiten sollte das Linux-System neu gestartet werden. Während der Bootsequenz werden die Meldungen

```
Turning on quota:
/dev/sda3 user quotasturned on
/dev/sda3 group quotas turned on

Starting rquota daemon done
```

ausgegeben. Hier wurde die benutzer- und gruppenbezogene Quotierung für die Partition */dev/sda3/* wurde erfolgreich eingerichtet.

Quotierung für einen Benutzer

Zur Konfiguration und Verwaltung der Quotierung steht eine Reihe von Befehlen zur Verfügung, die kurz vorgestellt werden sollen.

Die Konfiguration soll durchgeführt werden für den bereits im Linux-System angelegten Benutzer *willi*. Begrenzt werden soll der ihm zur Verfügung stehende Speicherplatz auf */dev/sda3* und die Anzahl der Dateien, die dieser Benutzer dort speichern darf. Mit

```
# edquota -u willi
```

wird der in der Umgebungsvariblen *$EDITOR* angegebene Texteditor gestartet, in diesem Beispiel *vi*, um die aktuelle Einstellung für diesen Benutzer anzuzeigen und bearbeiten zu können (Abbildung 3-38).

Die Angabe *blocks in use* zeigt die Anzahl der Blöcke (in Kilobytes), die dieser Benutzer bereits in */dev/sda3* belegt hat, *limits* enthält die für ihn aktuell gültige Begrenzung. Die Angabe für *inodes* zeigt, dass bereits 99 Dateieinträge für diesen Benutzer vorhanden sind; auch in dieser Zeile enthält *limits* die aktuellen Grenzwerte. Ein Eintrag dieser Form erfolgt für jede eingerichtete Quotierung.

Deutlich wird, dass für die Quotierung immer zwei Grenzwerte angegeben werden: *soft* und *hard*. Der Grenzwert für *soft* liegt immer deutlich unter dem vorgegebenen Wert für *hard*.

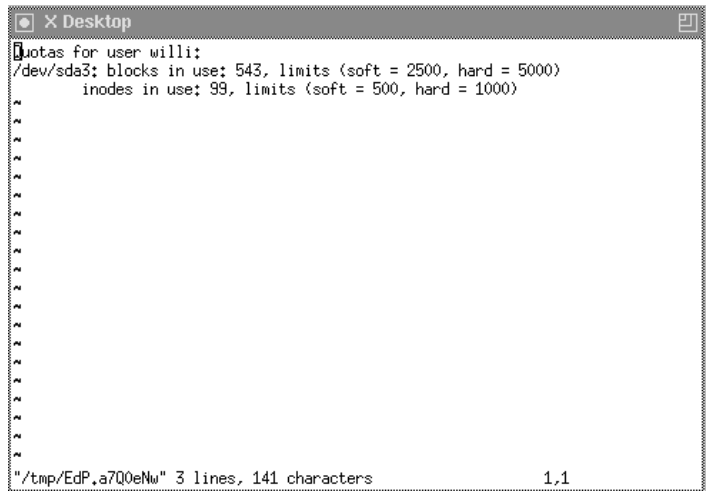


Abbildung 3-38 Bearbeitung der Quotierung für den Benutzer willi

Die Linux-Quotierung bietet die Möglichkeit, nach dem Erreichen des mit *soft* festgelegten Limits zunächst nur eine Warnmeldung auszugeben. Die echte Begrenzung mit dem Unterbinden weitere Speicherversuche erfolgt dann, wenn der mit *hard* definierte Grenzwert erreicht wurde. Dieser Mechanismus ist allerdings nur aktiv, wenn tatsächlich eine zeitlich begrenzte *Grace Period* vom Systemverwalter zugestanden wurde. Ist die nicht der Fall, wird bereits beim Erreichen des mit *soft* definierten Limits jeder weitere Speicherversuch unterbunden. Der Wert für *hard* spielt in diesem Falle keine Rolle mehr. Mit

```
# edquota -t
```

wird wiederum der Standard-Editor gestartet, um die aktuelle Vorgabe anzuzeigen und bei Bedarf ändern zu können (Abbildung 3-39).

Die Zeitangabe für *Grace Period* kann in *sec* (Sekunden), *min* (Minuten), *hour* (Stunden), *day* (Tagen), *week* (Wochen) oder *month* (Monaten) erfolgen und wird durch einfache Änderung des jetzt angzeigten Eintrags festgelegt.

Quotierung für eine Gruppe

Die Einrichtung und Verwaltung einer Quotierung für eine Gruppe erfolgt mit

```
# edquota -g fibu
```

wobei in diesem Beispiel *fibu* dem Namen einer auf dem Linux-Server eingerichteten Gruppe entspricht. Die weitere Konfiguration der Quotierung entspricht exakt dem bereits für einen einzelnen Benutzer dargestellten Ablauf. Auch hier werden die aktuellen Werte im Standard-Editor angezeigt und bearbeitet. Die Voreinstellung für die *Grace Period* gilt auch für die gruppenbezogenen Quotierungen.

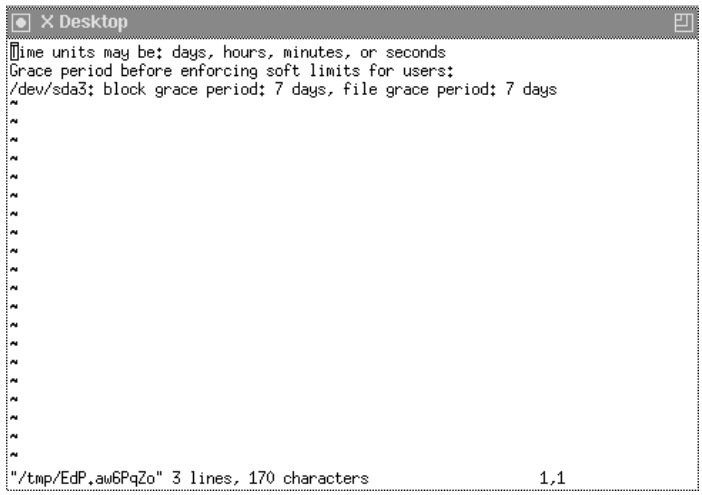


Abbildung 3-39 Bearbeitung der »Grace Period«

Weitere Funktionen zur Verwaltung der Quotierung

quotacheck

Der Befehl *quotacheck* prüft die aktuelle Belegung des gesamten Dateisystems auf dem Linux-Server und aktualisiert die Dateien *quota.user* und *quota.group*. Damit wird in jedem Fall die Quotierung auf die aktuellen Werte gesetzt, *quotacheck* sollte z. B. einmal wöchentlich über *cron* automatisch gestartet werden.

repquota

Der Befehl *repquota* erzeugt eine Gesamtübersicht der Quotierungen für das gesamte Filesystem eines Linux-Servers. Die auf dem Bildschirm ausgegebene Liste umfasst alle auf diesem Server eingerichteten Benutzer:

```
# repquota -a
                    Block limits                File limits
User      used    soft    hard  grace    used  soft  hard  grace
root   1064224       0       0          64519     0     0
bin      13829       0       0           3687     0     0
daemon       1       0       0              1     0     0
tty        380       0       0           1198     0     0
(...)
willi      543    2500    5000             99   500  1000
petra     1482       0       0            119     0     0
(...)
```

quotaon und quotaoff

Mit *quotaon* wird die Quotierung eingeschaltet, *quotaoff* beendet diesen Dienst. Normalerweise werden diese Befehle automatisch beim Starten und Beenden des Systems ausgeführt.

Verwaltung der Quotierung mit webmin

Eine sehr einfache und effektive Möglichkeit, Quotierungen auf einem und auch mehreren Linux-Servern zu verwalten, bietet das schon vorgestellte Programm *webmin*. Voraussetzung ist lediglich, dass die Installation und Grundkonfiguration der Quotierung, wie oben beschrieben, bereits durchgeführt wurde. Die Dateien *quota.user* und *quota.group* brauchen in diesem Fall nicht manuell angelegt zu werden, *webmin* übernimmt diese Aufgabe beim ersten Start der Quotierung.

Die Verwaltung der Quotierung erfolgt über das *webmin*-Modul *Disk Quotas* (Abbildung 3-40).

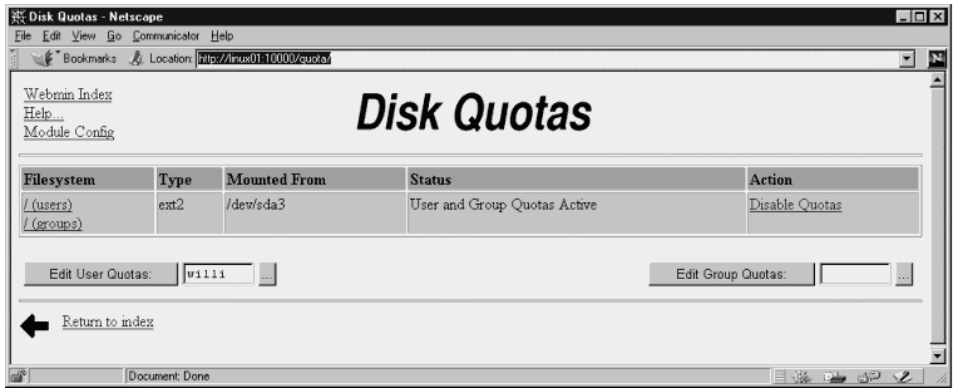


Abbildung 3-40 Anzeige der aktuellen Quotierungen mit webmin

Auch hier kann die Quotierung für einen einzelnen Benutzer oder für eine Gruppe konfiguriert werden, in diesem Beispiel werden die aktuellen Einstellungen für den benutzer *willi* abgerufen (Abbildung 3-41).

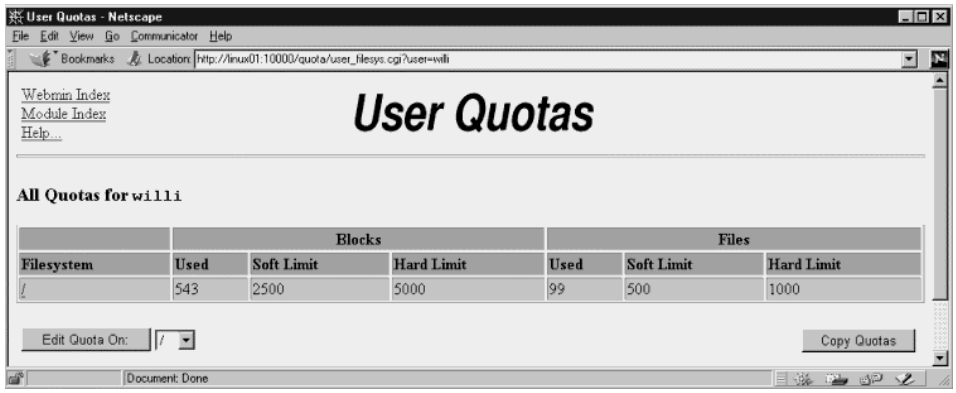


Abbildung 3-41 Aktuelle Quotierungen für den Benutzer willi

Die oben gezeigte Form enthält alle für diesen Benutzer eingestellten Quotierungen. Bei Bedarf können die aktuell eingestellten Vorgaben geändert werden (Abbildung 3-42).

Abbildung 3-42 Quotierung ändern

Wenn notwendig, können aus der Startform (Abbildung 3-39) durch Klick auf die Einträge *users* bzw. *groups* auch Gesamtübersichten abgerufen werden. In diesen Listendarstellungen steht eine Schaltfläche zur Verfügung, über die die Vorgaben für die *Grace Period* geändert werden können:

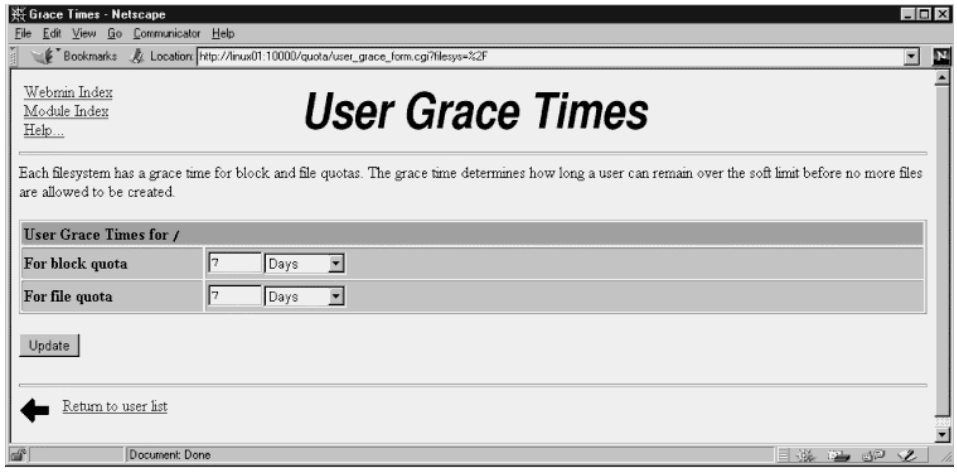


Abbildung 3-43 Konfiguration der Grace Period

Damit können alle zur Verwaltung der Quotierung notwendigen Einstellungen über das Programm *webmin* komfortabel ausgeführt werden.

3.3.2 CD-ROM- / DVD-Server

Die netzwerkweite Bereitsstellung von CD-ROMs und DVDs erfolgt in der Regel nicht über Novell- oder Windows-NT-Server, sondern über speziell zu diesem Zweck konfigurierte Geräte, die dann diese Ressourcen im Netzwerk bereitstellen.

Diese spezialisierten CD-ROM-Server sind teuer, diese Aufgaben können aber leider nicht optimal von Servern unter Novell oder Windows NT gelöst werden. Die Begründung liegt zum einen in den hohen Speicheranforderungen, die durch das »Cachen« dieser Datenbestände entstehen. Viel wichtiger ist aber die in diesen Systemen fehlende Möglichkeit, eine Vielzahl von Laufwerken so zu mounten, dass diese im Netzwerk unter einer Ressource (oder im Windows-Sprachgebrauch: Freigabe) geführt werden können. Ansonsten müßte der Windows-Client für jedes CD-ROM-Laufwerk einen Laufwerkbuchstaben zuordnen oder derselbe Buchstabe immer wieder gewechselt werden. Beide Möglichkeiten sind, wenn viele CD-ROMs benötigt werden, nicht durchführbar.

Benötigt wird also ein Server, der das *mounten* dieser Laufwerke beherrscht, z. B. ein Linux-Server. Damit ist dieser Server auch hervorragend für diese Art der Bereitstellung von Netzwerkressourcen geeignet, in vielen CD-ROM-„Spezialservern« arbeiten Systeme auf Linux-Basis.

Soll nur ein CD-ROM-Server realisiert werden, kann tatsächlich ein relativ alter Rechner mit ein oder zwei SCSI-Interfaces ausgestattet werden, der dann viele CD-ROM-Laufwerke über TCP/IP im Netzwerk bereitstellt.

Netzwerk-Clients können dann sofort z. B. über NFS diese Ressource nutzen, noch effizienter ist aber die Bereitstellung über eine Novell- oder Windows-Emulation, die dann keine weitere Spezialsoftware auf den Clients erfordert.

3.3.3 Netzwerk mit Microsoft-Clients: SAMBA

Das 1991 entstandene Programmpaket SAMBA (*www.samba.org*) des Australiers Andrew Tridgell verwendet das Windows-eigene SMB-Protokoll (Server Message Block). Es macht beliebige Unix-Rechner zu einem leistungsfähigen Datei- und Druckserver für DOS- und Windows-Rechner.

SMB baut auf NetBIOS auf, einem *Netzwerkprotokoll*, das über *Transportportokolle* wie NetBEUI, IPX/SPX und TCP/IP (genauer gesagt NetBIOS über TCP/IP gem. RFC 1001 und RFC 1002) übertragen werden kann. SMB ist ein Client-Server-Protokoll, mit dem der Client eine Anfrage an den SMB-Server stellt, die dieser dann beantwortet (Abbildung 3-44).

Hat der SMB-Client eine Verbindung zum Server hergestellt, werden SMB Kommandos gesendet, die den Zugriff auf Freigaben, das Öffnen von Dateien und beliebige Schreib-und Leseaktionen ermöglichen.

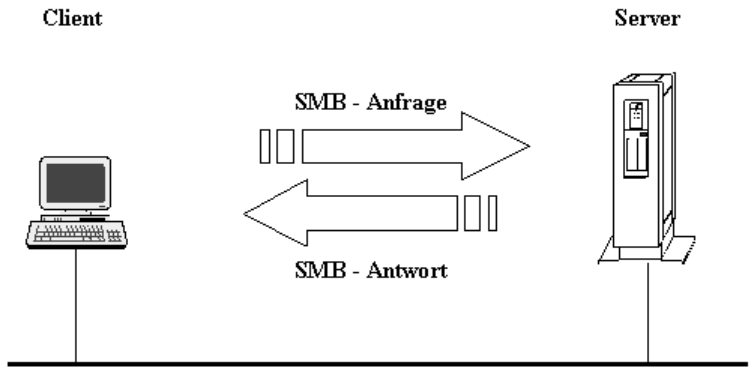


Abbildung 3-44 Client-Server-Struktur des SMB Protokolls

SAMBA bietet die Möglichkeit, einen Linux-Server einfach und fast unbemerkt von den Client-Rechnern in bestehende PC-Netzwerke mit Microsoft-Clients zu integrieren. Einzige Voraussetzung: Das Protokoll TCP/IP muss auf allen Clients installiert sein, eine automatische Adresszuweisung über DHCP ist empfehlenswert.

Die Client-Rechner unter Windows 9x/NT benötigen keinerlei Zusatzsoftware für den Zugriff auf einen *SAMBA*-Server. Mit entsprechender TCP/IP-Netzwerksoftware können selbst ältere Rechner unter MS-DOS oder Windows 3.x sofort in das Netzwerk integriert werden. Besonders interessant ist die Möglichkeit, Netzwerkverbindungen über einfache Boot-Disketten herstellen zu können.

Jeder Client-Rechner kann sofort auf die vom einem *SAMBA*-Server freigegebenen Verzeichnisse und Drucker zugreifen, ohne den Umweg über *NFS*, *FTP* oder andere Hilfsprogramme. Der Zugriff auf gespeicherte Daten ist im direkten Vergleich zur Arbeit mit *NFS* deutlich schneller!

Als wichtiges Argument für den Einsatz von SAMBA unter Linux werden immer wieder Preisvergleiche angestellt, die bei vorgegebener Netzwerkgröße die Kosten für Hardware, Software und *Total Cost of Ownership* (Folgekosten für Administration, Wartung, Softwareupdates ...) umfassen. Solche Berechnungen zeigen, dass bereits in kleinen, aus nur 25 Arbeitsplätzen bestehenden PC-Netzwerken schon bis zu. 10.000 DM für Serversoftware gespart werden kann, da keine Lizenzen gekauft werden müssen. Dies ist zunächst richtig; berücksichtigt werden in diesen Aufstellung aber tatsächlich nur die Kosten für die Anschaffung der benötigten Serversoftware, genauer gesagt: der Lizenzen. Insbesondere die Kosten für die professionelle Installation und die Folgekosten der Adminstration werden nicht berücksichtigt.

Der SAMBA-Server besteht aus zwei Programmen, die beide als Hintergundprozess (Daemon) laufen:

- Das eigentliche Serverprogramm ist *smbd*. Für jeden Benutzer mit einer aktiven Verbindung erstellt *smbd* eine Kopie, die dann als zusätzlicher Dienst temporär läuft.
- Der NetBIOS-Nameserver *nmbd* kann in Windows-Netzwerken auch als WINS-Server (*Windows Internet Name Server*) eingesetzt werden. Weitere Angaben dazu können der ManPage *nmbd (8)* entnommen werden.

Planung der Systemkonfiguration

Das Programmpaket SAMBA bietet die Möglichkeit, den Linux-Server zunächst einmal als zusätzliche Komopnente in eine bestehendes Netzwerk zu integrieren. Einzige Voraussetzung ist, dass die Windows-Clients unter TCP/IP betrieben werden. Bei Bedarf können so zunächst Datei- und Druckdienste zusätzlich bereitgestellt werden, um sich ein eigenes Bild von der Leistungsfähigkeit und Zuverlässigkeit der SAMBA-basierten Lösung machen zu können.

Die beiden nachfolgend vorgestellten Konfigurationen erfüllen bereits grundlegende Anforderungen an die Serversicherheit und die Absicherung der Datei- und Druckdienste. Die oft in der einführenden Literatur beschriebenen und für den kommerziellen Bereich unakzeptabelen Minimalkonfigurationen sollen hier keine Rolle spielen.

Die Vorgaben zur Systemkonfiguration sind:

- Der SAMBA-Server soll möglichst vor unbefugten Zugriffsversuchen geschützt werden. Dazu wird festgelegt, dass nur Windows-Clients aus explizit angegebenen TCP/IP-Teilnetzwerken zugreifen dürfen. Wenn notwendig, soll der Zugriff auf alle SAMBA-Ressourcen als zusätzliche Absicherung nur bestimmten Benutzern oder Gruppen erlaubt werden.
- Durch Nutzung von Optimierungsmöglichkeiten vor allem im Bereich der TCP/IP-Kommunikation soll die Reaktionsgeschwindigkeit des SAMBA-Servers möglichst hoch werden.
- Der Server soll Standard-Zeitserver für alle Windows-Clients im Netzwerk sein.
- Dateinamen sollen bis zu 255 Zeichen lang sein dürfen. Die Unix-typische Unterscheidung nach Groß-/Kleinschreibung sollte nicht erfolgen, alle Datums- und Zeitangaben für Dateien sollen den Windows-Vorgaben entsprechen.
- Jeder Zugriff auf eine vom SAMBA-Server erstellte Freigabe erfordert eine vollständige Authentifizierung. Nur Benutzer, die sich mit Name und Kennwort identifizieren, erhalten Zugriffsrechte. Die Benutzerkonten werden als Linux-Bewnutzerkonten geführt; die Übertragung der Kennworte im Netzwerk erfolgt in keinem Fall im Klartext. Eine Ausnahme sollen nur Freigaben darstellen, die für den Zugriff auf CD-ROMs erstellt werden, hier wird jedem Benutzer im Netzwerk der Zugriff erlaubt.

- Der SAMBA-Server ist Mitglied in der Arbeitsgruppe IBSAMULAT, zu der auch alle Windows-Clients im Netzwerk gehören.
- Jeder Benutzer soll ein privates Verzeichnis erhalten. Abhängig vom angemeldeten Benutzer soll der Server eine Ressource über eine Freigabe bereitstellen, in der nur dieser Benutzer Schreib- und Leserechte hat. Alle anderen Benutzer dürfen hier keinerlei Rechte erhalten, die Ressource darf für Unbefugte nicht als Verzeichnis oder Freigabe sichtbar sein.
- Über eine versteckte Freigabe INSTALL soll ein Installationspunkt für Software bereitgestellt werden, der dem Systemverwalter die Einrichtung von Programmen auf den Windows-Clients ermöglicht. Benutzer sehen diese Freigabe nicht in ihrer Netzwerkumgebung und haben keinerlei Rechte auf dieser Ressource.
- Der Server soll über eine Freigabe mit dem Namen *Daten* eine von allen Benutzern gemeinsam genutzte Verzeichnisstruktur bereitstellen, auf der z.B. Dokumentvorlagen oder Texte verfügbar gemacht werden können. Alle Benutzer sollen hier Schreib- und Leserechte haben.
- Jeder Benutzer soll auf Druckressourcen zugreifen können, die vom SAMBA-Server bereitgestellt werden. Es sollen aber nicht alle in */etc/printcap* definierten, sondern nur ein paar ausgewählte Drucker freigegeben werden.

Stand-Alone-Server

In der ersten Beispielkonfiguration wird ein SAMBA-Server konfiguriert, der die eben zusammengefassten Standardanforderungen als »Stand-Alone«-Server erfüllt, also keine Funktion in einer zentralen Benutzerverwaltung übernimmt. Ein Server dieses Typs kann ohne Konflikte zusätzlich in bestehende PC-Netzwerke unter Novell oder Windows NT eingefügt werden; er ist mit seinen Freigaben in der Netzwerkumgebung der Windows-Clients sichtbar. Eine Ressource kann verwendet werden, wenn ein Benutzer diese z.B. per Doppelklick in seiner Netzwerkumgebung anfordert. Voraussetzung dafür ist, dass dieser Benutzer über die erforderliche Berechtigung verfügt, also die Authentifizierung an der Linux-Benutzerdatenbank erfolgreich durchführen kann.

Da keine Benutzerkennworte im Klartext übertragen werden dürfen, müssen als Windows-Clients Rechner unter Windows 95b, Windows 98 oder Windows NT (Service Pack 3 oder höher) eingesetzt werden. Ältere Windows-Clients übertragen Benutzer-Kennworte im Klartext! Unter TCP/IP kann schon mit einem einfachen Analyseprogramm (*Paket-Sniffer*) alles mitgelesen werden!

NT 4.0 Primary Domain Controller PDC

In der später beschreibenen, zweiten Beispielkonfiguration übernimmt der SAMBA-Server die Funktion der zentralen Benutzerdatenbank im Netzwerk, d.h. realisert werden die Funktionen eines NT 4.0 *Primary Domain Controller* PDC. Der hier beschriebene SAMBA-Server stellt eine Erweiterung der »Stand-Alone«-Lösung dar, die zusätzlichen Vorgaben entsprechen den Leistungsmöglichkeiten eines PDC:

- Die Benutzerdatenbank des PDC soll zur zentralen Anmeldung aller Benutzer im Netzwerk verwendet werden. Zur Absicherung dieser Funktion hat Microsoft das Dömanen-Modell eingeführt, das durch die Einrichtung einer Vertrauensstellung zwischen allen Rechnern einer Domäne bestimmt wird. Windows-Clients benötigen dazu ein *Rechnerkonto*. Diese Konten werden auf dem PDC zentral geführt und dienen der gegenseitigen Identifikation von PDC und der zur eigenen Domänen gehörenden Windows-Clients. Wichtiger Bestandteil dieser Indentifizierung sind die für jeden Rechner bei der Installation eindeutig bestimmten SID, die als »Kennwort« für diese Konten verwendet werden. Der als PDC konfigurierte SAMBA-Server wird beim ersten Start eine eindeutige SID erhalten und diese zur Identifikation gegenüber den anderen Rechner in der eigenen Domäne verwenden.
- Die Benutzerdatenbank des Linux-Servers, auf dem der SAMBA-Dienst läuft, wird zentrale Benutzerdatenbank im Netzwerk. Ist ein Windows-Client Mitglied in der Domäne, so kann der Benutzer sich direkt an der Domäne anmelden.
- Bei jeder Benutzeranmeldung wird ein Anmeldskript (*Login-Skript*) ausgeführt, um die Ressourcen des PDC über logische Laufwerke verfügbar zu machen. Jeder PDC besitzt dazu die Standardfreigabe NETLOGON., in der diese Skripte im Netzwerk bereitgestellt werden. Benutzer haben in NETLOGON nur Leserechte.
- Profile speichern benutzerbezogene Einstellungen der Arbeitsoberfläche. Dies kann bereits sehr hilfreich sein, wenn sich verschiedene Benutzer nacheinander an einer Windows-NT-Workstation anmelden, denn sie erhalten eine eigene, immer wieder gleiche Umgebung. Über die Freigabe PROFIL des PDC werden servergespeicherte Benutzerprofile für die Windows-NT-Workstation bereitgestellt; die Verwaltung der Profile wird auf dem SAMBA-Server zentralisiert. Damit wird das Benutzerprofil auch unabhängig von dem Rechner bereitgestellt, an dem sich dieser Benutzer anmeldet. Die persönliche Umgebung wird »mitgenommen«, bei jeder Abmeldung wird das zentrale Profil automatisch aktualisiert.
- Bei Bedarf werden Systemrichtlinien (*Policies*) über die Standardfreigabe NETLOGON des PDC im Netzwerk bereitgestellt, um die Benutzerumgebung einer Windows-NT-Workstation benutzer- und rechnerbezogen einzuschränken.

Die Konfigurationsdatei /etc/smb.conf

Dies ist die Konfiguration des SAMBA-Dienstes über eine einzige Datei. Der Aufbau der Textdatei *etc/smb.conf* erinnert stark an die typischen Windows-Dateien vom Typ *.ini*. Parameter und deren Inhalte werden in der Schreibweise *name=wert* definiert, die Datei ist strukturiert durch Abschnitte, deren Überschriften in eckigen Klammern stehen. Kommentare werden durch Hash oder Semikolon eingeleitet.

Die Abschnitte definieren jeweils ein freigegebenes Verzeichnis. Der Name der Freigabe ist der Name der Überschrift. Eine Ausnahme stellen hier [global], [homes] und [printers] dar, die besondere Funktionen übernehmen und nicht direkt eine Freigabe darstellen (Tabelle 3-12).

[global]	enthält globale Voreinstellungen für das gesamte System.
[homes]	definiert eine automatisch erstellte Freigabe für das private Verzeichnis des jeweils angemeldeten Benutzers. Damit muss nicht jedem Benutzer allein zu diesem Zweck eine eigene Freigabe erstellt werden. Der Name der Freigabe ist immer der Benutzername. Freigegeben wird das Linux- Home-Verzeichnis des angemeldeten Benutzers.
[printers]	definiert Freigaben für die am Server verfügbaren Drucker. Alle in */etc/printcap* erstellten Drucker können bei Bedarf automatisch freigegeben werden.

Tabelle 3-12 SAMBA Konfiguration

Erste SAMBA-Konfiguration: Stand-Alone-Server

Der nachfolgende Text zeigt beispielhaft den Inhalt einer */etc/smb.conf* für einen Stand-Alone-Server. Die angegebenen Parameter werden bei Bedarf kommentiert:

```
[global]
  workgroup = IBSAMULAT
```

ist der Name der Arbeitsgruppe. Dies ist zunächst nur eine »Gliederungsebene«, innerhalb derer dieser SAMBA-Server mit allen anderen dieser Arbeitsgruppe zugeordneten Windows-Rechnern zusammen angezeigt wird. Trotzdem ist es sinnvoll, bereits hier eine sinnvolle Zuordnung zu treffen. Das gilt besonders dann, wenn dieser SAMBA-Server später die Aufgaben eines Windows NT 4.0 *Primary Domain Controlles* PDC übernehmen soll.

```
guest account = nobody
```

Wird eine Freigabe für »Jedermann« verfügbar gemacht (*public* = *yes*), so wird das hier angegebene Linux-Benutzerkonto verwendet. Sollen ohne Ausnahme nur authentifizierte Benutzer auf Ressourcen dieses SAMBA-Servers zugreifen können, so darf kein *guest account* angegeben werden!

```
keep alive = 30
```

gibt an, in welchem Zeitabstand in Sekunden *Keepalive*-Pakete gesendet werden, um zu Prüfen, ob der Client weiter verfügbar ist. Der Standardwert für diesen Parameter ist Null, in diesem Fall werden keine Prüfungen durchgeführt. Die Prüfung kann aber dann sinnvoll sein, wenn Windows-9x-Clients-Ressourcen des SAMBA-Servers verwenden. »Hängt« ein solcher Rechner, wird mit *keep alive* = *30* die Verbindung nach spätestens 30 Sekunden zwangsweise getrennt.

```
Log file /var/log/samba-log.%m
```

Für jeden Windows-Client wird eine Logdatei mit dem Namen *samba-log.Rechnername* im Verzeichnis */var/log* geführt. Der Parameter *%m* steht für den NetBIOS-Namen des Windows-Clients, vom dem aus der aktuelle Zugriff auf Ressourcen erfolgt (eine Beschreibung der in */etc/smb.conf* verwendbaren Parameter folgt später).

```
hosts allow = 192.168.100/255.255.255.0
hosts allow = 201.100. EXCEPT 201.100.10.1
host allow = samulat.de testdomain.de
```

Der Parameter *host allow* dient der Absicherung vor unbefugtem Zugriff und stellt damit einen sehr wichtigen Eintrag dar. Nur Clients aus den hier angegebenen Teilnetzen sind befugt, auf Ressourcen dieses SAMBA-Servers zuzugreifen. Wie oben dargestellt, kann die Angabe der Teilnetze über IP-Adressen oder über Internet-Namen erfolgen, mehrere Teilnetze können gemeinsam, durch Komma getrennt, angegeben werden. Bei Bedarf können Teile des zugelassenen Netzes mit *EXCEPT* explizit gesperrt werden. Bei Bedarf kann *host allow* für jede Freigabe angegeben werden, um weitere Einschränkungen vorzunehmen.

```
valid users = samulat,jan,michael
```

Damit erfolgt die Beschränkung des Zugriffs auf einzelne, im Parameter *valid users* angegebene Benutzer. Auch dies ist auf jeder einzelnen Freigabe möglich, es sollte dann aber unbedingt der Parameter *public=no* eingetragen werden.

```
valid users = @lager
```

Damit erfolgt die Beschränkung des Zugriffs auf alle Benutzer, die der Linux-Gruppe *@lager* angehören.

```
security = user
```

Jede mit dem SAMBA-Server erstellte und im Netzwerk verfügbare Freigabe basiert immer auf einem lokalen Verzeichnis dieses Linux-Servers, natürlich sind dafür auch die Linux Berechtigungen definiert. Jeder Benutzer, der sich an einem SAMBA-Server anmeldet, muss also unbedingt über ein Linux-Konto verfügen. Erhält dieser Benutzer unter Linux keine Rechte auf dem freigegebenen Verzeichnis, kann er auf keinen Fall darauf zugreifen, auch wenn später in der */etc/smb.conf* mehr Berechtigungen eingestellt werden. Der Parameter *security = user* erzwingt die Authentifizierung des Benutzers. Name und Kennwort müssen unter Linux und, wenn mit geschlüsselten Kennworten gearbeitet wird, in */etc/smbpasswd* gespeichert sein.

```
encrypt passwords = yes
```

Hier wird festgelegt, dass der SAMBA-Server mit verschlüsselter Kennwortübertragung arbeitet. So wird das Anmelde-Kennwort des Benutzers in keinem Fall im Klar-

text über das Netzwerk gesendet. Die verschlüsselten Windows-Kennworte müssen in diesem Fall aber in der zusätzlichen Datei */etc/smbpasswd* gepflegt werden.

```
printing = bsd
printcap name = /etc/printcap
load printers = no
```

Als Drucksystem wird BSD verwendet, die Druckerdefinition erfolgt in */etc/printcap*. Der Parameter *load printers = no* stellt sicher, dass nicht alle auf diesem Server verfügbaren Drucker automatisch angezeigt werden. Der Abschnitt [printers] wird also hier nicht verwendet, vielmehr wird an späterer Stelle direkt eine einzelne Druckerfreigabe erstellt.

```
socket options = TCP_NODELAY
```

Dieser Eintrag dient der Geschwindigkeitsoptimierung, SAMBA arbeitet mit vielen kleinen Netzwerkblöcken, die Unix aber puffert, um eine effiziente Übertragung zu ermöglichen. Für SAMBA ist dieses Verhalten aber nicht vorteilhaft, der obige Parameter schaltet diese Funktion aus. Gerade im Bereich der TCP/IP-Kommunikation sind weitere Optimierungsmöglichkeiten enthalten, die aber dann z.B. auch sehr stark vom Typ der verwendeten Netzwerkkarte abhängen:

```
oplocks = YES
```

Mit *oplocks = yes* werden Schreibzugriffe auf geänderte Dateien erst dann ausgeführt, wenn ein weitere Client die gleiche Datei lesen möchte (Schreib-Cache).

```
write raw = YES
read raw = YES
```

Mit *raw write = yes* und *read raw = yes* werden bis zu 65.535 Bytes in jedem Datenpaket gesendet bzw. empfangen Dies kann, je nach Typ der verwendeten Netzwerkkarte, einen erheblichen Geschwindigkeitszuwachs ergeben.

Im Einzelfall muss die weitere Optimierung experimentell durchgeführt werden. Details dazu enthält z.B. die man-Page zu *smb.conf*. In jedem Fall sollte in der Konfigurationsdatei */etc/rc.config* die Dummy-device mit *SETUPDUMMYDEF = no* deaktiviert werden, um unnötige Protokollabläufe zu vermeiden.

```
interfaces = 192.168.100.1/255.255.255.0
```

gibt die tatsächlich vorhandenen Teilnetze bekannt. Dies ist besonders dann wichtig, wenn einer Netzwerkkarte virtuelle Adressen zugeordnet wurden. Die Reaktionsgeschwindigkeit des SAMBA-Servers wird aber in jedem Fall erheblich erhöht, wenn hier die IP-Adresse des Servers mit der dazu gehörenden Subnet-Mask eingetragen wird.

```
time server = yes
```

Der SAMBA-Server ist Standard-Zeitserver im lokalen Netzwerk. Windows-Clients können mit dem Befehl *net time /s /y* ihre Uhrzeit direkt synchronisieren.

```
mangle case = no
case sensitive = no
short preserve case = yes
preserve case = yes
```

Windows-Systeme gehen mit Dateinamen anders um als Unix. Beide unterstützen lange Dateinamen mit bis zu 255 Zeichen, allerdings unterscheidet Unix zwischen Groß- und Kleinbuchstaben. Wird eine Windows-Datei mit einem Namen gespeichert, der nur aus Großbuchstaben besteht, so zeigt Windows diese Datei anschließend mit führendem Großbuchstaben, der Rest ist kleingeschrieben. Die o.a. Parameter passen die Unix- und die Windows-Namenskonvention aneinander an. Mit *case sensitive = no* unterscheidet der SAMBA-Server nicht mehr zwischen Groß- und Kleinschreibung, die anderen Parameter sorgen dafür, dass die Orginalschreibweise beim Abspeichern eingehalten wird.

```
dos filetimes = yes
dos filetime resolution = yes
```

Auch im Bereich der Datums- und Zeitangaben weisen Unix- und Windows-Systeme Unterschiede auf. Windows speichert Zeitangaben nur auf zwei Sekunden genau, Unix arbeit auf die Sekunde genau. Mit den obigen Parametern wird der SAMBA- Server auf die Windows-Vorgabe beschränkt.

```
[homes]
  comment = Heimatverzeichnis
  browseable = no
  read only = no
  create mode = 0750
```

Erstellt die automatische Freigabe für das private Verzeichnis des angemeldeten Benutzers. Der Kommentar ist *Heimatverzeichnis*, dieser wird in der Netzwerkumgebung der Windows-Clients mit angezeigt. Der *create mode* legt die Unix-Zugriffsrechte für vom Client angelegte Dateien und Verzeichnisse in der schon bekannten oktalen Schreibweise fest. Wie auch mit dem Befehl *chmod* setzt so beispielsweise die Angabe 644 Lese- und Schreibrechte für den Eigentümer; die Gruppe und andere Nutzer erhalten nur Leserechte und keine Rechte auf neu angelegte Dateien oder Verzeichnisse.

```
[install]
   comment = Installationspunkt
   path = /home/install
   browseable = no
   read only = yes
```

Erstellt eine Freigabe mit dem Namen INSTALL, die auf */home/install* verweist. Alle Benutzer erhalten mit *read only = yes* höchstens Leserechte, die Freigabe wird als »versteckte Freigabe« nicht in der Netzwerkumgebung der Windows-Clients angezeigt (*browseable = no*). Diese Ressource ist als netzwerkweit verfügbarer Installationspunkt gedacht, über den der Systemverwalter Standardsoftware auf den Windows-Clients einrichten kann. Zweckmäßigerweise erhält nur der Systemverwalter Rechte auf */home/install*.

```
[daten]
   comment = Datenverzeichnis
   path = /home/daten
   browseable = yes
   read only = no
   create mode = 0777
```

Erstellt eine Freigabe mit dem Namen DATEN zur gemeinsamen Nutzung durch alle Benutzer. Basis ist das lokale Verzeichnis */home/daten*. Alle Benutzer erhalten Schreib- und Leserechte *(read only = no)*, Berechtigungsdetails werden durch die Linux-Berechtigungen festgelegt.

```
[cd1]
   comment = CD-ROM Laufwerk 1
   path = /cd1
   public = yes
   locking = no
   read only = yes

[cd2]
   comment = CD-ROM Laufwerk 2
   path = /cd2
   public = yes
   locking = no
   read only = yes
```

Der Server dieser Beispielkonfiguration verfügt über zwei CD-ROM-Laufwerke, die je nach Bedarf in den Verzeichnissen */cd1* und */cd2* gemountet werden. Mit *public = yes* benötigen diese Freigaben keine Anmeldung, jeder Benutzer im Netzwerk kann auf diese Ressourcen des SAMBA-Servers nur lesend zugreifen (*read only = yes)*. Der SAMBA-Server verbietet bereits in der Voreinstellung den Schreibzugriff auf eine Freigabe, sodass die Angabe von *read only = yes* auch entfallen kann.

```
[hpl6]
   printable = yes
   comment = HP Laserjet 6L
```

In dem Abschnitt [hpl6] wird der in */etc/printcap* definierte Drucker *lp2* als Netzwerkressource bereitgestellt; der Name der Freigabe ist *hpl6*.

Mit *printable* = *yes* wird festgelegt, dass der Drucker nicht als Verzeichnis exportiert werden soll, der Kommentar zu dieser Freigabe ist *HP Laserjet 6L*.

```
printer driver = HP Laserjet IV
```

definiert den Namen des Druckertreibers, der von Windows-9x/NT-Systemen verwendet werden soll. Voraussetzung für die richtige Funktion ist, dass der hier angegebene Name der Druckertreibers so geschrieben wird, wie er auch im Konfigurationsmenü des Windows-Clients angegeben wird.

```
path = /var/tmp
public = yes
writeable = yes
browseable = yes
lpq command = /usr/bin/lpq -Plp2
lprm command = /usr/bin/lprm -Plp2 %j
print command = /usr/bin/lpr -Plp2 -rs %s
```

Der Parameter *-Plp2* für die Programme *lpq, lprm* und *print* gibt an, dass als Drucker der in */etc/printcap* definierte Eintrag für *lp2* verwendet wird. Der Drucker ist für jeden Benutzer verfügbar (*public* = *yes*, *writeable* = *yes*). Die Druckdaten werden temporär im lokalen Verzeichnis */var/temp* zwischengespeichert.

Parameter in /etc/smb.conf

Tabelle 3-13 zeigt eine Auswahl der Parameter, die zur Programmierung in */etc/smb.conf* zur Verfügung stehen.

Parameter	Inhalt
%U	in dieser Sitzung verwendeter Benutzername (genau der vom Client angegebene Benutzername, nicht also in jedem Fall der tatsächlich verwendete)
%G	die primäre Gruppe vom Benutzer %U
%H	das Home-Verzeichnis des Benutzers %U
%v	SAMBA-Versionsnummer
%h	der Internet-Hostname des Servers, auf dem der SAMBA-Dienst läuft
%m	der NetBIOS-Name des Client-Rechners
%L	der NetBIOS-Name des SAMBA-Servers
%M	der Internet-Hostname des Client-Rechners
%d	die Prozess-ID des SAMBA-Dienstes
%a	das Betriebssystem des Client-Rechners. Aktuell werden nur Samba, WfW, WinNT und Win95 erkannt, alle anderen werden als »UNKNOWN« angezeigt.
%I	die IP-Adresse des Client-Rechners
%T	aktuelles Datum und Uhrzeit

Tabelle 3-13 Parameter in /etc/smb.conf

Da smbd für jede aktive Verbindung einen neuen Prozess startet, der dann die aktuelle /etc/smb.conf auswertet, können über diese Parameter computer- oder benutzerbezogene Aktionen ausgeführt werden.

Test der Konfigurationsdatei /etc/smb.conf

Vor dem ersten Start des SAMBA-Servers sollte zweckmäßigerweise die korrekte Syntax der Konfigurationsdatei *etc/smb.conf* mit dem zu SAMBA gehörenden Programm *testparm* überprüft werden. Mit

```
# /usr/bin/testparm
Load smb config files from /etc/smb.conf
Processing section "[netlogon]"
Processing section "[homes]"
Processing section "[install]"
Processing section "[daten]"
Processing section "[cd1]"
Processing section "[cd2]"
Processing section "[hpl6]"
Loaded services file OK.
Press enter to see a dump of your service definitions
(...)
```

wird diese Überprüfung durchgeführt. Die abschließende Meldung *Loaded services file OK* zeigt, dass die aktuelle Konfigurationsdatei keine Fehler enthält.

Installation und Start

Der *SAMBA*-Daemon wird mit

```
rcsmb start
```

manuell gestartet. Zum automatischen Start während des Systemboot sollte in */etc/rc.config* der Eintrag

```
START_SMB=YES
```

vorgenommen werden. Nach Änderungen in der Konfigurationsdatei */etc/smb.conf* werden die aktuellen Einstellungen erst nach einem Neustart des Daemons mit

```
rcsmb restart
```

übernommen.

Benutzerkennworte verwalten – smbpasswd

Damit Benutzer auf die Ressourcen des SAMBA-Servers zugreifen können, ist es notwendig, diese auf dem Server einzurichten. Jeder Benutzer muss dazu zunächst mit *useradd* oder über *YaST* auf dem Linux-Server angelegt werden, wobei bereits jetzt unbedingt ein Kennwort zugewiesen werden sollte. Soll ein Benutzer nur

über SAMBA auf Ressourcen dieses Servers zugreifen und sich nicht direkt anmelden können, wird als Login-Shell */bin/false* zugewiesen. Damit ist die Anmeldung über *telnet, rlogin* oder *ftp* unmöglich.

Da das Kennwortsystem von Windows-Systemen und Unix nicht direkt kompatibel ist, werden die Windows-Kennworte aller Benutzer in der zusätzlichen Datei */etc/smbpasswd* eingetragen.

Der Shell-Befehl *smbpasswd* verwaltet die SAMBA-Kennworte, die für die geschlüsselte Anmeldung notwendig sind. Ähnlich wie mit dem Linux-Shell-Befehl *passwd* kann der Systemverwalter *root* neue Einträge in der Kennwortdatei erstellen oder einem beliebigen Benutzer ein neues Kennwort zuweisen. Ein Benutzer kann über *smbpasswd* bei Bedarf sein eigenen Kennwort ändern. Mit

```
smbpasswd -a <benutzername>
```

wird eine neuer Eintrag in */smb/smbpasswd* für *<benutzername>* erstellt. Das dazu gehörende Kennwort wird über die Konsole abgefragt.

Mit zwei weiteren Optionen kann der Systemverwalter einen bestehenden Eintrag in */smb/smbpasswd* aktivieren (*-e*) oder deaktivieren (*-d*) werden.

Standardmäßig muss jedem Domänen-Benutzer ein Kennwort zugewiesen werden, um die Anmeldung überhaupt zu ermöglichen. Im Ausnahmefall kann ein leeres Kennwort zugelassen werden, dies geschieht mit

```
# smbpasswd -n <benutzername>
```

Zusätzlich muss im Abschnitt [global] von */etc/smb.conf* der Eintrag

```
null passwords = true
```

aufgenommen werden.

Betrieb und Test

Der SAMBA-Server stellt ein komplexes System dar, das in jedem Fall schrittweise installiert und getestet werden sollte. Typische Probleme entstehen z. B. durch die »Mischung« der Berechtigungen auf der Ebene des Linux-Dateisystems mit den SAMBA-Berechtigungen, die in */etc/smb.conf* zugewiesen werden.

Nach dem ersten Start des SAMBA-Servers kann es sinnvoll sein, die nachstehend beschriebenen Tests schrittweise abzuarbeiten.

