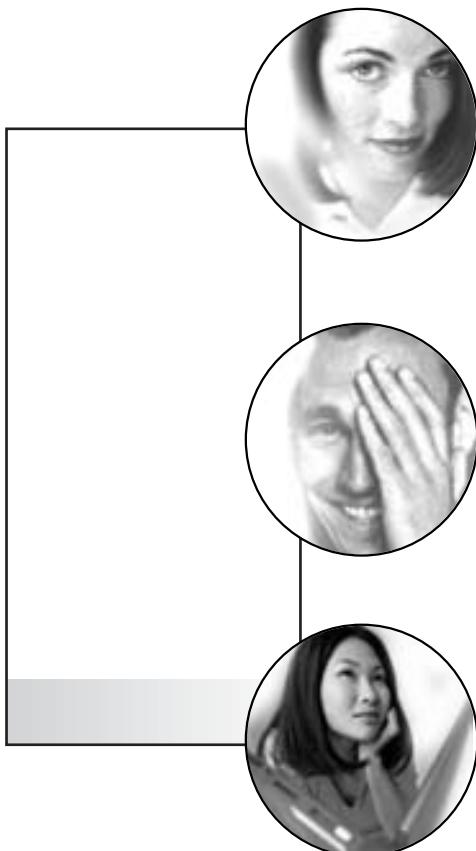


Frank Gehrke/Günther Klappheck/Peter Gelinski

Linux – Das Buch



Fast alle Hard- und Software-Bezeichnungen, die in diesem Buch erwähnt werden, sind gleichzeitig auch eingetragene Warenzeichen und sollten als solche betrachtet werden. Der Verlag folgt bei den Produktbezeichnungen im Wesentlichen den Schreibweisen der Hersteller.

Der Verlag hat alle Sorgfalt walten lassen, um vollständige und akkurate Informationen in diesem Buch bzw. Programm und anderen evtl. beiliegenden Informationsträgern zu publizieren. SYBEX-Verlag GmbH, Düsseldorf, übernimmt weder die Garantie noch die juristische Verantwortung oder irgendeine Haftung für die Nutzung dieser Informationen, für deren Wirtschaftlichkeit oder fehlerfreie Funktion für einen bestimmten Zweck. Ferner kann der Verlag für Schäden, die auf eine Fehlfunktion von Programmen, Schaltplänen o.Ä. zurückzuführen sind, nicht haftbar gemacht werden, auch nicht für die Verletzung von Patent- und anderen Rechten Dritter, die daraus resultiert.

Projektmanagement: Katja Roth

Endkontrolle: Mathias Kaiser Redaktionsbüro, Düsseldorf

Satz: reemers publishing services gmbh, Krefeld

Farbreproduktionen/Umschlaggestaltung: Guido Krüsselberg, Ratingen

Belichtung, Druck und buchbinderische Verarbeitung: Bercker, Grafischer Betrieb, Kevelaer

ISBN 3-8155-0311-6

1. Auflage 2001

Dieses Buch ist keine Original-Dokumentation zur Software der Firma Microsoft. Sollte Ihnen dieses Buch dennoch anstelle der Original-Dokumentation zusammen mit Disketten verkauft worden sein, welche die entsprechende Microsoft-Software enthalten, so handelt es sich um Raubkopien der Software. Benachrichtigen Sie in diesem Fall umgehend Microsoft GmbH, Edisonstr. 1, 85716 Unterschleißheim. Auch die Benutzung einer Raubkopie kann strafbar sein. Der Verlag und Microsoft GmbH

Alle Rechte vorbehalten. Kein Teil des Werks darf in irgendeiner Form (Druck, Fotokopie, Mikrofilm oder in einem anderen Verfahren) ohne schriftliche Genehmigung des Verlags reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

Printed in Germany

Copyright © 2001 by SYBEX-Verlag GmbH, Düsseldorf



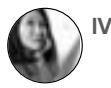
Vorwort	XXIII
Erstinstallation	Kapitel 1 1
Bootstrapping the new user	Kapitel 2 73
Linux-Dateisystem	Kapitel 3 93
Benutzerumgebung	Kapitel 4 213
Systemadministration – der Bootvorgang	Kapitel 5 349
Userverwaltung	Kapitel 6 369
Systemverwaltung	Kapitel 7 385
Konfiguration und Kompilierung des Kernels	Kapitel 8 395
Windows für Linux? – Emulatoren	Kapitel 9 401
Sicherheit ist Trumpf: Datensicherung	Kapitel 10 429
Wenn es mal nicht sofort klappt: Troubleshooting	Kapitel 11 441
X-Konfiguration und KDE	Kapitel 12 455
Linux im Netzwerk	Kapitel 13 495
Mit Linux ins Internet	Kapitel 14 575
Programmierung	Kapitel 15 617
Shell-Programmierung	Kapitel 16 625
Programmierung in Perl	Kapitel 17 671
HTML und CGI	Kapitel 18 719
Die Tastaturbelegung	Anhang A 779



Projektübersicht



Kernel-Parameter	Anhang B	781
HTML-Sonderzeichen	Anhang C	799
Die GPL		802
Index		803



Inhalt



Vorwort	XXIII
Erstinstallation	Kapitel 1
Installation von Linux und anderen Betriebssystemen auf der gleichen Festplatte	3
FIPS	4
Installation der mit dem Buch mitgelieferten SuSE-Distribution – YaST1 oder YaST2?	5
Erstellen einer Linux-Bootdiskette	5
Start der Installation	8
Die eigentliche Installation	8
Linux als einziges System	16
Linux als Installation neben (einem) anderen System(en)	20
Einige Grundlagen zur Partitionierung	22
Die SuSE-Pakete	27
Programmpakete	31
Weitere Installationsschritte	33
Konfiguration von LILO	34
Wahl der Zeitzone	38
Netzwerkkonfiguration	39
Erster Start des Systems	46
Benutzereinrichtung	46
Konfiguration der Maus	48
Die Installation mit YaST2	52
Start mit der dem Buch beiliegenden CD	53
Bootstrapping the new user	Kapitel 2
Die Shell	74
Dateinamen	75
Multiuser	76
Benutzerrechte	76

v



Inhalt



Multitasking	77
Das Filesystem	77
Linux und andere Filesysteme	79
Editoren	80
Einfache Dateioperationen	80
Listen von Dateien	80
Kopieren von Dateien	80
Verschieben von Dateien	81
Umbenennen von Dateien	81
Löschen von Dateien	81
Erzeugen bzw. Löschen von Inhaltsverzeichnissen	82
Onlinehilfe	82
Aliase	83
Hilfe – die Statusmeldungen laufen während des Bootens zu schnell vorbei!	85
Dienstprogramme	86
<i>unix2dos</i>	86
<i>df</i>	86
<i>gzip</i>	86
<i>gpm</i>	87
<i>whoami</i>	88
<i>history</i>	88
<i>find</i>	89
<i>mount</i> und <i>umount</i>	90
<i>clear</i>	91
<i>cal</i>	91
Oh Wunder ...	91
Midnight Commander	92
Herunterfahren des Systems	92





Linux-Dateisystem	Kapitel 3	93
Allgemeines		94
Überblick über das Linux-Dateisystem		95
Dateitypen		96
Die Dateinamensregel		99
Dateinamenskonventionen		102
Ein typischer Linux-Verzeichnisbaum		105
Arbeiten im Dateisystem		110
Verzeichnisse		110
<i>pwd</i>		111
<i>cd</i>		111
<i>mkdir</i>		113
<i>rmdir</i>		115
Der Befehl <i>ls</i>		116
Dateien		129
<i>cat</i>		129
<i>cp</i>		131
<i>mv</i>		134
<i>rm</i>		135
<i>ln</i>		136
Besondere Dateibefehle		144
<i>file</i>		144
<i>find</i>		145
Ausgabe von Dateiinhalten, Druck		151
<i>more</i>		151
<i>less</i>		154
<i>head</i> und <i>tail</i>		154
<i>wc</i>		155
<i>grep</i>		156
Druck von Dateien		161
<i>lpr</i>		162
<i>lpd</i> – der Druckdämon		163
<i>lpq</i>		163
<i>lprm</i>		164
<i>lpc</i>		164