(1) Test der Server-Dienste

Konnte der SAMBA-Server erfolgreich gestartet werden, so wird mit

```
# ps x | grep smbd
```

ein oder auch mehrere Einträge aus der Prozesstabelle ausgegeben. Der Dienst *smbd* sollte mindestens einmal auftauchen. Da *smbd* für jeden angeforderten Dienst eine Kopie von sich selbst erstellt und dann wiederum temporär als eigenen Dienst startet, kann aus der Anzahl der mit *ps* gefundenen Einträge auch abgeleitet werden, wie viele Anforderungen aktuell existieren.

(2) Test der Server-Grundfunktion

Die Grundfunktionen des laufenden *SAMBA*-Servers können mit dem Programm *smbclient* überprüft werden. Mit

```
# smbclient -L localhost
Added interface ip=192.168.100.1 bcast=192.168.100.255 nmask=255.255.255.0

Domain=[IBSAMULAT] OS=[Unix] Server=[Samba 2.0.5a]

    Sharename      Type      Comment
    ---------      ----      -------
    install        Disk      Installationspunkt
    daten          Disk      Datenverzeichnis
    cd1            Disk      CD-ROM Laufwerk 1
    cd2            Disk      CD-ROM Laufwerk 2
    hpl6           Printer   HP Laserjet 6L
    IPC$           IPC       IPC Service (Samba 2.0.5a)

    Server               Comment
    ---------            -------
    NTSQL1
    LINUX01              Samba 2.0.5a
    NT40SAM

    Workgroup            Master
    ---------            -------
    IBSAMULAT            LINUX01
```

wird der Zustand des auf diesem Server laufenden *SAMBA*-Daemons abgefragt. Angezeigt werden die IP-Adressinformationen, die Arbeitsgruppe (*Domain*), in der der SAMBA-Server arbeitet, und die erstellten Freigaben (*Shares*) für Verzeichnisse und Drucker. Im obigen Beispiel arbeitet im Netzwerk nicht nur der *SAMBA*-Server *Linux01*, sondern es wurden noch zwei weitere Rechner mit den Namen *NTSQL1* und *NT40SAM* gefunden (Betriebssystem dieser beiden Rechner ist Windows NT).

Wird das Programm *smbclient* nicht mit *localhost*, sondern mit dem Hostnamen oder der IP-Adresse eines im Netzwerk laufenden entfernten *SAMBA*- oder Windows-9x/NT-Servers, so sollte die entsprechende Zustandsanzeige ausgegeben werden.

(3) Aktuelle Serververbindungen abfragen

Mit dem Befehl *smbstatus* kann festgestellt werden, welche Clients zur Zeit eine Verbindung zum SAMBA-Server haben und welche Ressourcen aktuell verwendet werden:

```
# smbstatus
Samba version 2.0.5a
Service      uid      gid      pid     machine
----------------------------------------------
samulat      samulat  users    17132   nt40sam  (192.168.100.10) Sat Nov 27
16:31:04 1999
daten        samulat  users    17132   nt40sam  (192.168.100.10) Sat Nov 27
16:31:22 1999

Locked files:
Pid    DenyMode   R/W        Oplock           Name
--------------------------------------------------
17132  DENY_NONE  RDWR       EXCLUSIVE+BATCH  /home/daten/addison-wesley/awl-8-
o.dot   Sat Nov 27 16:35:21 1999
17132  DENY_NONE  RDWR       EXCLUSIVE+BATCH  /home/daten/addison-wesley/
Buch.doc   Sat Nov 27 17:38:27 1999
17132  DENY_NONE  RDWR       NONE             /home/daten/addison-wesley/
~WRL0002.tmp   Sat Nov 27 16:44:56 1999

Share mode memory usage (bytes):
   1048128(99%) free + 344(0%) used + 104(0%) overhead = 1048576(100%) total
```

Die Share *mode maemory usage* gibt eine Information über die tatsächliche Speicherbelegung und die frei verfügbaren Ressourcen.

(4) Funktion der NetBIOS-Namensauflösung prüfen

Mit

```
# nmblookup -B linux01
Sending queries to 192.168.100.1
```

wird geprüft, ob die Auflösung des NetBIOS-Rechnernamens für den SAMBA-Server richtig funktioniert. Als Antwort sollte die IP-Adresse des angegebenen Servers angezeigt werden.

(5) Zugriff auf Ressourcen mit smbclient

Der Befehl *smbclient* kann auch verwendet werden, um eine Verbindung zu einer Freigabe herzustellen, die von einem SAMBA-Server oder einem anderen Windows-9x/NT-Server im Netzwerk bereitgestellt wird. Testweise sollte zunächst versucht werden, eine der Freigaben des SAMBA-Servers zu verwenden. Mit

```
# smbclient \\\\linux01\\daten -Usamulat
```

wird versucht, die Freigabe *daten* auf dem SAMBA-Server *linux01* zu verwenden. Die Authentifikation soll mit dem Benutzernamen *samulat* durchgeführt werden. Die Angabe der zu verwendenden Freigabe erfolgt nach der *Universal Naming Convention* UNC. Der vollständige Name einer Ressource ist dabei durch die Syntax

\\Servername\Freigabename

festgelegt. Dem Servernamen ist ein doppelter Backslash vorangestellt, nach dem Servername folgt, durch einen Backshlash getrennt, der Name der Freigabe. Da der Backslash unter Unix standardmäßig ein »Fluchtzeichen« darstellt, müssen diese Sonderzeichen in der Kommandozeile doppelt geschrieben werden. Kann *smbclient* die angeforderte Verbindung herstellen, wird das Benutzerkennwort abgefragt:

```
Added interface ip=192.168.100.1 bcast=192.168.100.255 nmask=255.255.255.0
Password:
```

Nach erfolgreicher Anmeldung wartet *smbclient* jetzt auf die Eingabe von Steuerbefehlen, mit denen direkt auf der angegebenen Freigabe gearbeitet werden kann. Mit »?« kann eine Übersicht der möglichen Befehle abgerufen werden:

```
Domain=[IBSAMULAT] OS=[Unix] Server=[Samba 2.0.5a]
smb: \> ?
ls             dir          du           lcd          cd
pwd            get          mget         put          mput
rename         more         mask         del          open
rm             mkdir        md           rmdir        rd
prompt         recurse      translate    lowercase    print
printmode      queue        cancel       quit         q
exit           newer        archive      tar          blocksize
tarmode        setmode      help         ?            !
smb: \> q
```

Nach Eingabe von »q« oder »quit« wird *smbclient* beendet. Konnte auch dieser Test erfolgreich abgeschlossen werden, sollte auch versucht werden, mit *smbclient* auf eine Freigabe eines beliebigen Windows-Rechners im Netzwerk zuzugreifen.

(6) Auswertung der SAMBA-Logdateien

Der SAMBA-Server führt während des Betriebs eigene Logdateien; für die beiden Dienste *snmd* und *nmbd* sind dies */var/log/log.smb* und */var/log/log.nmb*. Zusätzlich wird, entsprechend des Eintrages *log file=/var/log/samba-log.%* in der Konfigurationsdatei */etc/smb.conf* für jeden Windows-Client eine eigene Logdatei geführt.

Die Logdatei des SAMBA-Dienstes *nmbd* ist */var/log/log.nmb*:

```
*****

  SAMBA-Server LINUX01 is now a domain master browser for workgroup IBSAMULAT
on subnet UNICAST_SUBNET
```

```
  *****
[1999/11/21 18:22:30, 0] nmbd/
nmbd_become_dmb.c:become_domain_master_browser_bcast(294)
  become_domain_master_browser_bcast:
  Attempting to become domain master browser on workgroup IBSAMULAT on subnet
192.168.100.1

(...)

[1999/11/27 11:36:13, 1] nmbd/nmbd_processlogon.c:process_logon_packet(69)
  process_logon_packet: Logon from 192.168.100.10: code = 12
```

Diese Logdatei enthält Informationen, die mit der Auflösung der NetBIOS-Namen im Zusammenhang stehen. Der im o.a. Beispiel gezeigte Auszug enthält so einen Eintrag, der den Server *linux01* als Master-Browser für die Arbeitsgruppe IBSAMULAT ausweist.

Die Logdatei des SAMBA-Dienstes *smbd* ist */var/log/log.smb*:

```
nt40sam (192.168.100.10) connect to service samulat as user samulat (uid=501,
gid=100) (pid 14519)
[1999/11/26 18:44:31, 1] smbd/service.c:close_cnum(557)
  nt40sam (192.168.100.10) closed connection to service samulat
[1999/11/27 11:36:14, 1] smbd/service.c:make_connection(521)
  nt40sam (192.168.100.10) connect to service samulat as user samulat (uid=501,
gid=100) (pid 16653)
[1999/11/27 11:36:14, 1] smbd/service.c:make_connection(521)
  nt40sam (192.168.100.10) connect to service netlogon as user samulat
(uid=501, gid=100) (pid 16653)
[1999/11/27 11:36:18, 0] smbd/nttrans.c:call_nt_transact_ioctl(2387)
```

Protokolliert werden Ereignisse im Zusammenhang mit der Identifikation von Benutzern und der Verwendung der vom SAMBA-Server bereitgestellten Ressourcen.

Da in der Konfigurationsdatei */etc/smb.conf* festgelegt wurde, dass auch für jeden Windows-Client eine Logdatei geführt werden soll, sind dort die auf den einzelnen Rechner bezogenen Details aus */var/log/log.smb* eingetragen.

Werden die Dienste *smbd* und *nmbd* mit der Option *-d* gestartet, so wird der interne Debugger eingeschaltet; der Umfang der protokollierten Vorgänge steigt an. Die Menge der Informationen kann in */etc/smb.conf* mit dem Parameter *debug level* im Bereich von *1* bis *100* eingestellt werden.

Es sollte in jedem Fall vermieden werden, dass das Logfile des Servers übermäßig anwächst. Sobald der SAMBA-Server stabil läuft,kann mit *debug level = 1* das Logging minimiert und zustätzlich mit *max log size = 1000* die Größe der Logdatei auf 1 Mbyte beschränkt werden. Erreicht das Logfile diese Größe, wird ein *.old* angehängt und es wird ein neues Logfile angefangen.

Tipps zur Fehlersuche

Nicht alle Client-Betriebssysteme unterstützen Freigabenamen, die länger als acht Zeichen sind. Dieses gilt für nahezu alle Clients unter 8- und 16-Bit-Betriebssystemen, wie DOS oder Windows for Workgroups.

Eine mögliche Fehlerquelle könnte sein, dass einige Clients Benutzerkennworte, bevor diese über das Netzwerk gesendet werden, in Großbuchstaben umwandeln und damit die Authentifizierung unmöglich machen.

Bei sehr hartnäckigen Problemen kann oft die Analyse der übertragenen SMB-Blöcke weiterhelfen. Eine Version von *tcpdump*, die auch SMB versteht, kann über *ftp://samba.anu/edu.au/pub/samba/tcpdump-smb* bezogen werden (in der CD-ROM im Verzeichnis *\samba\tcpdump-smb*).

SAMBA-Clients unter DOS, WfW Windows 9x/NT und OS/2

Um unter DOS, Windows for Workgroups (WfW) oder Windows9x/NT auf die SAMBA-Serverdienste zugreifen zu können, muss das Protokoll TCP/IP und ein NetBIOS-Client installiert sein. Obwohl in den heutigen Netzwerken kaum noch Arbeitsplatzrechner mit diesen Betriebssystemen zu finden sind, kann ein unter DOS laufender NetBIOS-Client bereits auf einer einzelnen Diskette untergebracht werden. Ist diese Diskette bootfähig, so kann ein beliebiger Rechner eine Netzwerkverbindung zu einem SAMBA-Server herstellen, um dann z. B. über einen Installationspunkt neu eingerichtet zu werden.

DOS und WfW

Bei DOS- oder WfW-Clients kann die notwendige zusätzliche Netzwerk-Software direkt über Microsoft bezogen werden (diese Clients sind auch auf der NT-4.0-Server-CD enthalten). Der Zugriff auf die Ressourcen des SAMBA-Servers erfolgt zweckmäßigerweise über *net*-Befehle. Mit

```
C:\> net view
Server-Name                    Beschreibung
-------------------------------------------------------------
\\linux01                      Samba 2.0.5a
\\nt40sam
Der Befehl wurde erfolgreich ausgeführt
```

werden alle SMB-Server im aktuellen Netzwerk angezeigt. Die Einrichtung eines logischen Laufwerkes *M:* zur Nutzung einer Ressource *\\linux01\daten* kann mit

```
C:\> net use M: \\linux01\daten /PERSISTENT:NO
Der Befehl wurd erfolgreich ausgeführt.
```

erfolgen. Die hier angegebene Option */PERSISTENT:NO* stellt sicher, dass die Laufwerkszuordnung nur für die aktuelle Sitzung gültig ist.

MSDOS Bootdiskette mit TCP/IP und NetBIOS-Client

Für Neuinstallationen, Migrationen auf ein neues Arbeitsplatz-Betriebssystem oder für den Fall, dass ein Netzwerk-Arbeitsplatz keine Verbindung mehr zu einem Server herstellen kann, ist eine Bootdiskette sehr hilfreich. Damit kann rechnerunabhängig vom tatsächlichen Betriebssystem gebootet und eine frei programmierbare Verbindung im Netzwerk hergestellt werden.

In Novell-Netzwerken war es schon fast selbstverständlich, DOS-Bootdisketten mit der Client-Software für den Novell-Server (dem *DOS-Requester*) vorzubereiten, um dann bei Bedarf schnell eine Netzwerkwerkverbindung herstellen zu können. Ist eine unter DOS formatierte, aber sonst leere Boot-Diskette verfügbar, so kann dies im SAMBA-Netzwerk auch mit einem NetBIOS-Client erreicht werden.

Voraussetzung ist eine unter MS-DOS formatierte, bootfähige Diskette. Da der Microsoft-NetBIOS-Client unter TCP/IP fast den gesamten Diskettenplatz benötigt, können hier auch keine unter Windos 9x formatierten Disketten zum Einsatz kommen, der Speicherplatz reicht dann nicht aus.

Die Speicherung der Client-Software erfolgt am einfachsten mit dem zum Windows-NT-4.0-Server gehörenden *Netzwerk-Client-Manager*. Wichtig ist hier, dass als Protokoll TCP/IP mit Adresszuweisung über DHCP eingestellt wird, der Typ der Netzwerkkarte ist in den meisten Fällen zunächst unerheblich, da der tatsächlich eingesetzte Netzwerkkartentyp oft (noch) nicht in der Auswahlliste vorhanden ist. Die gesamte Software paßt auf eine Diskette, deren Inhaltsverzeichnis dann so aussieht:

Abbildung 3-45 Inhaltsverzeichnis der DOS-NetBIOS-Client-Diskette

In der Datei *autoexec.bat* wird der NetBIOS-Client gestartet. Bei Bedarf können hier weitere Aktionen, z.B. das Verbinden von Netzwerklaufwerken, Kopieraktioen und das Laden vollständiger Festplattenimages eingetragen werden.

Das Verzeichnis *a:\net* enthält die benötigten Netzwerkkartentreiber und die Konfigurationsdateien. Der NetBIOS-Client muss für die im Netzwerkarbeitsplatz eingesetzte Netzwerkkarte konfiguriert werden, im obigen Beispiel wird der Treiber *EL90X.DOS* für eine 3COM-Karte vom Typ 3C905 verwendet. Um diese Boot-Diskette an die eigenen Erfordernisse anzupassen, sollte also zunächst der richtige Trieber beschafft und in *a:\net* gespeichert werden.

Alle weiteren Einstellungen erfolgen dann in den zwei Konfigurationsdateien (Listings 3-1 und 3-2).

```
[network.setup]
version=0x3110
netcard=ms$el90x,1,MS$EL90X,1
transport=tcpip,TCPIP
lana0=ms$el90x,1,tcpip

[ms$el90x]
drivername=EL90X$
; INTERRUPT=3
; IOBASE=0x300
; SlotNumber=1

[protman]
drivername=PROTMAN$
PRIORITY=MS$NDISHLP

[tcpip]
NBSessions=6
DefaultGateway0=
SubNetMask0=
IPAddress0=
DisableDHCP=0
DriverName=TCPIP$
BINDINGS=ms$el90x
LANABASE=0
```

Listing 3-1 PROTOCOL.INI

In *a.\net\PROTCOL.INI* (Listing 3-1) muss der Name des tatsächlich verwendeten Netzwerkkartentreibers an den grau hinterlegten Stellen eingetragen werden, weitere Anpassungen sind in der Regel nicht notwendig, wenn mit DHCP gearbeitet werden kann.

```
 [network]
filesharing=no
printsharing=no
autologon=yes
computername=DOS_WS01
lanroot=A:\NET
username=Administrator
workgroup=IBSAMULAT
reconnect=no
dospophotkey=N
lmlogon=0
logondomain=IBSAMULAT
preferredredir=full
autostart=full
maxconnections=8

[network drivers]
netcard=el90x.dos
transport=tcpdrv.dos,nemm.dos
devdir=A:\NET
LoadRMDrivers=yes

[Password Lists]
*Shares=a:\net\ShareOOO.PWL
ADMINISTRATOR=A:\NET\ADMINIST.PWL
```

Listing 3-2 SYSTEM.INI

In *a.\net\SYSTEM.INI (Listing 3-2)* wird der Name des Netzwerkartentreibers nur an einer Stelle eingetragen (grau hinterlegt). Die fett gedruckten Zeilen sollten dann die für das eigene Netzwerk zutreffenden Angaben für Arbeitsgruppennamen etc. enthalten und müssen entsprechend angepasst werden. Wird mit mehreren Bootdisketten gleichzeitig gearbeitet, muss der NetBIOS-Name des Clients immer eindeutig bleiben. Der Inhalt von *computername* muss für jede Diskette eindeutig vorgegeben werden.

Die Datei *autoexec.bat* und die Konfigurationsdateien aus dem oben gezeigten Konfigurationsbeispiel finden Sie auf der CD-ROM im Vereichnis */util/dos_client*.

Windows 9x/NT

Netzwerkarbeitsplätze unter Windows9x/NT benötigen keine Zusatzsoftware. Der TCP/IP-Protokollstack ist im Lieferumfang enthalten. Die IP-Adresszuweisung sollte grundsätzlich über DHCP erfolgen.

Nach dem Start des Dienstes *smbd* sollte der SAMBA-Server und seine Freigaben in der Netzwerkumgebung zu sehen sein (Abbildung 3-46).

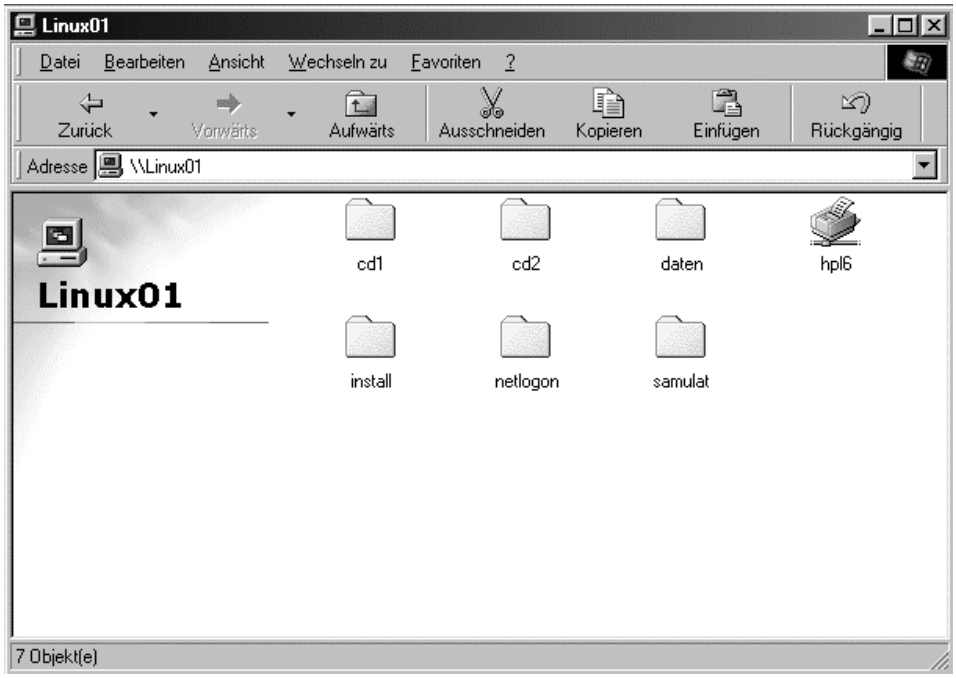


Abbildung 3-46 Freigaben des SAMBA-Servers Linux01 in der Netzwerkumgebung von Windows 98

In der grafischen Bedieneroberfläche können die Ressourcen sofort in gewohnter Weise verwendet werden. Auch die *net*-Befehle stehen weiterhin zur Verfügung, z. B. zur Erstellung von Benutzer-Anmeldeskripten.

OS/2

Mit der Komponente *Networking* bietet *OS/2 Warp Connect* eine den Microsoft-Bestriebssystemen vergleichbare Funktionalität. Für den Zugriff auf den SAMBA-Server werden die Dienste

- Multiprotocol Transport Services MPTS,
- IBM TCP/IP Version 3.0 for OS/2 und
- IBM LAN Requester

installiert. Sollen zusätzlich auch Verzeichnisse und Drucker des OS/2-Rechners im Netzwerk bereitgestellt werden, wird zusätzlich der *IBM Peer for OS/2* benötigt.

Der Dienst *IBM Peer for OS/2* sollte immer als letzter installiert werden. Weitere Details können bei Bedarf der mit OS/2 gelieferten Online-Dokumentation entnommen werden.

Zugriff auf einem SAMBA-Server in einem fremden Teilnetz

Wenn ein Windows-9x/NT-Client den SAMBA-Server über seinen NetBIOS-Namen anspricht, muss bekannt sein, welche IP-Adresse zu diesem Namen gehört. Die Namensauflösung erfolgt standardmäßig über NetBIOS-Broadcasts, die aber nicht über Router oder Gateways weitergeleitet werden. Damit ist ein SAMBA-Server in einem anderen Teilnetz für den Windows-Client unsichtbar.

Um trotzdem den Zugriff zu ermöglichen, ist eine Tabelle notwendig, die für den Client bei Bedarf die Namensauflösung ermöglicht. Diese Aufgabe übernimmt die Datei *lmhosts*, die ähnlich wie *hosts* zeilenweise IP-Adresse und Namen zuordnet. Die Namen sind in diesem Falle aber NetBIOS-Namen der Rechner.

Der Aufbau der Datei *lmhosts* ist damit sehr einfach: In jeder Zeile steht ein Eintrag, dieser beginnt jeweils mit einer IP-Adresse (oder Internet-Namen). Getrennt durch ein Leerzeichen folgt des NetBIOS-Name:

```
# Beispiel fuer die NetBIOS-Namensaufloesung mit lmhosts
Linux01.samulat.de linux01
192.168.100.5 linux05
```

SAMBA als WINS-Server

Um die NetBIOS-Namen anderer Netzwerkrechner in der eigenen Netzwerkumgebung anzeigen zu können, führt jeder Windows-Rechner einen Computersuchdienst (Browser) aus. Sind mehrere Windows-Rechner im Netzwerk, so wird einer dieser Rechner zum Master-Browser. Welcher Rechner das ist, hängt vom Betriebssystem ab, entscheidend ist eine zum Betriebssystem gehörende Kennziffer, das *os level* mit einem Wertebereich von 1 bis 255 (Tabelle 3-14).

Betriebssystem	os level
Windows for Workgroups	1
Windows 9x	1
Windows NT Wokstation 3.51	16
Windows NT Workstation 4.0	17
Windows NT Server 3.51	32
Windows NT Server 4.0	33

Tabelle 3-14 NetBIOS-Kennziffer »os level«

Sind mehrere Rechner mit gleichem *os level* im Netzwerk, so konkurrieren diese um die Funktion Master-Browser, aktuell hat aber immer nur eine Maschine diesen Status. Diese Zuordnung wird bei Bedarf immer wieder über NetBIOS-Broadcasts »geklärt«, erzeugt also eine absolut unnötige Netzwerklast.

Beim Einsatz von SMB auf NetBEUI und TCP/IP spielt NetBIOS als Programmierschnittstelle und Namensdienst immer eine wichtige Rolle. Die RFCs 1001 und 1002 legen einen Namensdienst für »NetBIOS over TCP/IP« fest, der bei Microsoft als WINS realisiert wurd.

Die NetBIOS-Namensauflösung (Browser-List) erfolgt im einfachsten Fall über Broadcasts, die das Netzwerk mit zunehmender Rechneranzahl erheblich belasten. Wird ein Windows-Client für die Namensauflösung über WINS konfiguriert, meldet er sich beim Rechnerstart beim WINS-Server an. Der WINS-Server erhält so im Laufe der Zeit eine vollständige Tabelle der NetBIOS-Namen, der dort verfügbaren Dienste und der dazu gehörende IP-Adressen. Sucht ein Windows-Client einen bestimmten Rechner im Netzwerk, sendet er keinen Broadcast, sondern schickt die Anfrage direkt an den WINS-Server.

SAMBA kann bereits in der Standardkonfiguration als WINS-Server arbeiten, zusätzliche Software braucht nicht geladen zu werden. Für einen reibungslosen Betrieb sollte der WINS-Server auch gleichzeitig als Master-Browser der Arbeitsgruppe arbeiten. Mit den Einträgen

```
wins support = yes
wins proxy = yes
local master = yes
preferred master = yes
os level = 42
```

in */etc/smb.conf* wird der SAMBA-Server Master-Browser und als Windows-Name-Server WINS konfiguriert. Zusätzlich müssen auch die Windows-Clients so konfiguriert werden, dass sie auf diesen Dienst zugreifen können. Dies geschieht zweckmäßigerweise über die IP-Adresszuweisung über DHCP:

```
option netbios name-servers 192.168.100.1
```

Mit dem Parametern *local master* = *yes* und *preferred server* = *yes* gibt der Dienst *nmbd* bekannt, dass dieser SAMBA-Server bereit ist, im Netzwerk als Master-Browser zu arbeiten. Mit *os level* = *33* wird der SAMBA-Server dann tatsächlich zum Master-Browser in diesem Netzwerk, zumindest solange, wie kein NT-4.0-Server aktiv ist. Mit einer noch höheren Priorität, z. B. *os level* = *42*, kann auch diese Konkurrenz abgehängt werden.

Wird der SAMBA-Server nicht als WINS-Server konfiguriert, so sollte der WINS-Dienst nicht gestartet werden. Ein anderer Windows-Rechner im Netzwerk sollte diese Aufgabe übernehmen, um unnötige Namensauflösungen über NetBIOS-Broadcasts in jedem Fall zu vermeiden. Mit

```
Wins support = no
wins server = 192.168.100.21
```

verwendet SAMBA den angegebenen Server.

SAMBA als CD-ROM-Server

Da im Linux-Dateisystem Laufwerke in ein Verzeichnis gemountet werden können, ist der SAMBA-Server auch dafür geeignet, im Netzwerk die Aufgaben eines CD-ROM-Servers zu übernehmen.

Eine typische Konfiguration soll z.B. aus sechs CD-ROM-Laufwerken bestehen; die dort enthaltenen CDs sollen allen Benutzern verfügbar gemacht werden. Auf dem Linux-Server wird zunächst ein Verzeichnis angelegt, ab dem später die CDs gemountet werden sollen (hier: */cdserver*). Das Verzeichnis wird mit ausreichenden Berechtigungen versehen. In der Konfigurationsdatei */etc/smb.conf* wird festgelegt, dass der SAMBA-Server dieses Verzeichnis »freigegeben« soll:

```
[cdserver]
   comment = CD-ROM Server
   path = /CD-ROM
   locking = no
   read only = yes
```

Soll jetzt der Systemverwalter *root* ein CD-ROM-Laufwerk unterhalb von */cdserver* mounten, so erfolgt dies jetzt zweckmäßigerweise über ein kleines Shell-Skript:

```
mkdir /CD-ROM/$1
mount -tiso9660 /dev/$2 /CD-ROM/$1 ro
echo .
echo Die aktuelle Konfiguration ist:
echo .
mount
```

Dieses Skript mit dem Namen *cdin* wird mit zwei Parametern (intern $1, $2) aufgerufen: Dem Namen des Verzeichnisses, in dem die CD-ROM gemountet werden soll (dieses Verzeichnis wird in */CD-ROM* neu erstellt), und dem Namen der Geärtedatei, über die das Laufwerk angesprochen werden kann. Die CD-ROM wird dann in dem neuen Verzeichnis gemountet, abschließend wird der aktuelle *Mount*-Status auf dem Bildschirm ausgegeben.

Damit sieht der Benutzer in der Freigabe *cdserver* des SAMBA-Servers eine Liste von Namen, die den Inhalt der dahinter liegenden CDs eindeutig angeben. Wechselt der Benutzer in ein »CD-Verzeichnis«, so kann er direkt auf den Inhalt der CD zugreifen. Die Orientierung, auch in großen CD-ROM-Servern, ist damit sehr einfach möglich.

Soll eine CD dismountet werden, so sollte dies auch wieder über ein einfaches Shell-Skript erfolgen:

```
umount /CD-ROM/$1
rmdir /CD-ROM/$1
echo .
echo Die aktuelle Konfiguration ist:
echo .
mount
```

Dieses Skript *cdout* benötigt nur einen Parameter: Angegeben wird der Name des Verzeichnisses in */cd-rom*, für das dismountet werden soll (intern $1). Diesmal wird als einziger Parameter nur der Name des Verzeichnisses angegeben, in dem die gewünschte CD gemountet ist. Die CD-ROM wird dismountet, das nicht mehr benötigte Verzeichnis wird gelöscht. Abschließend wird wieder der aktuelle Mount-Status auf dem Bildschirm ausgegeben.

Das nachfolgende Beispiel zeigt den vollständigen Arbeitsgang beim Mounten einer CD-ROM mit den Skripten *cdin* und *cdout*. Die CD-ROM */dev/sr1* soll im Verzeichnis */CD-ROM/telefon* gemountet werden:

```
# cdin telefon sr1
.
Die aktuelle Konfiguration ist:
.
/dev/sda3 on / type ext2 (rw,usrquota,grpquota)
proc on /proc type proc (rw)
/dev/sda1 on /boot type ext2 (rw)
none on /dev/pts type devpts (rw)
/dev/sr1 on /CD-ROM/telefon type iso9660 (ro)
```

Abschließend soll die CD-ROM wieder dismountet werden, die »Freigabe« wird damit beendet:

```
# cdout telefon
.
Die aktuelle Konfiguration ist:
.
/dev/sda3 on / type ext2 (rw,usrquota,grpquota)
proc on /proc type proc (rw)
/dev/sda1 on /boot type ext2 (rw)
none on /dev/pts type devpts (rw)
```

SAMBA-Ressourcen nutzen – smbclient

Mit dem zum SAMBA-Paket gehörenden Programm *smbclient* können Ressourcen beliebiger SMB-Server genutzt werden. Es ist damit auch möglich, von einem Linux- Server direkt auf Freigaben eines Windows-9x/NT-Servers zuzugreifen, wenn die dafür notwendigen Berechtigungen vorhanden sind. Die Angaben, mit welchem Server und mit welcher Ressource die Verbindung hergestellt werden soll, erfolgt in der schon beschriebenen UNC-Schreibweise, wobei die »\« in der Li-

nux-Shell jeweils doppelt geschrieben werden müssen. Um in jedem Fall das richtige Benutzerkonto zur Authentifizierung zu verwenden, sollte standardmäßig immer der Benutzername mit angegeben werden. Mit

```
# smbclient \\\\nt40ws\\temp -Usamulat
```

stellt *smbclient* eine Verbindung zum NT-Netzwerkarbeitsplatz *nt40sam* her, zugegriffen wird auf die Freigabe mit dem Namen *temp*. Während des Verbindungsaufbaus wird das Kennwort des angegebenen Benutzers *samulat* abgefragt.

Tabelle 3-15 stellt eine Auswahl der dann im *smbclient* möglichen Befehle dar, mit denen z.B. auch Dateien zwischen dem Linux-Rechner und dem SMB-Server kopiert werden können. Die Syntax erinnert an *ftp*, die Bedienung ist aber einfach und schnell erlernbar. Zur Strukturierung der Befehlsübersicht werden die Begriffe SMB-Server und SMB-Client verwendet, sie stehen für

- Rechner, auf dem der SAMBA-Server-Dienst läuft:(SMB-Server),
- Rechner, auf dem *smbclient* ausgeführt wird:(SMB-Client).

Kommando	Beschreibung
? <befehl>	Ist <*befehl*> angegeben, so wird eine Beschreibung des Kommandos ausgegeben. Ohne Parameterangabe wird eine Liste der möglichen Befehle angezeigt.
cd <verzeichnis>	wechselt in das Verzeichnis <*verzeichnis*> auf dem SMB-Server. Wird *cd* ohne Parameter aufgerufen, so wird das aktuelle Arbeitsverzeichnis angezeigt.
del <dateiname>	Der SMB-Server soll die angegebene Datei <*dateiname*> löschen. Die Platzhalter »*« und »?« können verwendet werden.
dir <maske>	Der SMB-Server zeigt den Inhalt des aktuellen Arbeitsverzeichnisses, bei Bedarf kann die Anzeige mit <*mask*> eingeschränkt werden. Auch hier können Platzhalter verwendet werden.
exit	beendet dieVerbindung zum SMB-Server, das Programm *smbclient* wird verlassen.
get <quelle> <ziel>	kopiert die Datei <*quelle*> aus dem Arbeitsverzeichnis des SMB-Servers zum Rechner, auf dem *smbclient* ausgeführt wird. Ist <*ziel*> angegeben, so erhält die kopierte Datei den dort angegebenen Namen.
lcd <verzeichnis>	wechselt in das Verzeichnis <*verzeichnis*> auf dem SMB-Client. Ohne Parameterangabe wird das aktuelle Arbeitsverzeichnis auf dem SMB-Client angezeigt.
md <verzeichnis>	erstellt das Verzeichnis <*verzeichnis*> auf dem SMB-Server.

Tabelle 3-15 Befehlsübersicht smbclient

Kommando	Beschreibung
mget <maske>	kopiert alle Dateien, die zur angegebenen Maske <*maske*> passen, vom SMB-Server zum SMB-Client.
mput <maske>	kopiert alle Dateien, die zur angegebenen Maske <*maske*> passen, vom SMB-Client zum SMB-Server
print <dateiname>	Die SMB-Client-Datei <*dateiname*> wird vom SMB-Server ausgedruckt.
put <quelle> <ziel>	kopiert die SMB-Client-Datei <*quelle*> zum SMB-Server. Ist <*ziel*> angegeben, so erhält die kopierte Datei auf dem SMB-Server den hier angegebenen Namen.
rm <maske>	löscht alle Dateien auf dem SMB-Server, die zur angegebenen <*maske*> passen.
rmdir <verzeichnis>	löscht auf dem SMB-Server das Verzeichnis mit dem Namen <*verzeichnis*>.

Tabelle 3-15 Befehlsübersicht smbclient

Zweite SAMBA-Konfiguration: Windows NT 4.0 Domänencontroller PDC

Das Programmpaket SAMBA ermöglicht die Emulation eines NT 4.0 *Primary Domain Controller* PDC, wobei in den aktuellen Versionen die wesentlichen PDC-Dienste vollständig und stabil implementiert sind.

Auch wenn damit der SAMBA-Server als PDC noch nicht perfekt ist, weil z.B. die NT-Print-Server-Funktion noch nicht fertiggestellt wurde, so kann bereits jetzt die Rolle des zentralen Anmeldeservers übernommen werden. Insbesondere unter Kostengesichtspunkten ist dies oft ein wichtiges Argument, wenn es um die Einsatzmöglichkeiten von Linux im kommerziellen Bereich geht.

Für das nachstende Konfigurationsbeispiel wird ein SAMBA-Server vorausgesetzt, der bereits als »Stand-Alone«-Server konfiguriert wurde und im Netzwerk erfolgreich betrieben werden konnte. Die dazu notwendigen Arbeitsschritte wurden bereits ausführlich beschreiben.

Grundkonfiguration in /etc/smb.conf

In */etc/smb.conf* sind nur wenige neue Parameter notwendig, um den SAMBA-Server als PDC der Domäne IBSAMULAT betreiben zu können:

```
[global]
   workgroup = IBSAMULAT
   domain logons = yes
   domain master = yes
```

Der bereits festgelegte Name der Arbeitsgruppe *workgroup = IBSAMULAT* ist hier auch der Name der Windows-NT-Domäne, kann also unverändert beibehalten werden. In der NT-Domäne werden alle Rechner in einer Vertrauensstellung orga-

nisiert, sie sind also jetzt mehr als nur eine »Gliederungsebene«, innerhalb derer dieser SAMBA-Server mit allen anderen dieser Arbeitsgruppe zugeordneten Windows-Rechnern zusammen angezeigt wird. Der Name der NT-Domäne sollte möglichst nicht mehr geändert werden.

```
[netlogon]
   path = /home/netlogon
```

erstellt eine Freigabe mit dem Namen *NETLOGON* , das auf dem lokalen Verzeichnis */home/netlogon* basiert. Ein als PDC konfigurierter SAMBA-Server kann hier Anmeldskripte, Benutzerprofile und Systemrichtlinien (Policies) zur Verfügung stellen.

Nach dem ersten Start von *smbd* als PDC wird die Datei *etc/MACHINE.SID* erstellt, sie enthält die *SID* des von SAMBA nachgebildeten Domänencontrollers und dient als »Kennwort« in der Einrichtung der Vertrauensstellung zu Netzwerk-Clients.

Rechnerkonto anlegen

Damit ein Netzwerk-Client in der NT-Domäne arbeiten kann, muss eine *Vertrauensstellung* zwischen dem PDC und dem Netzwerk-Client eingerichtet werden. Dies geschieht durch die Einrichtung eines vom PDC verwalteten Rechnerkontos, in dem die NetBIOS-Rechnernamen und die für jeden Rechner eindeutige SID zur beidseitigen Identifizierung verwendet werden. Die Arbeitsschritte dazu, am Beispiel der NT-4.0-Workstation *nt40sam*, im Einzelnen:

Zunächst wird das »Rechnerkonto« angelegt, d.h. für die Arbeitsstation *NT40WS* wird ein Linux-Benutzerkonto (Account) angelegt, als Benutzername wird der Rechnername mit einem nachgestellten »$« angegeben. Das Rechnerkonto erhält kein Kennwort:

```
# useradd nt40ws$
```

Die Datei */etc/passwd* sollte danach einen Eintrag der Form

```
nt40sam$:x:506:100:nt40 workstation:/home/nt40ws$:/bin/false
```

enthalten. Dieser Vorgang wird für alle Netzwerk-Clients wiederholt.

Zum Standard-Lieferumfang des Programmpaketes SAMBA gehört ein Shell-Skript, */usr/sbin/mksmbpasswd.sh*, mit dem aus der Systemdatei */etc/passwd* die SAMBA-Kennwortdatei */etc/smbpasswd* erstellt werden kann. Mit

```
# chmod +x mksmbpasswd.sh
```

wird das Skript zunächst einmal ausführbar gemacht, danach kann die SAMBA-Kennowortdatei mit

```
# cat /etc/passwd | mksmbpasswd.sh > /etc/smbpasswd
```

erstellt werden. Der vollständige Pfad zu */usr/sbin/mksmbpasswd.sh* muss bei Bedarf natürlich mit angegeben werden. In *etc/smbpasswd* muss der Eintrag für das Rechnerkonto jetzt noch auf den Typ »Workstation« eingestellt werden. Dies geschieht mit

```
# smbpasswd -m -a nt40ws$
```

wobei auch hier der Rechnername mit nachgestelltem »$« angegeben wird. In */etc/smbpasswd* sollte danach ein Eintrag der Form

```
nt40sam$:506:F8DA2163C81BB70FAAD3B435B51404EE:826B86F2F20A40922D662A03204E03C2:
[W          ]:LCT-38397DF7:nt40 workstation
```

zu finden sein. Die hexdezimalen Angaben repräsentieren Rechnerdetails bzw. enthalten das Rechnerkennwort. Wichtig ist, dass in den eckigen Klammern der Typ »W« angegeben ist.

Sollen für viele PC Netzwerkarbeitsplätze Rechnerkonten erstellt werden, bietet sich die Erstellung eines Shell-Skriptes an. Mit dem Skript

```
# Skript fuer Rechnerkonto (Gruppe: pcws)
adduser -g pcws -s /bin/false -c "nt40 workstation" "$1\$"
smbpasswd -m -a "$1\$"
```

kann dieses unter */bin/bash* automatisiert werden. Das Skript wird dazu einfach mit dem Rechnernamen aufgerufen, das »$« wird automatisch angehängt. Das Rechnerkonto gehört dann der Gruppe *pcws* an, die vor dem ersten Start des Skriptes bereits existieren sollte. Wird das Skript unter dem Name *createpc.sh* gespeichert und ausführbar gemacht, so kann mit

```
# ./create.sh nt40ws
```

das Rechnerkonto für NT40WS automatisch erstellt werden.

Meldet sich der Rechner NT40WS jetzt in der Domäne an, wir standardmäßig als Kennwort zunächst der Rechnername in Kleinbuchstaben verwendet, nach der ersten erfolgreichen Anmeldung wird dann ein zufällig bestimmtes Kennwort verwendet.

Benutzerkonto anlegen

Soll ein Benutzer zur Domäne hinzugefügt werden, so wird zunächst ein Benutzerkonto (immer mit Kennwort) auf dem Linux-Server angelegt. Zusätzlich muss jeweils noch der entsprechende Eintrag in */etc/smbpasswd* mit

```
smbpasswd -a <benutzername>
```

erstellt werden. Das Kennwort muss hierbei nochmals angegeben werden. Für den neuen Benutzer wird dann noch sein privates Verzeichnis angelegt und mit ausreichenden Berechtigungen versehen:

```
chown <benutzername> users <verzeichnisname>
```

Kennworte unter Linux und Windows synchronisieren

Im Abschnitt [Global] der Konfigurationsdatei */etc/smb.conf* wird ergänzt:

```
unix password sync = yes
passwd program = /usr/bin/passwd %u
passwd chat = *password* %n\n *password* %n\n *successfull*
```

Das Programm *passwd* muss dabei im angegeben Pfad liegen, die Angabe der Parameter ist lediglich ein Vorschlag und kann bei Bedarf variiert werden. Eine Auswahl der möglichen Parameter zeigt Tabelle 3-16.

Variable	Inhalt
%o	altes Kennwort
%n	neues Kennwort
\n	neue Zeile
\t	Tabulator

Tabelle 3-16 Parameter für passwd

NT-Workstation in der Domäne anmelden

Die Anmeldung der NT-4.0-Workstation, in diesem Beispiel mit dem Namen *nt40sam*, erfolgt über *Netzwerkumgebung -> Eigenschaften*. Nur der Name der Domäne muss angegeben werden, ein Rechnerkonto muss nicht erstellt werden (Abbildung 3-47).

Nach der Meldung *Willkommen in der Domäne IBSAMULAT* und dem abschließenden Neustart der Workstation sollte es möglich sein, sich als Benutzer in der Domäne anzumelden.

Anmeldeskript (Login Script)

Mit einem Anmeldeskript *(Login Script)* werden Details der Benutzerumgebung spezifiziert, z.B. die Verwendung von Netzwerk-Ressourcen über logische Laufwerke. Um während jeder Benutzeranmeldung ein Anmeldskript automatisch abzuarbeiten, ist der Eintrag

```
[global]
   logon script = start.bat
```

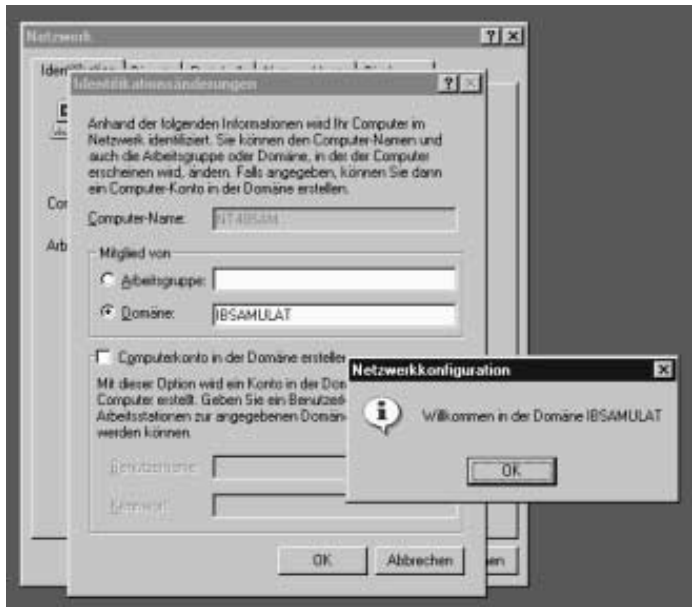


Abbildung 3-47 Domänenanmeldung

in */etc/smb.conf* vorzunehmen. Mit *logon script = start.bat* wird der Name des Anmeldeskripts festgelegt, das während jeder Benutzeranmeldung aus der Freigabe *NETLOGON* des PDC geladen wird. Die Datei *start.cmd* ist im Verzeichnis */home/netlogon* des Linux-Servers abzuspeichern. Mit

```
# chmod 644 start.bat
```

muss abschließend die Leseberechtigung für jeden Benutzer erteilt werden. Soll für jeden User ein eigenes Anmeldeskript ausgeführt werden, so wird dies über die Verwendung der Variablen %U in */etc/smb.conf* realisiert.

Das nachfolgende Beispiel zeigt ein kleines Anmeldeskript, dass die Zeitsynchronisierung durchführt und jedem Benutzer ein »privates« Laufwerk P: und ein Laufwerk M: für den gemeinsamen Zugriff auf Datenbestände zuweist:

```
REM Anmdeldeskript start.bat
REM
net time /s /y

net use P: \\linux01\%USERNAME% /PERSISTENT:NO
net use M: \\linux01\daten  /PERSISTENT:NO
```

Servergespeicherte Benutzerprofile

Benutzerprofile speichern Details der Umgebung, in der der aktuell angemeldete Benutzer auf der NT-4.0-Workstation arbeitet. Dazu gehören z.B. Details der

farblichen Gestaltung des Desktops, Ordner, Verknüpfungen und die aktuellen Netzwerkverbindungen. Auch viele Programmpakete, wie z.B. Microsoft Office, speichern benutzerspezifische Parameter innerhalb dieser Profile.

Es kann zweckmäßig sein, dem Benutzer bei seiner Anmeldung ein eigenes Profil zuzuweisen. Nahezu unabhängig von dem PC-Netzwerkarbeitsplatz, an dem diese Anmeldung erfolgt, erhält er immer seine gewohnte Arbeitsumgebung.

Voraussetzung dafür ist es, dass ein beliebiger Server im Netzwerk diese Profile zentral speichert. Auf dem SAMBA-Server *linux01* soll hierzu die Freigabe *profil* erstellt werden, basierend auf dem Verzeichnis */home/profil*. Alle Benutzer speichern dort ihre Profile, dazu sind natürlich die entsprechenden Linux-Berechtigungen einzustellen.

```
[global]
   logon path = \\linux01\profile\%U\profil

[profil]
   comment = Benutzerprofile
   path = /home/profil
   browseable = yes
   read only = no
   create mode = 0700
```

Nach dem Neustart des Dienstes *smbd* werden bei der nächsten Benutzeranmeldung automatisch servergespeicherte Profile erstellt und dann für alle weiteren Anmeldungen verwendet.