Inhalt



Datensicherung	166
<i>tar</i>	166
Rechte	166
Rechte und ihre Bedeutung	167
Rechte und Gruppen	171
Befehle für die Rechtevergabe	172
<i>chmod</i>	173
<i>umask</i>	175
<i>chown</i>	177
<i>chgrp</i>	177
Besonderheiten des x-Rechts	177
Einrichten des Drucksystems	179
Grundlagen	180
<i>lpd</i> automatisch starten	181
Einrichten des Spool-Bereichs	182
Bereitstellen der Informationen für <i>lpd</i> – die Datei <i>/etc/printcap</i>	185
Druckerfilter	188
Der Filter <i>apsfilter</i>	190
Aufbau eines Dateisystems	197
Zugriff auf eine Datei	199
Arbeiten mit Dateisystemen	202
Erstellen eines Dateisystems	202
<i>mkfs</i>	202
Mounten von Dateisystemen	205
<i>mount</i>	206
<i>umount</i>	207
Prüfen von Dateisystemen	209
<i>fsck</i>	209
<i>df</i>	210
<i>du</i>	210



Inhalt



Benutzerumgebung	Kapitel 4
Grundlegende Befehle	214
Besonderheiten bei der Eingabe	215
Informationen über Benutzer	218
Die Datei <i>/etc/passwd</i> und das Login	222
Benutzergruppen	225
Identitätswechsel und Passwortvergabe	228
Onlinehilfen	230
Weitere nützliche Befehle	233
Kommunikation	237
Arbeiten mit der bash	243
Dateinamensexpression	245
Datenumleitung	247
<i>Here-Dokumente</i>	252
Pipelines	254
Wie eine Shell arbeitet	257
Maskierung der Metazeichen	262
Die Befehlseingabe	264
Der Befehlszeileneditor	265
Einfache Befehlsverkettungen	270
Bedingte Befehlsausführung	271
Befehlssubstitution	272
Hintergrundverarbeitung	274
Alias-Namen	276
Funktionen	277
Shell-Variable	278
Andere Shells	284
Grafische Benutzeroberflächen	288
Editoren	289
<i>ed</i>	289
<i>vi</i>	295
<i>joe</i>	305
<i>sed</i> (Stream-Editor)	312
Linux-Werkzeuge	317
Der Midnight Commander	317
<i>awk</i> – ein Mustererkennungsprogramm	325
<i>bc</i> – ein einfacher Taschenrechner	337
<i>Xcdroast</i>	343
Überblick über weitere Werkzeuge	346



Inhalt



Systemadministration – der Bootvorgang	Kapitel 5	349
Der Bootvorgang		351
Warum Bootparameter?		351
Bootparameter		353
Allgemeine Einstellungen		353
Das Root-Laufwerk		353
Kein Autoprobing bestimmter Speicherbereiche		354
Defekter Co-Prozessor		354
Defekte CPU-Instruktion <i>hlt</i>		354
RAM > 64 MByte		354
SCSI		355
Adaptec-Host-Adapter AHA-1505/1520/1510/1515/1522 und Soundblaster 16-SCSI		355
Adaptec-Host-Adapter AHA-1540/1542		356
Adaptec-Host-Adapter AHA-274x, AHA-284x und AHA-294		356
BusLogic-SCSI Host-Adapter		356
Future Domain TMC-8xx und TMC-950		357
Pro Audio Spectrum		357
Seagate ST-01/02		357
Trantor T128		357
SCSI-Bandlaufwerke		357
Proprietäre CD-ROMs		358
Aztech		358
Sony CDU-31A und CDU-33A		358
Sony CDU-531 und CDU-535		359
GoldStar		359
Mitsumi		359
Mitsumi XA/MultiSession		359
Optics Storage		360
Philips CM206		360
Sanyo H 94 A		360
SoundBlaster Pro		360
Diskettenlaufwerke		360
IBM Thinkpad		361
HP Omnibook		361
IBM L 40 SX		361



X

Inhalt



PCI-Boards	361
Schnell oder sicher?	362
Nur ein Floppycontroller	362
Zwei Floppycontroller	362
Veraltete Optionen	362
Mehr als zwei Laufwerke	362
IDE/EIDE/ATAPI	363
Standard-IDE-Festplatten	363
Chipsatzunterstützung	364
ATAPI-CD-ROM-Laufwerke	365
Ethernet	365
Sonstige Geräte	366
Logitech Busmaus	366
XT-Festplatte	366
Weitere Informationen	366
Userverwaltung	Kapitel 6
User-Account	369
<i>useradd</i>	370
<i>passwd</i>	371
<i>groupadd</i>	375
<i>groups</i>	377
Modifikation von Benutzern und Gruppen	378
Modifikationsbefehle	378
Lösrbefehle	382
Systemverwaltung	Kapitel 7
Prozesse	386
Zeitversetzte Prozesse	387
<i>at, batch, cron</i>	388
<i>at</i>	388
<i>batch</i>	389
<i>cron</i>	389
<i>mount und umount</i>	391