Nach der ersten Anmeldung des Benutzers *samulat* erstellt der SAMBA-Server einen neuen Ordner mit gleichem Namen in */home/profil*. Darin enthalten ist nur der zunächst leere Ordner *profil*. Nach der nächsten Abmeldung des Benutzers wird dort die typische NT-Benutzerprofilstruktur automatisch erstellt und dann für alle weiteren Anmeldungen verwendet.

Systemrichtlinien (Policies)

Auch Systemrichtlinien (*Policies*) werden vom Anmeldeserver in der Standardfreigabe NETLOGON bereitgestellt. Da das Programmpaket SAMBA leider über kein Werkzeug zur Erstellung von Policies verfügt, muss die Datei *NTConfig.POL* über den *Systemrichtlinien-Editor* eines Microsoft-Domänencontrollers erstellt werden. Die fertige Datei kann dann ohne Einschränkungen auch in NETLOGON des SAMBA-PDC bereitgestellt werden.

Tipps zur Fehlersuche

Ist es nicht möglich, die Workstation in der SAMBA-Domäne anzumelden oder schlägt die Benutzeranmeldung fehl, so sollte zunächst das *debug level* von *smbd* erhöht werden, um zusätzliche Informationen zu erhalten.

Mit *tcpdump* sollte es möglich sein, ein NETLOGON zu sehen (SAMLOGON auf UDP Port 138). Ist dies nicht der Fall, fehlt vielleicht der Parameter *domain logons = yes* in */etc/smb.conf*.

Über Port 139 sollte ein LSA_OPEN_POLICY und zwei LSA_QUERY_INFO sichtbar sein (einer für SID S-1-3-..., einer für S-1-5-...).

SAMBA-Administration

Zur Konfiguration und Administration des SAMBA-Servers gibt es eine ganze Reihe von grafischen Oberflächen, die diese Arbeiten erleichtern sollen. Eine Auswahl soll nachfolgend dargestellt werden.

ksamba

Das KDE-Programm *ksamba* setzt direkt auf der Konfigurationsdatei */etc/samba.conf* auf und zeigt die aktuelle Konfiguration an. *ksamba* ist damit weniger ein Werkzeug zur Aministration eines laufenden SAMBA-Servers, es ist spezialisiert auf die Grundkonfiguration des SAMBA-Systems.

Nach dem Start von *ksamba* und der Auswahl der Konfigurationsdatei werden die aktuell definierten Freigaben angezeigt (Abbildung 3-48).

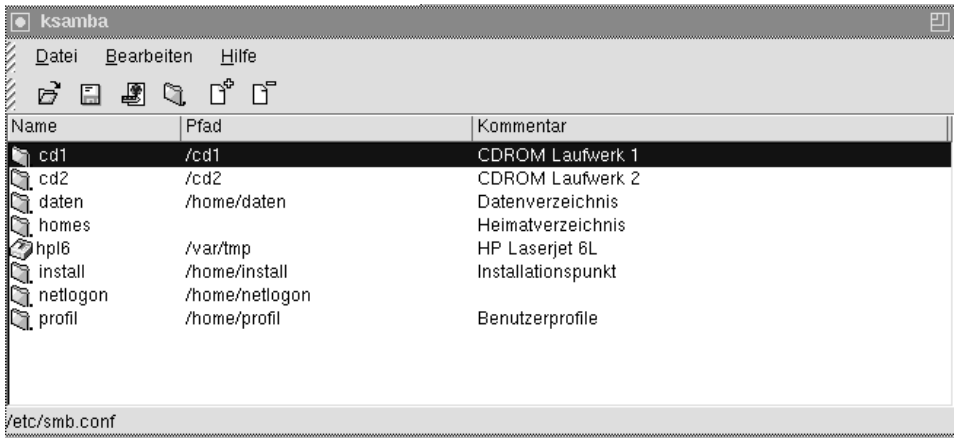


Abbildung 3-48 Bearbeitung von /etc/smb.conf mit ksamba

Hie kann jetzt ein neuer Eintrag erstellt oder ein bestehender gelöscht werden. Per Doppelklick wird ein Eintrag zur Bearbeitung ausgewählt (Abbildung 3-49).

In den »Karten« *Main, Misc, Name Mangling* und *File/Dir Mask* können alle Parameter dieser Freigabe bearbeitet und in */etc/smb.conf* gespeichert werden. Für die Bearbeitung von Details zu Druckerfreigaben steht eine ähnliche Form zur Verfügung.

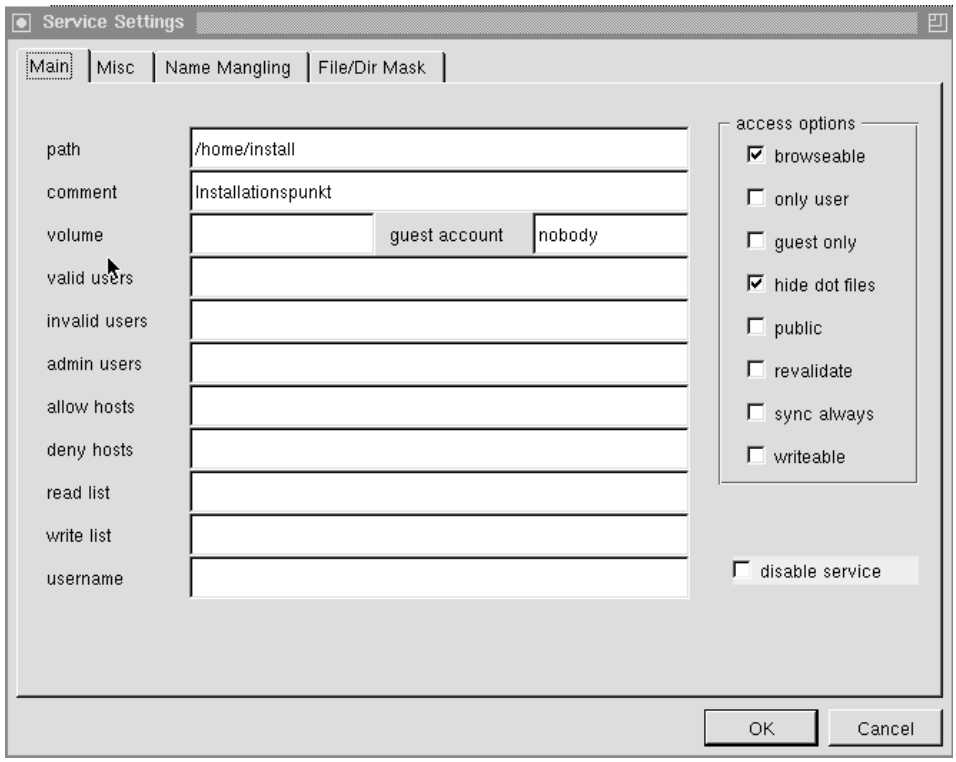


Abbildung 3-49 Freigabe bearbeiten

Aus der Startform heraus kann auch eine Funktion gestartet werden, mit der die im Abschnitt [global] definierten Parameter bearbeitet werden können. Interessant ist hier vor allem, dass *ksamba* alle möglichen Konfigurationsoptionen zeigt, die dann per Klick ausgewählt werden können (Abbildung 3-50).

Auch hier stehen wieder insgesamt vier Konfigurationsseiten zur Verfügung.

swat

Ein weiteres Werkzeug zur Konfiguration und bedingt zur Administration eines SAMBA-Servers ist das Programm *swat,* das zum SAMBA-Programmpaket gehört. swat ermöglicht es, die Datei */etc/smb.conf* über einen beliebigen WWW-Browser zu bearbeiten. Zusätzlich bietet swat interessante Links zu Dokumentationen an, die bei der Konfiguration sehr hilfreich sein können.

Beachtet werden sollte, dass *swat* die Datei */etc/smb.conf* reorganisiert. Die Einträge werden nach den Vorgaben von swat neu sortiert. Alle Kommentare, sowie *include* = und *copy* = Optionen werden entfernt! Vor dem erstmaligen Aufruf sollte daher immer eine Sicherung der Konfigurationsdatei angelegt werden.

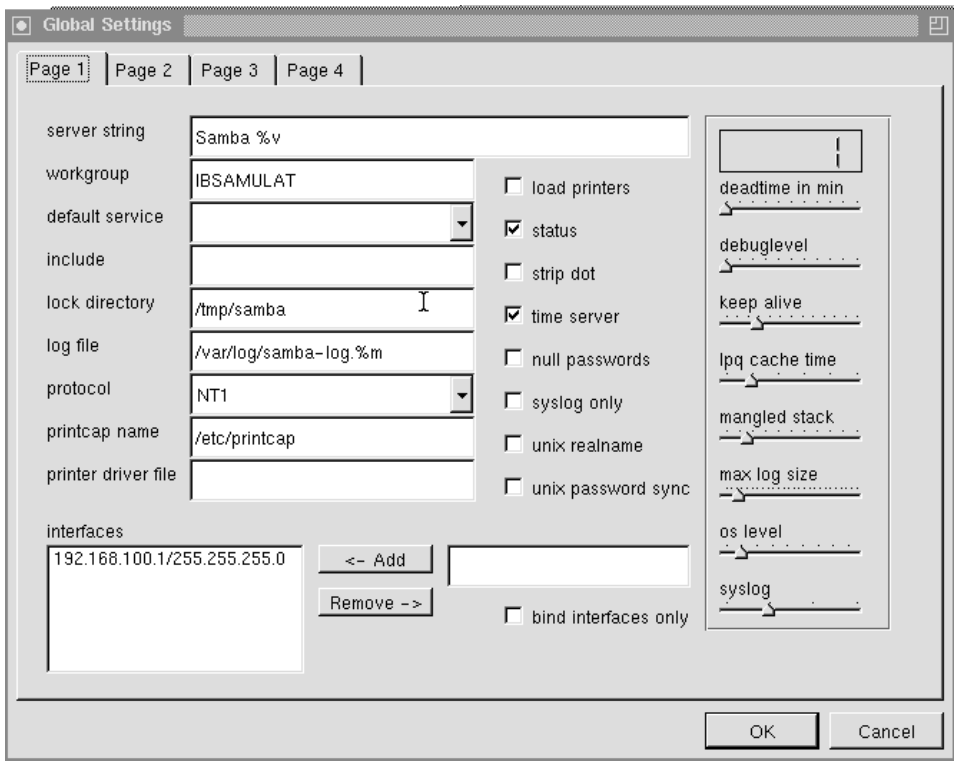


Abbildung 3-50 Parameter im Abschnitt [global] bearbeiten

Der Start vom *swat*-Daemon erfolgt automatisch bei jedem Systemstart, wenn */etc/rc.config* den nachfolgenden Eintrag enthält:

```
swat stream tcp nowait.400 root /usr/local/samba/bin/swat swat
```

Um *swat* zu verwenden, muss man dann lediglich mit einem beliebigen WWW-Browser auf die Portnummer 901 des gewünschten SAMBA-Servers zugreifen (Abbildung 3-51).

Die Bedieneroberfläche von *swat* ermöglicht nicht nur den Zugriff auf die Konfigurationsdaten (hier auch wieder getrennt nach Abschnitten), sondern auch die Anzeige von aktuellen Statusinformationen des SAMBA-Servers. Unter *Documentation* werden nahezu alle wichtigen Informationen zu Konfiguration und Betrieb von SAMBA zur Verfügung gestellt, zusätzlich werden die wichtigsten Hilfsprogramme vorgestellt (Abbildung 3-52).

Im oben gezeigten Beispiel der Konfiguration im Abschnitt [global] wird auch deutlich, dass *swat* für jeden einzelnen Parameter wiederum einen Hilfetext anbietet (Abbildung 3-53).

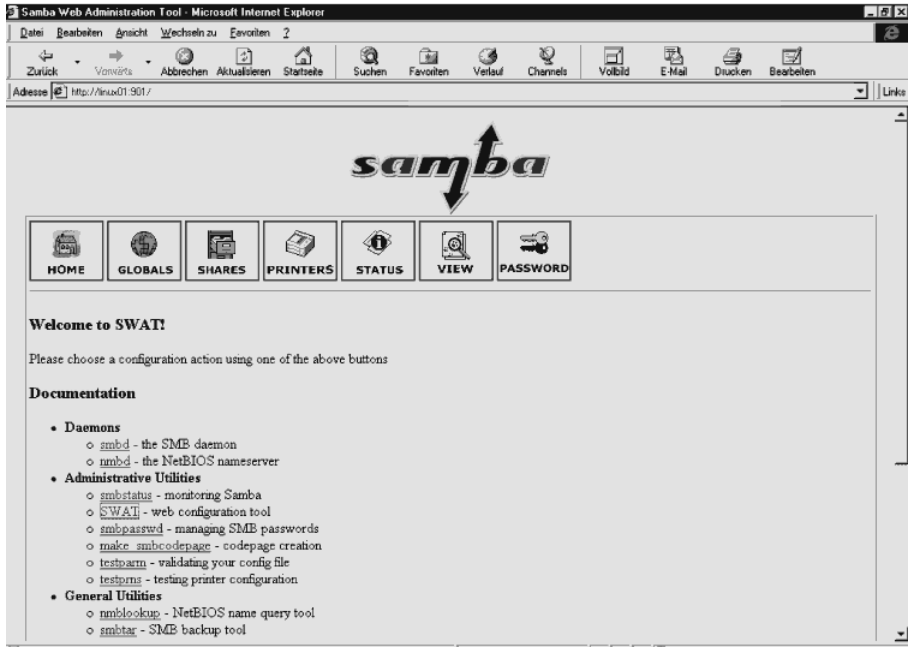


Abbildung 3-51 Administrationstool für SAMBA: swat

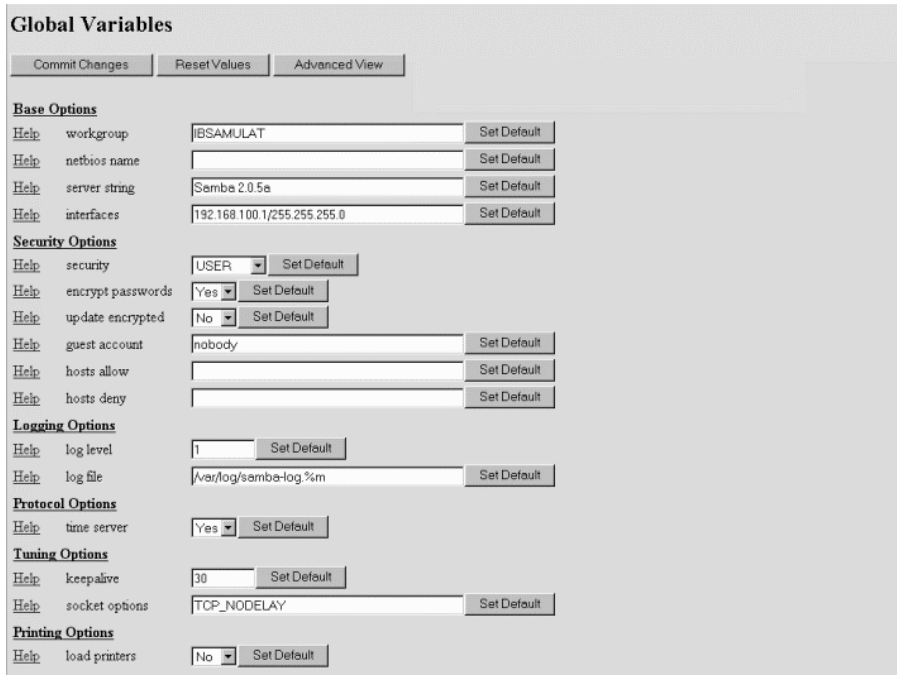


Abbildung 3-52 Beispiel: Parameterbearbeitung im Abschnitt [global]

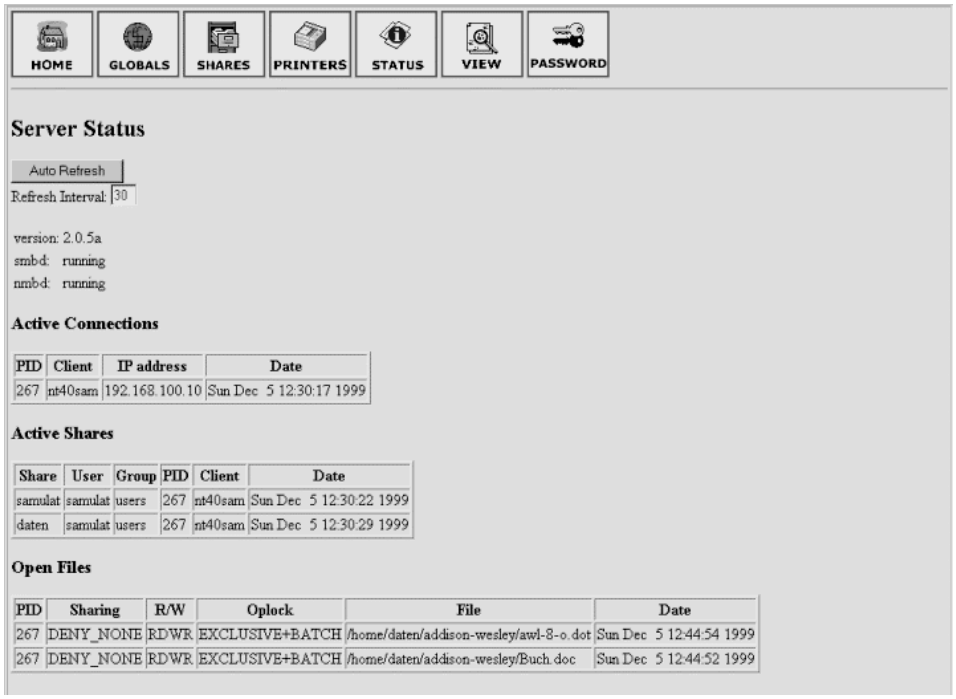


Abbildung 3-53 Anzeige von Statusinformationen

Interessant ist die Möglichkeit, über *swat* Informationen über den aktuellen Serverstatus abrufen und darstellen zu können. So kann schnell herausgefunden werden, welche Freigaben von welchen Benutzern verwendet werden (*Active Shares*) bzw. welche Dateien gerade verwendet werden (*Open Files*).

webmin

Die Konfiguration und Administration eines SAMBA-Servers kann über das schon mehrfach vorgestellte zentrale Administrationstool *webmin* erfolgen, verwendet wird das Modul *Samba Windows File Sharing*.

Verwendet werden sollte *webmin* in der Version 0.74 oder höher, ältere Versionen können Fehler beim Zugriff auf den SAMBA-Server zeigen (Versionsnummer aus */usr/sbin/smbd*); (Abbildung 3-54).

Die Startform zeigt die aktuellen Freigaben, über Tasten können weitere Konfigurationsoptionen abgerufen werden. Wird in der Spalte *Share Name* ein Eintrag ausgewählt, so werden die Details dieser Freigabe in einer weiteren Form angezeigt und können bearbeitet werden (Abbildung 3-55).

webmin kann auch die aktuellen Verbindungen zeigen, aufgerufen wird dazu der Link *View All Connections* aus dem Startformular (Abbildung 3-56).

Abbildung 3-54 webmin, Modul Samaba Windows File Sharing

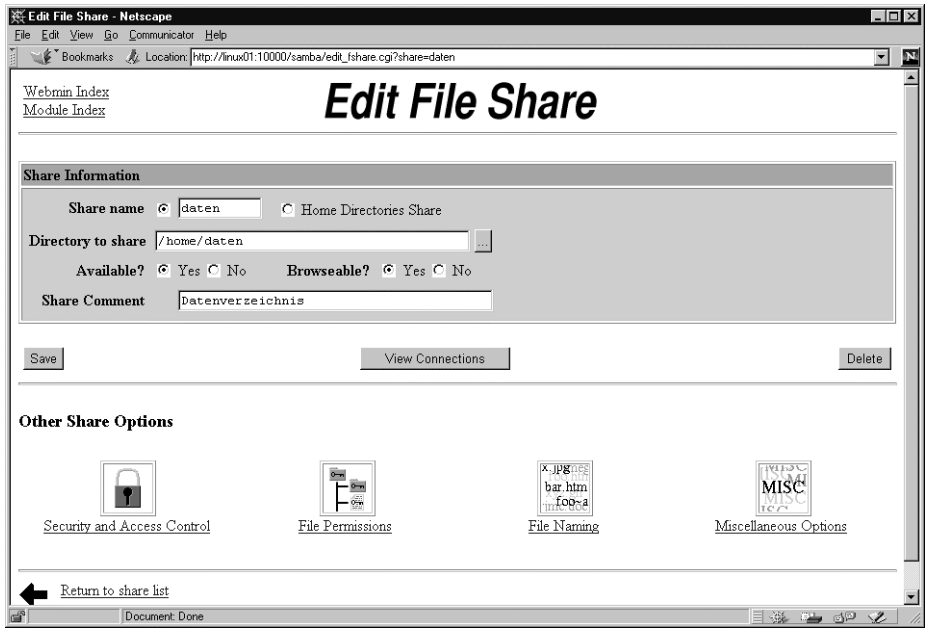


Abbildung 3-55 webmin, SAMBA Freigabe bearbeiten

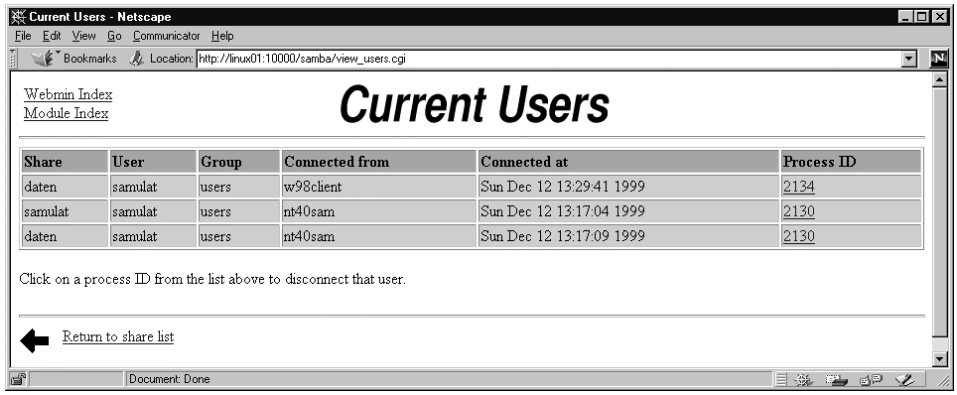


Abbildung 3-56 webmin, aktuelle Verbindungen zeigen

webmin bietet damit im Vergelich zu *swat* eine ähnliche Funktionalität, hier allerdings eingebettet als eines von vielen Modulen zur zentralen Systemverwaltung.

3.3.4 Netzwerk mit Linux-Clients

Bisher wurde fast immer davon ausgegangen, dass als Netzwerk-Clients Rechner unter Windows 9x/NT eingesetzt werden, der Linux-Rechner hatte nur die Funktion des Datei- und Druckservers.

Wie eingangs dargestellt, sprechen auch heute noch ein paar gewichtige Gründe gegen den Einsatz von Linux als Arbeitsplatz-Betriebssystem. Da aber auch hier der Preisdruck steigt, kann es sinnvoll sein, im Einzelfall Linux-Rechner einzusetzen. Auf den ersten Blick problematisch zu sein scheint dann der Zugriff auf die Ressource *Datei* eines Linux-Servers, unkritisch ist die bereits dargestellte gemeinsame Verwendung von Druckgeräten.

Die Bereitstellung von Datei- und Verzeichnisressourcen für Linux-Clients müsste sich zunächst an den technischen Möglichkeiten orientieren, die Unix-Systeme grundsätzlich zum Zugriff auf externe Datenbestände bieten. Möglich wäre dies z.B. schon über Programme wie ftp. Die Programmbedienung entspricht aber kaum noch den heute Anforderungen, sodass auf diese Möglichkeit hier nicht weiter eingegangen werden soll.

Praxisrelevanter sind Möglichkeiten, Verzeichnisse beliebiger Linux-Server in das eigene Dateiusystem zu *mounten*, um dann direkt auf diese Ressourcen zugreifen zu können.

NFS

Das *Network File System* NFS ist ein auf RPC basierender Netzwerkdienst, der es ermöglicht, auf Dateien entfernter Hosts so zuzugreifen, als wären sie im lokalen Fetsplattensystem gespeichert. Auch NFS ist ein »asymmetrischer« Dienst, es gibt

NFS-Server und NFS-Clients. Jeder Linux-Server kann dabei gleichzeitig beide Funktionen übernehmen, d.h. er kann Dateisysteme im Netzwerk bereitstellen (exportieren) und Dateisysteme anderer Rechner mounten (importieren).

NFS-Server

Auf einem NFS-Server müssen die Dienste

RPC-Portmapper (portmap),
RPC-Mount-Daemon (rpc.mountd)
RPC-NFS-Daemon (rpc.nfsd)

gestartet sein, bei Bedarf sollte dies mit *ps ax* überprüft werden. Wurde in der Datei /etc/*rc.config* der Eintrag

```
NFS_SERVER="YES"
```

vorgenommen, wird der NFS-Server automatisch bei jedem Systemstart mit gestartet. Jetzt muss nur noch festgelegt werden, welche Dateissysteme an welche Rechner exportiert werden sollen. Dies geschieht in der Konfigurationsdatei */etc/exports*, dazu ein Beispiel:

```
#
# /etc/exports
#
/home/install          linux06(rw) linux07(ro)
/home/netlogon         (ro)
```

In obigen Beispiel werden zwei Verzeichnisse per NFS exportiert. Auf */home/install* dürfen nur die beiden hier angegebenen Rechner zugreifen, der Host *linux06* hat Lese- und Schreibrechte, *linux07* nur Leserechte. Auf */home/netlogon* dürfen alle Hosts zugreifen, allerdings nur lesend.

Die Datei */etc/exports* wird von *mountd* und *nfsd* ausgewertet. Nach jeder Änderung dieser Konfigurationsdatei müssen dieses Daemons neu gestartet werden, am einfachsten erfolgt dies mit

```
# rcnfsserver restart
```

NFS-Client

Einzige Voraussetzung auf dem NFS-Client ist, dass der RPC-Portmapper gestartet ist. Ist dies der Fall, können fremde Dateisysteme, die von einem NFS-Server bereitgestellt werden, mit dem Befehl *mount* in das lokale Dateisystem eingebunden werden. Mit

```
# mount -t nfs linux01:/home/daten /nfs
```

wird das Verzeichnis */home/daten* des NFS-Servers *linux01* in das lokale Verzeichnis */nfs* gemountet.

SAMBA-Client

Läuft auf dem Linux-Server, auf den zugegriffen werden soll, der SAMBA-Dienst, so kann auch ein SMB-Client für den Zugriff auf dort gespeicherte Dateien verwendet werden.

Interessant ist dabei, dass die Dateitransfers deutlich schneller als unter NFS möglich sind, sodass dieser Art des Zugriffs auf Ressourcen des Linux-Servers sogar der Vorzug gegeben werden sollte.

smbmount

Der Befehl *smbmount* ermöglicht es, eine SAMBA-Freigabe direkt in ein lokales Verzeichnis zu mounten. Die Syntax ist (Auszug):

```
smbmount //server/share mountpoint [options ...]
Version 2.0.5a
    -d debuglevel         set the debuglevel
    -n netbios name.      Use this name as my netbios name
    -N                    don't ask for a password
    -I dest IP            use this IP to connect to
    -E                    write messages to stderr
    -U username           set the network username
    -W workgroup          set the workgroup name
(...)
```

Mit *smbmount* können Freigaben von Windows-9x/NT-Rechnern und von SAMBA-Servern direkt von Netzwerk-Clients unter Linux verwendet werden, idealerweise erfolgt das tatsächliche Mounten dann über eine grafische Oberfläche wie *knetmon*.

knetmon

Das KDE-Programm *knetmon* ermöglicht den Zugriff auf SMB-Server im Netzwerk. Nach dem Start des Programms und der eventuell notwendigen Angabe der eigenen Arbeitsgruppe werden die aktuell verfügbaren Server angezeigt (Abbildung 3-57).

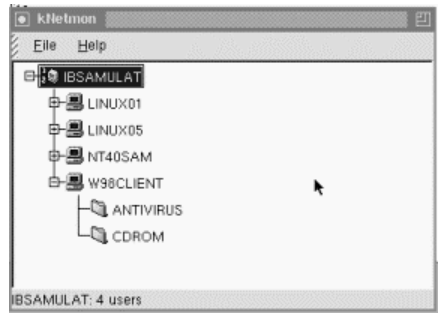


Abbildung 3-57 SMB-Ressourcen verwenden mit knetmon

Für jeden Server können die Freigaben dargestellt und bei Bedarf auch direkt gemountet werden. Damit ist ein sehr einfach zu konfigurierender Zugriff auf externe Dateiressourcen möglich. Über den Menüpunkt *File* können alle Details festgelegt werden, die für eine *Mount*-Aktion notwendig sind, z.B. auch der Anmeldename am entfernten Server.

3.4 Zeitserver

Die Synchronisation der Uhren aller Rechner im Netzwerk ist aus verschiedenen gründen zwingend notwendig. So werden die in den Protokollen dokumentierten Systemereignisse und Fehler mit der Angabe des Zeitpunktes gespeichert und der Systemadministrator benötigt eine homogene Systemzeit, um Ereignisse über mehrere Rechner hinweg verfolgen zu können. Auf jedem Rechner werden die Dateien mit Zeitstempeln versehen, um Versions- und Zustandskontrollen zu ermöglichen. Stimmen diese Zeiten nicht überein, weil die Uhren unterschiedliche Zeiten halten, so muss es zu Inkonsistenzen im Dateisystem kommen.

Im eigenen Netzwerk sollte ein Rechner als Zeitserver betrieben werden, alle weiteren Rechner synchronisieren dann ihre Uhren mit der Uhrzeit dieses Zeitservers.

Dieser Zeitserver braucht eine möglichst genaue Uhrzeit, der Wechsel von Sommer- auf Winterzeit und umgekehrt sollte automatisch erfolgen. Dazu gibt es verschiedene Möglichkeiten, den Zeitserver mit einer externen Referenz zu synchronisieren:

- im Internet erreichbarer Zeitserver
- ISDN-Zeitmarken
- am eigenen Zeitserver angeschlossene Empfänger für Funkuhren (DCF 77 oder GPS)

Zur Beurteilung der Qualität einer Uhrzeit wird ein Qualitätsmerkmal geführt, das *STRATUM*. Eine *STRATUM* von 1 haben alle Rechner, die ihre Zeit direkt von einer autorisierten Quelle wie z.B. einer Funkuhr beziehen. Rechner, die ihre Uhrzeit über einen im Internet erreichbaren Zeitserver synchronisieren, haben ein *STRATUM* von 2. Ein *STRATUM-3*-Rechner bezieht seine Zeit von einem *STRATUM-2*-Rechner und so weiter bis zu allen Rechnern, deren Uhrzeit überhaupt nicht synchronisiert ist (diese erhalten den maxmial möglichen Wert *STRATUM 16*).

3.4.1 Timeserver

Damit der Server im eigenen Netzwerk als Zeitserver (Timeserver) für Linux- und Windows-Clients arbeiten kann, sind in der Konfigurationsdatei */etc/inetd.conf* lediglich die vier Zeilen

```
daytime stream tcp nowait root internal
daytime dgram udp wait root internal
time stream tcp nowait root internal
time dgram udp wait root internal
```

durch Entfernen des Kommentarzeichens zu aktivieren. Nach dem nächsten Systemstart können die übrigen Rechner als *Timeclient* diesen Zeitserver nutzen.

Linux-Client

Mit dem Skript

```
/usr/sbin/netdate -v linux01
/sbin/clock -wu
```

kann ein beliebiger Linux-Rechner die Uhrzeit mit dem angegebenen Zeitserver synchronisieren. Der Zeitserver ist in diesem Beispiel der Host *linux01*. Nach dem Setzen der Systemzeit mit *netdate* ist mit *clock* auch die CMOS-Uhr entsprechend zu setzen.

Dem Befehl *netdate* kann aus Gründen der Fehlertoleranz auch eine Liste mit mehreren Zeitservern übergeben werden, aus denen dann je nach Verfügbarkeit und »Güte« der Zeitinformation der Zeiterver bestimmt wird. Welcher Zeitserver verwendet wird, hängt dabei nicht von der Reihenfolge in der Liste ab, sondern *netdate* prüft, welche Gruppe von Servern untereinander die geringste Zeitabweichung hat. Aus dieser Gruppe wird dann ein Zeitserver zur Synchronisation verwendet.

Soll *netdate* automatisch bei jedem Rechnerstart eines Linux-Clients ausgeführt werden, so sollte ein entsprechender Eintrag in */etc/rc.d/rc.lokal* erfolgen. Zusätzlich kann es sinnvoll sein, ein Shell-Skript (z. B. */sbin/timesync*) zu erstellen, sodass die Zeitsynchronisation zu jedem beliebigen Zeitpunkt vom Systemverwalter manuell ausgelöst werden kann.

Windows-Client

Rechner unter den Betriebssystemen Windows 9x und Windows NT synchronisieren ihre Uhren mit dem Befehl

```
net time \\linux01 /s /y
```

wobei der Name des Zeitservers gemäß UNC (*Universal Naming Convention*) hier mit zwei vorangestellten Backslashes eingetragen werden muss. Die Optionen */s* und */y* bewirken, dass die Rechneruhr ohne weitere Bestätigung eingestellt wird. Dieses Verfahren funktioniert auch für die automatische Umstellung auf Winter- und Sommerzeit, wobei der in den Eigenschaften der Windows-Systemuhr standardmäßig gesetzte Haken bei *Uhr automatisch von Sommer- auf Winterzeit umstellen* unbedingt entfernt werden muss.

3.4.2 Das Protokoll xntp

NTP v4 (*Network Time Protocol Distribution Version 4*) ist ein Client-Server-Protokoll zur Synchronisation der Uhren von Computersystemen mit fremden Zeitquellen. Der Daemon *xntp* gleicht die Zeit ständig mit einem Timserver ab und berichtigt die Systemzeit kontinuierlich in kleinen Schritten, um Inkonsistenzen zu vermeiden. Weicht die Systemzeit zu stark von der Zeitserverzeit ab, muss der Systemverwalter in der Regel manuell eingreifen.

Eine abgespeckte Version von *xntp* ist *SNTP (Simple Network Time Protokoll)*.

Neben dem eigentlichen Daemon *xntp* gibt es eine Reihe von Treibern für diverse Funkuhren und Modem-Ports, Verwaltungsprogramme und Konfigurationshilfen.

Installation und Start

Mit der Installation des Paketes *xntp* aus der Serie *n* (3,4 MByte), wird der *Network Time Protocol Daemon (Version 4)* auf dem Linux-Server eingerichtet. Soll *xntp* bei jedem Booten automatisch gestartet werden, ist in */etc/rc.config* der Eintrag

```
START_XNTP=YES
```

vorzunehmen. Der *xntp*-Daemon kann aber auch mit

```
rcxntpd start
```

manuell gestartet werden.

Konfiguration in /etc/ntp.conf

Die Haupt-Konfigurationsdatei für *xntp* ist */etc/ntp.conf*, daneben gibt es noch */etc/ntp.drift* zum Ausgleich der Gangungenauigkeiten der internen CMOS-Uhr und */etc/ntp.keys* für kryptographische Schlüssel zur Absicherung des eigenen Zeitserversystems vor unbefugten Zugriffen. Ein einfache Konfiguration (Minimalkonfiguration für einen `xntp`-Client) könnte z. B. so aussehen:

```
server linux01.samulat.de
driftfile /etc/ntp.drift
logfile /var/log/xntp
```

Die Zeile *Server* gibt an, welcher Rechner als Zeitreferenz dient. Hier kann ein im eigenen Netzwerk betriebener Host angebeben werden oder auch ein Verweis auf einen im Internet erreichbaren Zeitserver.

Das nächste Beispiel zeigt eine wesentlich ausführlichere */etc/ntp.conf* , die mit Fehlertoleranz und Zugriffsschutz für die Konfiguration eines eigenen Zeitservers ausreichend sein sollte:

```
# ntp.conf fuer eigenen Zeitserver
#
server linux01.samulat.de     # Hauptserver
server linux02.samulat.de     # Ersatz (1)
server linux03.samulat.de     # Ersatz (2)
server 127.127.1.0            # Lokale Uhr, falls alles
                              # andere ausfaellt
#
# Im Driftfile wird die errechnete Drift der Maschine
# gesichert, so dass diese nicht bei jedem Start neu berechnet
# werden muss. Monitor aktivieren.
#
driftfile /etc/ntp.drift
enable monitor
#
# Andere Maschinen duerfen die lokale Konfiguration
# nicht modifizieren
#
restrict default notrust nomodify
#
# Dieser Host duerfen die Zeit beziehen, aber nicht
# die Konfiguration aendern
#
restrict 192.80.x.y nomodify
#
# Konfigurationsaenderungen vom lokalen Host werden akzepiert
#
restrict 127.0.0.1
```

Der Daemon *xntpd* enthält ein leistungsfähiges Monitor-Modul, das in der obigen Konfigurationsdatei mit *enable monitor* aktiviert wird.

Betrieb und Test

Systemmeldungen von *xntp* werden in */var/log/messages* und in *var/log/ntp* ausgegeben, zusätzlich können detailliertere Informationen bis hin zu statistischen Auswertungen über das Monitor-Modul erstellt werden.

xntpd-Status abfragen – ntpq

Mit dem Shell-Befehl *ntpq* kann der Status eines laufenden *xntp*-Servers abgefragt und eingestellt werden. Wird *ntpq* ohne weitere Optionen aufgerufen, so wird *localhost* als *xntp*-Server angenommen, die Steuerung von *ntpq* erfolgt dann über die Kommandozeile:

```
linux01:/ # ntpq
ntqp>
```

Mit dem Kommando *pe* (*peers*) werden die von *xntpd* geführten Zeitquellen in einer Tabelle angezeigt, ein der Zeile vorangestelltes »*« zeigt an, dass dieser *peer* die aktuelle Zeitreferenz ist:

```
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
 GENERIC(0)      .DCFa.           0 u    -   64    0    0.000    0.000 4000.00
*LOCAL(0)        LOCAL(0)        10 l   14   64    1    0.000    0.000   0.000
```

Nach Eingabe von *q* (*quit*) wird *ntpq* beendet. Der Shell-Befehl ntpq kann auch mit Optinen gestartet werden. Die vollständige Syntax ist

```
ntpq [ -inp ] [ -c command ] [ host ] [ ... ]
```

wobei *host* der Zeitserver ist, zu dem *ntpq* eine Verbindung herstellen soll. Weitere Details, insbesondere auch Informationen zu den möglichen *commands*, können mit *man ntqp* abgefragt werden.

Linux-Rechner als ntp-Client – ntpdate

Der Shell-Befehl *ntpdate* stellt die Systemzeit eines Linux-Clients durch Aufruf eines *xntp*-Servers. Auch dieser Befehl kann nur vom Systemverwalter *root* ausgeführt werden. Auch *ntpdate* verfügt über eine Vielzahl von möglichen Optionen, die vollständige Syntax ist

```
ntpdate [ -bBdosu ] [ -a key ] [ -e authdelay ] [ -k keyfile ] [ -o version ] [
-p samples ] [ -t timeout ] server [ ... ]
```

ntpdate nutzt im Vergleich zu *xntp* einfachere Algorithmen zur Optimierung der Zeitsynchronisation, ermöglicht aber auch die fehlertolerante Verwendung von mehreren Zeitservern, aus denen dann die optimale Zeitquelle ermittelt wird. Die Genauigkeit, die *ntpdate* erreichen kann, ist unter anderem davon abhängig, dass der oder die Zeitserver in ausreichend kurzen Intervallen abgefragt werden. Daher sollte *ntpdate* nicht nur einmal bei Rechnerstart aufgerufen werden, sondern danach auch immer wieder in festen Zeitabständen, z.B. durch das später vorgestellte *cron*.

Windows-Rechner als ntp-Client

Windows-Clients können das in RFC2030 definierte *Simple Network Time Protocol* (SNTP) nutzen.

Für ein *xntp*-System ist die Genauigkeit der Systemuhr von Windows 9x und Windows NT zu gering. Intern arbeiten Windows 9x und Windows NT mit einer Genauigkeit von nur 10 ms. Damit die Fähigkeiten von *xntp* genutzt werden können, sollte die Auflösung der Systemuhr aber im Microsekundenbereich liegen. Mit SNTP steht ein Protokoll zur Verfügung, das zwar mit *xntp*-Servern zusammenarbeiten kann, aber nicht die Komplexität von *xntp* besitzt. SNTP und *xntp* Client-Server-Systeme arbeiten ohne Probleme miteinander, da das verwendete

Datenformat identisch ist. Dafür stellt das SNTP im Gegensatz zum *xntp* nicht so hohe Anforderungen an das Betriebssystem und eignet sich daher besonders zur Synchronisation von PC-Systemen mit Windows 9x und Windows NT. Während ein *xntp* Client/Server zwischen 4 und 6 MB RAM benötigt, kommt ein SNTP-Client/Server mit 0,5 bis 1 MB RAM aus.

Zur Einrichtung eines SNTP-Clients muss für die aktuellen Windows-Systeme zusätzliche Software beschafft und installiert werden:

Das nur im *NT Server Ressourcekit* enthaltene Programm *timeserver.exe* ermöglicht die Synchronisierung von NT-Servern und NT-Workstations über *xntp*. In der dazu gehörenden Konfigurationsdatei TIMESERV.INI können Namen und Adressen von *xntp*-Servern angegeben werden, wie der nachfolgende Auszug aus dieser Datei zeigt:

```
(...)
REM NTPServer is the name or numeric address of an NTP server
REM No default is given, since you should contact a timekeeper
REM (If you don't know what this is, you shouldn't try NTP)
REM (The names BroadcastClient and MulticastClient are reserved)

NTPServer=
(...)
```

Es ist hier sogar möglich, ähnlich wie beim Daemon *xntp*, direkt Funkuhren oder Modem-Ports als Zeitquelle zu verwenden.

Daneben steht eine Reihe von kommerziellen oder als Freeware erhältlichen SNTP-Clients für Windows 9x und Windows NT zur Verfügung. So z.B. das kommerzielle Paket LS-SNTP für Windows 95/98 und Windows NT der Firma Linum (Bezugsquelle: *www.linum.com*) oder der kostenlos nutzbare NTP-Client *Dimension 4* für Windows (auf der CD-ROM im Verzeichnis */tools/win9x/Dimension4*)

Windows 2000 enthält den neuen Dienst *W32Time*, der die vollständige Implementation des Simple Network Time Protocol (SNTP) nach RFC1769 darstellt. Das unter Windows 2000 zur Authentifizierung genutzte Protokoll (*MIT Kerberos version 5*) setzt auf diesen Dienst auf.

Novell-Server als ntp-Client

Auch Novell-Server können über *ntp* synchronisiert werden. Dazu gibt es als Freeware das Modul *RDATE.NLM*, das mit NTP umgehen kann. Mit

```
load rdate /p 360 /v 5 linux01
```

wird nach jeweils 360 Minuten die Uhrzeit beim Host *linux01* abgefragt. Weicht die Novell-Zeit mehr als 5 Sekunden ab, synchronisiert *rdate* die Uhrzeit des Novell-Servers.

3.4.3 Externe Zeitreferenzen nutzen

Zeitserver im Internet

Im Internet gibt es eine Reihe von öffentlich zugänglichen Rechnern, die mit Atomuhren verbunden sind, und deren Zeitinfomation zur Synchronisierung des eigenen Servers verwendet werden kann.

Die Verbindung zu einem Internet-Zeitserver sollte über ein Skript erfolgen, das mittels des später beschriebenen *cron* in größeren Zeitabständen automatisch abgearbeitet wird:

```
echo -n "time synchronisation " > /dev/tty1
/usr/sbin/nedate -v wrzx03.rz.uni-wuerzburg.de
/sbin/clock -wu
```

Der Befehl *netdate* fragt den öffentlichen Timeserver der Universität Würzburg ab und setzt das Systemdatum. Mit *clock -wu* wird auch die interne CMOS-Uhr auf die Systemzeit gesetzt.

Eine Liste der über das Internet erreichbaren Zeitserver *Public NTP Primary (stratum 1) Time Servers* kann z.B. unter *http://www.eecis.udel.edu/~mills/ntp/clock1.htm* abgerufen werden. Für jeden Server existiert dort ein Eintrag in der Form

```
ntp0.fau.de (131.188.34.75)
Location: University Erlangen-Nuernberg, D-91058 Erlangen, FRG
Geographic Coordinates: 49.573N 11.028E (from Meinberg GPS 166)
Synchronization: NTP V3 primary (GPS receiver (<<1us)), Sun SS12/Unix SunOS 5.6
Service Area: Germany/Europe
Access Policy: open access, pick one of ntp{0,1,2}.fau.de
Contact: The Timekeepers (time@informatik.uni-erlangen.de) Note: IP addresses
are subject to change; please use DNS
```

ISDN-Zeitmarken

Die Zeitsignale aus dem ISDN-Netz können auch für eine Uhrzeitsynchronisation verwendet werden. Voraussetzung ist eine vollständige und funktionierende ISDN-Konfiguration. Mit dem Befehl

```
isdnlog -t 1
```

wird bei der nächsten Einwahl in das ISDN-Netz die Systemzeit des Linux-Servers einmalig auf die von der Vermittlungsstelle gesendete Zeit gesetzt. Mit

```
isdnlog -t 2
```

wird die Systemzeit bei allen nachfolgenden Anrufen in das ISDN-Netz neu gesetzt.

Synchronisation über Funkuhr – DCF-77

Als Quelle der zur Uhrensynchronisation verwendeten Referenzzeit wird der Sender DCF-77 verwendet, den die Physikalisch-Technische Bundesanstalt in Mainflingen bei Frankfurt/Main betreibt. Das Langwellen-Signal des Senders DCF-77 ist in einem Umkreis von rund 2.000 km zu empfangen.

Low-Cost-Lösung

Grundsätzlich kann die Synchronisation über DCF-77 bereits mit geringem Hardwareaufwand realisiert werden. Für Linux sind eine Reihe von preiswerten (in der Regel unter 100 DM) DCF-77-Empfängermodulen verfügbar, die über serielle oder parallele Schnittstellen angeschlossen werden können. Als Software kann *xntp* eingesetzt werden. Nachteil dieser Lösungen ist meist die umständliche Ausrichtung der Empfangsantenne, wobei der optimale Standort oft erst nach langwierigen Tests ermittelt werden kann.

Eine preiswerte Lösung auf Basis einer externen DCF-77-Funkuhr wird z. B. von der Linum GmbH (*www.linum.com*) angeboten. Die *Sure-RPC-DCF-77-Funkuhr* ist ein kleines Empfangsmodul, das zweckmäßigerweise mit einem Verlängerungskabel von mindestens 7m Länge an einer seriellen Schnittstelle betrieben wird, um den optimalen Montageort möglichst weit vom Server entfernt finden zu können.

In dem nachfolgenden Konfigurationsbeispiel ist diese Funkuhr am seriellen Port COM2 angeschlossen.

Zunächst ist im Verzeichnis */dev* der symbolische Link */dev/refclock-0* zu erstellen. Mit

```
cd /dev
ln -s ttyS1 refclock-0
```

verweist der Link auf die zweite serielle Schnittstelle (*COM2*), bei Anschluss der DCF77-Uhr an *COM1* ist als Device *ttyS0* einzutragen.

Als Steuersoftware für die DCF-77-Uhr wird *xntp* verwendet. Dazu ist die Konfigurationsdatei */etc/ntp.conf* so zu bearbeiten, dass ein weiterer *peer* (127.127.8.0) als Zeitquelle zu Verfügung steht:

```
server 127.127.8.0 prefer mode 5  # DCF77 clock on serial port

server 127.127.1.0                # local clock (LCL)
fudge  127.127.1.0 stratum 10     # LCL is unsynchronized

0.0.0.0 mask 0.0.0.0 notrust      # no other synchronization

driftfile /etc/ntp.drift          # path for drift file
logfile   /var/log/ntp            # alternate log file
logconfig =syncstatus + sysevents # small log configuration
statsdir /var/log                 # place for statistics files
```

Mit dieser Konfiguration arbeitet der Server auch nach Ausfall der Funkuhr weiter als Zeitserver. Zeitsynchronisationen von außen werden nicht zugelassen.

Das DCF-Empfangsmodul arbeitet mit einer Betriebsspannungsversorgung über die serielle Schnittstelle, verwendet werden die Leitungen DTR und RTS. Um die dafür notwendigen Pegel richtig einzustellen, wird das nachstehende C-Programm-Skript *dcf77power.c* benötigt:

```
/* dcf77power.c v1.00 */

#include <errno.h>
#include <fcntl.h>
#include <termio.h>
#include <sys/ioctl.h>

int main()
{
  int dcf_dev, i;

  if((dcf_dev = open("/dev/refclock-0", O_RDWR|O_NOCTTY)) < 0)
  {
    perror("open /dev/refclock-x");
    return (-1);
  }
  i = TIOCM_DTR;
  ioctl(dcf_dev, TIOCMBIC, &i);      /* Clear DTR */
  i = TIOCM_RTS;
  ioctl(dcf_dev, TIOCMBIS, &i);      /* Set RTS */
  return (0);
}
```

Das Skript wird z. B. mit dem C-Compiler *cc*

```
cc dcf77power.c -o dcf77power
```

in das ausführbare Programm *dcf77power* übersetzt und dann im Verzeichnis */usr/sbin* gespeichert.

Das Programm *dcf77power* muss nach jedem Start von *xntp* automatisch gestartet werden. Dies wird durch das Einfügen der Zeile

```
/bin/stty 50 cs8 crtscts -ixon -ixoff ignpar parenb </dev/refclock-0
```

zum Stellen der seriellen Schnittstelle auf 50 Bit/s und der beiden Zeilen

```
/bin/sleep 15
/usr/sbin/dcf77power
```

zum Herstellen der Betriebsspannungsversorgung erreicht. Die Einträge sind in der Datei */sbin/init.d/xnptd* vorzunehmen (Listing 3-3).

```
(...)
test $link = $base && START_XNTPD=yes
test "$START_XNTPD" = yes || exit 0

return="$rc_done"

case "$1" in
 start)
 # Serielle Schnittstelle auf 50 Bit/s einstellen
 /bin/stty 50 cs8 crtscts -ixon -ixoff ignpar parenb </dev/refclock-0

 echo -n "Starting xntpd, please wait 15 seconds "
 if [ -n "$XNTPD_KERNEL_TICK" -a -x "/usr/sbin/tickadj" ]; then
    /usr/sbin/tickadj $XNTPD_KERNEL_TICK
 fi
 if [ -n "$XNTPD_INITIAL_NTPDATE" -a -x "/usr/sbin/ntpdate" ]; then
    /usr/sbin/ntpdate -bs $XNTPD_INITIAL_NTPDATE
.fi
 startproc /usr/sbin/xntpd || return=$rc_failed

 /bin/sleep 15
./usr/sbin/dcf77power

 echo -e "$return"
;;
(...)
```

Listing 3-3 Auszug aus /sbin/init.d/xntpd

Das Programm kann jetzt manuell mit

```
rcxntpd start
```

gestartet werden.. Die Meldung *Starting xntpd, please wait 15 seconds* sollte jetzt auf dem Bildschirm erscheinen und nach ca. 15 Sekunden sollte die Empfangsanzeige der Funkuhr im Sekundentakt blinken. Der abschließende Funktionstest kann jetzt mit dem bereits beschriebenen

```
/usr/sbin/ntpq
```

durchgeführt werden Dazu wird der Befehl *peers* eingeben. Wichtig ist, dass nach einiger Zeit die Zeile mit dem Eintrag GENERIC(0) mit einem »*« markiert wird. Dies zeigt an, dass diese Zeitquelle zur Synchronisation der Uhrzeit verwendet wird.

Je nach Empfangsqualität sollte nach 10-20 Minuten die Meldung

```
[...] synchronized to GENERIC(0), stratum=1
```

im Systemlogfile `/var/log/messages` auftauchen. Es ist möglich, dass *xntp* im Systemlogfile meldet, dass die Uhrzeit nicht gestellt werden kann, weil die Differenz von der Systemzeit zur Funkuhrzeit zu groß ist. In diesem Fall ist die Systemzeit einmal manuell zu teilen (eine minutengenaue Korrektur reicht). Dieses Problem kann auch bei einer fehlerhaften Einstellung der Zeitzone auftreten, gefordert ist UTC.

Kommerzielle Systeme

Kommerzielle Systeme zur Zeitsynchronisation auf Basis von DCF 77 sind von vielen Firmen verfügbar. So bietet z.B. die Firma IST-Software (*www.ist-software.de*) Lösungen auf Basis externer Geräte bzw. mit PC-Einsteckkarten an (Abbildung 3-58).

Abbildung 3-58 DCF-77-Funkuhrsystem, bestehend aus PC-Einsteckkarte und Antenne für Innenräume (Hersteller: HOPF, Bezugsquelle: www.ist-software.de)

Von der Firma Sonik GmbH (*www.sonik.de*) ist ein »Stand-Alone«-Zeitservermodul auf Linux-Basis (Abbildung 3-59) verfügbar, dass die Referenzzeit über *xntp* im Netzwerk zur Verfügung stellt.

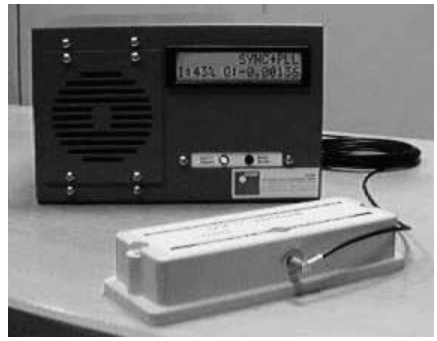


Abbildung 3-59 Timeserver der Firma Sonik GmbH

Synchronisation über Funkuhr – GPS

Als Quelle der zur Uhrensynchronisation verwendeten Referenzzeit wird das Satellitensystem GPS (*Global Positioning System*) des Amerikanischen Verteidigungsministeriums verwendet. Das GPS-Signal ist weltweit zu empfangen. Der Einsatz von GPS als Zeitquelle kommt in Frage, wenn ein befriedigender Empfang des DCF77- Signals wegen baulicher Gegebenheiten, nicht behebbarer Empfangsstörungen oder weger zu großer Entfernung des Empfängers vom DCF77-Sender nicht möglich ist. GPS eignet sich auch als Zeitquelle bei nicht-stationären Empfängern, wo eine permanente Ausrichtung der Antenne auf den DCF77-Sender nicht möglich ist, also z. B. auf Schiffen.

Abbildung 3-60 GPS-Funkuhrsystem, bestehend aus PC-Einsteckkarte und Außenantenne (Hersteller: HOPF, Bezugsquelle: www.ist-software.de)

In ca. 20.000 km Höhe bewegen sich die GPS-Satelliten auf unterschiedlichen Bahnen um die Erde. In jedem Satellit ist eine Atomuhr (Genauigkeit min. $1x10^{-12}$), deren Zeit kontinuierlich mit den Bahndaten ausgesendet wird. Der GPS-Empfänger registriert die Daten von max. sechs Satelliten und errechnet aus diesen Werten seine Position. Ist die Position berechnet, können die Laufzeiten der Daten von den einzelnen Satelliten ermittelt werden. Aus diesen Werten wird die GPS-Weltzeit im System bestimmt und über einen regelbaren Quarz mit einer Genauigkeit von ± 1 µs weitergeführt.

3.5 Programmdienste

Der Application-Server stellt Dienstleistungen im Netzwerk bereit, die über auf dem Server laufende Dienste oder Programme erbracht werden. Typische Beispiele hierfür sind Datenbankserver, WWW-Server sowie alle Serverdienste, deren Daten von Netzwerk-Cleints gezielt abgefragt werden können.

Der Application-Server stellt immer eine leistungsfähige Shell-Scriptsprache zur Verfügung, mit der der Systemverwalter Arbeitsabläufe auf dem Server per Programm automatisieren kann. Ein einfaches Beispiel sind die schon vorgestellten Skripte zur Einbindung von CD-ROMs in ein selbstentwickeltes CD-ROM-Server-Modell. Je leistungsfähiger diese Programmierschnittstelle ist, umso effizienter können Serverleistungen genutzt werden. Da Unix/Linux in C entwickelt wurden, sind nahezu alle C-Sprachkonstrukte in der Shell-Programmierung verfügbar!

3.5.1 Datenbanksystem

Der typische Programmdienst ist für viele Netzwerk-Server ein Datenbanksystem, fast alle angebotenen Softwarepakete entsprechen dem Standard SQL (*Standard Query Language*), mit dem Daten auf dem Server gespeichert, abgefragt und in Tabellenform anzeigt werden können. In einfachen Netzwerken und bei geringem Datenaufkommen können Access-basierte Lösungen tatsächlich noch ausreichend sein, dies gilt besonders, wenn auch hier bereits mit servergespeicherten Datenbeständen gearbeitet wird. Mit zunehmender Größe der Datenbank wird aber eine zentralisierte Lösung erforderlich, wobei die besondere Leistung der Datenbankserver tatsächlich darin liegt, in SQL programmierte Prozeduren auszuführen, die Datenaktionen direkt auf diesem Server ausführen können. Über Views werden Infomationen aus mehreren Tabellen zusammengeführt, verdichtet und an den anfragenden Client gesendet. Werden Daten geändert oder Datensätze neu erstellt bzw. gelöscht, können Trigger programmiert werden, die komplexe Datenbankaktionen durchführen, ohne weitere Netzwerklast zu erzeugen. Access kann hier am Netzwerkarbeitsplatz sehr wirkungsvoll als Datenbankfrontend für den SQL-Server eingesetzt werden,. Die Kopplung zwischen Datenbank und Access, Excel oder Word erfolgt dann über die Sofwareschnittstelle ODBC *(Open Database Connector)*, diese von Microsft definierte Schnittstelle ist für alle kommerziellen Datenbanksysteme verfügbar.

Innerhalb nur weniger Monate hat sich die Anzahl der auf Linux verfügbaren kommerziellen Datenbanksysteme explosionsartig erhöht, der Boom wurde so richtig spürbar auf der CeBit 1999. Waren es lange Zeit zwar schon sehr leistungsfähige, aber nur wenig bekannte Datenbanksysteme, die zum Lieferumfang der Distributionen gehörten, so sind es heute kommerzielle System wie *Adabas, db/2, Informix* oder *Sybase,* die für Linux verfügbar sind. Dabei sind diese hochwertigen Datenbanksysteme in der Regel nicht kostenlos verfügbar. Einige dürfen nur zur privaten Verwendung kostenlos eingesetzt werden, ansonsten wird auch hier der Kauf von Lizenzen fällig. Interessanteweise sind die Lizenzen im direkten Vergleich mit den anderen PC-Netzwerkbetriebssystemen oft deutlich preiswerter.

Industriestandard SQL

Die *Standard Query Language* SQL hat sich innerhalb weniger Jahre zu einem der wichtigsten Datenbankstandards entwickelt. SQL ermöglicht es zumindest in der Theorie Datenbankmodelle und prozedurale Aktionen unabhängig vom tatsächlich genutzten Datenbanksystemen einzurichten und zu betreiben. In der Praxis ist dies grundsätzlich möglich, es gibt aber leider immer wieder nennenswerte Unterschiede, z. B. in den Felddefinitionen oder in dem unterstützen SQL-Sprachumfang.

SQL ermöglicht die Definition von Datenbanken, Tabellen und die Führung einer vom Serverbetriebssystem unabhängigen Benutzerverwaltung; die Rechtevergabe kann hinunter bis auf die Feldebene durchgeführt werden. Mit dem bekannten SQL-Statement

```
select * from TO_Tabelle1
```

können Daten in Tabellenform angezeigt werden, z. B. als Abfrage in einem Access-Datenbankfrontend programmiert. Mit SQL können Datensätze eingefügt, gelöscht oder Feldinhalte gezielt geändert werden, bei Bedarf werden temporäre Tabelle aufgebaut oder über *Views* Daten verdichtet.

SQL kann aber noch wesentlich mehr: Es ist eine vollwertige Programmiersprache, in der auch komplexe Datenaktionen auf dem Server ausgeführt werden können. Enthalten sind logische Konstrukte, Schleifen und eine Vielzahl von Prozeduren, über die auf Server-Parameter zugegriffen werden kann. Wichtig ist die Möglichkeit, Trigger durch INSERT-, UPDATE- oder DELETE-Aktionen automatisch starten zu können. Diese speziellen Prozeduren prüfen die Integrität von Datenstrukturen, lösen komplexe Informationen mit Hilfe von Stammdatenbeständen auf oder geben Fehlermeldungen an den Datenbank-Client ab.

Kommunikation über ODBC

Die von Mircrosoft definiert Software-Schnittstelle *Open Database Connector* ODBC stellt die Verbindung zwischen Programmen wie Access, Excel oder Word und zentralen, serverbasierten Datenbanksystemen her. ODBC arbeitet mit SQL-Statements, die Schnittstelle übersetzt wechselseitig zwischen den beiden verbundenen »Welten« und ermöglicht so einen transparenten Dartenbankzugriff. ODBC-Treiber sind heute für nahezu alle gängigen Datenbankserver verfügbar. Interessant ist z. B. die Tabelleninhalte einer von AS/400 geführten Datenbank per ODBC in Access zu bearbeiten.

Soll auf eine spezielle Datenbank zugegeriffen werden, in unseren Beispielen auf linuxbasierten Systemen, so ist es notwendig, den passenden ODBC-Treiber zu beschaffen. Oft gehört diese Software bereits zum Standard-Lieferumfang der zu den Linux-Diustributionen gehörenden Datenbanksysteme, andernfalls ist die entsprechende Software mit großer Sicherheit über das Internet zu beziehen.

PostgreSQL

Das *Data Base Management System* DBMS *PostgreSQL* wurde ursprünglich als frei verfügbare Datenbank für den akademischen Bereich entwickelt, es ist der Nachfolger des relationalen Datenbanksystems INGRES. Entwickelt wurde das Programm an der University of California, Berkeley.

Die Programmpaket umfaßt ODBC, JDBC, APIs für diverse Programmiersprachen und eine leistungsfähige WWW-Unterstützung. Sehr häufig ist *PostgreSQL* in Verbindung mit dem WWW-Server *Apache* als Einstieg in datenbankorientierte Internetpräsentationen zu finden.

Weitere Informationen können über *www.postgresql.org* bezogen werden.

Installation und Start

Benötigt wird das Programmpaket *postgres* aus der Serie *ap,* installiert wird *PostgreSQL* Version 6.5.1-16 (8,6 MByte).

Um kommende Updates zu erleichtern oder um reine PostgreSQL-Client-Rechner einrichten zu können, wurde das Datenbanksystem *PostgreSQL* auf mehrere Pakete aufgeteilt (Tabelle 3-17).

postgres	PostgreSQL – die Datenbank
pg_datab	Initialisierungsdatenbank für PostgreSQL
pg_ifa	PostgreSQL-Interfaces (Basis)
pg_iface	zusätzliche PostgreSQL-Interfaces
pg_access	GUI für PostgreSQL-Datenbanken

Tabelle 3-17 Datenbanksystem PostgreSQL

Bei der Grundinstallation des *PostgreSQL*-Servers ist es wichtig, alle oben genannten Pakete zu installieren, das sonst die Datenbank nicht gestartet werden kann oder wichtige Werkzeuge zur Systemverwaltung fehlen.

Während der Installation wird in */etc/rc.config*

START_POSTGRES=“YES«

automatisch vorgenommen. Beim nächsten Systemstart wird der Datenbank-Daemon automatisch gestartet, wahlweise kann dies mit

```
# rcpostgres start
```

auch ohne Neustart sofort erfolgen.

Grundkonfiguration

Der Datenbankadministrator ist der Benutzer *postgres*. Nur er hat alle Rechte, um das Datenbanksystem zu verwalten. Das Benutzerkonto für *postgres* wird bereits bei der Installation angelegt, es muss allerdings noch, z.B. über YaST, ein Kennwort zugewiesen werden.

Nach der Anmeldung als Benutzer *postgres* sollte zunächst ein zweiter Datenbankbenutzer angelegt werden:

```
postgres@linux05:~ > createuser samulat
Enter user's postgres ID or RETURN to use unix user ID: 504 ->
Is user "samulat" allowed to create databases (y/n) y
Is user "samulat" a superuser? (y/n) y
createuser: samulat was successfully added
```

Der Datenbankbenutzer *samulat* erhält hier die Berechtigung, ebenfalls als Administrator arbeiten zu können.

Danach kann die erste *PostgreSQL*-Datenbank angelegt werden. Mit

```
postgres@linux05:~ > createdb adressen
```

wird die Datenbank *adressen* erzeugt. Der Benutzer, der die Datenbank anlegt, wird automatisch der *Data Base Administrator* DBA dieser Datenbank. Nur er und ein *Superuser* haben die Möglichkeit, diese Datenbank mit dem Befehl *destroydb* wieder zu löschen.

Arbeiten mit der Datenbank PostgreSQL

Zur Steuerung der Datenbank verfügt *PostgreSQL* über zwei Client-Programme:

psql ist ein textorientiertes SQL-»Terminal«, mit dem zeilenweise SQL-Anweisungen abgearbeitet werden können. Zum Start wird *psql* mit dem Namen der gewünschten Datenbank aufgerufen:

```
samulat@linux05: ~> psql adressen
Welcome to the POSTGRES ineractive sql monitor:
(...)
adressen=>
```

Durch Eingabe von *\q* wird *psql* beendet. Soll *psql* eine Textdatei abarbeiten, in der SQL-Kommandos gespeichert wurden, so wird beispielsweise mit

```
samulat@linux05: ~> psql -f tab1sql
```

die Datei *tabsql1* geladen und abgearbeitet.

Der zweite Client ist ein Tcl/Tk-Skript, das die grafische Oberfläche zu *psql* darstellt. Das Programm PostgreSQL Access *pgaccess* kann unter X11 gestartet werden, zunächst muss die Verbindung zu einer Datenbank hergestellt werden.

Abbildung 3-61 Bedieneroberfläche von pgaccess

Die Auswahl der gewünschten Datenbank erfolgt über das Menü *Database -> Open*. Angegeben werden müssen der Name der Datenbank sowie ein gültiger Account (Abbildung 3-62).

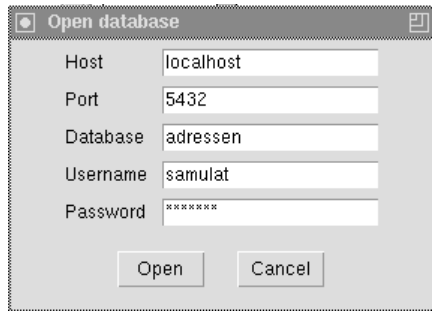


Abbildung 3-62 Datenbankanmeldung mit pgaccess

Auch ohne tiefergehende SQL-Kenntnisse kann über die grafische Oberfläche *pgaccess* schnell eine Tabelle angelegt werden; in Abbildung 3-63 ist dies die Tabelle *Gesamtliste*.

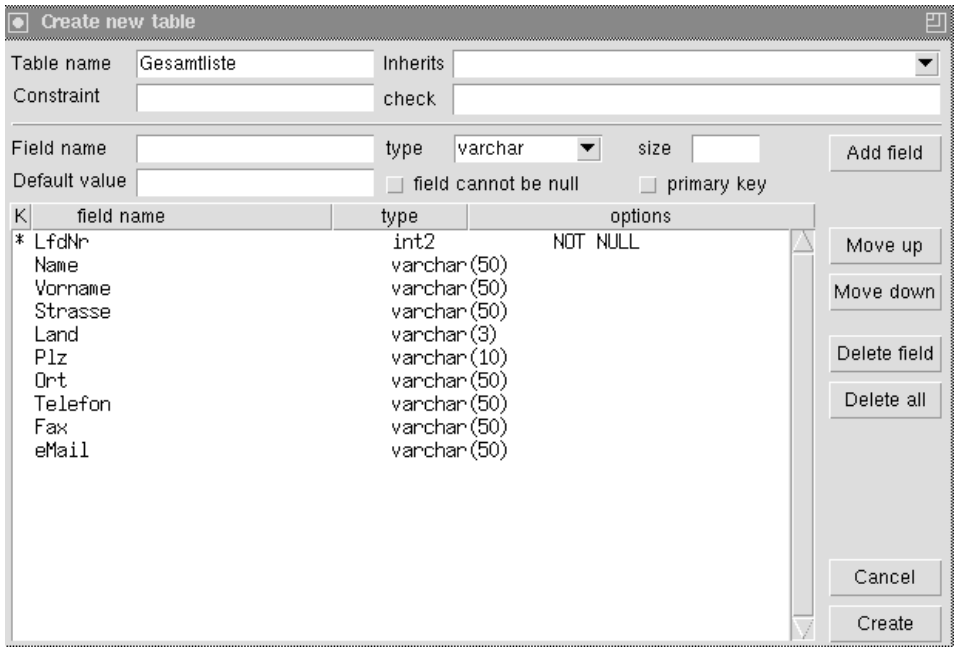


Abbildung 3-63 Tabellenerstellung mit pgaccess

MySQL

Die SQL-Datenbank MySQL (*www.mysql.org*) gehört zum Standardumfang vieler Linux-Distributionen. MySQL ist ein echtes Client-Server-Datenbanksystem mit einer Vielzahl von Schnittstellen, z.B. für C/C++, Perl, Eiffel, Java, PHP, ODBC und TCL. Neben dem Server-Dämon *mysqld* besteht das Programmpaket aus vielen Client-Anwendungen und vor allem für den Programmierer aus interessanten Bibliotheken.

An dieser Stelle wird die zur SuSE-Distribution gehörende MySQL Version 3.22.25 vorgestellt; weitere Informationen finden Sie in [Yarg99], [Kruc00] oder direkt unter *www.mysql.org*.

MySQL kann unter bestimmten Voraussetzungen ohne Lizenzkosten eingesetzt werden; weitere Lizenzinformationen sind auf der CD-ROM in der Datei */util/mySQL/Licensing_and_Support.htm* zu finden. Auch die Datenbankdokumentation ist als Postscript-File für Ausdruck im A4-Format in */util/MySQL* verfügbar.

Die Administration der Datenbank MySQL kann einfach über *phpMyAdmin* erfolgen. Dies ist eine auf PHP3-Skripten basierende grafische Oberfläche, die den Zugriff auf den Datenbankserver über einen WWW-Browser ermöglicht [Kruc00]. Einzige Voraussetzung dafür ist ein WWW-Server wie z.B. Apache, der PHP3-Skripte ausführen kann.

Hypertext Preproccessor PHP

Der *Hypertext Preproccessor* PHP ist eine in HTML »eingebettete« Skriptsprache, die eng an C, Java und Perl angelehnt ist und durch eigene Sprachelemente erweitert wurde. Wird ein WWW-Server wie Apache um ein Modul erweitert, das die Ausführung solcher PHP-Skripte ermöglicht, können PHP-Kommandos direkt in die HTML-Seiten geschrieben werden:

```
<html>
    <head>
        <title>Example</title>
    </head>
    <body>
        <?php echo "Hi, I'm a PHP script!"; ?>
    </body>
</html>
```

PHP ermöglicht es dem Programmierer, mit geringem Aufwand dynamisch erzeugte WWW-Seiten zu erstellen, in denen z. B. direkt auf einem Datenbankserver gespeicherte Informationen präsentiert *und* bearbeitet werden können. Im Gegensatz zu Javascript werden PHP-Skripte direkt auf dem WWW-Server ausgeführt.

PHP ermöglicht aber nicht nur Funktionen, die den Leistungen von CGI-Programmen vergleichbar sind, die besondere Leistung liegt im direkten Zugriff auf eine Vielzahl von Datenbanksystemen:

Adabas D	InterBase	Solid
dBase	mSQL	Sybase
Empress	MySQL	Velocis
FilePro	Oracle	Unix dbm
Informix	PostgreSQL	

PHP unterstützt auch den direkten Zugriff auf weitere Dienste. Die Programmierung auf der Basis von Protokollen wie IMAP, SNMP, NNTP, POP3 oder HTTP ist möglich.

Installation und Start

Zur Erweiterung des WWW-Servers Apache ist das Paket *mod_php* aus der Serie *n* notwendig. Mit einem kleinen Skript kann geprüft werden, ob der installierte WWW-Server PHP3 unterstützt:

```
<?
phpinfo()
?>
```

Dieses Skript wird unter dem Namen test1.php3 im Verzeichnis */usr/local/httpd/htdocs* gespeichert. Wird es mit einem beliebigen Browser aufgerufen, sollte die in Abbildung 3-64 gezeigte Liste der Systemeinstellungen erstellt werden.

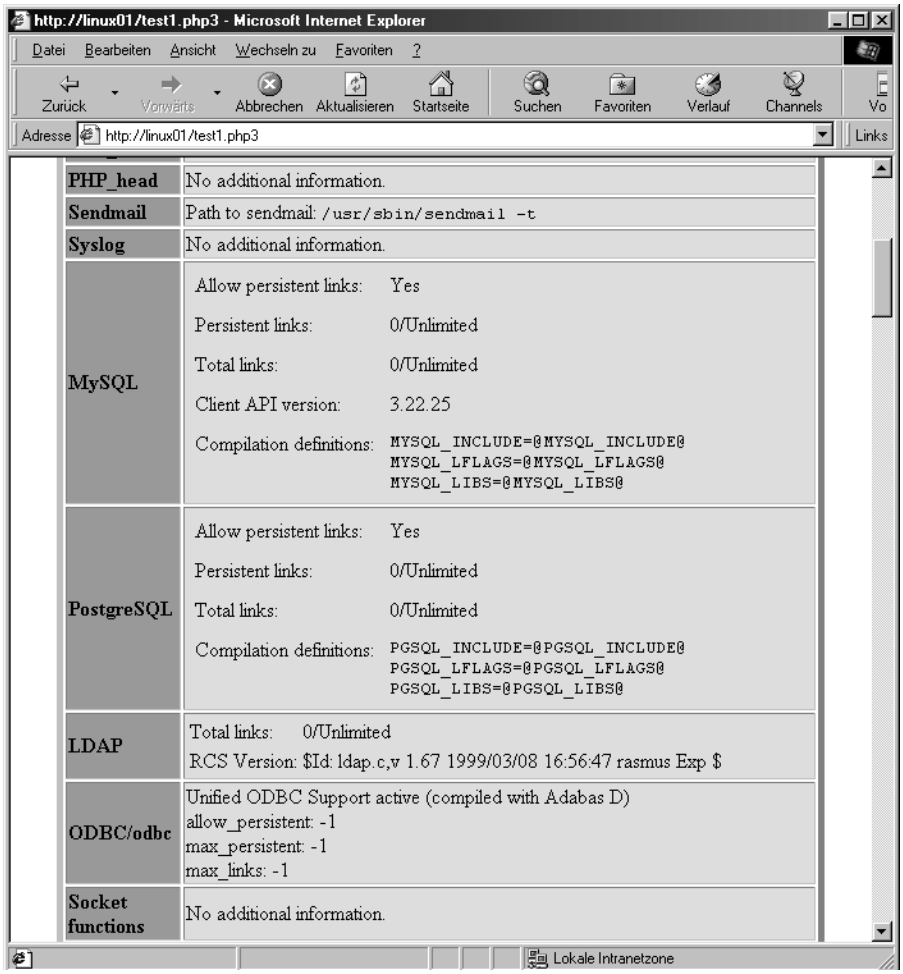


Abbildung 3-64 Abruf der Systemparameter mit PHP-Skript

Installation und Start von MySQL

Benötigt wird das Paket *mysql* aus der Serie *pay*. Zum automatischen Start des Datenbankdaemons ist der Eintrag

```
START_MYSQL="YES"
```

in der Konfigurationsdatei */etc/rc.config* erforderlich. Nach dem erfolgreichen Start des Datenbankdaemons kann der Serverstatus mit dem Befehl *mysql* abgefragt werden:

```
Linux01:~ # mysqladmin status
Uptime: 71  Threads: 1  Questions: 1  Slow Queries: 0  Opens: 6  Flush Tables:
1  Open Tables:  2
```

Grundkonfiguration

MySQL führt wie fast alle anderen Datenbanksysteme eine eigene Benutzerdatenbank, in der die Berechtigungen innerhalb des Datenbanksystems festgelegt werden. Der Datenbankverwalter von MySQL hat den (leider unglücklich gewählten) Namen *root*.

Um mit der Datenbank arbeiten zu können, muß diesem Benutzer mit dem Programm *mysqladmin* ein Kennwort zugewiesen werden. Mit

```
Linux01:~ # mysqladmin -u root password 'wurmloch'
```

wird das Kennwort *wurmloch* zugewiesen. Der Benutzer *root* hat als Datenbankadministrator zwar alle Berechtigungen; er darf aber nur von dem Server aus zugreifen, auf dem der MySQL-Daemon läuft. Für die hier vorgestellte Adminstration mit *phpMyAdmi*n ist dies zunächst ausreichend. Die dafür genutzten PHP3-Skripte werden auch dann lokal auf dem Datenbankserver ausgeführt, wenn die Datenbankverwaltung von einem Remote Host ausgeführt wird.

Die Grundkonfiguration wird mit einem *ping* geprüft:

```
Linux01:~ # mysqladmin -u root -p ping
Enter password:
mysqld is alive
```

Die Rückmeldung *mysql is alive* zeigt, dass die Datenbank in Funktion ist und die Anmeldung akzeptiert wurde.

Informationen über die installierte Version von MySQL können mit dem Programm *mysqladmin* abgerufen werden:

```
linux01:~ # mysqladmin -u root -p version
Enter password:
mysqladmin  Ver 7.11 Distrib 3.22.25, for pc-linux-gnu on i686
TCX Datakonsult AB, by Monty

Server version     3.22.25
Protocol version      10
Connection         Localhost via UNIX socket
UNIX socket        /tmp/mysql.sock
Uptime:         56 min 4 sec

Threads: 1  Questions: 7  Slow queries: 0  Opens: 7  Flush tables: 1  Open
tables: 3
```

Installation und Konfiguration von phpMyAdmin

Auf der CD-ROM finden Sie im Verzeichnis */util/phpmyadmin* das Installationsfile zu *phpMyAdmin* (Version 2.0.5 vom 5.12.1999). Aktuellere Versionen und weitere Informationen können über *http://www.phpwizard.net/phpmyadmin* bezogen werden.

Das Installationsfile ist in das Verzeichnis zu kopieren, in dem der WWW-Server seine Textdateien speichert und ist dort zu entpacken:

```
linux01:~ # cp phpMyAdmin_2_0_5_tar.tar /usr/local/httpd/htdocs
linux01: ~ # cd /usr/local/httpd/htdocs
linux01: /usr/local/httpd/htdocs # tar xzvf phpMyAdmin_2_0_5_tar.tar
```

Im dem neuen Verzeichnis */usr/local/httpd/htdocs/phpMyAdmin* werden die PHP-Skripte von phpMyAdmin gespeichert.

Die Konfigurationsdatei ist *config.onc.php3;* hier wird unter anderem festgelegt, welche Benutzer die von *phpMyAdmin* zur Verfügung gestellten Skripte benutzen dürfen. Grundsätzlich stehen zwei Anmeldeverfahren zur Auswahl:

- *Basic Authentification:* Jeder Benutzer hat über die URL *http://hostname/phpMyAdmin/index.php3* den vollen Zugriff auf den MySQL- Server. Dieses Verfahren ist selbst für kleine Netzwerke als unsicher zu betrachten und sollte nicht verwendet werden.
- *Advanced Authentification:* Jeder Benutzer muß sich mit Benutzer-ID und Kennwort anmelden. Listing 3-4 zeigt die dazu notwendigen Anpassungen in der Konfigurationsdatei *config.inc.php3*.

```
(...)
// MySQL hostname
$cfgServers[1]['host'] = 'localhost';
// MySQL port - leave blank for default port
$cfgServers[1]['port'] = '';
// Use advanced authentication?
$cfgServers[1]['adv_auth'] = true;
// MySQL standard user (only needed with advanced auth)
$cfgServers[1]['stduser'] = 'root';
// MySQL standard password (only needed with advanced auth)
$cfgServers[1]['stdpass'] = 'wurmloch';
// MySQL user (only needed with basic auth)
$cfgServers[1]['user'] = 'root';
// MySQL password (only needed with basic auth)
$cfgServers[1]['password'] = '';
// If set to a db-name, only this db is accessible
$cfgServers[1]['only_db'] = '';
// Verbose name for this host - leave blank to show the hostname
$cfgServers[1]['verbose'] = '';
(...)
```

Listing 3-4 Konfigurationsdatei für phpMyAdmin (Auszug)

Arbeiten mit der Datenbank MySQL

Die Datenbankverwaltung erfolgt mit *phpMyAdmin* (Abbildung 3-65). Mit einem beliebigen WWW-Browser wird dazu das PHP-Skript */phpMyAdmin/index.php3* aufgerufen. Nach der Identifizierung als berechtigter Benutzer (hier: *root* mit dem Kennwort *wurmloch*) kann in der Datenbank gearbeitet werden.

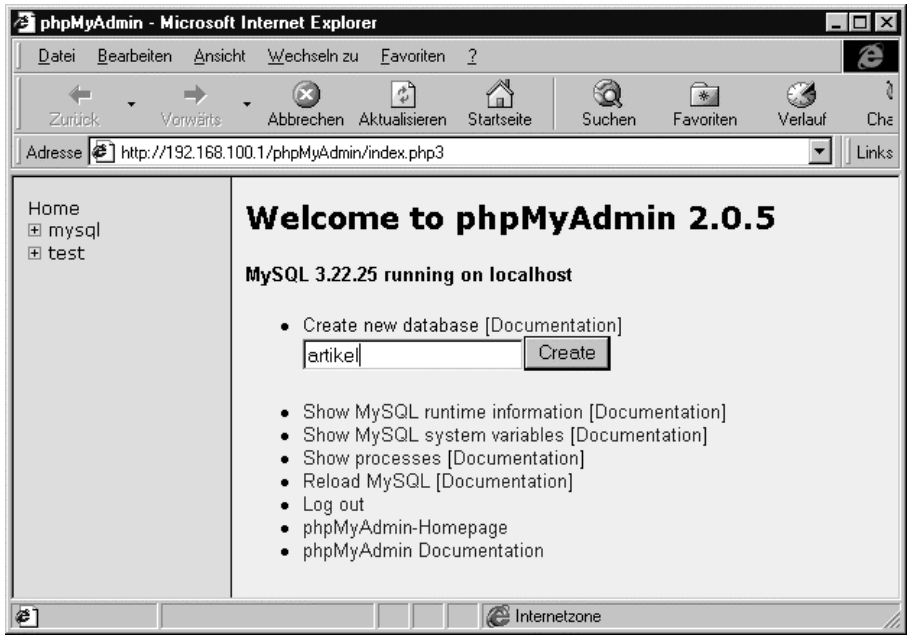


Abbildung 3-65 Startseite von phpMyAdmin

In Abbildung 3-66 wird die neue Datenbank *artikel* anlegt. Diese soll zunächst nur eine Tabelle mit dem Namen *Bestand* enthalten. Diese Tabelle wird in der Abbildung 3-67 mit einer CREATE-Anweisung erstellt.

Die erfolgreiche Ausführung der CREATE-Anweisung wird wie in Abbildung 3-62 bestätigt; sofort danach können Daten in diese Tabelle eingegeben und abgerufen werden (Abbildungen 3-67 und 3-68).

Die Anzeige der in der Tabelle *Bestand* enthaltenen Datensätze erfolgt über die Funktion *Browse;* mit *Select* kann der Anzeigeumfang genauer spezifiziert werden.

Datenbankzugriff über ODBC

Benutzerkonto einrichten

Nach der Grundinstallation der Datenbank MySQL existiert zunächst nur der Benutzer *root*, der als Datenbankverwalter alle Rechte besitzt. Eine Besonderheit von MySQL ist, dass diese Berechtigung zunächst aber auf den Hosts beschränkt ist, auf dem die Datenbank Daemon läuft.

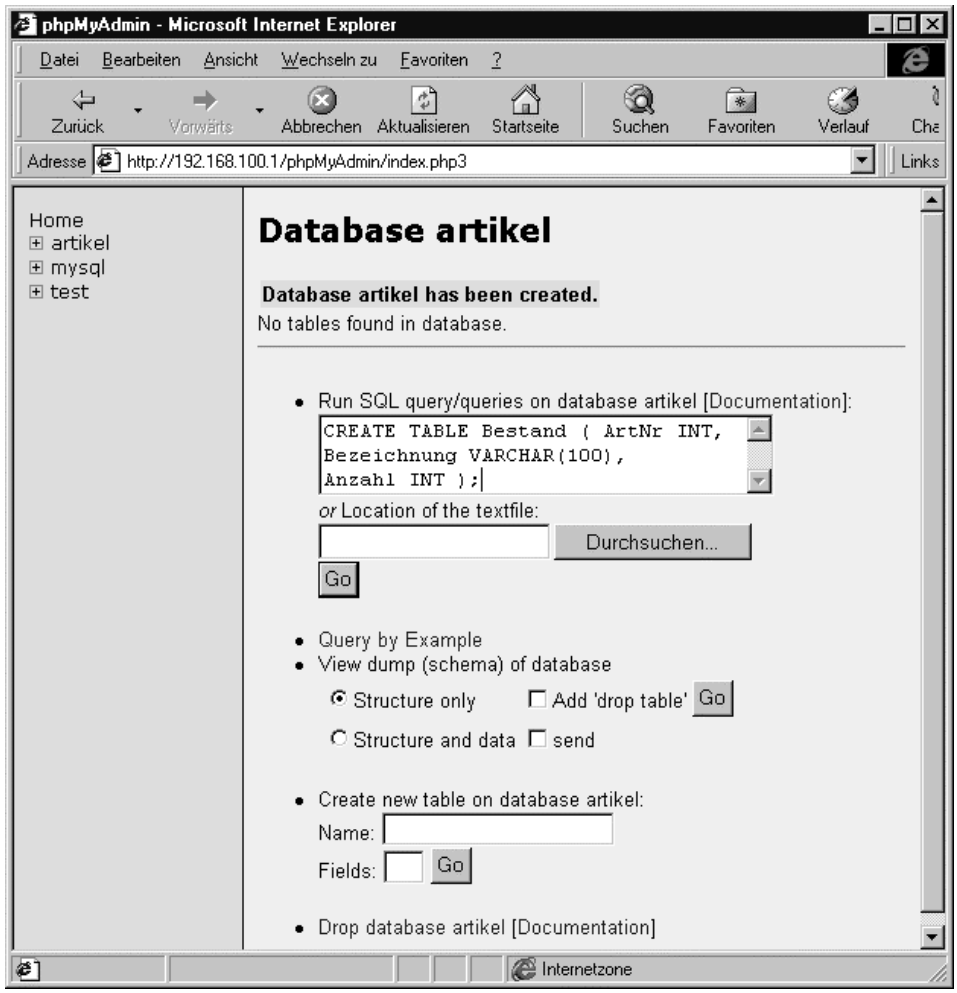


Abbildung 3-66 Tabelle erstellen(1)

Soll über ODBC von einem anderen Rechner im Netzwerk auf die Datenbank zugegriffen werden, so muß zunächst ein dafür geeignetes Benutzerkonto angelegt werden. Dies ist mit der grafischen Oberfläche *phpMyAdmin* nicht möglich, sodass dafür direkt mit dem zur Datenbank gehörenden SQL-Kommandozeilenwerkzeug *mysql* gearbeitet werden muß (An dieser Stelle sollen nur die tatsächlich benötigten Arbeitsschritte beschrieben werden; eine detaillierte Beschreibung von *mysql* finden Sie in [Yarg99] oder unter *http://www.mysql.org*).

Im folgenden Beispiel wird ein neuer Benutzer mit dem Namen *max* angelegt, der von jedem Rechner im Netzwerk auf die Datenbank zugreifen kann.

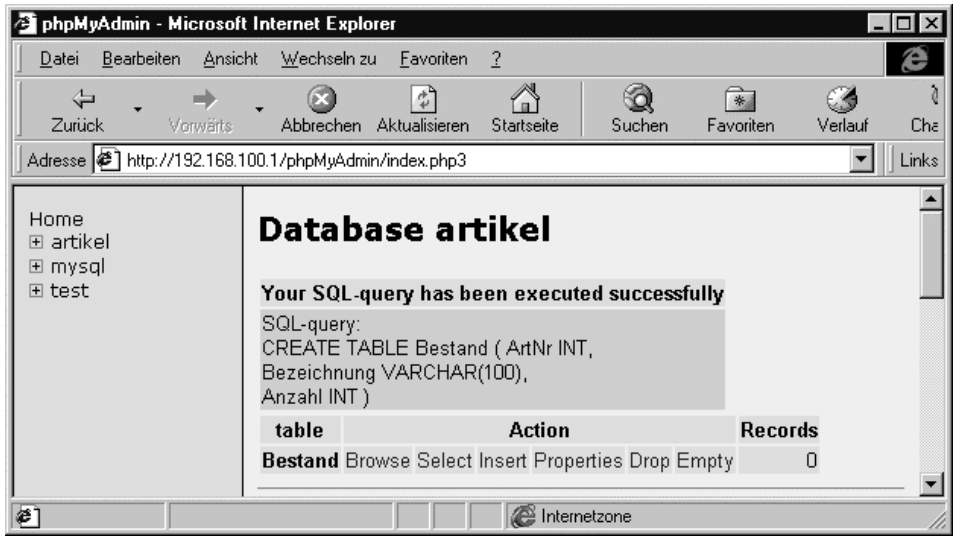


Abbildung 3-67 Tabelle erstellen (2)

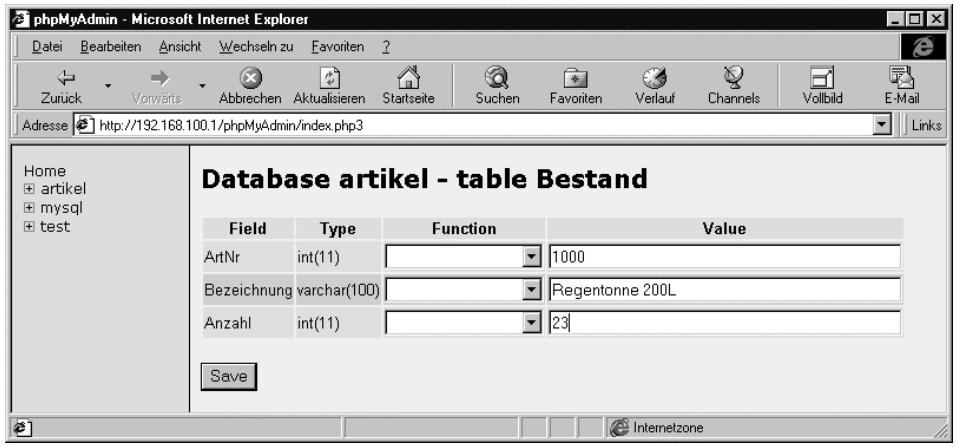


Abbildung 3-68 Dateneingabe

Das Programm *mysql* wird als Benutzer *root* gestarten; das bereits festgelegte Kennwort *wurmloch* wird abgefragt:

```
linux01:~ # mysql -u root -p
Enter password:
Welcome to the MySQL monitor. Commands en with ; or \g.
Your MySQL Connection id is 15 to server version: 3.22.25

Type 'help' for help.

mysql> _
```

Abbildung 3-69 Anzeige des Tabelleninhaltes

Das Programm kann durch Eingabe von *quit*, *exit* oder durch Strg D beendet werden. Jeder SQL-Befehl muß durch ein Semikolon abgeschlossen werden. Mit der ersten SQL-Anweisung werden die aktuell von MySQL geführten Datenbanken abgefragt:

```
mysql>  SHOW DATABASES;
+----------+
| Database |
+----------+
+artikel   +
+mysql     +
+test      +
+----------+
3 rows in set (0.00 sec)
```

Während der Grundinstallation wurden die Datenbanken *mysql* und *test* angelegt; *mysql* enthält die Systemtabellen; *test* ist eine leere Datenbank. Die Datenbank *artikel* wurde bereits mit *phpMyAdmin* zusätzlich erstellt.

Zur Benutzerverwaltung wird der Zugriff auf eine der Systemtabellen benötigt. Im nächsten Arbeitsschritt wird dazu die Datenbank *mysql* ausgewählt und die darin gespeicherten Tabellen abgerufen:

```
mysql > USE mysql;
Reading table information for completion of table and column names
You can turn off this feature to get quicker starup with -A

Database changed
mysql > SHOW TABLES;
+-----------------+
| Tables in mysql |
```

```
+-----------------+
+colums_priv      +
+db               +
+func             +
+host             +
+tables_priv      +
+user             +
+-----------------+
```

Die Taballe *user* enthält die Benutzerkonten (Tabelle 3-18). Hier wird festgelegt, unter welchem Namen sich ein Benutzer an der Datenbank anmelden darf, von welchen Hosts diese Anmeldung erfolgen darf und welche effektiven Berechtigungen entstehen.

Feldname	Beschreibung
Host	IP-Adresse(n) oder Hostname(n) der Rechner, von denen eine Anmedlung erfolgen darf.
User	Benutzername
Password	Kennwort (geschlüsselt)
Select_priv	Tabellenberechtigung
Insert_priv	Tabellenberechtigung
Update_priv	Tabellenberechtigung
Delete_pirv	Tabellenberechtigung
Create_priv	Datenbankberechtigung
Drop_priv	Tabellenberechtigung
Reload_priv	Serververwaltung
Shutdown_priv	Serververwaltung
Process_priv	Serververwaltung
File_priv	Dateiberechtigung
Grant_priv	Tabellenberechtigung
References_priv	Datenbankberechtigung
Index_priv	Datenbankberechtigung
Alter_priv	Datenbankberechtigung

Tabelle 3-18 Struktur der Tabelle user

Alle Berechtigungen werden in der Tabelle *user* über die Feldinhalte »Y« oder »N« eingestellt. Für die Felder *Host*, *User* und *Password* werden Zeichenketten (Strings) eingetragen. Mit dem SQL-Befehl

```
mysql >  SELECT host,user FROM user;
+-----------------+
```

```
| user | host      |
+------------------+
| root | localhost |
| root | linux01   |
|      | localhost |
|      | linux01   |
+------------------+
```

werden die aktuell eingetragenen Benutzer und die für sie zugelassenen Hosts angezeigt.