Inhalt



Dämonen	393
<i>mountd</i>	393
<i>nfsd</i>	393
<i>crond</i>	393
<i>syslogd</i>	393
<i>klogd</i>	394
<i>inetd</i>	394
<i>lpd</i>	394
<i>kerneld</i>	394
Konfiguration und Kompilierung des Kernels	Kapitel 8
Konfiguration	395
Das Kompilieren	396
Kernel mit Modulen	397
Große Kernel	398
Die Kernel-Parameter	399
Prozessoren	399
Laptops	399
USB	399
Sound	400
Video for Linux	400
... und, und, und	400
Windows für Linux? – Emulatoren	Kapitel 9
Warum werden Emulatoren benutzt?	401
VMware	402
Funktionsprinzip von VMware	404
Systemvoraussetzungen	406
Wann setzt man VMware ein?	407
Bezugsquellen	408
Paketinstallation	409
Konfiguration des Softwarepaketes	410
Installieren von <i>vmtools</i>	413
Einrichten eines virtuellen PCs	414





Installation des Gast-Betriebssystems	420
Arbeiten mit VMware	426
Sicherheit ist Trumpf: Datensicherung	Kapitel 10
Disketten	430
Wechselplatten	431
Bandlaufwerke	431
Festplatten	432
Das Archivierungsprogramm <i>tar</i>	432
Funktionen	433
Optionen	434
Beispiele für die Arbeit mit <i>tar</i>	438
Wenn es mal nicht sofort klappt: Troubleshooting	Kapitel 11
CD-ROM	443
Probleme während der Installation	443
Das CD-ROM-Laufwerk ist nach der Installation „nicht mehr vorhanden“	443
Der Mitsumi-Treiber meldet „mcd: init failed“	445
Timing-Probleme beim CD-ROM/RAM-Zugriff	445
<i>mount</i> funktioniert nicht	446
Probleme beim Floppy-Boot	446
Soundkarten und LILO	447
Probleme beim Starten mit LOADLIN	447
Zu wenig Speicher	447
„Einfrieren“ beim Hochfahren des Kernels	448
Probleme mit SCSI-Konfigurationen	449
Prinzipielle Probleme mit SCSI	449
SCSI und alte Netzwerkkarten	450
Designfehler bei Prozessoren	450
Ältere i486-DX/100-Prozessoren	450
Co-Prozessoren	451



Inhalt



Probleme mit dem EIDE-Controller	451
Probleme mit Laptop-Diskettenlaufwerken	451
Kein deutscher Tastaturtreiber	451
LILO ist verschwunden!	452
Probleme in der Handhabung von Linux	452
Ich kann mich nur als <i>root</i> einloggen	452
Der Bildschirm enthält nur wirre Zeichen, keine Buchstaben	453
Beim Kompilieren erfolgt die Fehlermeldung „signal 11: internal compiler error“	453
Fehler beim Kompilieren des Kernels	453
Als <i>root</i> erhalte ich die Fehlermeldung „command not found“	454
X-Konfiguration und KDE	Kapitel 12
Die Entwicklung der grafischen Betriebssystemaufsätze	456
Die Funktionsweise von X Window	457
Die X-Server	457
Der Window-Manager	458
Die Einrichtung der grafischen Oberfläche	458
Framebuffer	459
Sax	459
Mauseinstellungen	461
Tastatur	462
Grafikkarte	463
Monitor	463
Bildschirm	464
Die Konfiguration mittels <i>xf86config</i>	466
Grafikkarte zu neu?	466
KDE	467
K-Office 1.0	467