Der neue Benutzer *max* soll von allen Rechnern im Netzwerk zugreifen dürfen; im Feld *Host* wird dazu der Platzhalter »%« eingetragen. In diesem Beispiel erhält der Benutzer zunächst alle in Tabelle 3-18 dargestellten Berechtigungen. Das für *max* vorgesehene Kennwort *xxx* wird über die Funktion PASSWORD geschlüsselt in die Tabelle *user* eingetragen.

Mit dem SQL-Befehl INSERT wird ein neuer Benutzereintrag erstellt:

```
mysql > INSERT INTO user
     -> VALUES('%','max',PASSWORD('xxx'),
     -> 'Y','Y','Y','Y','Y','Y','Y',
     -> 'Y','Y','Y','Y','Y','Y','Y');
Query OK, 1 row affected (0.00 sec)
```

Damit ist der neue Benutzer *max* in der Lage, auch über ODBC von jedem beliebigen Rechner aus mit der MySQL-Datenbank in Verbindung zu treten. Die hier eingestellten Berechtigungen entsprechen denen eines Datenbankverwalters; sie müssen im praktischen Betrieb auf die tatsächlichen Erfordernisse reduziert werden.

Installation ODBC

Auf der CD-ROM finden Sie die vollständigen ODBC Setup-Routinen im Verzeichnis */util/mySQL/odbc*. In den Unterverzeichnissen sind ODBC-Treiber für Windows 9x und Windows-NT in der Version 2.50.29 enthalten. Akuellere Versionen, Treiber für andere Betriebssysteme und die Source-Codes können über *www.mysql.org* direkt bezogen werden. Die Datei */util/MySQL/odbc/README* enthält weitere Informationen.

Die Installation wird nach Aufruf des entsprechenden Setups in gewohnter Weise automatisch ausgeführt. Zur Erstellung eines neuen ODBC-Treibers sind dann die Anmeldedaten anzugeben (Abbildung 3-70).

Nach Abschluß dieser Konfiguration kann von einem Netzwerkarbeitsplatz unter Windows 9x/NT über Excel oder Access auf die MySQL-Datenbestände zugegriffen werden.

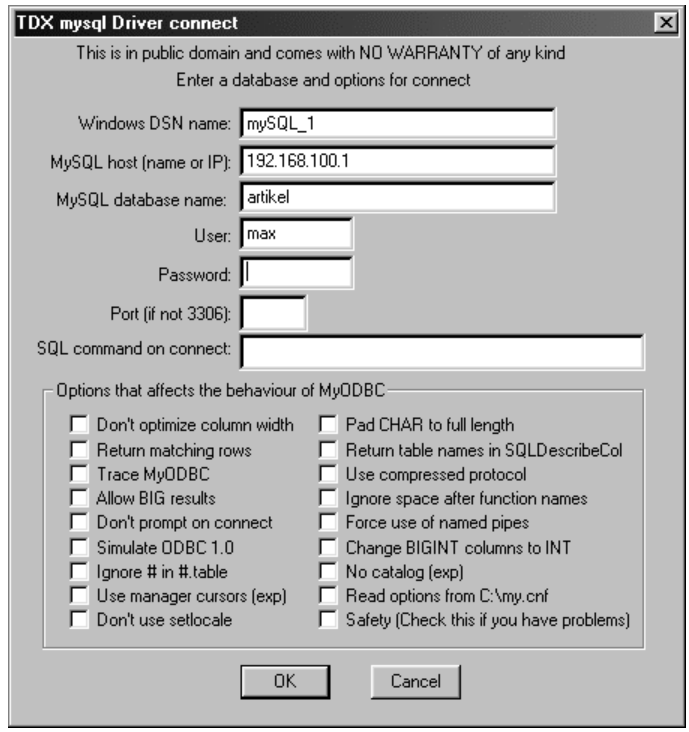


Abbildung 3-70 Anmelde- und Verbindungsdaten für einen MySQL-ODBC-Connector

Adabas D

Adabas D (*www.adabas.com*) ist ein professionelles Datenbanksystem der Firma Software AG, in dem das relationale Modell vollständig implementiert ist, einschließlich der Unterstützung für Domains, Primärschlüssel, änderbare Join Views, referenzielle Integrität, Trigger und Datenbankprozeduren. Sie ist eines der Datenbanksysteme, auf denen die Standardsoftware SAP R/3 läuft. Auch Adabas D ist vollständig über SQL steuerbar. Für Adabas D ist verfügbar:

- C/C++/COBOL-Precompiler
- *WebDB* zur Kopplung von Adabas D und WWW-Servern
- ODBC- und JDBC-Treiber
- Perl-, Tcl/Tk-Interface
- Windows-basiertes DBA-Tool: *DOMAIN*
- Abfragetool für Microsoft-Office-Anwendungen: *QueryPlus*
- Migrationstool für MS-Access-Datenbanken: *AccessPlus*

Adabas D bietet auch ein für den kommerziellen Einsatz günstiges Lizenzierungsmodell, z.B.:

ADABAS D Entry-Edition 11.0 für Linux, Lizenz für 10 User	ca. 500,00 DM
ADABAS D Business-Edition 11.0 für Linux, Lizenz für beliebig viele User	ca. 5000 ,00 DM

Die Lizenz für beliebig viele User wird z.B. dann benötigt, wenn diese Datenbank zusammen mit einem WWW-Server wie Apache für Internet-Präsentationen eingesetzt wird. Viele andere leistungsmäßig vergleichbare kommerzielle Datenbanksysteme sind deutlich teurer. Weitere Details können z.B. [Stik98] entnommen werden.

Installation und Start

Die Einrichtung der Datenbank erfolgt durch das Programmpaket *adabas* aus der Serie *pay;* installiert wird Adabas D 11.0 Personal Edition. Empfehlenswert ist es, gleichzeitig auch die vollständige Dokumentation mit zu installieren, dazu wird das Paket *adadocde* benötigt.

Der Systempfad für die Adabas-D-Dateien muss in der Umgebungsvariablen DBROOT eingetragen und für andere Shells exportiert werden. Dies geschieht mit

```
# DBROOT=/usr/lib/adabas
# export DBROOT
```

Nun sollte der während der Installation erstellte Eintrag für *START_ADABAS* in */etc/rc.config* noch manuell auf

START_ADABAS=“YES«

geändert werden. Beim nächsten Systemstart wird der Datenbank-Daemon dann automatisch gestartet. Wahlweise kann dies mit

```
# rcadabas start
Starting service ADABAS D remote SQL
Starting service ADABAS D server
```

auch ohne Neustart sofort erfolgen.

Grundkonfiguration

Der Datenbankadministrator ist der Benutzer *adabas*. Nur er hat alle Rechte, zur Verwaltung des Datenbanksystems. Das Benutzerkonto für *adabas* wird bereits bei der Installation angelegt; auch hier muss allerdings noch z.B. über YaST ein Kennwort zugewiesen werden. Die weiteren Konfigurationsarbeiten erfolgen dann als Benutzer *adabas*. Mit

```
root@linux01 # su adabas
ababas@linux01:~/bin >
```

kann der Systemverwalter *root* ohne Kennwortabfrage mit dem Benutzeraccount *adabas* arbeiten.

Auch hier muss zunächst eine Datenbank angelegt werden, um mit Adabas D arbeiten zu können. Die kann z.B. über das Kommandozeilenwerkzeug *xcontrol* erfolgen (Abbildung 3-71).

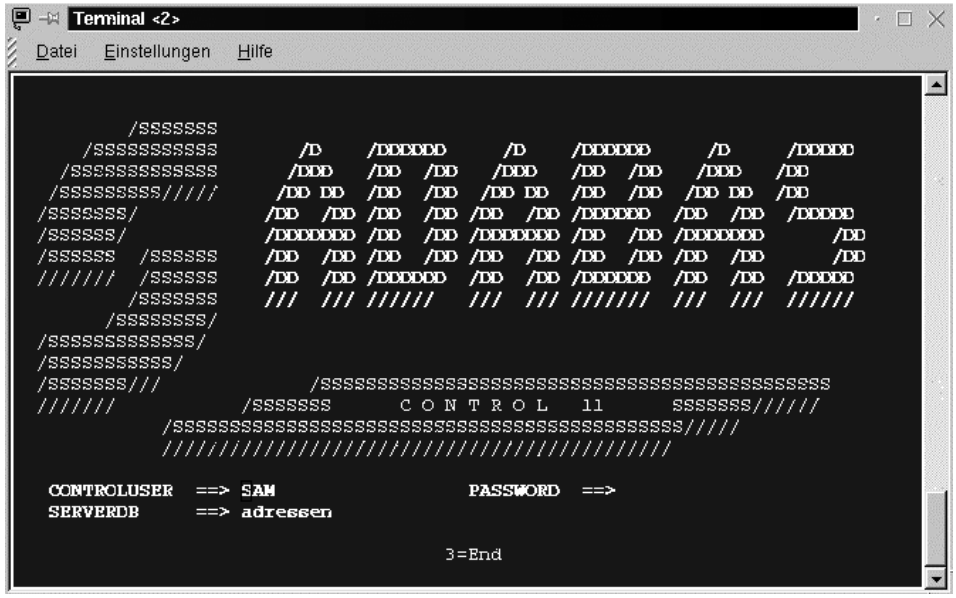


Abbildung 3-71 Anlegen einer Datenbank mit xcontrol

Im Startfenster von *xcontrol* wird zunächst der Name und das Kennwort des *CONTROLUSER*, des Datenbankverwalters, angegeben. Unter *SERVERDB* wird der Name der neuen Datenbank eingetragen. Die Eingabefelder können mit ⇆ gewechselt werden, [RETURN] beendet die Eingabe.

In den nachfolgenden Fenstern, die nur dann erscheinen, wenn tatsächlich eine neue Datenbank angelegt wird, werden zunächst die Namen und Kennworte für zwei weitere wichtige Konten zur Verwaltung dieser Datenbank eingetragen:

Sysdba	höchster Datenbankadministrator
Domain User	Besitzer des Datenbankkataloges, d.h. der Sytemtabellen

Danach werden die Konfigurationsparameter der neuen Datenbank aufgelistet und können bei Bedarf geändert werden. Tabelle 3-19 enthält eine Übersicht der wichtigsten Parameter.

Parameter	Standard	Beschreibung
MaxServerDB	1	Anzahl Datenbanken in verteilter Umgebung
MaxBackupDevs	2	Anzahl der Backup-Geräte (Bänder, Dateien, ...) für parallel ausgeführtes Save & Restore
MaxServerTasks	6	max. Anzahl paralleler Prozesse für Save & Restore
MaxUserTasks	3	max. Anzahl gleichzeitig arbeitender Benutzer (Sessions)
MaxCPU	1	Anzahl von Prozessoren, die diese Datenbank maximal nutzen darf
Data_Cache_Pages	1000	Anzahl von 4kB-Blöcken für den Daten-Cache
Proc_Data_Pages	30	Anzahl von 4kB-Blöcken für den DB-Prozeduren-Cache
Temp_Cache_Pages	30	Anzahl von 4kB-Blöcken für temporäre Tabellen
Catalog_Cache_Pages	32	Anzahl von 4kB-Blöcken für Strukturdaten
Log_Queue_Pages	50	Anzahl von 4kB-Blöcken für Loq-Queue
Log_Cache_Pages	20	Anzahl von 4kB-Blöcken für Log-Cache
Conv_Cache_Pages	100	Anzahl von 4kB-Blöcken für Konverter-Cache
MaxLocks	2500	max. Anzahl gleichzeitig möglicher Sperren
PNOPoolSize	10000	Größe des Memory-Pools für die Suche nach freien Seiten
RunDirectory	/usr/lib/adabas/wrk/toskana	Verzeichnisse für die Dateien *knldiag*, *knltrace*, *control.** und *Devspaces*, wenn für diese keine Pfad angegeben wird
DiagSize	100	Größe der Kernel-Diagnosedatei in 4kB-Blöcken
KernelTraceSize	200	Größe der Kernel-Tracedatei in 4kB-Blöcken
Default Code	ASCII	Default-Kodierung für CHAR-Felddatentypen
Date Format	INTERNAL	Datums- und Zeitformat. INTERNAL = YYYYMMDD, HHHHMMSS, EUR= DD.MM.YY, HH.MM.SS

Tabelle 3-19 Datenbank Konfiguration

Abschließend sind die Namen und Pfade der Dateien anzugeben, die jetzt tatsächlich die Datenbank darstellen. In der Beispielkonfiguration wird für alle vier Dateien der Typ »F« (*Datei*) eingestellt, *DEVSPACE PATH* gibt dann den tatsächlichen Speicherplatz und den Dateinamen an (Tabelle 3-20).

NAME	TYPE	SIZE	DEVSPACE PATH
SYSTEMDEF	F	1000	/home/daten/toskana/systemdef
TRANS LOG	F	2000	/home/daten/toskana/trans_log
ARCHLOG 1	F	2000	/home/daten/toskana/archlog_1
DATDEV 01	F	2000	/home/daten/toskana/datdev_01

Tabelle 3-20 Datenbankdateien

Die Konfiguration ist jetzt abgeschlossen, nun ist noch anzugeben, ob die Datenbank in einem Durchgang (*Install*), schrittweise (*Stepwise*) oder aus einer Datensicherung (*Restore*) heraus hergestellt werden soll. In diesem Beispiel sollte *Install* gewählt werden.

Während der Einrichtung der Datenbank werden die bearbeitenden Schritte angezeigt:

```
---> INSTALL PARAMETER............................ OK
     START SERVERDB COLD........................... ACTIVE
     INIT CONFIGURATION............................ --
     ACTIVATE SERVERDB............................. --
     LOAD SYSTEM TABLES............................ --
```

ACTIVE zeigt den gerade in Bearbeitung stehenden Schritt an; erfolgreich abgeschlossene Aktionen werden mit *OK* gekennzeichnet. Im *Arbeitsschritt LOAD SYSTEM TABLES* wird dann für jede Tabelle ein weiterer Eintrag ausgegeben.

Die Datenbank *toskana* wird abschließend gestartet, nach Adabas-Sprachgebrauch wird sie vom Zustand *COLD* in den Zustand *WARM* gebracht. Die Anzeige von *xcontrol* zeigt dann mit waagerechten Balken die prozentuale Auslastung von Daten- und Logbereichen sowie die aktuelle Anzahl der Benutzerconnects (Abbildung 3-72).

Die weitere Steuerung und Beobachtung der laufenden Datenbank *toskana* kann jetzt mit der grafischen Oberfläche *adcontrol* durchgeführt werden (Abbildung 3-73).

In der laufenden (*warmen*) Datenbank kann der Datenbankadministrator jetzt über *adcontrol* Benutzer, Tabellen, Indizes, Views, Trigger etc. anlegen und verwalten.

Der Datenbankzugriff für Clients wird am einfachsten über ODBC realisiert. Die zu ADABAS D gehörende Client-Software für Windows 9x/NT und ODBC-Trieber können aktuell über *www.adabas.com* geladen werden.

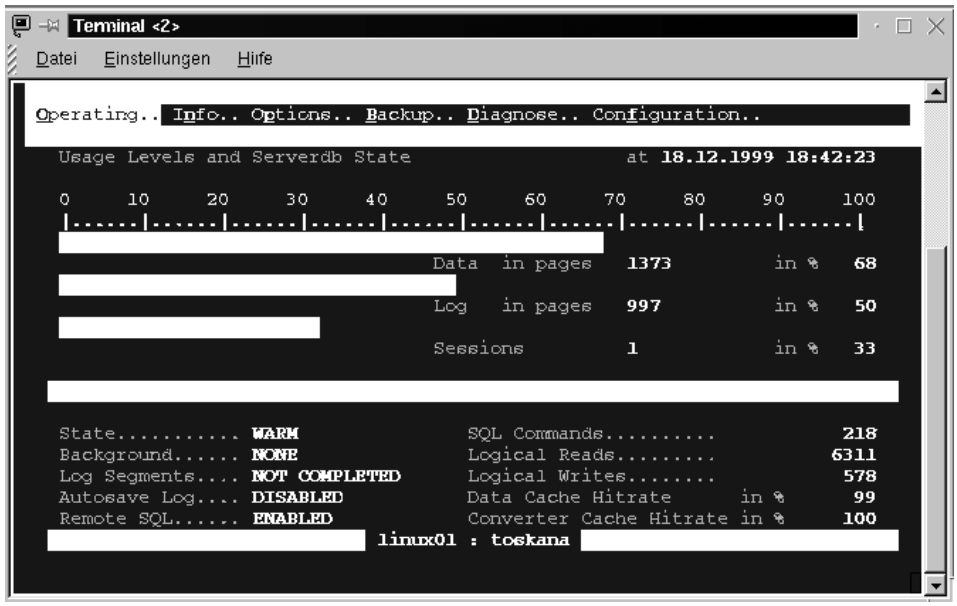


Abbildung 3-72 Statusanzeige der Datenbank toskana

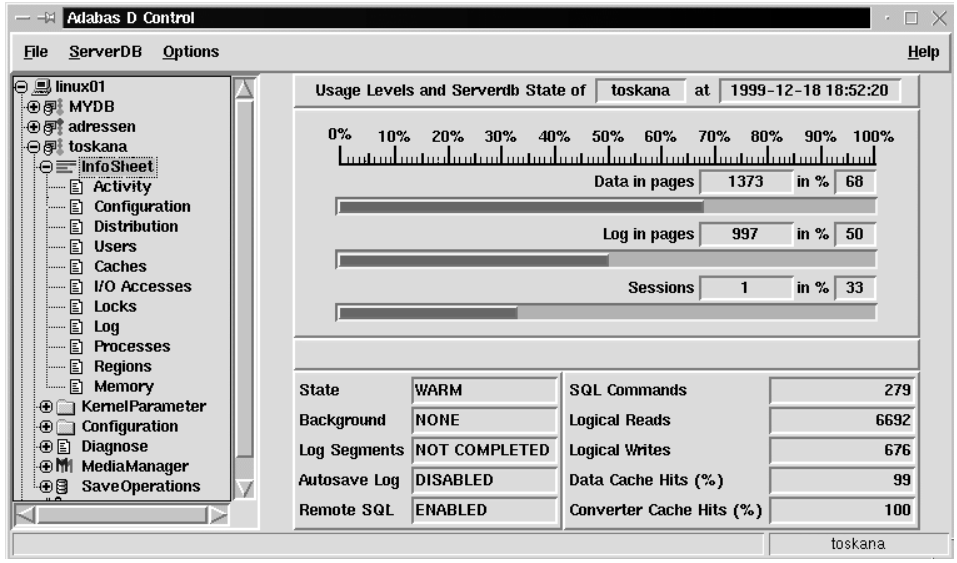


Abbildung 3-73 Bedieneroberfläche von adcontrol

3.5.2 Mainframe-Zugang

Der Zugriff auf Mainframes ware viele Jahre lang die Domäne der einfachen, textorientierten Terminalsysteme. Mit dem Aufkommen der DOS-Netzwerkarbeitsplätze wurden dann viele Programme erstellt, die zusätzliche Funktionalitäten in die Terminals integrierten und die Arbeitsabläufe vereinfachten. Heute ist oft festzustellen, dass gerade diese Programme auf Netzwerkarbeitsplätzen unter Windows NT nicht mehr laufen, da sie mit direkten Hardwarezugriffen arbeiten. Aber sie werden noch weiter benötigt, eine Neuprogrammierung (nur wegen des Wechsels des Arbeitsplatzbetriebssystems) scheidet aus Kostengründen aus.

Standard-Terminalemulationen

Es wurde bereits dargestellt, dass bei Linux eine Vielzahl sehr leistungsfähiger Terminalemulationen zum Standardumfang gehören. Diese Terminals, z.B. *xterm*, können in sehr vielen Details konfiguriert werden, zahlreiche Emulationen von älteren Standard-Terminals können direkt eingestellt werden.

Oft aus ganz speziellen Anforderungen heraus sind auch Lösungen verfügbar, die weit über diesen Rahmen hinausgehen. Dazu gehört das Programm *x3270*, ein *IBM 3270 Terminal Emulator für X Windows* (auf der CD-ROM im Verzeichnis */util/x3270)*.

Das Programm kann zur Kommunikation mit jedem IBM-Mainframe eingesetzt werden, der Sitzungen vom Typ 3270 über *telnet* unterstützt. Auch wird ein reiner ASCII-Modus unterstützt, der das erste Login ermöglicht. Danach wird im Vollbild-3270-Modus weitergearbeit. Die auf Hosts vom Typ 3278 oder 3279 verwendeten speziellen Zeichensätze können emuliert werden, dies gilt für fast jeden festen Zeichensatz aus dem X-System.

Terminalsoftware unter DOS- und Windows-Emulation

Wenn einmal gar nichts anderes mehr geht – unter Linux kann ältere Software für DOS oder Windows for Workgroups oft noch innerhalb einer leistungsfähigen Emulation dieser Betriebssysteme laufen. Einen sehr interessanten Ansatz ermöglicht hier das Programm *vmware*, mit dem diese Betriebssysteme sogar als weitere »Anwendung« innerhalb des laufenden Linuxsystems geführt werden, diese *virtuellen* DOS- oder Windows-Rechner können dann oft noch erfolgreich ältere Programme handhaben.

3.5.3 Schutz vor Viren

Oftmals werden Linuxserver als Mail- oder Fileserver für Netzwerke eingesetzt. Daher besteht die Wahrscheinlichkeit, dass sich auch auf Linuxsystemen Viren einnisten. Diese können dem System zwar nichts anhaben, verbreiten sich aber auf den zugreifenden Windowssystemen. Um ein Netzwerk sicherer zu machen, müssen die Arbeitsrechner und der Datenbestand auf den Dateiservern regelmäßig auf Viren untersucht werden.

Ist Linux durch Viren gefährdet?

Computerviren sind Programme, die in der Lage sind, sich in Programmcode hinein-zukopieren und sich auf diese Weise zu vermehren. Möglich ist das Vorhandensein von Schadens- oder Zerstörungsfunktionen. Die Palette möglicher Schäden reicht von unnötigem Speicherplatzverbrauch auf der Festplatte über lästige Störeffekte bei Tastatur und/oder Bildschirm bis hin zu Datenmanipulationen und -zerstörungen.

In den achtziger Jahren verbreiteten sich die Viren hauptsächlich über Programme, die von Diskette zu Diskette kopiert wurden, um schließlich durch manuelle Installation in den Computer zu gelangen. Dieser Art des Virenbefalls ist heute in den meisten Unternehmen ein wirkungsvoller Riegel vorgeschoben.

Die neueste Generation von Computer-Viren bevorzugt Benutzerdaten und nutzt, einmal in ein System eingedrungen, Standardbefehle der betreffenden Anwendung wie MS-Word-Makros oder Postscript-Befehlssequenzen.

Die zweite Neuentwicklung im Bereich der Viren betrifft die Art der Aktivität, die sie entfalten. Waren dies bis vor wenigen Jahren primär destruktive Operationen, die bis zur Löschung aller Daten führen konnten, so werden heute in zunehmendem Maße Viren entdeckt, die Systeme für einen Angriff über das Netzwerk vorbereiten, indem Systemdateien modifiziert oder Paßwörter aufgezeichnet werden.

Spezielle Linux- oder Unix-Viren sind zwar bisher nicht bekannt, die Darstellung der grundsätzlich von Viren auf das eigene Netzwerk ausgehenden Gefahren macht aber deutlich, dass hier unbedingt wirksame Schutzmechanismen eingerichtet werden müssen.

Haftung für Computerviren

Heutzutage wird man eine umfassende, dem Stand der Technik entsprechende Viren-Prüfung in der Regel als übliche (Vorsorge-) Maßnahme ansehen müssen, deren Nichteinsatz schon Fahrlässigkeit bedeutet!

Nicht nur das Unternehmen ist rechtlichen Risiken ausgesetzt, auch die im Unternehmen handelnden Personen – sowohl die Mitglieder der Geschäftsführung als auch die für die EDV verantwortlichen Mitarbeiter – können (zumindest zivilrechtlich) persönlich zur Verantwortung gezogen werden.

Niemand kann sich diesbezüglich auf seine Unkenntnis berufen; jedes betroffene Unternehmen muss, ggf. bei fehlendem eigenem Know-How, auch Dienstleistungen externer Fachbetriebe in Anspruch nehmen.

Virenscanner unter Linux

Mittlerweile gibt es eine hohe Anzahl brauchbarer Anti-Virensoftware auch für Linux. Die meisten dieser Programme sind darauf ausgelegt, Viren für DOS/Windows zu finden und zu eleminieren. Dies ist genau die Zielgruppe an Programmen, welche man für einen Fileserver ins Auge fassen sollte.

Ein automatisierter Check wird unter Linux zur einfachen Angelegenheit. Am sinnvollsten installiert man seine Checkprogramme so, dass diese automatisch Prüfungen starten. Dank *cron* ist dies jedoch kein Problem. Alle Programme lassen sich per Kommandozeile aufrufen und über Ausgabeumleitungen sollte es kein Problem sein, die erzeugten Meldungen in Log-Dateien zusammenzufassen oder direkt auf einer Konsole auszugeben.

Sinnvollerweise erweitert man die eigentlichen Anti-Virensoftware um einige kleine Sripte, so dass nur noch eventuelle Alarmierungen angezeigt werden.

Eine weitere häufige Quelle von Viren sind E-Mail-Attachments. Gerade *Makroviren* sind oft in so übertragenen Dokumenten enthalten. Von daher sollte eine automatisierte Prüfung eingegangener Texte auf Viren erfolgen. Auch hier ist für Linux Handlungsbedarf gegeben. Denn immer öfter werden Linuxserver als zentrale E-Mail-Server eingesetzt.

Nachstehend ein paar aktuelle Bezugsquellen für weitere Dokumentationen und für aktuelle Virenscanner, die auch für den Einsatz unter Linux geeignet sind:

```
http://www.ce.is.fh-furtwangen.de/~link/security/av-linux.htm
http://www.heise.de/ix/artikel/1998/02/136/default.html
http://satan.oih.rwth-aachen.de/AMaViS/amavis.html
```

3.5.4 Datensicherung

Regelmäßige Datensicherungen auf Band gehören zu den Standardarbeiten in jedem Netzwerk. Unter Linux können nahezu alle erhältlichen Bandlaufwerke (*Streamer*) verwendet werden, gleich ob diese an Floppy-, IDE- oder SCSI-Ports angeschlossen werden müssen.

Einbau und Konfiguration der Hardware

Die Konfiguration der Hardware beschränkt sich somit in der Regel auf den mechanischen Einbau des Bandlaufwerkes, nach dem nächsten Systemstart sollte die neue Komponente automatisch erkannt und im Boot-Log ausgewiesen werden. Dazu zwei Beispiele:

Das preiswerte interne *IDE-Bandlaufwerk HP Colorado 4/8 Gbyte* kann bis zu 70 MByte/min übertragen, es ist werkseitig als »IDE-Slave« konfiguriert. Das Speichermedium sind Bänder vom Typ TR-4, z.B. SONY QTR-4 (225,6 m). Dieses Laufwerk wird von Linux automatisch erkannt und ist z.B. als */dev/ht0* verfügbar.

Das SCSI-Bandlaufwerk *Python DAT* speichert 2/4 Gbyte, verwendet werden Bänder vom Typ DG-90M. Auch dieses Laufwerk wird von Linux automatisch erkannt und ist dann z.B. als */dev/st0* erfügbar.

Gerätedatei	Beschreibung
/dev/rmt0	erster SCSI-Streamer »rewinding« (spult automatisch zurück)
/dev/nrmt0	erster SCSI-Streamer »non rewinding«
/dev/ftape	ester Floppy-Streamer »rewinding«
/def/nftape	erster Floppy-Streamer »non rewinding«

Tabelle 3-21 Gerätedateien für Bandlaufwerke

Die im kommerziellen Bereich eher selten anzutreffenden *Floppy-Streamer* sind in der Konfiguration und im Betrieb nicht einfach handhabbar wie Geräte mit IDE- oder SCSI-Anschluss. Floppy-Streamer werden mit dem Kernel-Treiber *ftape* angesteuert, die Dokumentation zum gesamten *ftape*-System ist bei Bedarf unter *www.math1.rwth-aachen.de/~heine/ftape/* zu finden.

Bandlaufwerk ansteuern – mnt

Mit dem Befehl *mnt* können direkt Befehle zur Bandsteuerung an das Laufwerk gegeben werden:

```
linux01:~ # mnt -d /dev/ht0 retension
```

spannt das Band neu. Es wird dazu von Anfang bis Ende durchgespult.

```
linux01:~ # mnt -d /dev/ht0 rewind
```

spult das Band zurück.

```
linux01:~ # mnt -d /dev/ht0 eof
```

spult das Band hinter den letzten Datensatz. So können neue Daten an an bereits teilweise bespieltes Band angefügt werden.

```
linux01:~ # mnt -d /dev/ht0 erase
```

löscht das Band im Laufwerk, alle Daten auf diesem Band gehen verloren.

tar

Das Backup-Programm *tar* (*tape archiver*) ist ein universell verfügbarer und einheitlicher Standard zur Datensicherung. Das verwendete Backup-Medium muss nicht unbeding ein Bandlaufwerk sein. Es eignet sich jedes Gerät, auf das Linux über seine Gerätedateien zugreifen kann.

Der Befehl

```
tar -c -M -f /dev/fd0H1440 /daten
```

sichert in einer neuen Archivdatei das Verzeichnis */daten* und alle enthaltenen Unterverzeichnisse eine oder mehrere Disketten im Laufwerk A:.

Mit der zusätzlichen Option *-z* kann *tar* angewiesen werden, alle Dateien vor dem Sichern zu komprimieren. Aber hier ist Vorsicht geboten: Ist nur ein Block des Datenträgers fehlerhaft, wird das gesamte Backup unbrauchbar, da *tar* keine Möglichkeit bietet, Fehler im Sicherungsmedium zu überspringen. Bei einem unkomprimierten Backup ist lediglich die betroffene Datei nicht mehr zu restaurieren.

```
tar -c -N 1999-08-11 -f filename.tar /daten
```

archiviert nur die Dateien aus dem Verzeichnis */daten* im Archiv *filename.tar*, die nach dem 11. 08. 1999 geändert wurden. Damit wird bei regelmäßig durchgeführten Sicherungen Zeit und Speicherplatz eingespart.

```
tar -d -f filename.tar
```

vergleicht jede Datei im angegebenen Archiv *filename.tar* mit der Orginaldatei.

```
tar -x -f filename.tar
```

stellt alle im Archiv *filename.tar* gesicherten Dateien wieder her. Sollen nicht alle, sondern nur bestimmte Dateien wiederhergestellt werden, so wird mit dem Befehl

```
tar -x -f filename.tar /daten/text1
```

aus dem Archiv *filename.tar* nur die Datei */daten/text1* wiederhergestellt.

Bandsicherungen müssen regelmäßig auf wechselnde Sicherungsmedien durchgeführt werden. Zweckmäßigerweise werden dazu Shellskripte angelegt, die dann über *crontab* zu festgelegten Zeitpunkten ausgeführt werden. Ein typisches Sicherungsskript könnte dann z. B. so aussehen:

```
mt -f /dev/st0 rewind          # Band zurueckspulen
echo Start: `date` >> /var/log/backup.log
tar cvfz /dev/st0 /etc /home       # Daten sichern
echo Ende : `date` >> /var/log/backup.log
sleep 300
mt -f /dev/st0 rewind          # Band zurückspulen
mt -f /dev/st0 offline         # Band auswerfen
```

Dieses Sicherungsskript wird unter dem Namen

```
/etc/sicherung
```

gespeichert und mit

```
chmod 700 /etc/sicherung
```

ausführbar gemacht.

Die jetzt erforderliche zeitgesteuerte Aktion wird in */etc/crontab* eintragen: In diesem Beispiel soll die Datensicherung jeweils werktags um 02.00 Uhr erfolgen. Verwendet werden soll das Shell-Skript */etc/sicherung*:

```
00 02 * * 1-5 root /etc/sicherung
```

kdat

Das KDE Programm *kdat* ist die grafische Oberfläche für den Befehl *tar*. Es enthält eine einfache Bedieneroberfläche, die Planung der unterschiedlicher Sicherungsläufe kann über *Sicherungsprofile* automatisiert werden.

Wird *kdat* erstmalig verwendet, erfolgt zunächst die Auswahl des Bandlaufwerkes über das Menü *Bearbeiten -> Einstellungen -> Bandgerät*. Ein neues Band wird dann zunächst von *kdat* formatiert (Abbildung 3-74).

Abbildung 3-74 Bandformatierung mit kdat

Grundsätzlich ist es sogar möglich, mehrere Sicherungsdateien auf einem Band zu erstellen und zu verwalten. Die Konfiguration des Umfanges der Datensicherung erfolgt zweckmäßigerweise über *Sicherungsprofile*, um die einmal getroffenen Einstellungen unter einem vorgegebenen Namen jederzeit wieder abrufen zu können.

Sind die zu sichernden Verzeichnisse festgelegt, kann der eigentliche Sicherungslauf gestartet werden. *kdat* überprüft zunächst, ob der angegebene Umfang tatsächlich auf das Band passt. Reicht die Speicherkapazität aus, so beginnt die eigentliche Sicherung (siehe Abbildung 3-75).

Während der laufenden Datensicherung zeigt *kdat* die aktuell bearbeitenden Verzeichnisse/Dateien, zusätzlich geben Statusinformationen die bereits geschriebene Datenmenge bzw. die voraussichtliche Dauer des Sicherungslaufes.

Sollen auf einem Band gesicherte Daten mit *kdat* wiederhergestellt werden, können der gesamte Inhalt des Sicherungsbandes oder auch gezielt nur einzelne Verzeichnisse bzw. Dateien ausgewählt werden (Abbildung 3-76).

Im obigen Beispiel ist nur das Verzeichnis */home/samulat* mit allen darin enthaltenen Dateien ausgewählt und wird vom Band auf die Festplatte zurückgeschrieben.

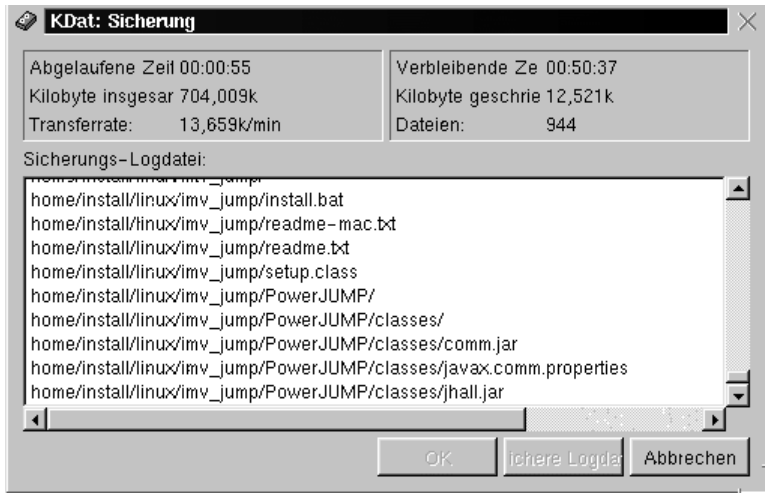


Abbildung 3-75 Datensicherung mit kdat

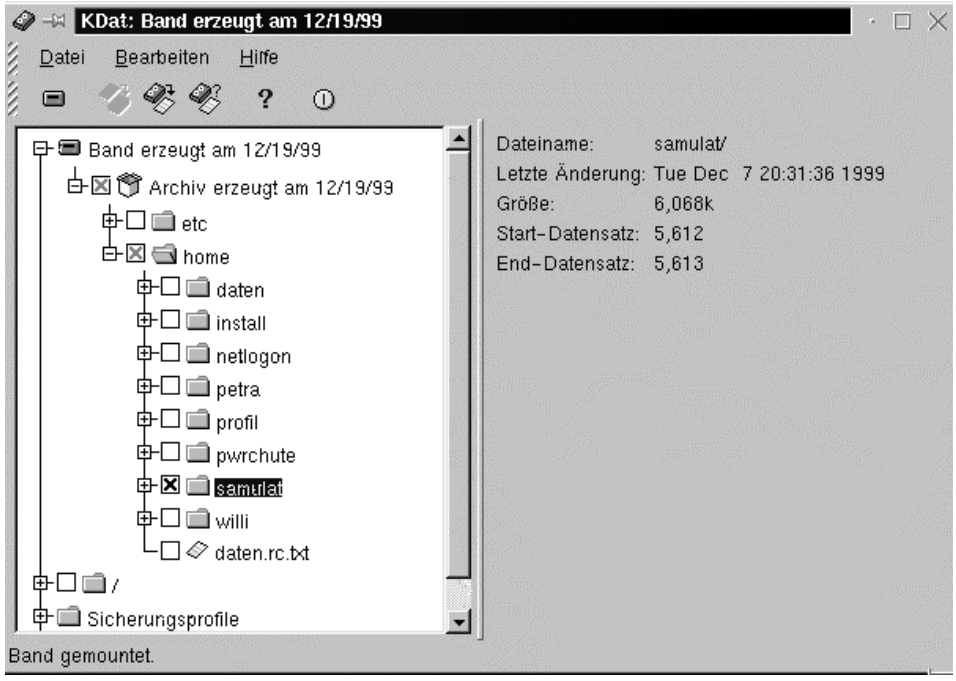


Abbildung 3-76 Wiederherstellung von Datenbeständen mit kdat

3.5.5 Netzwerkverwaltung

Für die Verwendung unter Linux sind eine Vielzahl leistungsfähiger Tools zur Netzwerkverwaltung verfügbar, der Umfang der angebotenen Funktionen ist beeindruckend:

- Grafische Darstellung der aktuellen Netzwerkstruktur mit allen aktiven Komponenten. Die tägliche Netzwerkarbeit sehr erleichtern kann auch die Steuerung von Test- und Diagnosefunktionen (z.B. *ping, tracert, ...*) durch einfache grafische Selektion der gewünschten Komponenten.
- Inventarisierung von Hard- und Software. Datenbankgestützter Nachweis der aktuell im Netzwerk eingesetzten Komponenten bis herunter zu der Hardwarekonfiguration in den einzelnen Rechnern.
- Prüfung der technischen Netzwerkfunktionen. Erkennen von Engpässen und rechtzeitige Warnung durch ständige Überwachung der technischen Parameter der einzelnen aktiven Netzwerkkomponenten. Erstellen von Netzwerkstatistiken, Ermittlung von Lastverteilungen und Anzeige des aktuellen Systemzustandes als Diagramm in der grafischen Darstellung der Netzwerkumgebung.
- Einsatz als »Network Control Server«: Netzwerk-Monitor.
- Implementierung eines IDS-Tools (*Intrusion Detection System*), das den laufenden Netzbetrieb überwacht, filtert und bei unzulässigen Aktionen Alarm schlägt
- Überwachung und Darstellung von dynamischen Parametern auf entfernten Hosts, wie CPU-Auslastung, Festplattenzugriffe, Swapping, Prozesse, Interrupts und vieles mehr
- Bearbeitung und netzwerkweite Verteilung von Konfigurationsdateien. Installation und Update von Programmen, Führung von Lizenznachweisen.

Realisiert werden diese netzwerkweiten Überwachungsfunktionen durch den Einsatz von RPC-Daemons und dem *Simple Network Management Protocol* SNMP, wobei grafische Frontends zur Präsentation des gesammelten Daten eingesetzt werden. Auch hier kommen immer mehr unter Java programmierte Oberflächen zum Einsatz, die die Anzeige der Informationen mit jedem beliebigen WWW-Browser ermöglichen.

Einge Beispiele sollen nachfolgend dargestellt werden, eine vollständige Abhandlung ist an dieser Stelle nicht möglich. Freie Programme wie GxSNMP (GNOME), Scotty/tkined, Cheops, sysmon, mon, InetRover, NOCOL, BigBrother, Pong3, Son of Pong, MyNMS, NORDUnet, HNMS, Halcyon und viele andere sind für Linux verfügbar und können im praktischen Betrieb eine sehr ernstzunehmende Konkurrenz zu kommerziellen Paketen darstellen.

RPC-Daemon

Das Verzeichnis */util/perf-rstat* der CD-ROM enthält den RPC-Daemon *rstatd* mit dem grafischen Frontend *perf* (*Performance Monitor*). Sie ermöglichen den direkten Zugriff und die Anzeige wichtiger Leistungsparameter entfernter Hosts, z.B. die CPU-Auslastung und Statistiken der Netzwerk-Interfaces einschließlich Kollisionserkennung auf dem Übertragungsmedium.

SNMP

In den 80er Jahren wurde das *Simple Network Management Protocol* (SNMP) für die einheitliche Verwaltung verschiedener Netzwerktypen entwickelt. SNMP operiert auf der Anwendungsebene, unter Einsatz von TCP/IP-Transportprotokollen, sodass es unabhängig von der zugrundeliegenden Netzwerk-Hardware arbeitet. SNMP ist in den RFCs 1155, 1157 und 1901 bis 1908 als Internet-Standard definiert.

SNMP ist eine Client-Server-Architektur, in der der *Agent* den Server und der *Manager* den Client repräsentieren.

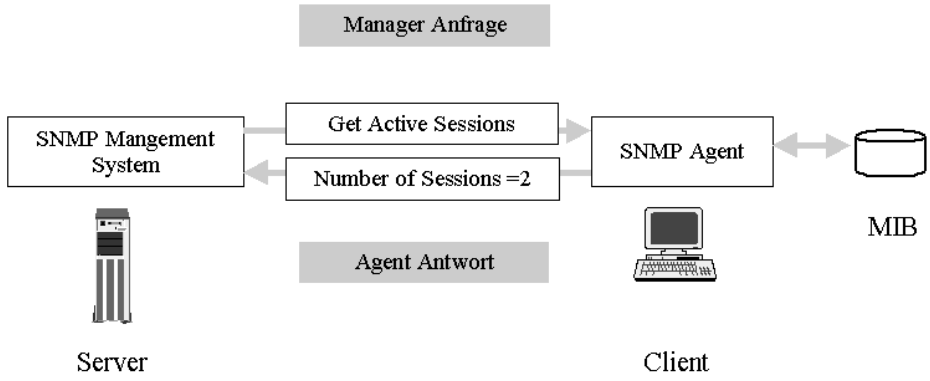


Abbildung 3-77 Kommunkation zwischen SNMP-Manager und Agent

Der *Agent* ist ein Programm, das auf jedem überwachten oder verwalteten Netzwerkelement läuft, es kann auf alle Elemente der Gerätekonfiguration zugreifen und aktuelle Statusinformationen abrufen. Die so ermittelten Daten werden in der *Management Information Base* (MIB) gespeichert. Bekannt ist vor allem die MIB-2 (RFC 1213), dieVariablen für das TCP/IP-Protokoll enthält.

Die *Manager*-Software läuft auf der Überwachungsstation des Netzwerkes, von hier werden die aktiven Agenten im Netzwerk abgefragt und die Daten übernommen.

Zusätzlich gibt es im SNMP-Befehlssatz ein spezielles Kommando namens `trap`, das dem Agenten erlaubt, auch unaufgefordert Daten an den Manager zu senden. Dadurch wird dieser über Ereignisse, wie z.B. Fehler oder Systemneustarts informiert.

Prinzipiell ist SNMP ein einfaches Protokoll, solange alle Aktionen auf Basis des Prinzips von Anfrage und Speicherung beruhen. Der Manager kann nur zwei verschiedene Operationen ausführen: den Wert einer *MIB*-Variable des Agenten erfragen (*Get-Request*) oder ihn setzen (*Set-Request*). Ein Kommando, das als Antwort auf einen *Get-Request* gedacht ist (*Get-Response*) wird nur vom Agenten verwendet.

Die Flexibiliät von SNMP hängt direkt mit der Fähigkeit der *MIB* zusammen, neue Elemente verwalten zu können. Soll z.B. ein neuer Router neue Informationen liefern oder einen erweiterten Befehlssatz erhalten, so muss der Hersteller die entsprechenden Variablen dem Datenbestand hinzufügen.

Viele Hersteller bieten für ihre aktiven Netzwerkkomponenten bereits serienmäßig SNMP-Agenten an, auch für fast alle Netzwerkbetriebssysteme sind SNMP-Agenten verfügbar oder gehören bereits zum Standardumfang.

Installation und Start von SNMP unter Linux

Um einen SNMP-Agent und Managerprogramme unter Linux einsetzen zu können, wird das Programmpaket *snmp* aus der Serie *n* benötigt.

Installiert wird *CMU SNMP* in der Version 3.6-26 (0,6 Mbyte). Unterstützt werden SNMP Version 1 (SNMPv1) und SNMP Version 2 (SNMPv2). *CMU SNMP* enthält einen SNMPv1/SNMPv2-Agenten und eine Reihe von Kommandozeilenwerkzeugen. SNMPv3 wird (noch) nicht unterstützt.

Zum automatischen Start des *CMU-SNMP*-Daemons ist der Eintrag

```
START_SNMPD="YES"
```

in */etc/rc.config* erforderlich. Die Konfigurationsdatei ist */etc/snmpd.conf*, hier sind im Wesentlichen die Codes für die Bezeichner in der MIB-Datenbank enthalten. Einzelheiten dazu können am besten den entsprechenden RFCs entnommen werden (auf der CD-ROM in */usr/doc/rfc*).

SNMP-Utilities

Tabelle 3-22 zeigt eine Auswahl der elementaren Utilities für den Zugriff auf die MIB-Datenbank.

Zu jedem Programm aus Tabelle 3-22 existiert eine gleichnamige Man-Page, über die bei Bedarf weitere Informationen abgerufen werden können.

Die direkte Verwendung dieser SNMP-Kommandozeilenbefehle ist aber eher die Ausnahme, in der Regel wird eine der nachfolgend besprochenen Applikationen eingesetzt werden, die dann oft direkt auf diesen Befehlen aufsetzen.

Name	Beschreibung
snmpget	Datenbankabfrage (GET Request)
snmpnetstat	Ähnlich *netstat*, zahlreiche Optionen sind möglich.
snmpset	Werte aktualisieren (SET-Request)
snmptest	einfache Bedieneroberfläche für GET, GET NEXT und SET-Requests
snmptranslate	Hilfsprogramm zur Umsetzung von Objektnamen (nur SNMPv2)
snmptrap	Benachrichtigung, wenn der Agent Systemereignisse selständig meldet
snmptrapd	Empfang und Protokollierung von Trap-Nachrichten (läuft als Daemon)
snmpwalk	inkrementelle Abfrage des MIB Informationsbaumes

Tabelle 3-22 SNMP-Utilities (Auswahl)

Network Flight Recorder NFR

Der *Network Flight Recorder* NFR der amerikanischen Firma NFR Software ist ein lesitungsfähiges Tool zur automatisierten Überwachung des gesamten TCP/IP-Netzwerkverkehrs. Das Programm »hört die übertragenen Netzwerkdaten mit«, dazu wird, wie z. B. auch beim Programm *tcpdump*, eine im *promiscuous Mode* betriebene Netzwerkkarte verwendet.

NFR ermöglicht den Schutz des eigenen Netzwerkes vor immer neuen Angriffstechniken durch eine gezielte Überwachung, ohne hierzu Eingriffe im System selber notwendig zu machen. Zusätzlich können auch Routineaufgaben im Netzwerk, wie die Abrechnung auf Basis der tatsächlichen Netzwerknutzung, aufgesplittet nach Service und Abteilung, mit diesem Programm einfach realisiert werden.