Inhalt



Die Bedienung des K-Desktops	468
Das Panel	469
Der Konqueror	471
Die Modi des Konquerors	473
Die URL-Leiste	473
Ansicht und Einstellungen	474
Dateioperationen	474
Hilfe!	475
K-Console – die Terminal-Emulation	475
Die virtuellen Bildschirme	476
Die Taskleiste	477
Icons auf dem Desktop	478
Das Kontextmenü	478
Der Disk-Navigator	480
Kandalf	481
Konfiguration des K-Desktops	481
Angeschlossene Geräte	482
Dateien	483
Energiekontrolle	483
Erscheinungsbild	483
Hilfe	486
Information	486
System	489
Webbrowser	489
YaST2	490
Das Einrichten von Icons auf dem Desktop	490
Nie mehr die Befehlszeile?	491
KDE vs. Gnome	492
KDE – Heute und morgen	493
Linux im Netzwerk	Kapitel 13
	495
Offene Thin-Ethernet-Verkabelung	497
Konfektionsware vs. Eigenbau	500
Thin-Ethernet mit Dosen	503
Twisted-Pair-Verkabelung	504
IP-Adressen	506
IP-Adressierung	506



Inhalt



Linux im heterogenen Netzwerk	514
NFS	515
Besondere Netzwerkimplementierungen	516
Der Auto-Mounter	517
Die Konfiguration der „Clients“	528
Weitere Auto-Mounts mit <i>/etc/auto.misc</i>	529
Die Konfiguration des SAMBA-Servers	531
Die Datei <i>/etc/smb.conf</i>	532
Spezielle Sektionen	534
Variablensubstitution	536
Parameter	537
<i>testparm</i>	544
<i>smbclient</i>	545
Einbindung eines SAMBA-Clients – a case study	546
Der MARS-Server	551
Installation	552
Sektion 1: Die Novell-Volumes	553
Sektion 2: Der Servername	554
Sektion 3: Interne Netzwerknummer	554
Sektion 4: IPX-Devices	555
Sektion 5: Besondere Geräteoptionen	556
Sektion 6: Serverversion	556
Sektion 7: Behandlung der Passwörter der DOS-Clients	556
Sektion 8: Verhalten beim Login und Dateioperationen	557
Sektion 9: Rechtevergabe beim Erzeugen von Verzeichnissen und Dateien	558
Sektionen 10 und 11 : User ID und Group ID vor dem Einloggen	558
Sektion 12: Supervisor-Login	559
Sektion 13: User-Logins	559
Sektion 14 wird derzeit nicht genutzt	560
Sektion 15: Die automatische Einordnung von Linux-Benutzern als MARS-Benutzer	560
Sektion 16: Überprüfungen während des Hochfahrens	560
Die Sektionen 17 bis 20 werden derzeit nicht benutzt	561
Sektion 21: Drucker-Queues	561
Sektion 22: Printserver	561
Starten des Servers	562
DOS-Clients für MARS	562



Inhalt



Linux-Client für MARS	563
Drucken unter MARS	564
Erfahrungen	565
Weitere Informationen	565
Up, up and away – die Anbindung an das Internet	565
Modem, ISDN-Karte oder ISDN-Modem?	566
Mit Linux ins Internet	Kapitel 14
Voraussetzungen für den Internetzugang	578
Mailing	579
E-Mail mit <i>elm</i>	579
Newsgroups	587
Newsreader	588
World Wide Web	588
Lynx	589
Arena	589
Mosaic	589
Netscape	589
Konqueror	598
Datentransfer mit FTP	602
anonymous FTP	602
Das Prinzip von FTP	603
<i>ftp</i>	603
<i>xftp</i>	610
Konqueror	614
<i>telnet</i>	615
Programmierung	Kapitel 15
C	618
C++	620
Shell-Programmierung	620



Inhalt



Perl	621
<i>Tcl/Tk</i>	622
Java	623
Andere Sprachen	623
 Shell-Programmierung	 Kapitel 16
Struktur eines Shell-Skripts	627
Starten von Shell-Skripten	628
Ein-/Ausgabefunktionen	630
<i>echo</i>	630
<i>read</i>	632
Argumentwerte	632
Kontrollstrukturen	635
<i>test</i>	635
Verzweigungen	639
<i>if</i> -Anweisung	639
<i>case</i> -Verzweigung	642
Schleifen	645
Die <i>while</i> -Schleife	645
Auswertung von Argumenten als Ausdrücke	647
Die <i>break</i> -Anweisung	648
Die <i>continue</i> -Anweisung	650
Die <i>until</i> -Schleife	650
Die Zählschleife mit <i>for</i>	651
Veränderung von Befehlszeilenargumenten mit <i>shift</i>	654
Dialoge	655
<i>dialog</i>	655
Beispiele	658
 Programmierung in Perl	 Kapitel 17
Aufbau von Perl-Programmen	679