NFR (*www.nfr.com*) erlaubt den kostenlosen Download des Programmes und Gebrauch der Quellen, solange ein Unternehmen das Tool im eigenen Hause zu nichtkommerziellen Zwecken einsetzt. Veränderungen und Erweiterungen für den eigenen Gebruach sind zulässig, die Lizenz darf allerdings nicht weitergegeben werden, sie muss in jedem Fall direkt von NFR bezogen werden [Fey99].

NFR ist zwar nicht als IDS-Tool (*Intrusion Detection Software*) entwickelt worden, eignet sich aber hervorragend dazu, unerlaubte Eindringversuche aus einem externen Netzwerk oder aus dem eigenen LAN festzustellen und Alarm zu schlagen. Der Anwender kann frei entscheiden, welche Daten und Netzwerkereignisse interessant sind, wie diese Informationen gefiltert und wie auf bestimmte Vorkommnisse reagiert werden soll. Hierzu kann NFR mit der eigenen Scripting-Sprache »N« programmiert und durch beliebige Module erweitert werden. NFR fordert eine hohe Rechenleistung, für die Protokolldateien müssen mehrere GByte freie Kapazität vorhanden sein.

Eine ausführliche Beschreibung des Programmes NFR finden Sie auf der CD-ROM im Verzeichnis */usr/doc/nfr* (Quelle: *www.nfr.com/nfr/).*

Big Brother

Big Brother verwendet einen lokal auf den Clients installierten Dienst, der die jeweiligen Systemparameter ermittelt und ständig die Verfügbarkeit der wichtigsten Netzwerkdienste prüft. Die so ermittelten Statusinformationen werden zu einem oder mehreren Auswerterechnern übertragen:

- *DISPLAY Server*: Er zeigt den aktuellen Zustand in Form einer einfachen WWW-Grafik.
- *PAGER Server*: Er meldet dem Systemveralter aktuelle Probleme im Netzwerk.

Spong

Ein einfaches Netzwerkmonitoring kann durch das Softwarepaket *spong* realisiert werden (*http://www.edsgarage.com/projects/spong/index.html*), das nach dem Client-Server-Prinzip arbeitet (Abbildung 2-78).

Ein auf dem *Control-Server* laufender Serverprozess prüft in einstellbaren Zeitabständen die Erreichbarkeit vorgegebener Rechner auf der Applikationsebene. Damit wird nicht nur die Erreichbarkeit eines Clients über das Netzwerk geprüft, sondern auch, ob die geforderten Dienste (FTP, WWW, ...) in Funktion sind.

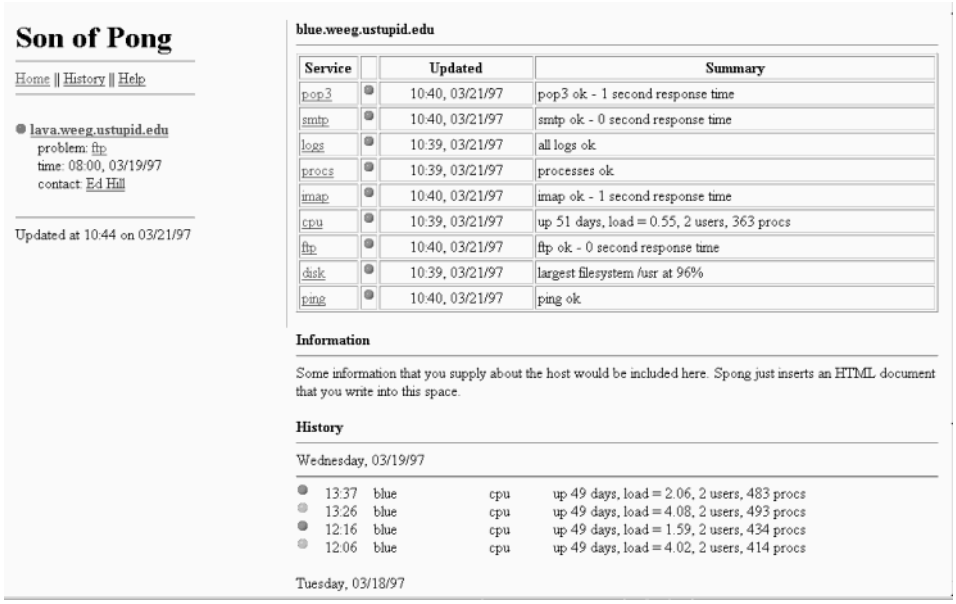


Abbildung 3-78 Darstellung von Host-Informationen mit spong

Der Test ist unabhängig davon, unter welchem Betriebssystem der Client läuft. Zusätzlich kann auf jedem Client über ein Perlskript eine Funktion installiert werden, die Systemdaten ermittelt und diese zum Control-Server überträgt.

cheops

Das Programm *cheops* (*http://www.marko.net/cheops*) ermittelt die Host-Betriebssysteme und zeigt die Routen zu den einzelnen Rechnern im Netzwerk. Automatisch werden alle auf einem Host verfügbareren Dienste ermittelt (*Port-Scanner*). Mit einem SNMP-Browser können alle Arten von Informationen von den entsprechenden Systemen eingeholt werden. Enthalten ist auch ein Netzwerkmonitor, der temporäre Ausfälle erkennt und anzeigt.

scotty und tkined

Von der TU Braunschweig (Autor: Jürgen Schönwalder) kommen die Programme *scotty* und *tkined*. Ähnlich einem vektororientierten Zeichenprogramm ist *tkined* ein grafischer Editor für Netzwerkstrukturen, das Programm arbeitet unter XWindows auf Basis von *Tcl* und *Tk*.

Eine mit *tkined* gezeichnete Netzwerkkarte (*map*) ist nicht nur ein »Plan« des eigenen Netzwerkes; jedes Objekt kann mit dynamisch aktualisierten Diagrammen versehen werden, die aktuelle Netzwerk- oder Rechnerparameter anzeigen, z.B. die Auslastung eines Netz-Interfaces, eines Netzwerksegmentes oder auch die CPU-Auslastung eines Servers.

Installation und Start

Installiert werden *scotty* und *tkined* über das Programmpaket *scotty* aus der Serie *tcl*. Aktuelle Programmversionen können auch über *ftp.ibr.cs.tu-bs.de* bezogen werden.

Das Programm *tkined* kann danach sofort in einem X-Terminal gestartet werden.

Netzwerk-map erstellen

Tkinetd kann ein vorgegebenes IP-Netzwerksegment nach aktiven Komponenten durchsuchen (*scannen*) und das Ergebnis grafisch darstellen. Mit etwas manueller Nachbearbeitung kann auf diese Art schnell eine grafische Ansicht des eigenen Netzwerkes erstellt werden.

Zunächst muss über das Menü *Tools -> IP-Discover* der neue Menüpunkt *IP-Discover* erstellt werden, danach wird *IP Discover -> Discover IP Network* ausgewählt. Es erscheint eine Form, in der das zu durchsuchende Netzwerk angegeben wird (Abbildung 3-79).

Abschließende Nullen brauchen nicht mit angegeben werden. Im hier gezeigten Beispiel soll *tkined* das Netzwerk 192.168.100.0 durchsuchen und die gefundenen aktiven Komponenten anzeigen. Nach ein paar Sekunden oder auch wenigen Minuten wird dann die aktuelle Netzwerkstruktur und alle automatisch ermittelten Daten grafisch dargestellt (Abbildung 3-80).

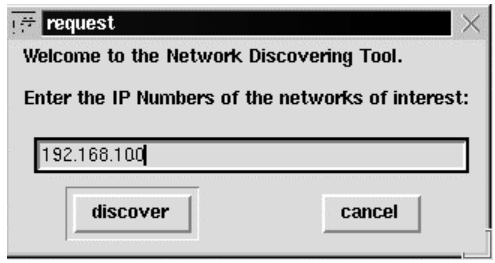


Abbildung 3-79 Netzwerk auswählen

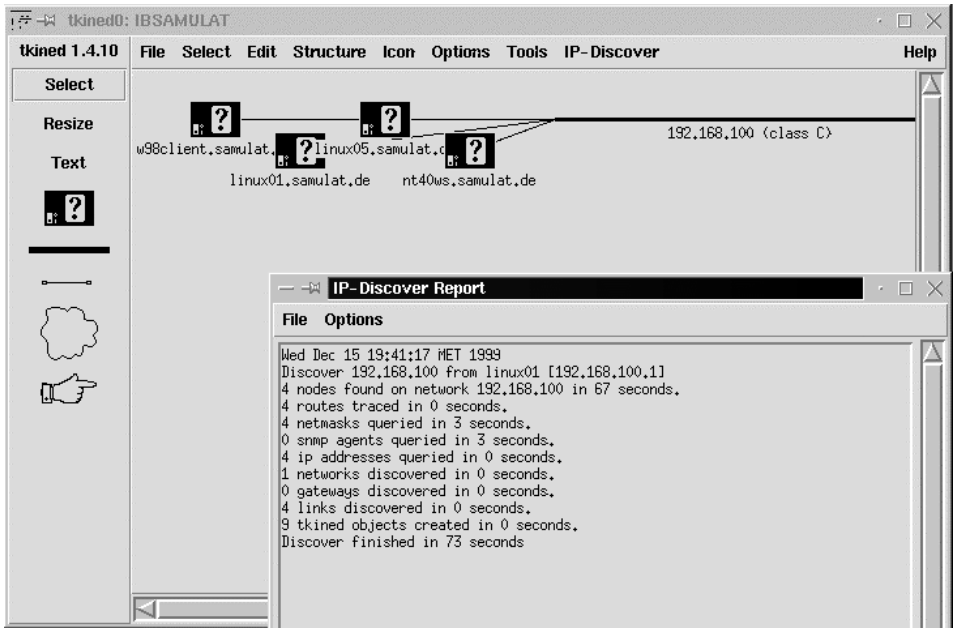


Abbildung 3-80 Netzwerk »Discover« mit tkined

Das Netzwerk 192.168.100.0 enthält vier aktive Komponenten (*Nodes*) mit den hier angegebenen Namen. Das Abbildung sollte jetzt manuell weiter bearbeitet werden, die typischen Arbeitsgänge können per Mausklick durchgeführt werden.

Linke Maustaste	Mittlere Maustaste	Rechte Maustaste
Objekte neu erstellen Objekte selektieren bei Netzwerken und Diagrammen die Größe verändern	selektierte Objekte bewegen	Kontextmenü des selektierten Objektes aufrufen

Tabelle 3-23 Steuerung von tkined

Nachdem die Netzwerkobjekte neu plaziert wurden, können zunächst verwendeten Symbole für die aktiven Netzwerkkomponenten über das Menü *Icon* durch Bilder ersetzt werden, die die tatsächliche Funktion oder den Gerätetyp darstellen. Mit wenigen Arbeitsschritten ensteht so auf Basis der automatisch ermittelten Daten ein aussagekräftiger Plan der Netzwerkumgebung (Abbildung 3-81).

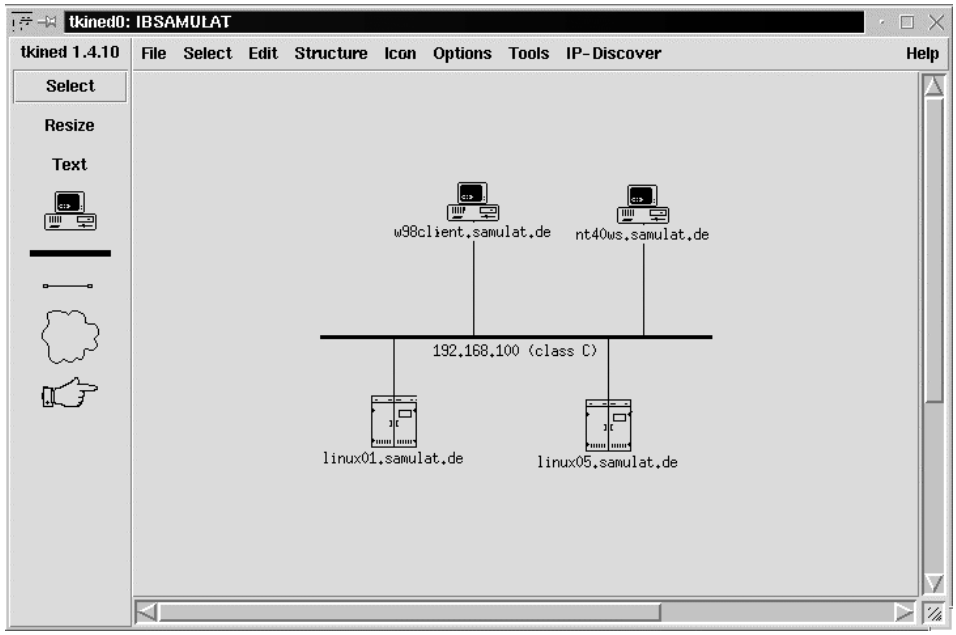


Abbildung 3-81 Überarbeiteter Netzwerkplan für IBSAMULAT

Aktuelle Netzwerkparameter abfragen

Um Netzwerkparameter abzufragen, wird zunächst ein einzelnes oder auch mehrere Objekte in der aktuellen *map* selektiert. Die danach angeforderte Information oder Aktion bezieht sich dann auf die aktuell selektierten Objekte. Sollen alle Objekte markiert werden, reicht dazu die Betätigung der Taste [A].

Mit *IP-Discover -> RPC-Server* oder *IP-Discover -> TCP-Server* kann gezielt nach den entsprechenden Diensten gesucht werden. Der Menüpunkt IP-trouble (erstellt mit *Tools -> IP-Trouble) stellt* Werkzeuge zur Verfügung, mit denen die TCP/IP-Netzwerkverbindungen schnell und effektiv geprüft werden können. Ein Ping auf einen einzelnen oder auf alle markierten Netzwerkrechner ist schnell ausgeführt und ausgewertet. Damit können Netzwerkprobleme und aktuelle Engpässe schnell analysiert und Fehlerursachen gesucht werden.

Dynamische Anzeige von Netzwerk- und Rechnerparametern

Von der *scotty*-Homepage stammt der in Abbildung 3-82 dargestellte fiktive Ausschnitt aus dem Netzwerk der Uni Braunschweig.

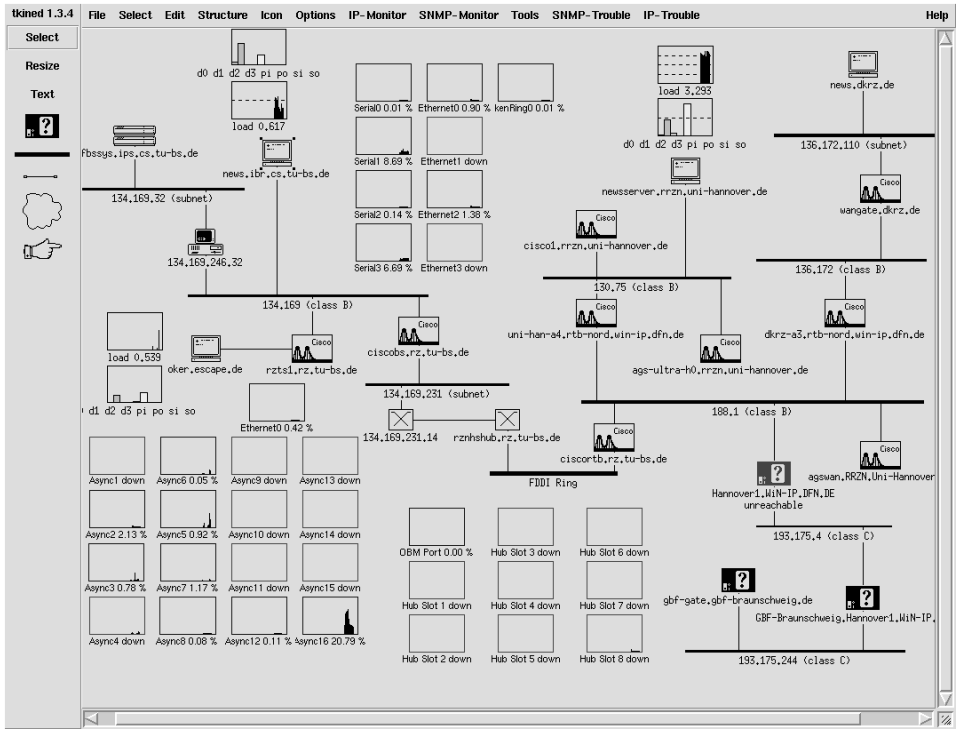


Abbildung 3-82 Netzwerkplan – ein Beispiel von der scotty-Homepage

Ganze Teilnetze wurden in dieser Ansicht zu »Icons« geschrumpft, um die Gesamtdarstellung übersichtlich zu halten. Diese Icons können jederzeit wieder expandiert werden. Über *Tools -> IP-Monitor* wurden vielen Netzwerkobjekten kleine Anzeigefelder zugeordnet, in denen Meßwerte in Form von Balkendiagrammen (*Barchart*) oder Kurven (*Stripchart*) angezeigt werden. Diese Darstellungen werden in der Ansicht dynamisch aktualisiert, alle wichtigen Netzwerkparameter können auf einen Blick erfasst werden.

3.6 Internetdienste

Die grundlegende Struktur PC-basierter Netzwerke hat sich in den vergangenen Jahren erheblich gewandelt. Waren es anfänglich noch relativ kleine Netzwerke, in denen oft ein einzelner Server unter Novell Netware 3.12 die zentralen Datei- und Druckdienste für wenige PC-Arbeitsplätze bereitstellten, so werden heute die

Netzwerkstrukturen zunehmend größer. Immer häufer bilden aus dem Internet bekannte Dienste wie E-Mail und Webserver zentrale Bausteine der eigenen Kommunkationsstruktur.

Intranet-Strukturen bieten die Möglichkeit, eine über die Jahre gewachsene inhomogene Rechner- und Betriebssystemumgebung tatsächlich weiterverwenden zu können. Mit den auf fast alle Betriebssystemen verfügbaren Internet-Browsern können Standard-Netzwerkdienste einfach gehandhabt werden. Selbst in der Schule lernt heute jeder schon der Umgang mit dem Informationssystem Internet. Die Verwendung von E-Mail, der Zugriff auf Informationen über Suchmaschinen und die Präsentation der eigenen Arbeitsergebnisse als Webseiten – fast jeder Schüler hat damit heute seine Erfahrungen gemacht. Was Teil der Ausbildung ist, wird natürlich auch am Arbeitsplatz erwartet. Das Internet zeigt Arbeitsmechanismen auf, die fast immer auf den eigenen Tätigkeitsbereich anzuwenden sind – die Standardwerkzeuge des Internet werden selbstverständlich für die eigene Arbeitsorganisation gefordert.

Internetdienste bereitszustellen, das ist immer mehr auch einer der Gründe zur Nutzung von linuxbasierten Systemen. Mit diesem preiswerten Betriebsssystem, ergänzt durch alle für den Internetbetrieb wichtigen Programme und Serverdienste können die PC-Netzwerkbetriebssysteme Novell oder Windows NT kaum konkurrieren.

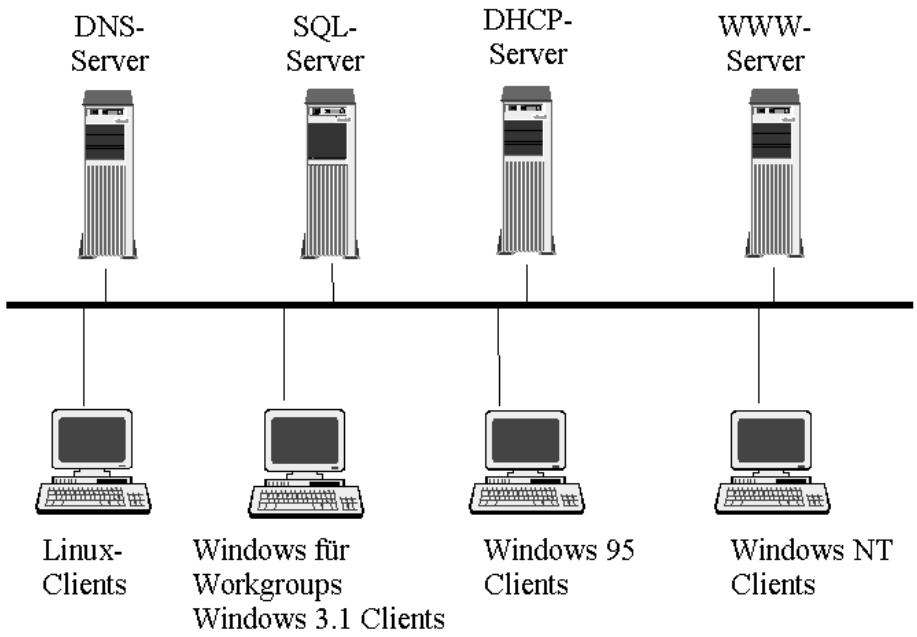


Abbildung 3-83 Typische Intranet Komponenten

Was soll jetzt unser Linux-Server tatsächlich leisten? Zunächst soll er die zentrale Komponente im Intranet sein, d.h. Internetdienste bereitstellen, die primär im lokalen Netzwerk verfügbar sind. Eine Verbindung in das Internet ist hier noch nicht gefordert; möglich sein sollte es aber bereits, vom eigenen Webserver Informationen abzuholen und mit einem Internetbrowser wie *Netscape* oder *Microsoft Internet Explorer* am PC-Netzwerkarbeitsplatz anzuzeigen. Sollen diese Informationen dynamisch mit Datenbankgespeicherten Firmendaten verknüpft werden, ist die Kopplung dieser beiden serverbasierten Dienste notwendig. Natürlich bieten auch Systeme unter Novell und Windows NT diese Möglichkeiten, nur dann müssen zumindest die dafür notwendigen Datenbanksysteme und die erforderlichen Lizenzen gekauft werden.

In weiteren Ausbauschritten erfolgt dann die Verbindung des eigenen Netzwerkes mit dem Internet. Über ISDN wird eine Verbindung zu einem Provider hergestellt; ab sofort ist die gemeinsame Nutzung von externen Internetdiensten möglich. Hier wird aber auch die Konfiguration des Linux-Servers zunehmend komplexer: Proxy-Server, die Absicherung des eigenen Netzwerkes vor unbefugtem Zugriff bis hin zum Firewall-System und die Verwendung von E-Mail als zentrale interne und externe Kommunkationskomponente erfordern einen immer höheren Arbeitsaufwand. Um stabile und im laufenden Betrieb sicher zu administriende Systeme erstellen zu können, sollen hier praxisorientierte Beispiele über die typischen Installations- und Konfigurationshürden hinweghelfen. Damit wird die vollständige Systemoptimierung nicht in jedem Fall im Vordergrund stehen können, wichtig ist eine funktionierende und ausreichend abgesichert Lösung.

Wo wird der Weg »Internet« hinführen? Für viele Firmen eröffnen sich über das Internet Möglichkeiten zur Präsentation und zum Verkauf der eigenen Produkte, die nachhaltig durch die Nutzung der neuen Techniken bestimmt werden. Der Weg vom internen Informationssystem über die gemeinsame Nutzung des Internet kann bis zum E-Commerce-System führen, in dem dann auch Kundenaktionen elektronisch abgewickelt werden können. Gerade Serversysteme unter Linux bieten hier hervorragende und vergleichsweise preiswerte Einstiegsmöglichkeiten. Leistungsfähige und praxiserprobte E-Commerce-Systeme sind bereits für einige tausend Mark zu haben. Die konkurrierenden Systeme beginnen nicht selten erst mit Einstiegspreisen im sechstelligen Bereich.

3.6.1 Webserver Apache konfigurieren

Die Funktion eines Webservers (*WWW-Server*) im Intranet oder Internet wahrzunehmen gehört heute zu den Standardaufgaben jedes Linux-Servers. Vermutet wird, dass bereits die Hälfte aller Webserver im Internet unter Linux laufen! Der unter Linux am häufigsten verwendete Webserver ist *Apache* (*http://www.apache.org*), der unter GNU GPL frei verfügbar ist.

Der WWW-Server *Apache* ist modular aufgebaut. Die Funktionen könen so auch während der Laufzeit durch das Hinzufügen neuer Module erweitert werden. Die Liste der verfügbaren Module ist inzwischen so umfangreich, dass nahezu jede Funktion eines Webservers realisierbar ist.

Installation und Grundkonfiguration

Der WWW-Server *Apache* gehört zum Standard-Lieferumfang fast jeder Linux-Distribution und wird im Rahmen der Grundinstallation mit eingerichtet. Der automatische Start erfolgt mit dem Eintrag

```
START_HTTPD="YES"
```

in */etc/rc.config*. Die Konfiguration des Apache erfolgt direkt in der Konfigurationsdatei */etc/httpd/httpd.conf* oder über spezielle Administrationswerkzeuge, wie z. B. das bereits mehrfach verwendete *webmin*.

Betrieb

Apache kann von jedem WWW-Browser über die Angabe des Hostnamens oder der IP-Adresse erreicht werden, standardmäßig wird zunächst die Startseite der *Apache*-Dokumentation angezeigt. In der Grundkonfiguration ist die Startseite unter dem Namen *index.html* in */usr/local/httpd/htdocs* gespeichert und kann einfach gegen einen eigenen Text ausgetauscht werden.

Bei Bedarf kann die Pfadangabe zur Startseite in */etc/httpd/httpd.conf* geändert werden, dazu muss nur der Eintrag

```
DocumentRoot   /usr/local/httpd/htdocs
```

auf den gewünschten Pfad geändert werden.

Private Benutzerseiten

Jedem Benutzer des Linux-Servers kann eine eigene WWW-Seite zugewiesen werden, die dieser dann selbst editieren darf. Der Zugriff erfolgt dann z. B. über die Adresse

http://www.linux01.de/samulat/index.html

Die HTML-Dateien sollen dabei in einem Verzeichnis unterhalb des jeweiligen Benutzer-Homeverzeichnisses liegen. In */etc/httpd/httpd.conf* ist dazu der Eintrag

```
UserDir   /home/*/public_html
```

notwendig. Für jeden Benutzer muss jetzt ein Verzeichnis mit dem Namen *~/public_html* existieren, dort können dann die HTML-Seiten gespeichert werden.

Kennwortabfrage

Sollen einzelne Seiten vor unbefugtem Zugriff geschützt werden, so kann *Apache* eine Benutzeranmeldung mit Name und Kennwort erzwingen. Dazu muss zunächst eine Datei mit dem Namen *.htaccess* im zu schützenden Verzeichnis angelegt werden. Dort werden die vier Zeilen

```
AuthType Basic
AuthName "Bitte geben Sie Benutzernamen und Kennwort an"
AuthUserFile /etc/httpd/conf/users
require vaild-user
```

eingetragen. Die Zeile *AuthType Basic* legt die Autorisierungsmethode fest, danach folgt ein Text, der bei der Abfrage angezeigt wird (*AuthName*). *AuthUserFile* gibt an, aus welcher Datei *Apache* die erlaubten Kombinationen von Benutzername und Kennwort lesen soll; *require valid-user* legt abschließend fest, dass jeder Benutzer zugelassen wird, der in */etc/htrtpd/conf/users* über einen Eintrag verfügt. Mit

```
# htpasswd -c /etc/httpd/conf/users michael
```

wird die Kennwortdatei neu angelegt, der Benutzereintrag für *michael* wird vorgenommen, sein Kennwort wird abgefragt. Alle weiteren Benutzereinträge erfolgen dann mit

```
# htpasswd /etc/httpd/conf/users <benutzername>
```

Administration über webmin

Auch zur Administration des WWW-Servers *Apache* kann *webmin* sehr gut verwendet werden. Alle schon dargestellten Beispiele können über diese grafische Oberfläche realisiert werden. *Webmin* erlaubt aber auch den dirketen Zugriff auf weitere Konfigurationsdetails (siehe Abbildung 3-84):

Apache ermöglicht es, auf einem Server mehrere, voneinander unabhängige virtuelle WWW-Server zu konfigurieren, die dann unter einer jeweils eigenen IP-Adresse reagieren. Diese Konfiguration manuell auszuführen ist schwierig, *webmin* bietet mit der Funktion *Create a New Virtual Server* dazu eine vergleichsweise einfache Möglichkeit.

Test

Die Logdateien des Webservers

/var/log/httpd.access_log
/var/log/httpd.error_log

können am einfachsten mit Programmen wie z. B. dem *httpd-analyzer* (*http://www.netstore.de*) ausgewertet werden. Gezeigt wird, aus welchen Domains die meisten Zugriffe kommen, welche der eigenen Seiten am häufigsten aufgerufen werden oder von welchen anderen Webservern »Besucher« kamen.

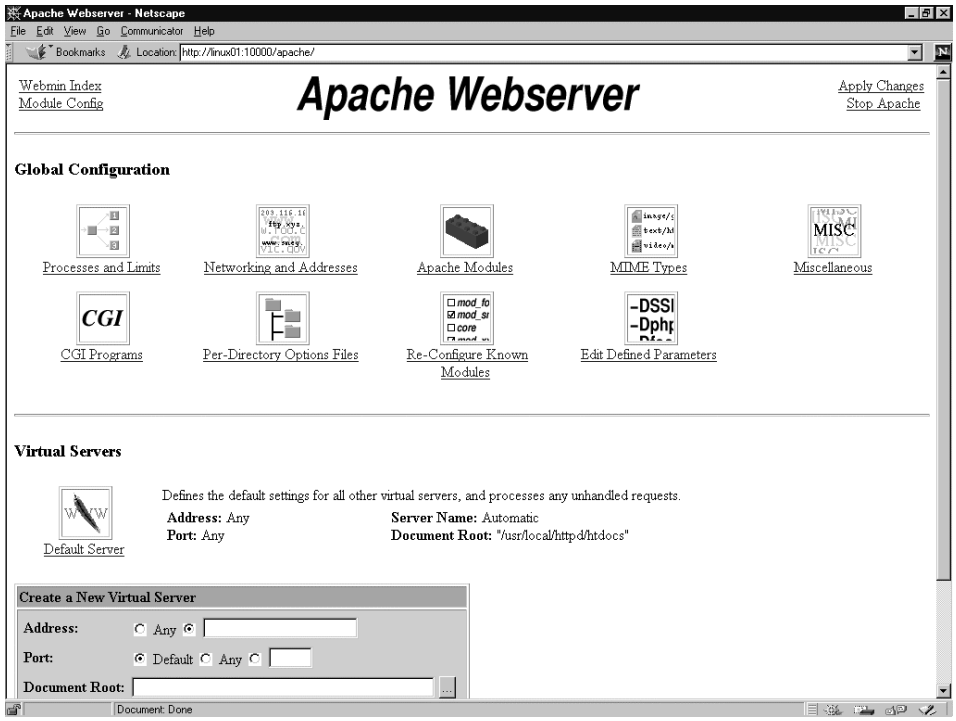


Abbildung 3-84 Administration von Apache über webmin

ISDN-Router

Zur Anbindung des eigenen Netzwerks an das Internet oder an andere entfernte Netzwerke gibt es auf dem Markt eine Reihe von fertigen Lösungen: Low-Cost-Router oder sogenannte Internet-Access-Router ermöglichen dies schon zu Preisen ab ca. 700 DM. Diese Geräte verfügen über einen ISDN- und einen Netzwerkanschluss, werden mit wenig Aufwand konfiguriert und stellen für kleine Netzwerke bereits leistungsfähige und einfach hanzuhabende Lösungen dar. Oft werden auch PC-Netzwerkarbeitsplätze mit einer ISDN-Karte ausgestattet. Die hier mitgelieferte Software stellt dann für diesen Arbeistplatz oder auch für ganze Teilnetzwerke die Verbindung zum Internet her (neben anderen Leitungen wie Faxserver, Telefondiesten, ...).

Für professionelle Netzwerke ist aber deutlich mehr an Leistungsumfang zu fordern. Der Router soll

- nicht nur eine Verbindung herstellen, sondern mehrere, möglichst sogar beliebig viele Teilnetze bei Bedarf automatisch über vorgegebene Wähl- oder Standleitungen verbinden.

- alternative Verbindungswege nutzen können. Dazu sind eine Reihe von Verbindungsparametern zu sammeln und auszuwerten, z. B. die aktuelle Verfügbarkeit, die bisher erreichten Übertragungsgeschwindigkeiten oder Fehlerhäufigkeiten.
- das eigene Netzwerk vor unbefugtem Zugriff schützen. Dazu sind Identifikationsmechanismen ebenso notwendig wie auch die Möglichkeit, den ein- und ausgehenden Datenstrom in Menge und Inhalt zu überwachen. Unerwünschte Dienste müssen abgeschaltet werden.

Einfache Router können diesen Anforderungen nicht mehr gerecht werden; wesentlich teurere Geräte bieten dann diese Leistungen an.

Die Möglichkeit bietet sich hier an, einen Rechner unter Linux als Router einzusetzen. Jeder Linux-Server, der ein LAN an das Internet oder an ein entferntes Netzwerk anbindet, stellt grundsätzlich nichts anderes dar als einen klassischen Router. Mit entsprechend leistungsfähiger Software sollte es möglich sein, alle Anforderungen erfüllen zu können.

Benötigt werden mindestens zwei Netzwerkinterfaces: die Netzwerkkarte für das eigene Netzwerk und eine ISDN-Karte zum Anschluss an das Telefonnetz. Nach der Installation und Einrichtung der Hardware muss dann die Routing-Software die Daten weiterleiten.

Konfiguration der ISDN-Karte

Um eine oder mehrere ISDN-Karten unter Linux verwenden zu können, müssen diese zunächst eingebaut und konfiguriert werden können. Linux unterstützt eine Vielzahl von aktiven und passiven ISDN-Karten. Bewährt haben sich in der Praxis preiswerte ISA-Karten wie *AVM Fritzcard Classic*, *ELSA* und *Teles Teledat*. Viele der modernen PCI-Karten machen immer wieder Probleme bzw. die Konfiguration gestaltet sich sehr schwierig.

Für den kommerziellen Einsatz ist es wichtig, dass die gewünschte Konfiguration auch für den Betrieb im Telefonnetz zugelassen ist. *Aktive* ISDN-Karten besitzen in der Regel mitsamt der Firmware eine solche Zulassung, die dann auch für den Betrieb unter Linux gilt. Bei passiven Karten gilt diese Zulassung in den meisten Fällen nur dann, wenn die Karte auch mit der Software des Herstellers betrieben wird. Eine Ausnahme sind die Karten des Herstellers ELSA, diese sind auch unter Linux zugelassen. Wer auf die Zulassung angewiesen ist, muss also eine aktive oder eine passive Karte von ELSA einsetzen, alternativ kann eine nicht-zugelassene passive Karte aber auch an einer Telefon-Nebenstellenanlage angeschlossen werden.

SuSE-Linux enthält das Programmpaket *isdn4linux* (I4L), bestehend aus Hardwaretreiber, Netzwerkinterface und Modem-Emulationen. Darüber hinaus sind viele Programme verfügbar, die auf I4L aufsetzen, so z. B. auch ein Anrufbeantworter. Die ISDN-Konfiguration erfolgt über das Programm *isdnctrl*. Für die Einbindung in

das lokale Netzwerk- und Routingsystem werden *ifconfig* und *route* benötigt. Sehr einfach können aber nahezu alle Konfigurationsarbeiten über YaST erledigt werden.

Da die internen ISDN-Karten leider über keine LED-Statusanzeigen verfügen, sollten während der gesamten Grundkonfiguration immer wieder die entsprechenden Log-Dateien ausgewertet werden, zumindest sollten auf einem Textterminal mit

```
linux01:~ # tail -f /var/log/messages
```

die Systemmeldungen angezeigt werden.

Nach dem Einbau einer ISDN-Karte muss zunächst die Hardware konfiguriert werden, dies kann am einfachsten über YaST erfolgen, dazu wird *Administration des Systems -> Hardware in das System integrieren -> ISDNHardware konfigurieren* gewählt. In dieser Form sind jetzt die technischen Daten der ISDN-Karte einzutragen, unter anderem der Kartentyp, I/O-Adressen und verwendeter Interrupt. Diese Einstellwerte sollten bekannt sein, eventuell müssen diese sogar vor dem Linux-Start mit geeigneten Testprogrammen ermittelt werden.

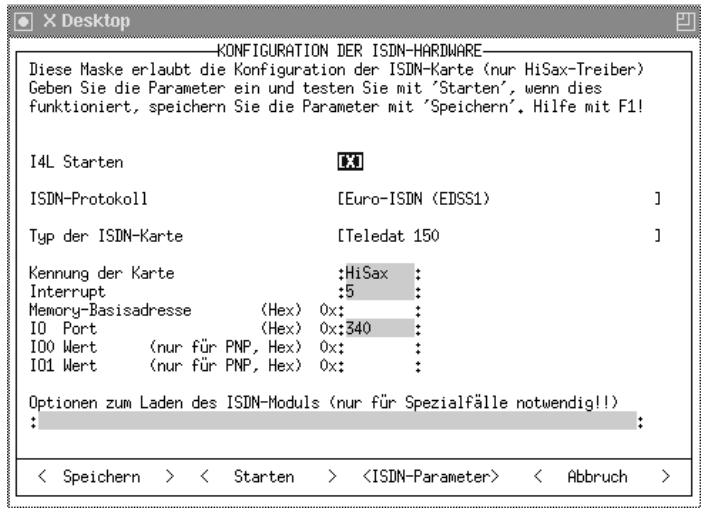


Abbildung 3-85 ISDN-Karte konfigurieren

Als ISDN-Protokoll sollte in der Regel *Euro-ISDN DSS1* angegeben werden können, nur bei Anschluss an eine Telefon-Nebenstellenanlagen (*TK-Anlage*) kann es vorkommen, dass die ältere nationale Protokollvariante *1TR6* gewählt werden muss.

Ein wichtiger Hinweis: Wird die ISDN-Karte an einer TK-Anlage angeschlossen, müssen die technischen Daten des Anschlusses , insbesondere Protokolltyp, Anschlussnummer und die tatsächlich von diesem Anschluss übergebene interne Rufnummer bekannt sein.

Ist die Hardware-Grundkonfiguration abgeschlossen, wird diese mit der Funktion <*Starten*> probeweise aktiviert.

Über die Funktion <*ISDN-Parameter*> kann jetzt eine Form aufgerufen werden, in der die verwendeten eigenen und die anzuwählenden Rufnummern angegeben werden können. Die *eigene Telefonummer* ist dabei in der Regel unkritisch:

- Bei *EURO-ISDN* ist es die *Multiple Subscribe Number* MSN, genauer gesagt die eigene Rufnummer ohne Vorwahl. Bei Nebenstellanlagen muss das tatsächliche Format der übermittelten Rufnummer bekannt sein.
- Bei *1TR6* wird anstelle der MSN die *Endgeräte-Auswahl-Ziffer* EAZ verwendet, in der Regel eine einzelne Ziffer im Bereich 1 bis 9 (die Null ist nicht zugelassen).

Im Feld *Nummern, die anrufen dürfen* braucht zunächst nichts eingetragen zu werden, dies wird erst dann notwendig, wenn dieser Linux-Server später als Einwählpunkt konfiguriert werden sollte.

Jetzt muss in jedem Fall noch angegeben werden, wohin dieses Interface eine Verbindung herstellen soll. Soll dies eine Verbindung zum Internet-Provider sein, werden abschließend die entsprechenden Zugangsdaten einzutragen.

Anzurufende Nummern	Rufnummer des Providers, bei Nebenstellenanlagen mit »Amtsleitungsholung«
Name des PPP-Logins	Benutzer-Account beim Provider
Passwort des PPP-Logins	Benutzerkennwort dazu

Die Angaben unter *Wählmodus* können in der Regel zunächst direkt übernommen werden; in der Regel ist hier keine Anpassung notwendig.

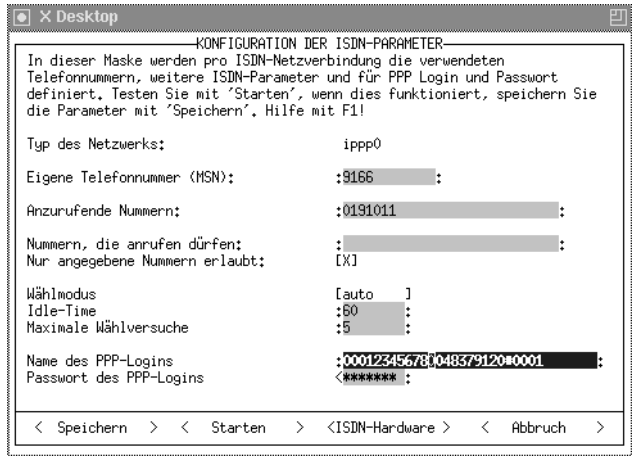


Abbildung 3-86 Konfiguration der ISDN-Parameter

Auch in dieser Form werden die Parameter zunächst mit <*Starten*> probeweise eingestellt. Erfolgt keine Fehlermeldung, ist dieser Konfigurationschritt abgeschlossen. Die aktuellen Einstellungen werden mit <*Speichern*> gesichert.

Das neue erstellte ISDN-Gerät muss jetzt in die Netzwerkumgebung eingetragen und aktiviert werden. Mit YaST erfolgt dies über *Netzwerk konfigurieren -> Netzwerk Grundkonfiguration*.

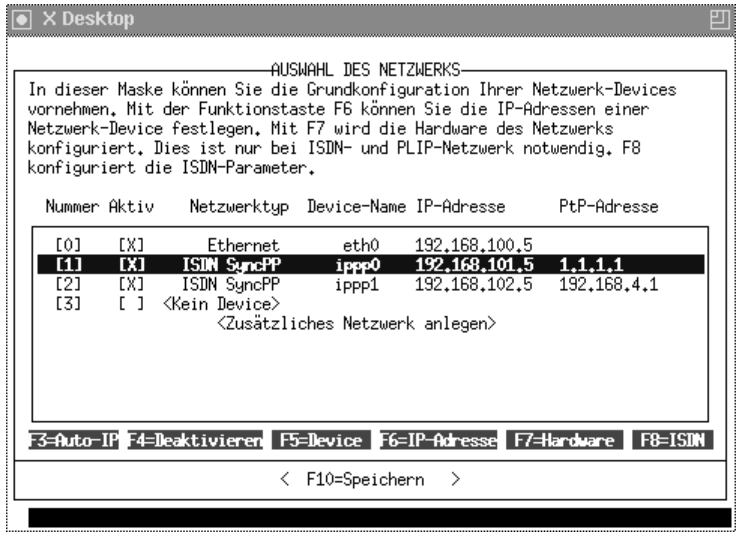


Abbildung 3-87 ISDN-Netzwerkkonfiguration über YaST, Schritt 1

Mit [F5] wird ein neues Device angelegt, der Typ ist *ISDN SyncPPP*, der Name des ersten ISDN-Device ist *ippp0*. Nach Eingabe von [RETURN] bzw. [F6] ist es möglich, die IP-Adressangaben vorzunehmen:

In diesem Beispiel (Abbildung 3-82) ist die IP-Adresse der Netzwerkkarte *eth0* 192.168.100.5. Das Device *ippp0* bekommt, um später vor allem die Routerprogrammierung optimal durchführen zu können, eine Adresse aus einem anderen Teilnetz zugewiesen, hier ist es 192.168.101.5. Die Parameter für *Default Gateway* und *IP-Adresse des PtP-Partners* werden auf die »Dummy«-Werte 1.1.1.1 gesetzt; hier sollen später die tatsächlichen Adressen dynmaisch vom Internet-Provider zugewiesen werden.

Mit <*Weiter*> wird die Eingabe der IP-Adressen beendet, die davor liegende Form wird wieder angezeigt. Hier sollte die neue ISDN-Device unbedingt noch mit [F4] aktiviert werden. Mit [F10] werden alle Einstellungen gespeichert.

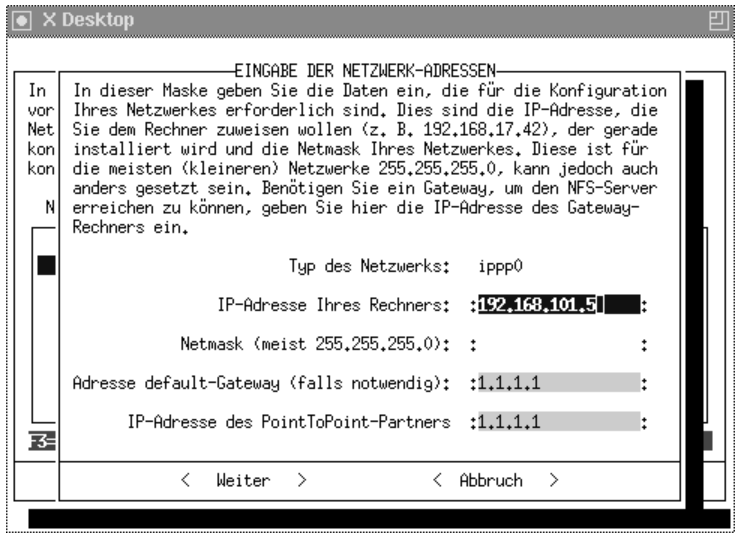


Abbildung 3-88 ISDN-Netzwerkkonfiguration über YaST, Schritt 2

Nach dem Verbindungsaufbau zum Internet-Provider ist die Auflösung der im Internet verwendeten Namen wichtig. Diese wird über einen *Nameserver* durchgeführt, der vom Provider bereitgestellt wird. YaST ermöglicht diese Angabe über das Menü *Konfiguration Nameserver*. Nachdem die Frage *Möchten Sie auf einen Nameserver zugreifen* mit »Ja« beantwortet wurde, kann der Name oder besser noch die IP-Adresse des Provider-Nameservers angegeben werden (Abbildung 3-89).

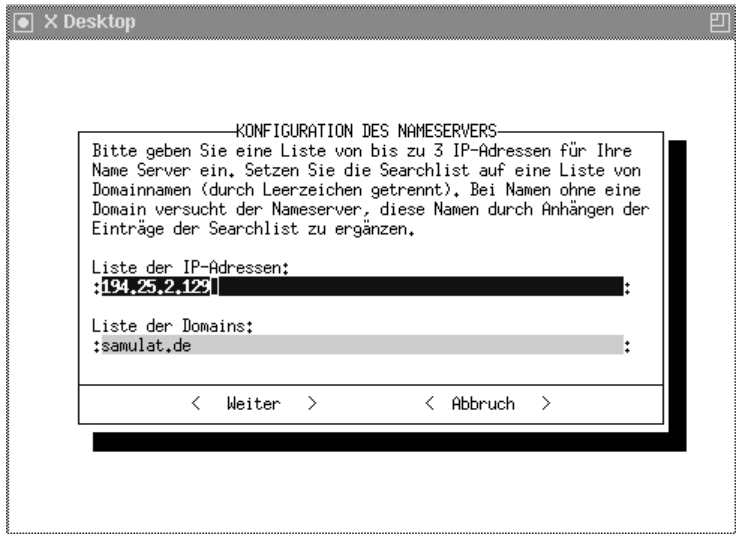


Abbildung 3-89 ISDN-Netzwerkonfiguration, Angabe Nameserver

Um die dynamsiuche IP-Adresszuweisung durch den Provider zu ermöglichen, muss in /etc/rc.config

```
IP_DYNIP="YES"
```

eingetragen werden.

Die Konfigurationsarbeiten sind nun abgeschlossenen, der Rechner sollte jetzt neu getartet werden. Wahlweise können die Netzwerkdienste auch mit

```
linux01: ~ # init 1
```

angehalten werden, wobei auch XWindows in jedem Fall beendet wird. Mit

```
linux01_ ~ # init 2
```

werden die Netzwerkdienste dann neu gestartet.