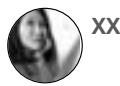
Ein- und Ausgabefunktionen	680	
Ausgabefunktionen	680	
Eingaben von der Standardeingabe lesen	682	
Eingaben als Befehlszeilenparameter lesen	682	
Variablen	683	
Zahlentypen	683	
Zeichenkettentypen	685	
Typkonvertierung	686	
Operatoren	687	
Arrays	690	
Kontrollstrukturen	691	
Verzweigungen	692	
Varianten	695	
Schleifen	695	
Sprunganweisungen	698	
Funktionen	701	
Die Berechnung der Quadratwurzel mit <i>sqrt()</i> und Ausgabe des ganzzahligen Anteils mit <i>int()</i>	702	
Die Funktionen <i>chop()</i> und <i>chomp()</i>	702	
Perl als Hilfswerkzeug für die Manipulation von Datenbankinhalten	704	
Das Datenbanksystem PostgreSQL	705	
Die Beispieldatenbank	705	
Voraussetzungen	710	
Handles	711	
Beispiele	713	
Zusammenfassung	716	
HTML und CGI	Kapitel 18	719
Holz vs. Bytes		720
Rechnerwelten		721
Die HTML-„Programmierung“		722
TAGS		722



Inhalt



Struktur des Dokuments	722
Kommentare	723
Header	723
Body	724
Die Farben der Links	725
Textfarbe	726
Steuerung des Textflusses	726
Zeilenschaltungen und Absätze	728
Bilder	730
URL	731
Hypertext Links	732
Farben	734
Umlaute und Sonderzeichen	735
Zentrieren	737
Der horizontale Strich	737
Listen	738
Tabellen	741
Pfade	744
Kombination der einzelnen Elemente	745
Troubleshooting	745
With a little help from my friends	746
Ein Beispiel	746
Die CGI-Schnittstelle	749
Die Übergabe von Formulardaten an CGI-Programme	750
<i>FORM</i>	751
<i>INPUT</i>	751
<i>SELECT</i>	752
<i>OPTION</i>	752
<i>TEXTAREA</i>	752
Senden von Formulareinträgen	752
Sonderzeichen	761
Perl und CGI-Skripte	763



Inhalt



Die Tastaturbelegung	Anhang A	779
Belegung der deutschen Tastatur ohne Treiber		780
Kernel-Parameter	Anhang B	781
README- und HOWTO-Dateien		782
HTML-Sonderzeichen	Anhang C	799
Die GPL		801
Index		803





Vorwort



Vorwort



Liebe Leserin, lieber Leser,

der stetigen weiteren Entwicklung von Linux Rechnung tragend, halten Sie die nun-mehr siebte überarbeitete Auflage von *LINUX – Das Buch* in Ihren Händen.

Wenn in der Ausgabe 6/1999 der *UNIXOpen* über die zum damaligen Zeitpunkt aktuelle Ausgabe geschrieben wurde: „[...] Wie ein gutes Linux-Buch aussehen muss, hat der Sybex-Verlag mit diesem Werk klar gezeigt. Informativ, nicht langweilig aber immer professionell geschrieben, zum Nachschlagen genauso geeignet wie zum Reinlesen und Ausprobieren [...]“, so hoffen wir, auch mit dieser Überarbeitung wieder den Nerv unseres Publikums getroffen zu haben.

Ach ja, unser „Publikum“ – wie in jedem unserer Bücher möchten wir auch in diesem abermals betonen, dass Sie, liebe Leserin, von uns selbstverständlich genauso wertgeschätzt werden wie diejenigen Vertreter der Spezies Mensch, die wir überwiegend direkt ansprechen: DER Leser, DER Systemadministrator, DER Anwender. Das „Jahrhundertwerk Rechtschreibreform“ wurde leider nicht dazu genutzt, die deutsche Sprache im gleichen Zug um die grammatischen Ergänzung einer geschlechtsneutralen Anredeform zu bereichern, und seien wir ehrlich: Ständig Formulierungen wie „Der/die Anwender/in“ lesen zu müssen, ist doch recht ermüdend. Verzeihen Sie uns also, dass wir „drei Kerle“ überwiegend von unserer geschlechtlichen Position her formulieren ...