Verbindungsaufbau testen – isdnctrl

Nachdem die ISDN-Karte konfiguriert und die Zugangsdaten zum Provider eingetragen sind, kann der erste Verbindungsaufbau erfolgen (Abbildung 3-90).

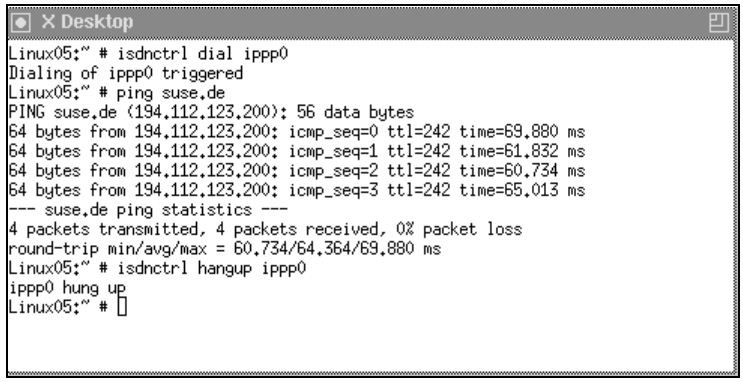


Abbildung 3-90 Verbindungsaufbau mit isdnctrl

Mit *isdnctrl dial ippp0* wird der Wählvorgang gestartet, kurz danach sollte die Verbindung zum Provider hergestellt sein. Der anschließende Befehl *ping suse.de* zeigt, ob die grundsätzliche Funktion gegeben ist und ob die Namensauflösung funktioniert. Mit *isdnctrl hangup ippp0* wird die Verbindung manuell getrennt.

Alle Details dieses Vorgangs können in der Log-Datei */var/log/messages* »mitgelesen« werden. Insbesondere bei einer notwendigen Fehlersuche sollte zunächst hier überprüft werden, ob überhaupt eine Einwahl in das ISDN-Netz möglich war.

Wichtig ist aber auch eine grafische Zustandsanzeige für die ISDN-Leitungen, die z. B. über das Programm *isdnmon* erfolgen kann (Abbildung 3-91).

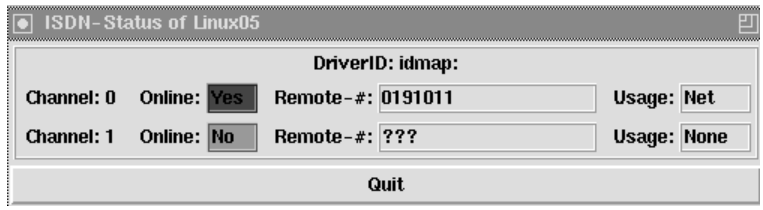


Abbildung 3-91 ISDN-Zustandsanzeige mit dem Programm isdnmon

Konnte der Verbindungsaufbau erfolgreich durchgeführt werden, muss auch für *ippp0* eine IP-Adresse vom Provider zugewiesen worden sein. Dies sollte in jedem Fall überprüft werden, z. B. mit dem Befehl *ifconfig* (Abbildung 3-92).

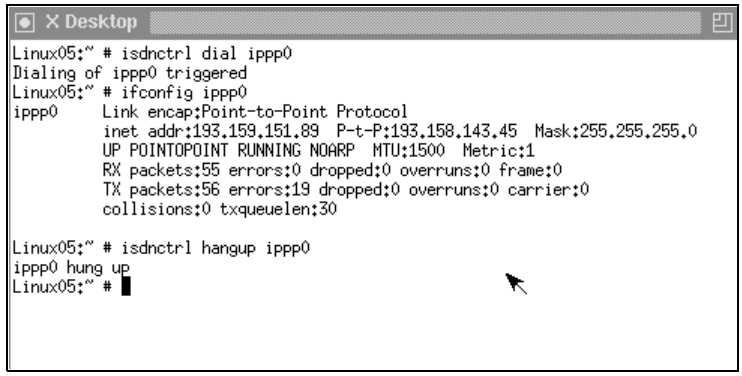


Abbildung 3-92 Zuweisung einer dynamsichen IP-Adresse durch den Provider

Steuerung und Statusabfrage über I4L-Befehle

Außer *isdnctrl* und *isdnmon* bietet das Softwarepaket *I4L* eine Reihe weiterer Werkzeuge für Betrieb und Konfiguration (Tabelle 3-24).

Name	Beschreibung
imon	zeigt in einem Textfenster den aktuellen Status der ISDN-Kanäle an
imontty	textorientiertes Monitorprogramm mit einmaliger Ausgabe
ippstats	zeigt ppp-Statistiken an
isdnlog	protokolliert den Datenverkehr auf dem S0-Bus
isdnrep	Reportprogramm, Auswertetool für isdnlog

Tabelle 3-24 Befehle zur ISDN Steuerung und Überwachung

Vor der erstmaligen Verwendung einiger Befehle sind zusätzliche Konfigurationsarbeiten notwendig: *isdnlog* setzt z. B. die Bearbeitung der Dateien */etc/isdn/isdn.conf* und */etc/isdn/isdn.conf* voraus, um dann auch Verbindungskosten darstellen zu können.

Fehlersuche bei unkontrolliertem Verbindungsaufbau

Die Funktion *Auto-Dial* der ISDN-Schnittstelle ist zwar für die geforderte Funktionalität sehr wichtig, führt aber häufig zu unkontrolliertem Aufbau einer Verbindung. Wichtig ist dann, herauszufinden, welcher Dienst diesen Verbindungsaufbau erzwungen hat. Mit

```
# isdnctrl verbose 3
```

werden sehr detaillierte Informationen zu jeder Verbindung in */var/log/messages* ausgegeben. Wird eine Verbindung aufgebaut, können so die Quell- und Ziel-IP-Adressen und die Nummer des anfordernden Ports ermittelt werden:

Port	Dienst
513	login
21	FTP
37	time
110	POP3

Hier einige Beispiele für häufige Fehlerursachen:

- Eine nicht richtig durchgeführte Namensauflösung, bei der dann unnötigerweise immer wieder versucht wird, Hostnamen über den DNS-Server des Providers aufzulösen. Wichtig ist, dass alle Rechner im eigenen Netzwerk über vollständige und aktuelle Dateien *hosts* verfügen.
- Vom SAMBA-Dienst erzeugte Broadcasts. In diesem Fall könnte der Parameter *interfaces* in der Konfigurationsdatei */etc/smb.conf* fehlen oder es wurde ein falscher Wert angegeben.
- Der Daemon für dynamisches Routing *routed* wurde versehentlich aktiviert.
- Der eigene DNS-Server ist falsch konfiguriert und stellt immer wieder Verbindungen zum Nameserver des Providers her.
- Der Dienst *sendmail* versucht, Mails abzuschicken, die nicht vom Zielserver angenommen werden.

Routing-Konfiguration

Soll der Verbindungsaufbau über die ISDN-Schnittstelle automatisch immer dann erfolgen, wenn eine IP-Adresse außerhalb des eigenen Netzwerkes verwendet wird, so erfordert dies eine Routerfunktion.

In */etc/rc.config* ist dazu zunächst der Eintrag

```
IP_FORWARD="YES"
```

erforderlich, damit wird das Routing grundsätzlich aktiviert. Wie und unter welchen Voraussetzungen Datenpakete geroutet werden sollen, wird in der Konfigurationsdatei */etc/route.conf* festgelegt.

```
#
# /etc/route.conf
#
1.1.1.1           0.0.0.0           255.255.255.255     ippp0
192.168.100.0     0.0.0.0           255.255.255.0       eth0
default           1.1.1.1
```

Mit der hier gezeigten Konfiguration werden alle Pakete aus dem eigenen Netzwerk 192.168.100.0, die nicht direkt zugestellt werden können, an das *default* Gateway 1.1.1.1 gegeben. Damit führt jetzt jeder Zugriff auf eine nicht zum eigenen Netzwerk gehörende IP-Adresse zum ISDN-Verbindungsaufbau. Dies gilt auch, wenn ein Hostname nicht lokal über die Datei */etc/hosts* aufgelöst werden kann. In diesem Fall wird versucht, den Namen über den Nameserver des Internet-Providers aufzulösen.

Wird jetzt

```
linux01: ~ # ping suse.de
```

eingegeben, sollte kurze Zeit nach dem automatisch erfolgenden Verbindungsaufbau auch eine Antwort von *suse.de* erfolgen.

IP-Masquerading

Mit der Zuweisung einer »echten« IP-Adresse durch den Provider ist dieser Rechner auch direkt aus dem Internet erreichbar. Bei jedem Verbindungsaufbau wird zwar eine andere Adresse zugewiesen (was diese Gefahr tatsächlich etwas mindert), aber zumindest für die Dauer der Verbindung besteht so tatsächlich die Möglichkeit, des unbefugten Zutritts zum eigenen Netzwerk.

Eine vergleichsweise einfache aber wirkungsvolle Schutzmaßnahme stellt das *IP-Masquerading* dar. Alle Rechner im eigenen Netzwerk werden »versteckt«, d.h. nach außen wird, gleich welcher Rechner diese Verbindung nutzt, nur mit einer IP-Adresse gearbeitet. Von außen ist also das eigene Netzwerk nicht sichtbar.

Zur Einrichtung dieses Dienstes wird das Programmpaket *firewall* aus der Serie *n* benötigt. Die Grundkonfiguration erfolgt in */etc/rc.config*:

MSQ_START	IP-Masquerading Daemon starten
MSQ_DEV	Masquerading-Device: idealerweise die ISDN-Karte des Servers
MSQ_NETWORKS	Liste der zu versteckenden Netzwerke
MSQ_MODULES	Protokolle, die maskiert werden sollen (die hier voreingestellten Module können in der Regel beibehalten werden)

Beispielkonfiguration für das eben erstellte ISDN-Device *ippp0*:

```
MSQ_START="YES"
MSQ_DEV="ippp0"
MSQ_NETWORKS 192.168.100.0/24
MSQ_MODULES "ip_amsq_cuseeme ip_masq_ftp ip_masq_irc ip_masq_quake
ip_maskq_rausio ip_maskq_vdolive"
```

Nach dem nächsten Systemstart bzw. Neustart des Netzwerkdienstes wird der Start des *IP-Masquerading*-Daemons angezeigt.

Zu Konfiguration und Start kann auch *ipfwadm* verwendet werden, eigentlich ein Werkzeug zu Einrichtung von einfachen Paketfiltern. Mit

```
ipfwadm -a -m -S 192.168.100.0/255.255.255.0
```

wird das Netzwerk *192.168.100.0* »maskiert«.

Noch ein Hinweis: Mit *IP-Masquerading* ist es technisch möglich, einen vom Provider gekauften Einzelplatzanschluss allen Rechnern im Netzwerk zur gemeinsamen Nutzung zur Verfügung zu stellen. Mit dem Provider sollte geklärt werden, ob dies zulässig ist.

Ein einfaches Firewallsystem

Das IP-Masquerading ist der erste Schritt auf dem Weg zu einer Firewall, von Hard- und Software, die das eigene Netzwerk vor unbefugten Zugriffen aus dem Internet schützt und trotzdem die Nutzung der gewünschten Dienste ermöglicht.

Sicherheit von TCP/IP-Netzwerken

Alle 20 Sekunden wird in den USA in ein Netzwerk eingebrochen. Der Einstieg in Firmennetze erfolgt in 80 Prozent aller Fälle über das Internet. Neuesten Schätzungen zufolge beläuft sich der daraus resultierende Schaden auf jährlich 100 Millionen US$.

Ein großer Teil der mutwilligen Beschädigungen von Computer-Systemen und Datennetzen wird erstaunlicherweise aber von Mitarbeitern des betroffenen Unternehmens begangen. Häufig genug versuchen frustrierte Mitarbeiter, sich über den Weg der Zerstörung von Datenmaterial oder durch Sabotage an Systemen am Arbeitgeber zu rächen.

Spione	Sind professionelle Hacker mit sehr hohem technischem Know-how. Sie verdienen ihr Geld in der Regel mit Industriespionage.
Vandalen	haben fast gleiches technisches Niveau wie Spione. Ihr Ziel ist es, möglichst viel Schaden anzurichten.

Tabelle 3-25 Vor wem muss das eigene Netzwerk geschützt werden?

Joyrider	hacken aus Neugier und Langeweile. Sie sind zwar nicht absichtlich böswillig, richten aber oft Schaden an, weil sie sich ignorant verhalten oder versuchen, ihre Spur zu verwischen.
Punktejäger	stehen mit anderen Hackern im Wettbewerb und bevorzugen besonders gut geschützte Standorte. Wer zuerst eindringen kann, punktet.
Anwender	richten aus Unwissenheit oder Nachlässigkeit Schäden an.

Tabelle 3-25 Vor wem muss das eigene Netzwerk geschützt werden?

Welche Methoden werden zum Eindringen in fremde Netzwerke genutzt? Tabelle 3-26 zeigt einige Möglichkeiten:

Password Guessing	Erraten von Paßworten mit Hilfe von geeigneten Daemonen, z.B. »finger«
Password Cracking	Mit Hilfe eines Crackprogrammes oder eines Wörterbuches können Paßworte verglichen und entschlüsselt werden.
E-Mail Angriffe	Verbreitung von Viren über E-Mail-Dateien
Mail Flooding	Verbreiten von rufschädigendem Material unter falschem Namen
IP-Spoofing	Verbindungsaufbau von außen mittels »scheinbar zulässiger« IP-Adresse
System-Angriffe	durch das Ausnutzen von System-, Konfigurations- oder Anwenderfehlern. Nutzung von »Hintertürchen« und »Trojanischen Pferden«
SYN-Flooding	»Lähmen« eines Dienstes oder des gesamten Systems durch eine absichtlich herbeigeführte Überlastung

Tabelle 3-26 Gefahren für das eigene Netzwerk (Auswahl)

Netzwerke auf der Basis von TCP/IP zeigen leicht nachvollziehbare Strukturen und Zugriffsmechanismen. Der Mechanismus zur Vergabe von IP-Adressen kann, ausgehend von nur wenigen Informationen über das zu analysierende Datennetz, leicht analysiert und für eigene Zwecke mißbraucht werden.

Datenverbindungen in TCP/IP-Netzen sind nicht verschlüsselt: Auch Benutzernamen und das dazugehörige Paßwort werden also grundsätzlich offen über die Verbindungsleitungen übertragen!

Um die dargestellten Gefahren zu minimieren, ist ein großer technischer und organisatorischer Aufwand im TCP/IP-Netzwerk notwendig. Sichergestellt werden muss die

- *Authentizität*: Sind die Kommunikationspartner tatsächlich die, die sie zu sein vorgeben?
- *Vertraulichkeit*: Sind die Informationen vor der Kenntnisnahme Unberechtigter geschützt?

- *Integrität*: Sind die Daten vor Manipulationen sicher?
- *Verbindlichkeit*: Sind getroffene Verabredungen verbindlich, und wie läßt sich das nachweisen?

Die Sicherheitsaspekte werden oft in zwei Kategorien unterschieden:

- Sicherheit des Netzes, einzelner Hosts sowie der dort gespeicherten Daten (umfasst Probleme, die mit dem Zugang zum Netz oder einzelnen Rechnern zusammenhängen, insbesondere mit der Verfügbarkeit und Sicherheit einzelner Dienste); Sicherheit der Kommunikationsbeziehungen und Anwendungen
- Authenzität von Daten und Kommunikationspartnern, Schutz der Privatsphäre sowie Geheimschutz

Angriffe erfolgen z. B. durch »*Spoofing*«, »*Faking*« oder »*Man-in-the-Middle-Attacks*«. In diesen Fällen geht es darum, falsche Identitäten vorzutäuschen, um so Einrichtungen oder Personen zu kompromittieren oder unberechtigt an Informationen zu gelangen.

Schutz von Datennetzen über Firewalls

Zum Schutz des firmeneigenen Intranet vor Angriffen aus dem Internet oder auch zum Schutz einzelner Teile des Intranet haben sich in den letzten Jahren »*Firewalls*« als geeigneter Mechanismus etabliert. Zentrale Aufgabe eines Firewalls ist das »Verstecken« interner Netze.

Es gibt *Firewalls* in zwei unterschiedlichen Ausbaustufen:

- einfache *Paketfilter* (prüfen die IP-Adressen ein- und ausgehender Pakete), die bereits in der Software der Router implementiert sind
- *Applikationsfilter*, die als eigenständige Softwareprodukte (*Proxy-Server*) auf speziell konfigurierten Rechnern laufen

Firewalls sind zentrale Datenschranken zum Schutz von Netzwerken. Ihre Aufgabe ist es, Gefahren aus dem Internet zu minimieren und gleichzeitig die Vorteile einer Netzanbindung zu ermöglichen.

Alle ein- und ausgehenden Daten passieren den Firewall. Je nach Art und Konfiguration des Firewalls wird die gewünschte Verbindung freigegeben oder gesperrt. Firewalls

- »Verstecken« interne Datennetze,
- koppeln gesicherte (kalte) an ungesicherte (heiße) Netzwerke,
- schützen das gesicherte Netzwerk und ermöglichen gleichzeitig den (möglichst) ungestörten Zugriff auf das ungesicherte Netzwerk,
- stellen den einzigen Zugang vom gesicherten zum ungesicherten Netzwerk dar.

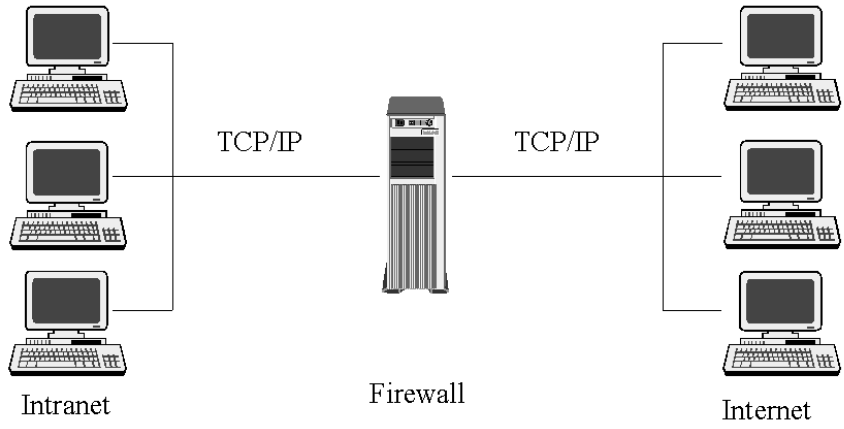


Abbildung 3-93 Struktur eines typischen Firewall-Systems

Die Sicherheit ist nur gewährleistet, wenn auch wirklich der gesamte Datenverkehr über die Firewall geleitet wird.

Paket-Filter

Paketfilter entscheiden, ob ein TCP- oder UDP-Packet weitergeleitet wird oder nicht. In der Regel können bis zu vier »Filter« gesetzt werden:

- Ziel (Empfänger-IP-Adresse)
- Quelle (Absender-IP-Adresse)
- Port-Number (Server TCP/UDP Port Number)
- Richtung (ein- oder ausgehend)

Paketfilter werten nicht den Inhalt eines Pakets aus, sondern entscheiden aufgrund der im Header geführten IP-Adressen und der Richtung, ob weitergeleitet wird oder nicht.

Packetfilter sind oft bereits im *Router* implementiert. Sie lassen interne IP-Adressen aber weiterhin von außen sichtbar. Die Anleitung zur Realisierung eines Paketfilter-Firewalls auf Basis eines Linux-Systems ist in /Rudo97/ veröffentlicht.

Application-Filter

Application-Filter werden auch als *Proxy-Server* (Stellvertreter) bezeichnet. Sie nehmen die Benutzeranfragen der Clients auf und bauen an deren Stelle die Verbindung zum gewünschten Server auf. Sie erlauben eine strenge Benutzer-Authentifikation und detaillierte Login-Informationen.

Die Filterregeln sind wesentlich einfacher zu konfigurieren und zu testen als bei Paketfiltern. Interne IP-Adressen werden nicht nach außen abgebildet.

Konfiguration des Proxy-Server squid

Squid ist ein *Proxy-Server* für WWW-Dokumente, anonymous FTP und Gopher, der die Möglichkeit bietet, auch größeren Netzwerken einen gemeinsam nutzbaren Internet-Zugang zur Verfügung zu stellen. Aus dem Internet abgerufene Seite werden durch *Squid* gespeichert, so dass die Zugriffsgeschwindigkeit erheblich erhöht werden kann, wenn z.B. mehrere Benutzer mit gleichen WWW-Dokumenten arbeiten.

Squid ist auch ein *Application-Filter;* er kann wichtige Aufgaben aus dem Bereich *Firewall* übernehmen, dazu sind vielfältige Konfigurationsmöglichkeiten vorhanden.

Installation und Start

Zur Einrichtung von *Squid* wird das Paket *squid2* aus der Serie *n* benötigt. Installiert wird Squid in der Version 2.2 (2,5 MB). In */etc/rc.config* ist der Eintrag

```
START_SQUID_"YES"
```

vorzunehmen, damit wird der Proxy-Server beim nächsten Booten automatisch gestartet wird.

Grundkonfiguration

Die Konfigrationsdatei von Squid ist */etc/squid.conf*. Die nachfolgende, sehr einfache Beispieldatei erlaubt allen Benutzern den Zugriff, nutzt 100 Mbyte Cache Speicher und 8 MByte RAM.

```
# squid.conf - a very basic config file for squid

# turn looging to it's lowest level
debug_options ALL,1

# defines a group (or access control list) that includes
# all ip adresses
acl all src 0.0.0.0/0.0.0.0

# allow all sites to use us as a sibling
icp_access allow all

# test the following sites to check that we are connected
samulat.de

# run as the squid user
cache_effective_user squid squid
```

Mit dieser Konfiguration antwortet *Squid* auf Anfragen über den Standard-Port 3128, Log-Dateien werden nur in minimalem Umfang erstellt und alle Dateien werden im Standardpfad */usr/local/squid/cache* abgespeichert.

Diese Konfiguration enthält noch keinerlei Absicherungsmöglichkeiten, sie sollte aber den ersten Start des *Squid*-Daemon ermöglichen. Während des Starts von *Squid* erfolgt bereits ein Verbindungsaufbau zum Provider, um die Verfügbarkeit der Internet-Anbindung zu prüfen (hier wird nach der Domäne *samulat.de* gesucht). Ist dies nicht möglich, kann *Squid* nicht gestartet werden.

Die vollständige Konfiguration von Squid ist sehr aufwändig und setzt in jedem Fall auch die Kenntnis der entsprechenden Dokumentation voraus (*www.squid.nl-anr.net/*).

Einige zur Konfiguration von *Squid* im kommerziellen Umfeld wichtige Parameter der Datei */etc/squid.conf* enthält Tabelle 3-27.

Parameter	Beschreibung
inside_firewall	ist eine Firewall im Netzwerk eingerichtet, so enthält dieser Eintrag den Namen oder die IP-Adresse des entsprechenden Rechners. Squid wird dann nur über diese Adresse seine Seiten holen.
local_domain	Liste der eigenen Domänen, durch Leerzeichen getrennt
source_ping off	Die Überprüfung der Erreichbarkeit des Quellservers muss für den Offline-Betrieb abgeschaltet werden.
cache_mem 8	weist Squid 8 MByte RAM zu (Minimalwert).
cache_swap 350	Auf der Festpalatten sollen max. 350 MByte für das Zwischenspeichern (Cache) der WWW-Seiten genutzt werden.
cache_dir /var/squid/cache	Pfad zum Festplattencache
debug_options ALL,1	bestimmt die Menge der abgespeicherten Debug-Informationen.
refresh pattern . 18400 50% 43200	Hier wird festgelegt, dass alle Objekte mindestens 18.400 Minuten als aktuell zu halten sind (ca. 14 Tage), nach 43.200 Minuten (ca. 30 Tagen) ist die Seite nicht mehr aktuell und neu zu holen. Dazwischen bestimmt der Füllgrad der Cache-Datei, ob die Seite weiter gehalten wird. Mit der Prozentangabe können z. B. für Bilder oder Texte unterschiedliche Wertigkeiten festgelegt werden: refresh_pattern \.gif$ 43200 100% 43200
connect_timeout 60	Nach 60 Sekunden wird bei nicht erreichbaren Seiten ein Fehler gemeldet.
read_timeout 5	Wenn eine bereits begonnene Übertragung 5 Minuten stillsteht, wird abgebrochen.

Tabelle 3-27 Konfigurationparameter für Squid (Auswahl)

Konfiguration über webmin

Auch für *Squid* stellt das zentrale Systemverwaltungstool *webmin* eine leistungsfähige Verwaltungsoberfläche dar (Abbildung 3-94).

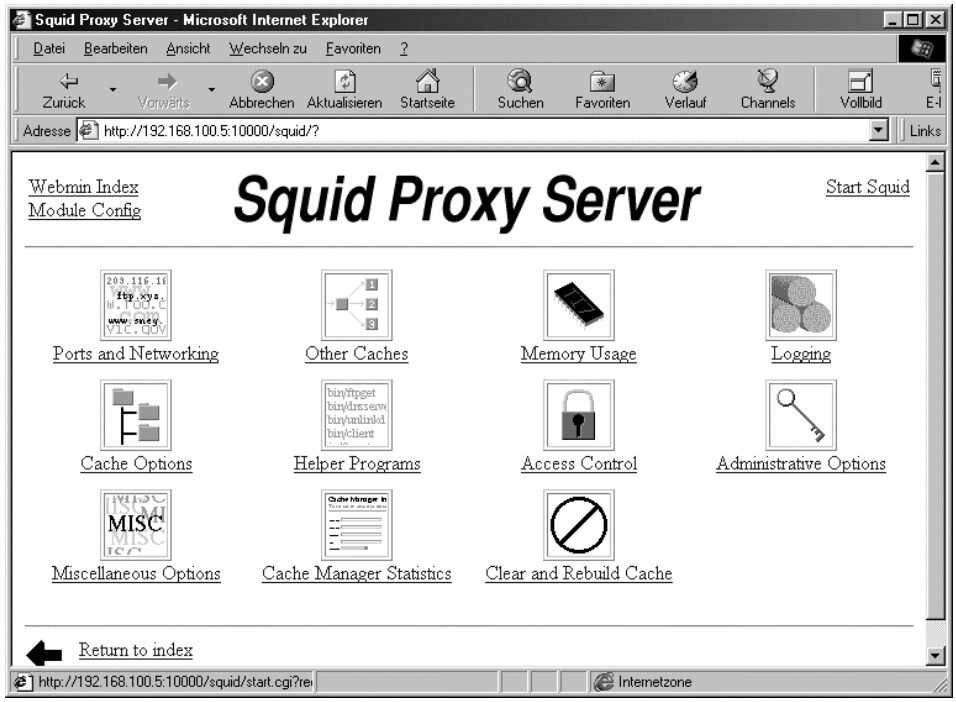


Abbildung 3-94 Administration von Squid über webmin

Über *webmin* können vor allem auch die Cache-Speicher überwacht und bei Bedarf initialisiert werden (*Clear and Rebuild Cache*). Wird Squid auf dem Linux-Server eingesetzt, ist es sehr empfehlenswert, die Cache-Speicher in eine eigene Partition zu legen.

Client-Konfiguration

Die Einrichtung der WWW-Browser beschränkt sich in der Regel auf die Angabe des Hostnames oder der IP-Adresse, auf dem der Proxy-Server läuft. Zusätzlich muss noch die Portnummer angegeben werden.

Microsoft Internet Explorer

Die Grundkonfiguration umfaßt die Dienste *HTTP, FTP* und *Gopher* (Abbildung 3-95).

Netscape unter Microsoft Windows

Auch hier werden die drei Dienste *HTTP*, *FTP* und *Gopher* konfiguriert (Abbildung 3-96).

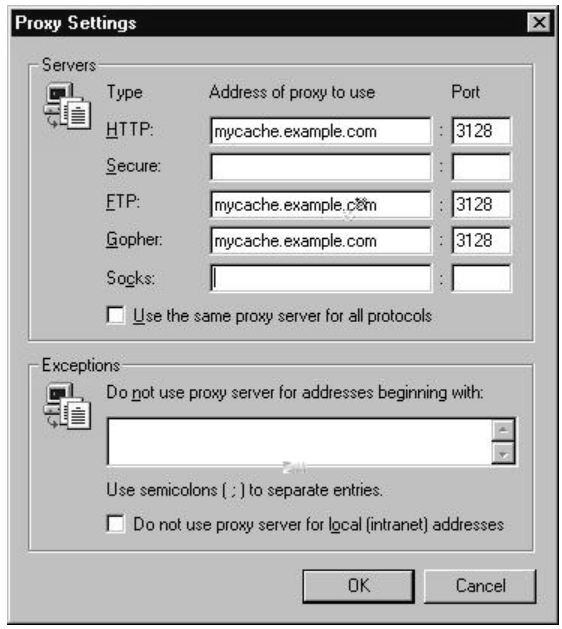


Abbildung 3-95 Konfiguration für den Microsoft Internet Explorer

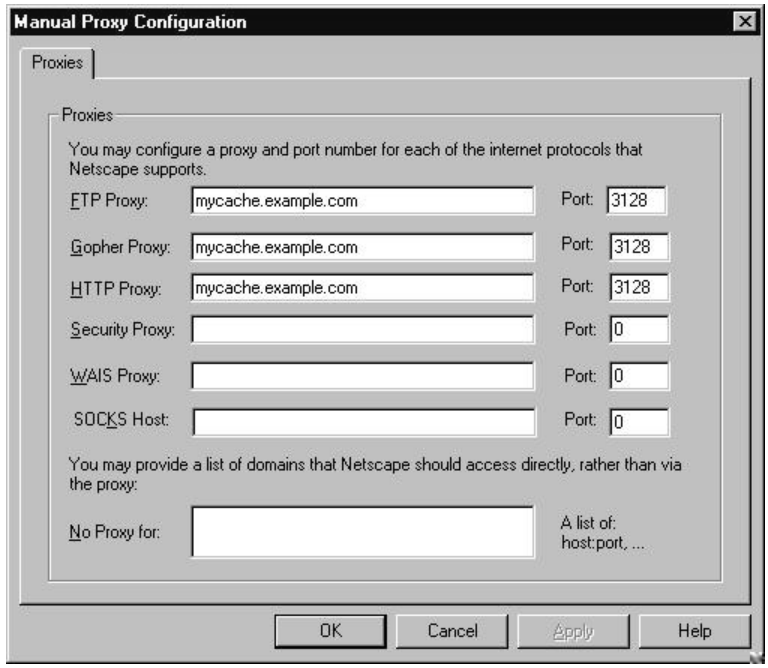


Abbildung 3-96 Client-Konfiguration für den Netscape Navigator 3

Statusinformationen und Hilfsprogramme

Standardmäßig führt Squid vier Log-Dateien:

1. */usr/local/squid/logs/access.log*
2. */usr/local/squid/logs/cache.log*
3. */usr/local/squid/logs/store.log*
4. */usr/local/squid/cache/log*

Die ersten drei Log-Dateien können jederzeit ausgewertet werden, die vierte Datei enthält einen dynamsichen Index zur Cache-Verwaltung.

Die Log-Dateien können mit der Zeit sehr groß werden und sollten daher reglmäßig überprüft werden. Sie können nicht gelöscht werden, solange Squid läuft. Ein kleiner Trick hilft hier allerdings weiter. Mit *kill -USR1* kann Squid gezwungen werden, die aktuellen Logdateien abzuschließen und mit der Erweiterung **.0* zu sichern. Diese Sicherungsdateien können dann gelöscht werden.

Log-Datei	Inhalt
logs/access.log	Wieviel Benutzer haben Daten von Squid angefordert, wieviel Seiten wurden jeweils abgerufen, welche Seiten werden am häufigsten abgerufen, etc.
logs/cache.log	Fehlermeldungen, Startinformationen
logs/store.log	Zeigt die Speicheraktionen im Cache

Tabelle 3-28 Log-Dateien unter Squid

Für die Log-Datei *access.log* gibt es eine Reihe von Programmen, mit denen spezielle Auswertungen ausgeführt werden können, z. B. mit:

```
access-extract.pl < access.log > summary
access-extract-urls.pl < access.log >> summary
access-summary.pl < summary > report.txt
```

Die Datei *report.txt* enthält danach alle wichtigen Informationen. Ein weiteres Skript erstellt dann daraus eine Datei im HTML-Format, die dann mit jedem beliebigen WWW-Browser angezeigt werden kann

```
calamaris.pl < access.log > stats.html
```

Es können aber auch direkt Informationen aus *access.log* im HTML-Format augegeben werden, z. B. mit:

```
squidclients -H < access.log > clients.html
squidtimes < access.log > times.html
```

Kwebwatch
Webseiten können mit dem KDE-Programm *kwebwatc* auf Veränderungen hin überwacht werden. Dazu werden die entsprechenden Seiten in einer Liste erfasst und dann in periodischen Abständen automatisch auf Änderungen überprüft.

3.7 E-Mail – elektronische Post

Die Übertragung elektronischer Nachrichten in Form von E-Mail ist eine der Kernkomponenten eines Netzwerks. Viele Benutzer arbeiten nur deshalb im Intranet oder Internet, um den Kommunikationsdienst »elektronische Post« nutzen zu können.

Grundsätzlich gibt es mindestens drei Wege, eine E-Mail zuzustellen:

- Sind Absender und Empfänger auf dem gleichen Host, so reicht bereits eine einfache Kopieraktion.
- Im TCP/IP-Netzwerk wird das *Simple Mail Transfer Protocol* SMTP eingesetzt.
- Sind zwei Hosts nur zeitweise über Wählleitungen miteinander verbunden, so wird *Unix to Unix Copy* UUCP verwendet.

Hier soll vor allem der Transport von E-Mail im lokalen Netzwerk oder über einen Provider weiter in das Internet betrachtet werden:

Der Benutzer (Client) wählt sich über das Telefonnetz bei dem Internet-Provider ein und nimmt Verbindung zum E-Mail-Server auf. Auf dem E-Mail-Server arbeiten zwei getrennte Daemons:

- ein SMTP-Server, der die E-Mail in Empfang nimmt und weiterleitet
- ein *Post Office Protocol* POP-Server, der die bisher eingegange E-Mail zur Abholung bereitstellt

Auf der Client-Seite gibt es drei Arten von Mail-Programmen:

- *Mail User Agent* MUA zum lokalen Erstellen, Lesen und Verwalten von E-Mail. Beispiele dafür sind Programme wie *mail, elm, pine* oder auch *emacs*.
- *Mail Transport Agent* MTA. Aufgabe dieser Programme ist das Weiterleiten der abgeschickten Nachrichten an einen Provider. Der unter Linux am häufigsten anzutreffende MTA ist *sendmail*.
- Programme, die Nachrichten von einem entfernten POP3-Server abholen und auf dem lokalen Host speichern, wie z. B. *popclient* oder den Nachfolger *fetchmail*

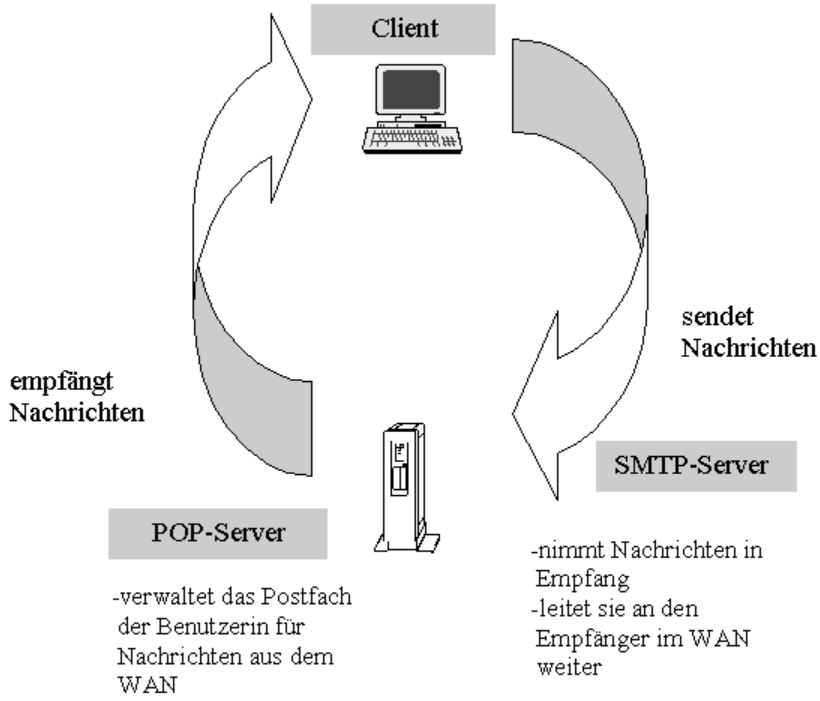


Abbildung 3-97 E-Mail-Transport

3.7.1 Konfiguration des E-Mail-Server

Mit den Diensten *sendmail*, *fetchmail* und *procmail* kann ein leistungsfähiger Mailserver aufgebaut werden.

POP-Server

Die interne Postfachverwaltung kann mit dem Programm *popper* erfolgen, einem *POP3-Server*. Benötigt wird dazu das Paket *pop* aus der Serie *n*.

Jedesmal, wenn ein POP-Client Mail zu diesem Server schickt, stellt *popper* diese direkt an den entsprechenden Benutzer zu. Dem Systemverwalter ist es z. B. möglich, alle Benutzer eines Servers per E-Mail zu erreichen, ohne tatsächlich eine Nachricht je Benutzer zu erstellen.

Das Programm *popper* wartet auf Anfragen mit der Portnummer 110, die grundlegende Funktion kann über eine einfache telnet-Verbindung geprüft werden (Abbildung 3-98).

Weitere Informationen zum Programm *popper*, insbesondere auch zu möglichen Optionen, können über *man popper* abgerufen werden.

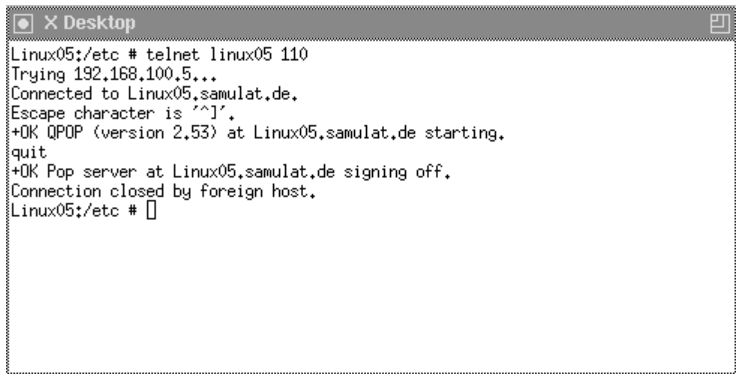


Abbildung 3-98 Test des popper Daemons mit telnet

Konfigration von sendmail

Der *Mail Transport Agent* MTA ist zuständig für die Auslieferung von E-Mails, in der Regel durch die Weiterleitung an den Internet-Provider.

Der am häufigsten unter Linux verwendte *Mail Transfer Agent* MTA ist *sendmail* (*http://www.sendmail.org/*). Dieses Programm ist zwar sehr umständlich zu konfigurieren, die SuSE-Distribution enthält aber bereits eine in den meisten Fällen sofort anwendbare Standardkonfiguration in */etc/sendmail.conf*. Die Client-Schnittstelle kann mittels *POP3* oder *IMAP4* realisiert werden.

Sendmail bietet statische E-Mail-Routen und unterstützt eine Reihe von Transportprotokollen (*smtp, uucp, ...*). Auch ist es möglich, *sendmail SPAM*- und *Relay*-sicher zu machen. Dazu werden mittels *Blacklists* Kriterien definiert, die Nachrichten schon beim Empfang zurückweisen und verhindern, dass unerwünschte Mail von außen kommend über den eigenen Server weitergeleitet wird (*Relayschutz*).

Die Basiskonfiguration erfolgt in */etc/rc.config*. Das Programm *sendmail* wird mit dem Eintrag

```
SMTP="YES"
```

automatisch als Daemon gestartet.

Sendmail verwaltet für den Transport von E-Mails eine Warteschlange. In einem einstellbaren Zeitraster (oder bei Bedarf auch sofort nach dem Erstellen einer Nachricht) werden alle in der Warteschlange gespeicherten E-Mails an die Empfänger weitergeleitet.

Das Zeitintervall kann bei Bedarf über eine Option eingestellt werden:

```
sendmail -q30
```

stellt eine Intervalldauer von 30 Minuten ein. Soll *sendmail* nicht im vorgegebenen Zeitraster automatisch den Verbindungsaufbau zum Provider auslösen, darf die Option *-q* nicht angegeben werden. In diesem Fall muss über einen Eintrag in *crontab* explizit festgelegt werden, wann die in der Warteschlange gespeicherten Nachrichten versendet werden sollen.

Ein wichtiger Hinweis: Im tatsächlichen Netzwerkbetrieb immer wieder problematisch sind nicht zustellbare Nachrichten, die wiederholt einen Verbindungsaufbau zum Provider erzwingen und sehr hohe Leitungskosten entstehen lassen.

Die Adminsitration von *sendmail* in einer grafischen Oberfläche, insbesondere die einfach zu handhabende Nutzung der vielfältigen Absicherungsmöglichkeiten ermöglicht das Programm *webmin* (Abbildung 3-99).

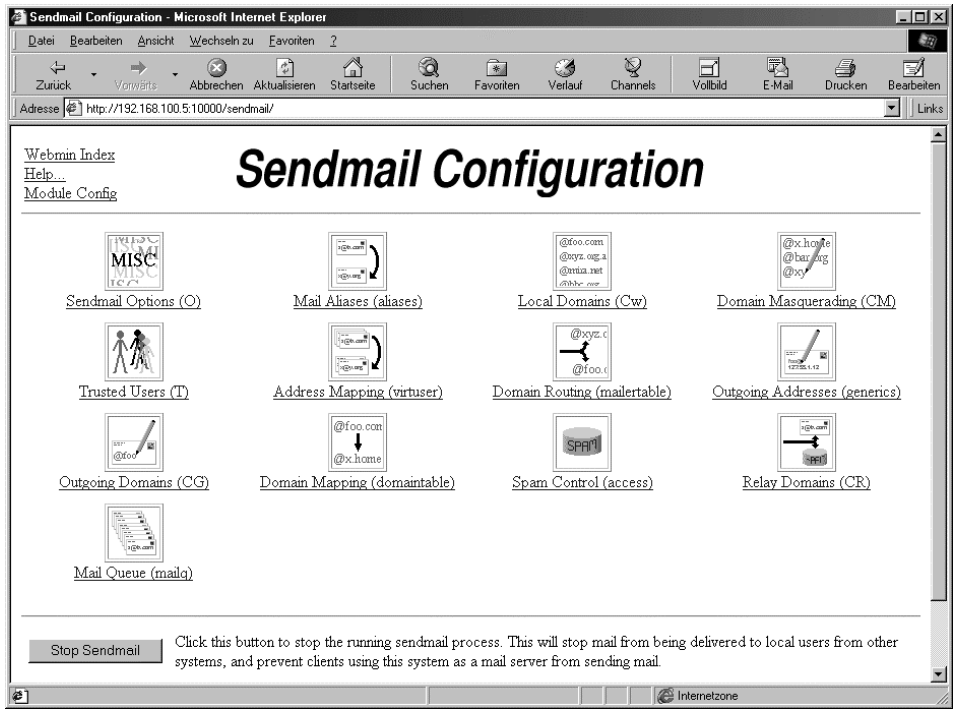


Abbildung 3-99 Konfiguration von sendmail über webmin

Konfiguration von fetchmail

Das vorstehend beschriebene *sendmail* ist ausschließlich für den Transport in eine Richtung zuständig: Es versendet nur E-Mails des lokalen Hosts an die einzelnen Empfänger. Um E-Mails von einem Mail-Account beim Provider (im nachfolgenden Beispiel von einem POP3-Server) abzuholen, kann z.B. *fetchmail* eingesetzt werden.

Die Konfiguration von *fetchmail* erfolgt über die Datei *.fetchmailrc* im Home-Verzeichnis des ausführenden Benutzers (hier *root*).

Da die Datei für den POP3-Account Kennworte im Klartext (!) enthält, sollte die Berechtigung mit

```
# chmod 600 .fetchmailrc
```

soweit wie möglich eingeschränkt werden. Ein Eintrag in *.fetchmailrc* könnte z. B. so aussehen:

```
poll pop.btx.dtag.de protocl POP3 user "000123456780483791 20#0001" password
"clown123" is samulat
```

In dem oben gezeigten Beispiel wird E-Mail für den hier angegebenen Account beim Server *pop.btx.dtag.de* abgeholt und an *procmail*, den lokalen *Mail Delivery Agent* MDA gegeben, der die Weiterleitung an die lokalen Benutzer *samulat* übernimmt.

Grundsätzlich können über die Konfiguration von *.fetchmail* auch mehrere E-Mail-Accounts abgearbeitet werden, dazu sind lediglich die entsprechenden Zeilen zu programmieren. Die Abholung ist dann aber relativ zeitaufwändig, besser ist es, wenn über einen »Sammelaccount« abgeholt werden kann. Ob dies möglich ist, muss mit dem Provider geklärt werden; entsprechende *fetchmail*-Konfigurationen sind möglich. Einzelheiten dazu können mit *man fetchmail* abgerufen werden.

Konfiguration von procmail

Der Dienst *procmail* verteilt die eingegangenen E-Mails im lokalen Netzwerk. Bei Bedarf können Filter eingerichtet werden, um unerwünschte Mail zu unterdrücken. Die Konfiguration erfolgt über die Datei *.procmail* im Verzeichnis des ausführenden Benutzers (auch hier wieder *root*).

Das nachfolgende Beispiel für *.procmail* untersucht alle eingehenden Mails auf die *to*-Angabe im Header und leitet sie dann an den entsprechenden Benutzer weiter.

```
(*** < file > *** text ***) .procmailrc

# Mails mit dem Header "to: info@samulat.de" werden an
# den lokalen Benutzer weitergeleitet

:0
* ^To:.*info@samulat.de
! info

# Mails mit dem Header "to: postmaster@samulat.de" werden an
# den lokalen Benutzer weitergeleitet
:0
```

```
* ^To:.*postmaster@samulat.de
! postmaster

# Mails mit dem Header "to: jan@samulat.de" werden an
# den lokalen Benutzer jan und an seine private
# E-Mail-Adresse weitergeleitet
:0 c
* ^To:.*jan@samulat.de
! jan.samulat@privatdomain.de
:0
* ^To:.*jan@samulat.de
! jan

# Alle Mails und alle "Carbon Copies" fuer michael@samulat.de
# werden weitergeleitet an michael
:0
* (^To:.*michael@samulat.de)|(^CC:.*michael@samulat.de)
! michael

# Alle unzustellabren Mails gehen an den Absender zurueck
# In diesem Fall wird die Datei "nosuchuser" angehaengt. Diese
# Diese Textdatei muss bereits existieren!

:0
  |(/usr/bin/formail -r -k \
     -A"X-loop: mailchef@samulat.de "| \
       /usr/bin/gawk '{print }\
       /^/ && !HEADER \
         { system("/bin/cat nosuchuser"); \
         print"--" ;\
         HEADER=1 }' ) |\
         /usr/bin/sendmail -t

exit
```

Steuerung des Mailtransfers

Wie bereits dargestellt, sind grundsätzlich zwei Möglichkeiten zur Steuerung des Mailtransfers denkbar:

- Zeitgesteuert über den *cron*-Daemon. Damit werden unnötige Verbindungsaufbauten zum Internet-Provider unterbunden. Empfehlenswert vor allem, wenn immer wieder Probleme mit unkontrollierten Anwahlversuchen auftreten.
- Bei jedem ISDN-Verbindungsaufbau. Dazu ist eine Erweiterung des IP-UP-Skriptes */etc/ppp/ip-up* notwendig. In den Beispiel-Skripten der SuSE-Konfiguration sind diese Zeilen schon enthalten und brauchen bei Bedarf nur noch auskommentiert werden:

E-Mail mit t-online

Auch die Telekom bietet breits seit 1997 den Internetzugang direkt über PPP an. Der zur Einwahl notwendige Benutzername wird aus den *t-online*-Anmeldedaten gebildet:

Anschlusskennung + Teilnehmernummer + Mitbenutzernummer

Die Anschlusskennung beginnt in der Regel mit vier Nullen und ist zwölfstellig, danach folgt die Teilnehmernummer. Die Mitbenutzernummer ist in der Regel »0001«, vor dieser ist ein »Hash«, wenn die Teilhnehmernummer weniger als 12 Stellen umfasst. Das Kennwort ist das *t-online*-Paßwort.