Wir hoffen, Ihnen allen ein Arbeitsmittel in die Hände gegeben zu haben, das sich nicht nur technisch und konzeptionell durch seine Überlegenheit gegenüber vielen anderen Systemen auszeichnet, sondern einen weiteren wesentlichen Vorteil ihnen gegenüber hat: Es ist frei!

Diese „Freiheit“ wird oftmals in sehr prägnanter Form wie folgt definiert: „*By this, think free speech rather than free beer* (etwa: *Freiheit im Sinne freier Rede und nicht im Sinne von Freibier*)“. Projekte wie Linux selber und die vielen weiteren freien Komponenten, die in diesem Buch besprochen werden, haben damit immer auch einen (gesellschafts-) politischen Anspruch.

Wir wünschen Ihnen viel Spaß und Erfolg im Umgang mit Ihrem neuen System, welches hoffentlich in der Zukunft auch dasjenige Ihrer Wahl sein wird, so wie es das unsere ist!

F. Gehrke, P. Glinsky, G. Klappheck

Kiel, im Februar 2001





Erstinstallation



Installation von Linux und anderen Betriebssystemen auf der gleichen Festplatte	3
Installation der mit dem Buch mitgelieferten SuSE-Distribution – YaST1 oder YaST2?	5
Start der Installation	8
Erster Start des Systems	46
Die Installation mit YaST2	52



1

Erstinstallation



Auch wenn man nun mit seinem Buch vor dem Rechner sitzt und schnellstmöglich Linux installiert haben möchte, legen wir nahe, zumindest dieses Kapitel erst einmal komplett durchzulesen, um einen Überblick zu erhalten, und dann im zweiten Schritt mithilfe dieses Buches Linux auf die Festplatte zu bannen.

Die Installation von Linux ist kein trivialer Vorgang. Wenige, unüberlegt gedrückte Tasten und die vorher prächtig funktionierende DOS- (OS/2-, Windows-...) Welt ist im Daten-Nirvana verschwunden. „Lasst uns mal eben Linux installieren ...“ sollte niemals der Ausgangspunkt für Neulinge sein. Wir empfehlen an dieser Stelle eindringlich, alle Daten des betreffenden Rechners zu sichern. Die Möglichkeiten sind je nach Systemumgebung vielfältig; wichtig ist jedenfalls, dass die Daten zu Beginn der Installation sicher auf einem Band, auf einem Server, auf einer Wechselplatte, einer selbstgebrannten CD oder auf 348 Disketten (bei 500 MB) vorliegen.

Sehr praktisch für den Betrieb von verschiedenen Betriebssystemen in einem PC sind Wechselrahmen für Festplatten, so dass man für jedes Betriebssystem einen Festplatteneinschub besitzt. Andererseits stehen dann unter Linux die DOS-Daten nicht mehr zur Verfügung.

Wer weiterhin sein bisheriges Betriebssystem neben Linux auf seinem Rechner betreiben will, der sollte sich eine Notfall-Bootdiskette für DOS, Windows oder OS/2 erstellen – falls die nicht sowieso schon griffbereit in der Schublade liegt.

Prinzipiell gibt es mehrere Möglichkeiten von Umgebungen, in die Linux eingefügt werden soll:

- Linux soll das einzige Betriebssystem auf dem Rechner sein. Dies ist die einfachste Möglichkeit. Relevante Daten müssen auf Band gesichert bzw. auf einen anderen Rechner kopiert sein, danach kann frei experimentiert werden, ohne Angst haben zu müssen, dass man sein System mit wichtigen Daten lahm legt. Für den Anfänger empfiehlt sich hier, den alten Pentium I zu reaktivieren, um mit Linux zu experimentieren.
- Linux soll neben DOS, OS/2 oder Windows auf dem Rechner installiert sein, und das Betriebssystem der Wahl soll abwechselnd gestartet werden können. Dies dürfte bei Linux-Neulingen oft der Fall sein, da man sich nicht so ohne weiteres von seinen bisherigen Programmen trennen möchte. Hier empfehlen wir aus Gründen der Datensicherheit und der Einfachheit, Linux eine eigene Festplatte zu spendieren, was bei den heutigen Festplattenpreisen kein großes Problem mehr darstellt. Falls noch eine ungenutzte alte Festplatte herumliegt, so sollte man diese für Linux nutzen. Sie sollte dann allerdings mindestens 500 MByte besitzen.