Grundkonfiguration

Die Grundkonfiguration zur Einwahl über *t-online* erfolgt am einfachsten über YaST. Hier sind nur wenige Arbeitsschritte notwendig:

Für die ISDN-Konfiguration sind die Einwahlnummer 0191011, Benutzername und Kennwort anzugeben (Abbildung 3-100).

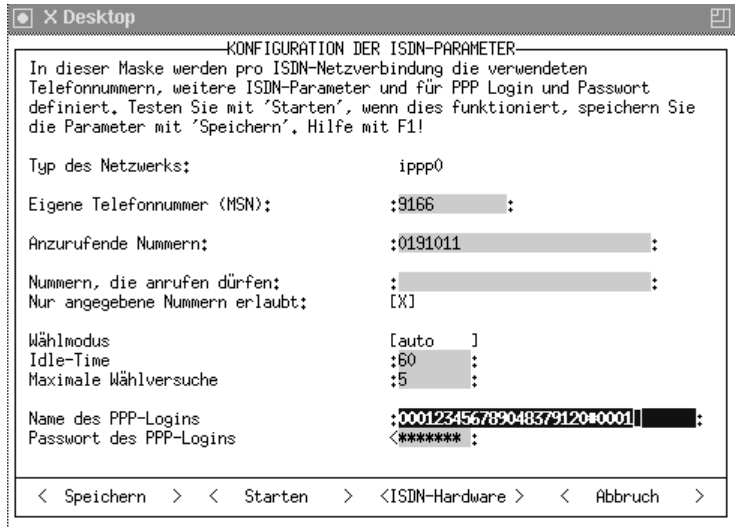


Abbildung 3-100 t-online Konfiguration mit YaST

Diese Parameter werden gespeichert und dann von YaST in */etc/ppp/pap-secrets*, */etc/ppp/options.ippp? und /etc/rc.config* eingetragen. Geprüft werden sollte, ob in */etc/rc.config* der Eintrag

```
IP_DYNIP=YES"
```

gesetzt ist. Zur vollständigen Konfiguration der Internetanbindung über *t-online* folgen die wichtigsten Angaben:

Nameserver	dns00.btx.dtag.de	194.25.2.129
SMTP Server	mailto.btx.dtag.de	
POP3 Server	pop.btx.dtag.de	
NNTP Server	news.btx.dtag.de	
FTP Proxy	ftp-proxy.btx.dtag.de	
HTTP Proxy	www-proxy.btx.dtag.de	
Wais Proxy	wais-proxy.btx.dtag.de	
Gopher Proxy	Gopher-proxy.btx.dtag.de	

Das E-Mail-System von *t-online* beherrscht die Protokolle SMTP und POP3 und kann mit allen dargestellten Mailprogrammen arbeiten.

E-Mail abholen

Der POP3-Server von *t-online* ist *pop.btx.dtag.de*. Als Benutzerkennung ist die BTX-Kennung + Mitbenutzernummer anzugeben. Mit

```
popclient -a -3 -u USER-ID -v -p PASSWORD pop.btx.dtag.de
```

wird das Abholen über das Programm *popclient* realisiert, eine entsprechende Konfiguration für *fetchmail* wurde bereits dargestellt.

E-Mail versenden

Hierfür wird SMTP Server *mailto.btx.dtag.de* verwendet. In */etc/rc.config* ist dazu

```
SENDMAIL_TYPE=smtp
SENDMAIL_SMARTHOST="mailto.btx.dtag.de"
```

einzutragen.

3.7.2 Konfiguration der E-Mail-Clients

Der Shell-Befehl mail

Mit dem Kommandozeilenbefehl *mail* kann bereits eine einfache Verwaltung eigener E-Mail realisiert werden. Natürlich ist damit auch der Zugriff auf alle diesem Benutzer gehörenden E-Mails möglich, ebenso das Versenden von Nachrichten über das Internet. Damit ist *mail* auch sehr sinnvoll, um nach dem Einrichten des Mail-Servers die ersten Funktionsprüfungen durchzuführen.

Wird *mail* aufgerufen, stehen eine Reihe von einfachen Kommandos zur interaktiven Bedienung zur Verfügung (Tabelle 3-29).

Taste	Aktion
P	zeigt die aktuelle Nachricht an (print). Bei mehreren Nachrichten kann mit [+] und [-] navigiert werden. Eine Nachricht kann auch direkt durch Angabe ihrer laufenden Nummer angezeigt werden.
D	löscht die aktuelle Nachricht (delete). Ist die laufende Nummer dieser Nachricht bekannt, kann diese über die Funktion [u] bei Bedarf sogar wiederhergestellt werden. delete kann auf mehrere Nachrichten gleichzeitig angewendet werden.
R	auf eine Nachricht antworten (reply)
Q	Sitzung beenden (quit)

Tabelle 3-29 mail Kommandos (Auswahl)

Eine neue Nachricht kann dadurch erstellt werden, dass *mail* mit der Empfängeradresse als Parameter aufgerufen wird. Mit

```
linux01: ~ # mail samulat@samulat.de
```

kann direkt eine E-Mail erstellt werden, die Texteingabe wird mit Strg D beendet.

Graphische E-Mail-Clients

E-Mail-Clients stehen heute für alle grafisch-orientierten Betriebssysteme zur Verfügung. Sie orientieren sich an den dargestellten Standards und ermöglichen die Erstellung, das Versenden und den Empfang von elektronischen Nachrichten.

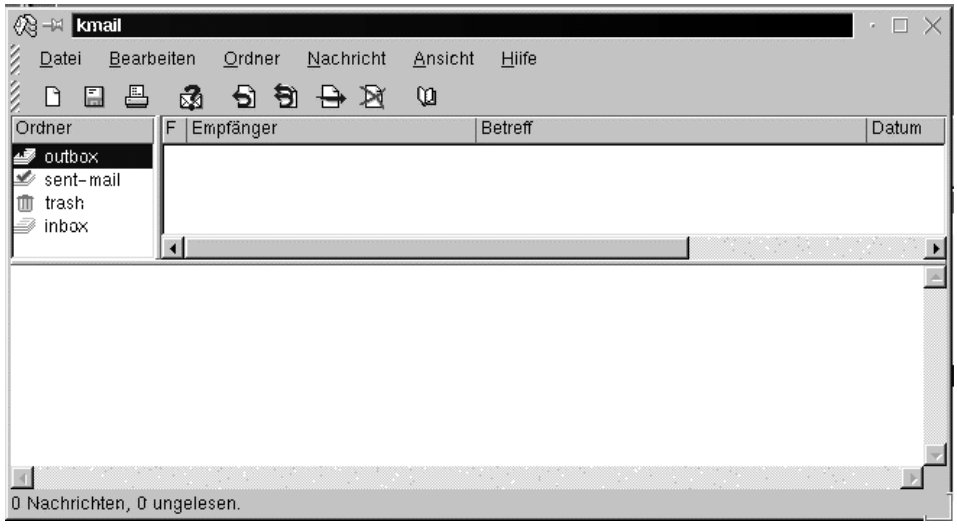


Abbildung 3-101 Bedieneroberfläche von kmail

Für KDE ist z. B. das Mail-Programm (*Mail-Reader*) *kmail* verfügar. *knewmail* ist eine Ergänzung, die den Transport von E-Mail als *Mail Transport Agent*, MTA durchführt (Abbildung 3-101).

Wesentlich häufiger anzutreffen sind aber Mail-Clients wie Microsoft Exchange Client, Outlook oder auch WWW-Explorer. Auch hier beschränkt sich die Konfiguration auf die Angabe des verwendeten Mailservers, zusätzlich muss die eigene Identifikation gewährleistet sein.

3.8 Router zur Netzwerkverbindung

Der Linux-Server eignet sich nicht nur als Router zur Internet-Anbindung, er ist auch hervorragend als Einwahlserver oder zur Verbindung von lokalen Netzwerken über das WAN geeignet.

Windows 9x/NT bietet dafür den *Remote Access Service* RAS, über den sich bis zu 256 Benutzer über Telefonleitungen mit einem NT-4.0-Server verbinden können. Auch RAS bietet umfangreiche Konfigurationsmöglichkeiten, die tatsächlichen Möglichkeiten werden aber erst deutlich, wenn man die Dokumentation und die Hilfsprogramme ansieht, die leider nicht zum Standard-Lieferumfang von NT gehören, sondern in der Technischen Referenz zum NT-Server zu finden sind.

Das Linux sehr gute und einfach zu handhabende Routingmöglichkeiten bietet, wurde bereits bei der Internetanbindung deutlich. Die Konfiguration als Einwahlserver bzw. als Router zu Verbindung in andere entfernte Netzwerke bietet sich geradezu an.

Soll Linux nur eine ISDN-Verbindung bei Bedarf herstellen können, so kann dies entsprechend der schon dargestellten Anbindung an einen Internet-Provider ausgeführt werden. Interessanter ist für den täglichen Betrieb aber eine andere Variante:

- Der Linux-Server verfügt über eine ISDN-Karte mit zwei Kanälen. Die ISDN-Device *ipppp0* ist fertig konfiguriert und wird als Verbindung zum Internet-Provider genutzt.
- Ein zweites ISDN-Device soll jetzt eingerichtet werden, dass bei Bedarf die Verbindung zu einem zweiten lokalen Netzwerk herstellt. Dieses zweite Netzwerk soll als Einwählpunkt ebenefalls einen Linux-Server erhalten, beide Teilnetze sollen über einen Router verbunden werden.

In Tabelle 3-30 ist die hier vorgestellte Konfiguration grafisch dargestellt, verbunden weden sollen zwei Netzwerke, die Einwahlpunkte sind die Server *linux01.forum.de* und *linux05.samulat.de*.

linux01.forum.de		linux05.samulat.de
etho 192.168.2.1 gw: 1.1.1.1		eth0 ip: 192.168.100.5 gw: 1.1.1.1
ippp0 192.168.3.1 PtP: 1.1.1.1	*Internet*	ippp0 192.168.101.5 PtP: 1.1.1.1
ippp1 192.168.4.1 gw: kein Eintrag PtP: 192.168.102.5 Tel: 01234 xxxxx User: isdn/isdn	*Netzwerkkopplung*	ippp1 192.168.102.5 gw: kein Eintrag PtP: 192.168.4.1 Tel: 01234 xxxxx User: isdn/isdn

Tabelle 3-30 Routerplanung

Tabelle 3-30 zeigt die Konfigurationen für die Devices *eth0, ippp0* und *ippp1. gw* kennzeichnet den Eintrag für das default-Gateway. Der zur Einwahl genutze Account auf beiden Servern ist *isdn* mit dem Kennwort *isdn*.

Etwas problematisch bei dieser Idee ist, dass auch das zweite ISDN-Device leider nur als RawPPP ausgeführt werden kann. Damit wird die Konfiguration umständlicher, es müssen Start- und Stoppskripte erstellt werden.

Die Einrichtung der zweiten ISDN-Device erfolgt wiederum einfacherweise über YaST (Abbildung 3-102).

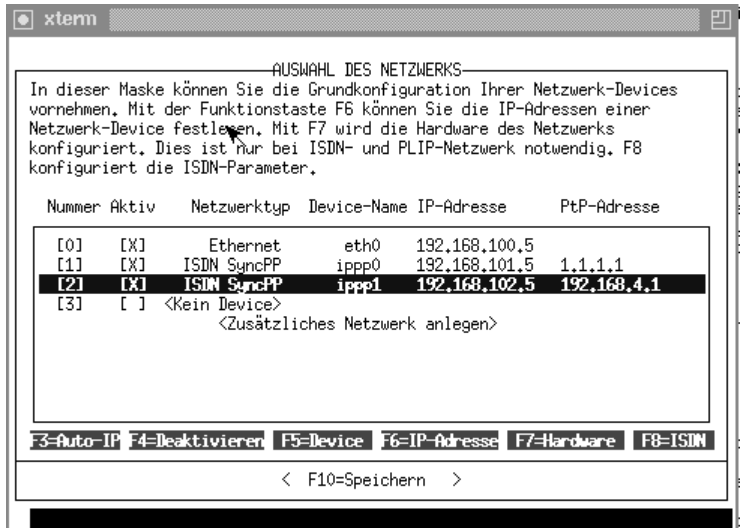


Abbildung 3-102 Konfiguration der ISDN-Device ippp1

Auch hier wird der Device *ippp1* mit 192.168.102.5 ein eigenes Teilnetz zugewiesen, die Adresse 192.168.4.1 ist die Adresse des PtP-Partners im entfernten Netzwerk. Die übrigen Konfigurationsdetails entsprechen den Arbeiten für *ippp0*, die Rufnummern müssen zum entfernten Netzwerk passen.

Wichtig ist jetzt die Konfiguration in */etc/route.conf,* mit der der Router bei Bedarf das interne Netzwerk entweder mit dem Provider oder mit dem zweiten Netzwerk verbindet:

```
#
# /etc/route.conf
#
1.1.1.1           0.0.0.0         255.255.255.255         ippp0
192.168.4.1       0.0.0.0         255.255.255.255         ippp1
192.168.2.0       192.168.4.1     255.255.255.0           ippp1
192.168.100.0     0.0.0.0         255.255.255.0           eth0
default           1.1.1.1
```

Das entfernte Netzwerk 192.168.2.0 wird über die Device *ippp1* erreicht, die Kommunikation wird über die *PtP-Partner* 192.168.4.1 abgewickelt. Alle Adressen, die in eigene Netzwerk oder in das Teilnetz 192,168.2.0 gehören, werden an das *default* Gateway 1.1.1.1 weitergeleitet, also an *ippp0*. Mit

```
linux01: ~ # ping 192.168.2.1
```

wird jetzt automatisch eine Verbindung über *ippp1* zum entfernten Netzwerk aufgebaut.

Pearson Education

ADDISON-WESLEY

A Anhang

A.1 Die GPL im Einzelnen

Das Linux-Betriebssystem, die meisten der in diesem Buch beschriebenen Programme und Teile des Handbuches selbst unterliegen der GNU General Public License (GPL). Mit dieser Lizenz machen die Urheber und Inhaber des Copyright ihr Produkt zu Freier Software. Das Wichtigste an der GPL ist die dahinter stehende Idee der Freien Software. Um dieser Idee auch in der wenig ideellen Welt des Softwaremarktes eine standfeste Position zu geben, hat Richard Stallman die Free Software Foundation gegründet und die General Public License herausgegeben.

Die General Public License ist die ausgefeilteste aller Lizenzen für Freie Software. Um Ihnen den Inhalt der GPL leichter verständlich zu machen, drucken wir hier einen im Wesentlichen inhaltlich mit der GPL übereinstimmenden Text.

Bei diesem Text handelt es sich nicht um eine durch die Free Software Foundation bestätigte Übersetzung der General Public License. Ähnlichkeiten mit dem Original sind beabsichtigt, sollen aber nicht zu einer Verwechslung dieses Textes mit der GPL selbst führen.

Im Vorwort zur GPL werden die wesentlichen Punkte der Lizenz ohne die verbindlichen Formulierungen des eigentlichen Lizenztextes eingeführt, im eigentlichen Lizenztext stehen die genauen Bedingungen für die Vervielfältigung, Verbreitung und Bearbeitung:

1.) Die General Public License (GPL) gilt für jedes Programm und jedes andere Werk, in dem ein entsprechender Vermerk des Urhebers darauf hinweist, dass das Werk unter den Bestimmungen der General Public License verbreitet werden darf. Im Folgenden wird jedes derartige Programm oder Werk als »das Programm« bezeichnet. Als „auf dem Programm basierendes Werk“ wird das Programm sowie jegliche Bearbeitung im Sinne des Urheberrechts bezeichnet; das bedeutet, ein Werk, das das Programm, auch auszugsweise, unverändert oder verändert, und/oder in eine andere (Compiler-) Sprache übersetzt, enthält. (Im Folgenden wird die Übersetzung ohne Einschränkung in »Bearbeitung« eingeschlossen.) Jeder Lizenznehmer wird im Lizenztext als »Sie« angesprochen. Andere Handlungen als Vervielfältigung, Verbreitung und Bearbeitung werden von der General Public License nicht berührt; sie fallen nicht in ihren Anwendungsbereich. Der Vorgang der Ausführung des Programms wird nicht eingeschränkt, und die Ausgabe des Programms unterliegt der Lizenz nur, wenn der Inhalt ein auf dem Programm basierendes Werk darstellt (unabhängig davon, dass die Ausgabe durch die Ausführung des Programms erfolgte). Ob dies zutrifft, hängt davon ab, was das Programm tut.

2.) Sie dürfen auf beliebigen Medien unveränderte Kopien des Quellcodes vom Programm, wie Sie ihn erhalten haben, anfertigen und verbreiten. Voraussetzung hierfür ist, dass Sie mit jeder Kopie einen entsprechenden Copyright-Vermerk, sowie einen Haftungsausschluß veröffentlichen. Sie müssen alle Vermerke, die sich auf die Lizenz und das Fehlen einer Garantie beziehen, unverändert lassen. Desweiteren müssen Sie allen anderen Empfängern des Programms zusammen mit dem Programm eine Kopie der GPL geben. Sie dürfen für den eigentlichen Kopiervorgang eine Gebühr verlangen, und es steht Ihnen frei, gegen Entgelt eine Garantie für das Programm anzubieten.

3.) Sie dürfen Ihre Kopie(n) des Programms oder einen Teil davon verändern, wodurch ein auf dem Programm basierendes Werk entsteht; Sie dürfen derartige Bearbeitungen unter den Bestimmungen des Abschnitts 1 vervielfältigen und verbreiten, vorausgesetzt, dass zusätzlich alle folgenden Bedingungen erfüllt werden:

(a) Sie müssen die veränderten Dateien mit einem auffälligen Vermerk versehen, der auf die von Ihnen vorgenommene Modifizierung und das Datum jeder Änderung hinweist.

(b) Sie müssen dafür sorgen, dass jede von Ihnen verbreitete oder veröffentlichte Arbeit, die ganz oder teilweise von einem freien Programm oder Teilen davon abgeleitet ist, Dritten gegenüber als Ganzes unter den Bedingungen der GPL ohne Lizenzgebühren zur Verfügung gestellt wird.

(c) Wenn das veränderte Programm normalerweise beim Lauf interaktiv Kommandos einliest, müssen Sie dafür sorgen, dass es, wenn es auf dem üblichen Wege für solche interaktive Nutzung gestartet wird, eine Meldung ausgibt oder ausdruckt, die einen geeigneten Copyright-Vermerk enthält sowie einen Hinweis, dass es keine Gewährleistung gibt (oder anderenfalls, dass Sie Garantie leisten) und dass die Benutzer das Programm unter diesen Bedingungen weiter verbreiten dürfen. Auch muss der Benutzer darauf hingewiesen werden, wie er eine Kopie der GPL ansehen kann. (Ausnahme: Wenn das Programm selbst interaktiv arbeitet, aber normalerweise keine derartige Meldung ausgibt, muss Ihr auf dem Programm basierendes Werk auch keine solche Meldung ausgeben).

Diese Anforderungen betreffen das veränderte Werk als Ganzes. Wenn identifizierbare Teile des Werkes nicht von dem Programm abgeleitet sind und vernünftigerweise selbst als unabhängige und eigenständige Werke betrachtet werden können, dann erstrecken sich die General Public License und ihre Bedingungen nicht auf diese Teile, sofern sie als eigenständige Werke verbreitet werden. Wenn Sie jedoch dieselben Teile als Teil eines Ganzen verbreiten, das ein auf dem Programm basierendes Werk darstellt, dann muss die Verbreitung des Ganzen nach den Bedingungen der GPL erfolgen. Hierbei werden die Rechte weiterer Lizenznehmer auf die Gesamtheit ausgedehnt, und damit auf jeden einzelnen Teil – unabhängig von der Person des Verfassers.

Es ist nicht der Zweck dieses Absatzes, Rechte für Werke zu beanspruchen oder Ihre Rechte an Werken zu bestreiten, die komplett von Ihnen geschrieben wurden; vielmehr ist es die Absicht der GPL, die Rechte zur Kontrolle der Verbreitung von Werken, die auf einem freien Programm basieren oder unter seiner auszugsweisen Verwendung zusammengestellt worden sind, auszuüben.

Die einfache Zusammenstellung eines anderen Werkes, das nicht auf dem freien Programm basiert, gemeinsam mit dem Programm oder einem auf dem Programm basierenden Werk, auf einem Speicher- oder Vertriebsmedium, fällt nicht in den Anwendungsbereich der GPL.

4.) Sie dürfen das Programm (oder ein darauf basierendes Werk wie in Abschnitt 2) als Objectcode oder in ausführbarer Form unter den Bedingungen von Abschnitt 1 und 2 vervielfältigen und verbreiten – vorausgesetzt, dass Sie dabei das folgende tun:

(a) Liefern Sie zusätzlich den vollständigen, zugehörigen, maschinenlesbaren Quellcode auf einem Medium, das üblicherweise für den Datenaustausch verwendet wird, wobei die Verteilung unter den Bedingungen der Abschnitte 1 und 2 erfolgen muss; oder

(b) Liefern Sie das Programm mit dem mindestens drei Jahre gültigen schriftlichen Angebot, jedem Dritten eine vollständige, maschinenlesbare Kopie des Quellcodes zu einem Preis, der die Kosten für die materielle Durchführung der Verteilung nicht übersteigt, zur Verfügung zu stellen. Der Quellcode muss unter den Bedingungen der Abschnitte 1 und 2 auf einem für den Datenaustausch üblichen Medium verbreitet werden; oder

(c) Liefern Sie das Programm mit der Information, die auch Sie als Angebot zur Verteilung des korrespondierenden Quellcodes erhalten haben. (Diese Alternative gilt nur für nicht-kommerzielle Verbreitung und nur, wenn Sie das Programm als Objectcode oder in ausführbarer Form mit einem entsprechenden Angebot nach Unterabschnitt b erhalten haben.)

Unter Quellcode eines Werkes wird die Form des Werkes verstanden, die für Bearbeitungen vorzugsweise verwendet wird. Für ein ausführbares Programm bedeutet vollständiger Quellcode: der gesamte Quelltext aller Module, die das Programm beinhaltet, zusätzlich alle zugehörigen Schnittstellendefinitionen, sowie die Scripte, die die Kompilierung und Installation des ausführbaren Programmes kontrollieren. Als besondere Ausnahme braucht der verteilte Quellcode nichts zu enthalten, was normalerweise (entweder als Quellcode oder in binärer Form) mit den Hauptkomponenten des Betriebssystems (Kernel, Compiler usw.) verteilt wird, unter dem das Programm läuft – es sei denn, diese Komponente gehört zum ausführbaren Programm.

Wenn die Verbreitung eines ausführbaren Programms oder des Objectcodes erfolgt, indem der Kopierzugriff auf eine dafür vorgesehene Stelle gewährt wird, so gilt die Gewährung eines gleichwertigen Zugriffs auf den Quellcode als Verbreitung des Quellcodes, auch wenn Dritte nicht gezwungen sind, die Quellen zusammen mit dem Objectcode zu kopieren.

5.) Sie dürfen das freie Programm nicht vervielfältigen, verändern, weiter lizenzieren oder verbreiten, sofern es durch die General Public License nicht ausdrücklich gestattet ist. Jeder anderweitige Versuch der Vervielfältigung, Modifizierung, Weiterlizenzierung und Verbreitung ist nichtig und beendet automatisch Ihre Rechte unter der GPL. Jedoch werden die Lizenzen Dritter, die von Ihnen Kopien oder Rechte unter der GPL erhalten haben, nicht beendet, solange diese die Lizenz voll anerkennen und befolgen.

6.) Sie sind nicht verpflichtet, die General Public License anzunehmen, da Sie sie nicht unterzeichnet haben. Jedoch gibt Ihnen nichts anderes die Erlaubnis, das Programm oder von ihm abgeleitete Werke zu verändern oder zu verbreiten. Diese Handlungen sind gesetzlich verboten, wenn Sie die Lizenz nicht anerkennen. Indem Sie das Programm (oder ein darauf basierendes Werk) verändern oder verbreiten, erklären Sie Ihr Einverständnis mit der General Public License und mit allen ihren Bedingungen bezüglich der Vervielfältigung, Verbreitung und Veränderung des Programms oder eines darauf basierenden Werkes.

7.) Jedesmal, wenn Sie das Programm (oder ein auf dem Programm basierendes Werk) weitergeben, erhält der Empfänger automatisch vom ursprünglichen Lizenzgeber die Lizenz, das Programm entsprechend den in der GPL festgelegten Bestimmungen zu vervielfältigen, zu verbreiten und zu verändern. Sie dürfen keine weiteren Einschränkungen für die Ausübung der darin zugestandenen Rechte des Empfängers vornehmen. Sie sind nicht dafür verantwortlich, die Einhaltung der Lizenz durch Dritte durchzusetzen.

8.) Sollten Ihnen infolge eines Gerichtsurteils, des Vorwurfs einer Patentverletzung oder aus einem anderen Grund (nicht auf Patentfragen begrenzt) Bedingungen (durch Gerichtsbeschluß, Vergleich oder anderweitig) auferlegt werden, die den Bedingungen der General Public License widersprechen, so befreien Sie diese Umstände nicht von den Bestimmungen in der GPL. Wenn es Ihnen nicht möglich ist, das Programm unter gleichzeitiger Beachtung der Bedingungen in der GPL und Ihrer anderweitigen Verpflichtungen zu verbreiten, dann können Sie als Folge das Programm überhaupt nicht verbreiten. Wenn zum Beispiel ein Patent nicht die patentgebührenfreie Weiterverbreitung des Programmes durch diejenigen erlaubt, die das Programm direkt oder indirekt von Ihnen erhalten haben, dann besteht der einzige Weg, das Patent und diese Lizenz zu befolgen, darin, ganz auf die Verbreitung des Programms zu verzichten. Sollte sich ein Teil dieses Abschnitts als ungültig oder unter bestimmten Umständen nicht durchsetzbar erweisen, so soll dieser Abschnitt seinem Sinne nach angewandt werden; im Übrigen soll dieser Abschnitt als Ganzes gelten. Zweck dieses Abschnittes ist nicht, Sie dazu zu bringen, irgendwelche Patente oder andere Eigentumsansprüche zu verletzen oder die Gültigkeit solcher Ansprüche zu bestreiten; dieser Abschnitt hat einzig den Zweck, die Integrität des Verbreitungssystems der freien Software zu schützen, das durch die Praxis öffentlicher Lizenzen verwirklicht wird. Viele Leute haben großzügige Beiträge zum weiten Bereich der mit diesem System verbreiteten Software im Vertrauen auf die konsistente Anwendung dieses Systems geleistet. Es liegt am Autor/Geber, zu entscheiden, ob er die Software mittels irgendeines anderen Systems verbreiten will; ein Lizenznehmer hat auf diese Entscheidung keinen Einfluß.

9.) Wenn die Verbreitung und/oder die Benutzung des Programmes in bestimmten Staaten entweder durch Patente oder durch urheberrechtlich geschützte Schnittstellen eingeschränkt ist, kann der Urheberrechtsinhaber, der das Programm unter die GPL gestellt hat, eine explizite geographische Begrenzung der Verbreitung angeben, indem diese Staaten ausgeschlossen werden, so dass die Verbreitung nur innerhalb und zwischen den Staaten erlaubt ist, die nicht ausgeschlossen sind. In einem solchen Fall beinhaltet die GPL die Beschränkung, als wäre sie im Lizenztext niedergeschrieben.

10.) Die Free Software Foundation kann von Zeit zu Zeit überarbeitete und/oder neue Versionen der General Public License veröffentlichen. Solche neuen Versionen werden vom Grundprinzip her der gegenwärtigen entsprechen, können aber im Detail abweichen, um neuen Problemen und Anforderungen gerecht zu werden. Jede Version der Lizenz hat eine eindeutig unterschiedliche Versionsnummer. Wenn das Programm angibt, welcher Version und »any later version« es unterliegt, so haben Sie die Wahl, entweder den Bestimmungen dieser Version zu folgen oder denen jeder beliebigen späteren Version, die von der Free Software Foundation veröffentlicht wurde. Wenn das Programm keine Versionsnummer angibt, können Sie eine beliebige Version wählen, die je von der Free Software Foundation veröffentlicht wurde.

11.) Wenn Sie den Wunsch haben, Teile des Programmes in anderen freien Programmen zu verwenden, deren Bedingungen für die Verbreitung anders sind, schreiben Sie an den Autor, um ihn um die Erlaubnis zu bitten. Für Software, die unter dem Copyright der Free Software Foundation steht, schreiben Sie an die Free Software Foundation; die FSF macht zu diesem Zweck manchmal Ausnahmen. Die Entscheidung darüber wird von den beiden folgenden Zielen geleitet: dem Erhalten des freien Status von allen abgeleiteten Werken der freien Software und der Förderung der Verbreitung und Nutzung von Software generell.

12.) Da das Programm ohne jegliche Kosten lizenziert wird, besteht keinerlei Gewährleistung für das Programm, soweit dies gesetzlich zulässig ist. Sofern nicht anderweitig schriftlich bestätigt, stellen die Urheber und/oder Dritte das Programm so zur Verfügung, „wie es ist“, ohne irgendeine Gewährleistung, weder ausdrücklich noch implizit, einschließlich, aber nicht begrenzt auf die Tauglichkeit und Verwendbarkeit für einen bestimmten Zweck. Das volle Risiko bezüglich Qualität und Leistungsfähigkeit des Programmes liegt bei Ihnen. Sollte das Programm fehlerhaft sein, übernehmen Sie die Kosten für notwendigen Service, Reparatur oder Korrektur.

13.) In keinem Fall, außer durch geltendes Recht gefordert oder schriftlich zugesichert, ist irgendein Urheber oder irgendein Dritter, der das Programm wie oben erlaubt modifiziert oder verbreitet hat, Ihnen gegenüber für irgendwelche Schäden haftbar, einschließlich jeglicher genereller, spezieller, zufälliger oder Folgeschäden, die aus der Benutzung des Programmes oder der Unbenutzbarkeit des Programmes folgen (einschließlich, aber nicht beschränkt auf Datenverluste, fehlerhafte Verarbeitung von Daten, Verluste, die von Ihnen oder einem Dritten erlitten werden, oder einem Versagen des Programms bei der Zusammenarbeit mit irgeneinem anderen Programm), selbst wenn ein Urheber oder Dritter über die Möglichkeit solcher Schäden unterrichtet worden war.

Die General Public License schließt mit einer Anleitung, wie Sie eigene Werke unter die GPL stellen können.

A.2 Wie wird die GPL auf eigene neue Programme angewendet?

Wenn Sie ein neues Programm entwickeln und wollen, dass es für größtmöglichen Nutzen für die Allgemeinheit ist, dann ist der beste Weg, dies zu erreichen, es zu freier Software zu machen, die jeder unter diesen Bestimmungen weiterverbreiten und verändern kann.

Um dies zu erreichen, fügen Sie die folgenden Anmerkungen zu Ihrem Programm hinzu. Es ist am sichersten, sie an den Anfang einer jeden Quelldatei zu stellen, um den Gewährleistungsausschluß möglichst deutlich darzustellen; außerdem sollte jede Datei mindestens eine *Copyright*-Zeile besitzen sowie einen kurzen Hinweis darauf, wo die vollständige Lizenz gefunden werden kann.

eine Zeile mit dem Programmnamen und einer kurzen Beschreibung

```
Copyright  ( C )  20yy            Name des Autors
This programm is free software; you can redistribute it and/or modify it under
the terms of the GNU General Public License as published by the Free Software
Foundation; either version 2 of the License, or
( at your option ) any later version.
This programm is distributed in the hope that it will be useful, but WITHOUT
ANY WARRANTY; without even the implied warrenty of MERCHANTABILITY or FITNESS
FOR A PARTICULAR PRUPOSE. See the GNU General Public License for more details.
You shoud have received a copy of the GNU General Public License along with the
program; if not, write to the Free Software Foundation, Inc., 675 Mass Ave,
Cambridge, MA 02139, USA.
```

Fügen Sie auch eine kurze Notiz hinzu, wie Sie postalisch (normal oder per E-Mail) erreichbar sind.

Wenn Ihr Programm interaktiv ist, sorgen Sie dafür, dass es nach dem Start einen kurzen Vermerk ausgibt:

```
Gnomovision version 69, Copyright ( C ) 19yy Name des Autors
Gnomovision comes with ABSOLUTELY NO WARRENTY; for details type ´show w ´ .
   This is free software, and you are welcome to redistribute it
   under certain conditions; type ´show w ´ for details.
```

Die hypothetischen Kommandos *show w* und *show c* sollten die entsprechenden Teile der GPL anzeigen. Natürlich können die von Ihnen verwendeten Kommandos anders heißen als *show w* und *show c*; es könnten auch einfach Mausklicks sein – was immer am besten in Ihr Programm paßt.

Wenn nötig, sollten Sie auch Ihren Arbeitgeber (wenn Sie als Programmierer arbeiten) oder Ihre Schule dazu bringen, einen Copyright-Verzicht für das Programm zu unterschreiben. Hier ist ein Beispiel mit geänderten Namen:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision` (witch makes passes at compilers) written by James Hacker.

Unterschrift von Ty Coon, 1 April 1989

Ty Coon, Vizepräsident

Diese General Public License erlaubt es nicht, das Programm in proprietäre Programme einzubinden. Wenn Ihr Programm eine Bibliotheksfunktion ist, kann es sinnvoller sein, das Binden proprietärer Programme mit dieser Bibliothek zu gestatten. Wenn Sie dies tun wollen, sollten Sie die *GNU Library General Public License* anstelle dieser Lizenz verwenden.

In HTML gewandelt 04-Aug-96 BeH

A.3 Die beiliegende CD-ROM

Die diesem Buch beiliegende CD-ROM wurde vom Autor zusammengestellt. Die CD enthält die Linux-Dokumentation, HOWTOs und weitere Informationen zur Installation und Betrieb eines Linux-Systems.

Viele Programme, die in diesem Buch vorgestellt werden, finden Sie auf dieser CD.

README	Aktuelle Informationen zu dieser CD
/usr/doc	Linux-Dokumentation, HOWTOs uvm.
/samba/tcpdump-smb	Das Programm tcpdump mit Erweiterung für SMB
/usv	USV Software der Hersteller APC und IMV
/util	Im Buch vorgestellte Linux Programme, die nicht zum Standardumfang der gängigen Distributionen gehören
/util/dos_client	NETBIOS-Client für MSDOS
/util/MySQL	Lizenzinformationen, Dokumentation und ODBC Treiber für die Datenbank MySQL
/util/perf-rstat	Programm perfstat
/util/phpMyAdmin	Installationsfile für phpMyAdmin
/util/webmin	Programm webmin in der Version 0.74
/util/x3270	Terminal-Emulation für IBM 3270
/etc	Konfigurationsdateien der im Buch beschreibenen Linux-Server
/scripts	Im Buch beschriebene Skripte und weitere Textdateien

Pearson Education

ADDISON-WESLEY

Quellenverzeichnis

Bücher	
[BoDe98]	Borkner-Delcarlo, O.: KDE 1.1 programmieren und anwenden, MITP- Verlag, Bonn, 1998.
[BoDe99]	Borkner-Delcaro, O.: Linux im kommerziellen Einsatz mit Samba 2.0.3. Hanser-Verlag, München, Wien, 1999.
[Gers95]	Gerschau, L.: Strukturierte Verkabelung. Komponenten, Übersichten,Standards. DATACOM-Verlag, Bergheim, 1995.
[GlHL95]	Glogau, D.; Hein, M.; Ladner, R.: Netzwerkausschreibung. Konzepte, Planung, Realisation. DATACOM-Verlag, Bergheim, 1995.
[Gonc00]	Goncalves, M.: Linux im Unternehmen. MITP-Verlag, Bonn, 2000.
[Hein98]	Hein, J.: Linux-Systemadministration, 3. Auflage, Addison-Wesley, München, 1999.
[Hsti99]	Holtz, H.; Schmitt, B.; Tikart, A.: Linux für das Internet und Intranet, 3. Auflage, MITP-Verlag, Bonn, 1999.
[Kirh96]	Kirch, O.: Linux. Wegweiser für Netzwerker, O'Reilly/International Thomson Verlag, Bonn, 1996.
[Kofl98]	Kofler, M.: Linux. Installation, Konfiguration, Anwendung. 4. Auflage, Addison-Wesley, München, 1999.
[Rais99]	Raison, A.; Kirsch,C.: Linux im Einsatz, dpunkt-Verlag, Heidelberg, 1999.
[Rosh96]	Roscher, A.: LINUX als Windows Server. PC-Vernetzung mit LINUX/UNIX und dem Internet. dpunkt Verlag, Heidelberg, 1996.
[Stik98]	Stickdorn, R.: ADABAS D. Die professionelle Datenbank für Linux und Windows, dpunkt Verlag, Heidelberg, 1998.
[Tikg99]	Tikart, A.; Kühnle, J.; Gentara, L.: Linux für Anwender. MITP-Verlag, Bonn, 1999.
[WaEv97]	Wasburn, K.; Evans, J.: TCP/IP. Aubau und Betrieb eines TCP/IP-Netzwerkes. 2. Auflage, Addison-Wesley, München, 1997.
[Yarg99]	MySQL & mSQL. Reese,G.; King,T. O'Reilly, Bonn, 1999.
[Zenk94]	Zenk, A.: Lokale Netze – Kommunikationsplattform der 90er Jahre. Addison-Wesley,München , 1994.

Zeitschriften / Aufsätze	
[Dilu99]	Diedrich, O.; Lubitz, H.: Sichere Reise. Ein Firewall mit Linux. In: c't 3/99, S. 154.
[DuPR99]	Dupke, K.; Povel, M.; Renner, F.: Heiße Linie. Telefon-Support für Linux im Test. In: iX 11/99, S. 90 ff.
[Endr98]	Endres, J.: Druckausgleich. Linux als Multiprotokoll-Printserver. In: c't 24/98, S. 186 ff.
[Fey99]	Fey, J.: Schweizer Messer. Freies Analyse-Tool für Netzwerkverwalter. In: Gateway 5/99, S. 56.

[Herm99] Hermans, R.: Subito! Einmal Intranet in 24 Stunden. In: Linux-Magazin 9/99, S. 42.

[Kirs99] Kirsch, Ch.: Windows unter Linux nutzen. In: iX 5/99, S. 90.

[Köhn98] Köhntopp, K.: Spinne im Netz. Samba macht Linux zum Windows Server. In: c't 20/98, S. 204.

[Kruc00] Kruck,W.: Daten im Griff. MySQL-Datenbankadminstration mit phpMyAdmin. In: Linux-Magazin 2/2000, S. 64-66.

[KuSc99] Kuhn, B.; Schwaller, T.: Doppelt hält besser. Hochverfügbarkeitssystem für Linux. In: Linux-Magazin 4/99, S. 36-39.

[PC14] PC Magazin Spezial 14: 1999 Linux für Anwender.

[Such99] v. Suchodoletz, D.: Gut gebootet. Diskless X-Terminals unter Linux. In: Linux-Magazin 8/99, S. 38.

[Wöjü99] Wölfer, T.; Bernau, J.: Back to the Roots. Einführung in die Linux-Systemverwaltung. In: PC Magazin 9/99, S. 173.

Stichwortverzeichnis

A
a2ps *179*
Absicherung gegen Viren *15*
Apache *252*, *286*
Application-Server *16*, *247*
Applikationsfilter *301*
apsfilter *170*
ARP *133*
ARP-Spoofing *68*
Auslagerungsdatei *27*

B
Bandsicherung *20*
BogoMips *78*
Bootdiskette *79*
bootp *140*
Brandabschnitt *20*

C
CDIR *130*
CD-ROM *327*
CD-ROM-Server *15*, *193*, *217*
cgi *159*
Cluster *68*
cron-Daemon *120*

D
DAT *20*, *31*
DCF *242*
DHCP *12*, *140*, *162*
DHCP relay agent *144*
DHCP-Client *144*
DHCP-Server *143*
DIN 50173 *39*
DLC *19*
DNS *11*, *297*
Druckerkonfiguration *167*

E
ECC *28*
EMV *36*
eth0 *48*
export *51*
ext2fs *42*

F
FAQ *98*
fetchmail *311*
Fileserver *16*
Firewall *21*, *299*
FQDN *11*
FTP *303*
Funkuhr *13*, *242*
fvwm95 *50*

G
Gateway *19*
Gopher *303*
GPS *246*

H
Hostname *10*
hosts *10*
Hot Standby *67*
Hot Swap *31*
HOWTO *98*
HTML *253*

I
I4L *290*
IBM 3270 *269*
IDS-Tool *276*, *279*
ifconfig *48*
Installationspunkt *15*
Intranet *285*
Inventarisierung *21*
IP-Adresse *10*
IP-Masquerading *298*
ISDN *22*, *241*, *289*
isdnctrl *295*
isdnmon *295*

K
KDE
- grafisches Login *51*
- Grundkonfiguration *53*
- Installation *50*
- kcmdhcpd *144*
- kcrontab *121*
- kdat *274*
- kdf *113*
- kdm *51*
- kfile system control *115*
- kfinger *146*
- khelp *55*
- klpq *178*

kmuser *96*
knetmon *234*
knu *146*
k-panel *55*
kpm *111*
kpstree *107*
ksamba *226*
ksniffer *147*
ktail *103*
ktalk *146*
ktop *110*
kwebwatc *308*
kdm *51*
Konfigurationsdateien
/etc/dhcpd.conf *142*
/etc/exports *233*
/etc/fstab *113*, *187*
/etc/host.conf *138*
/etc/hosts *136*
/etc/hosts.allow *137*
/etc/hosts.deny *137*
/etc/hosts.equiv *138*
/etc/inetd.conf *235*
/etc/inittab *73*, *74*
/etc/lilo.conf *71*
/etc/networks *138*
/etc/printcap *167*, *181*
/etc/protocols *139*
/etc/rc.config *295*
/etc/resolv.conf *139*
/etc/route.conf *298*, *319*
/etc/services *139*
/etc/smb.conf *197*, *227*
Benutzerverwaltung *96*

L
LAN *38*
LILO *41*, *71*
Logdateien *208*
Loopback *165*

M
MAC-Adresse *48*
Mainframe *25*
Makroviren *271*
marsnwe *186*
mc *92*, *156*
MTBF *29*
Multiprotokollrouter *23*
Multitaskingsystem *5*
mvm *50*

N
netatalk *186*
net-Befehle *210*
NetBIOS-Client *210*
Netzwerkstatistik *20*
NFS *41*, *232*
NIC *10*, *129*
NNTP *253*
NTP *237*

O
ODBC *247*, *258*, *267*
Orange Book *66*
OS/2 Warp Connect *214*

P
Paketfilter *299*
PHP *253*
POP3 *253*, *308*
popclient *315*
popper *309*
PostgreSQL *249*
Postscript *170*
Printserver *15*
procmail *312*
Proxy-Server *21*, *301*, *303*

Q
Quoting *14*, *114*, *186*

R
Rack Monitoring Systems *38*
RAID *30*, *45*, *69*
RAS *317*
Relayschutz *310*
Reliability *24*
RFB *154*
RFC *99*, *127*
Root-Filesystem *42*
Router *22*, *290*
RPC-Daemon *277*

S
Samba
Benutzerprofil *224*
Broadcasts *297*
ksamba *226*
lmhosts *215*
Login Skript *223*
mksmbpasswd.sh *221*
NETLOGON *224*

- nmbd *195*
- nmblookup *207*
- PDC *196*, *220*
- SMB Protokoll *193*
- smbclient *206*, *207*, *218*
- smbd *195*
- smbmount *234*
- smbpasswd *205*
- smbstatus *207*
- swat *227*
- Systemrichtlinie *225*
- testparm *204*
- WINS Server *215*

sax *50*
SCSI *29*
sendmail *297*, *310*
Server-Up-Time *24*
Shellbefehle
- adduser *222*
- alias *94*
- apropos *86*
- arp *133*, *184*
- cat *89*, *221*
- cd *83*
- chmod *93*, *188*, *224*, *273*, *312*
- chown *223*
- clock *118*, *236*
- cp *90*
- cron *271*, *313*
- crontab *121*
- date *119*
- df *114*
- echo *95*
- env *95*
- export *151*
- fdformat *116*
- find *92*, *114*
- grep *92*
- gunzip *98*
- ifconfig *134*, *296*
- init *73*, *125*, *295*
- kill *109*
- less *90*
- ln *91*
- lpc *177*
- lpq *177*
- lpr *176*
- lprm *177*
- ls *89*
- lsmod *166*
- mail *315*
- man *86*
- mkdir *89*
- mkfs *117*
- mknod *84*
- mnt *272*
- mount *93*, *117*, *217*, *233*
- mv *90*
- netdate *236*
- netstat *135*
- nice *109*
- nohup *152*
- passwd *96*
- ping *49*, *134*, *298*
- ps *105*, *205*
- pstree *107*
- pwd *81*
- rdev *79*
- rm *90*
- rmdir *89*
- setterm *88*
- shutdown *75*, *79*
- sleep *273*
- su *78*, *150*
- tail *102*, *291*
- tar *272*
- tee *152*
- tkman *87*
- top *106*
- touch *188*
- wall *95*
- watch *107*
- xman *87*

Shell-Skript *217*
SID *221*
Skalierbarkeit *24*
SMS-Nachricht *125*
SMTP *308*
SNMP *21*, *38*, *253*, *276*
SNTP *237*, *240*
Socket *133*
SPAM *310*
Special Files *83*
SQL *18*, *247*
Squid *303*
SSL-Verbindungen *159*
startx *51*
STRATUM *235*
Streamer *271*
Striping-System *30*
strukturierte Verkabelung *38*
Subnet-Mask *130*

Systembefehle
zcat *98*

T
TCP/IP *9*
tcpdump *210*, *226*
telnet *14*, *150*, *269*, *309*
TeraTerm Pro *152*
Timeserver *235*
TK-Anlage *291*
t-online *314*
Top-Level-Domain *11*

U
Überspannungsschutz *36*
Umwandlung von Textdateien *82*
UNC *208*, *218*, *236*
USV *34*, *56*
UUCP *308*

V
Virenscanner *270*
VNC *154*

W
WAN *16*, *127*
webmin *122*, *159*, *191*, *230*, *288*, *304*, *311*

X
X11 *50*
xntp *237*

Z
Zeitserver *235*
Zeitsynchronisierung *13*