Zwar kann man Linux auch auf einer 200er Festplatte installieren, aber der Anfänger dürfte in diesem Fall Schwierigkeiten bekommen, zu entscheiden, welche der Programm Pakete nicht notwendigerweise installiert werden müssen. Derzeitig benötigt die dem Buch beiliegende SuSE-Distribution mit ihren Voreinstellungen ca. 800 MByte.





Der dornigere Weg ist es, Linux auf der gleichen Festplatte wie das bisherige System unterzubringen. Der Einwand „Ich hab da doch noch zwei Gigabyte, ich brauche keine neue Platte“ ist natürlich richtig, aber diese Sparmaßnahme ist mit zusätzlichen Arbeiten verbunden.

Installation von Linux und anderen Betriebssystemen auf der gleichen Festplatte

Wer Linux keine eigene Festplatte spendiert, dem sei noch einmal ans Herz gelegt, die Daten seines bisherigen Betriebssystems zu sichern.

Als Nächstes müssen wir den für Linux benötigten Festplattenplatz abschätzen. Nebst den oben erwähnten ca. 450 MByte für die Grundinstallation der SuSE-Distribution benötigt Linux prinzipiell eine *Swap-Partition*. Dies ist die Partition, die benutzt wird, um Daten aus dem Arbeitsspeicher auszulagern, falls hier der Platz knapp wird.

In ihrer Funktion ähnelt sie also der Windows-Auslagerungsdatei, nur das hier der untrügliche Vorteil besteht, dass diese Daten in einer eigenen Partition lagern und mit einfachen Bordmitteln nicht eingesehen werden können. Im Gegensatz dazu kann man sich mit einem beliebigen Hex-Editor die Windows-Auslagerungsdatei ansehen und feststellen, was in den letzten Tagen mit diesem Rechner getrieben wurde.

Die Swap-Partition ist nicht mehr wie früher auf eine Größe von 128 MByte begrenzt; in diesen Zeiten der rapide abstürzenden Festplattenpreise empfehlen wir diese Größe, mindestens jedoch das Doppelte des physikalisch vorhandenen Arbeitsspeichers, wenn dieser kleiner 64 MByte ist.

Außerdem benötigen Programme unter der später zu installierenden grafischen Oberfläche reichlich Speicherplatz: Während der *Acrobat Reader* noch mit ca. 10 MByte auskommt, benötigt z.B. die Vollinstallation des StarOffice-Pakets ca. 160 MByte. Außerdem sollte natürlich noch genügend Platz für die Arbeit unter Linux vorhanden sein: die Briefe, Kalkulationen, Bilder, Videos und Sounds, die man selber erstellt oder weiterbearbeiten möchte, brauchen auch ihren Platz. Zusätzliche Linux Programme und die eigenen Daten müssen aber nicht auf der gleichen Partition wie die Systemdaten liegen.

Falls nun auf der Festplatte kein Platz mehr für eine Linux- und eine Swap-Partition vorhanden ist, muss zunächst einmal Platz dafür geschaffen werden.

Es gibt zum einen die Möglichkeit, die Festplatte mit dem DOS-Dienstprogramm Fdisk komplett neu zu partitionieren. Fdisk zerstört alle Daten, so dass man diese später wieder von seinem Sicherungsmedium zurück auf die Win/DOS-Partition kopieren muss. Dieser unpraktische Weg ist gangbar, aber heutzutage nicht mehr notwendig. Einfacher ist der Weg über das Dienstprogramm FIPS, das auf der SuSE-CD im Verzeichnis *dosutils\fips* liegt.



Erstinstallation



FIPS

Bevor FIPS zum Einsatz kommt, muss die Festplatte defragmentiert werden.

Durch häufiges Löschen und Schreiben von Daten entstehen „Löcher“ auf der Festplatte. Außerdem befinden sich Daten im hinteren Bereich der Festplatte, und diesen wollen wir ja gerade für Linux nutzen. Hier kommt jetzt das Windows-Dienstprogramm DEFrag zum Einsatz. Dieses schiebt alle Daten in den vorderen Bereich der Festplatte und stellt seine Arbeit auch noch anschaulich dar. Der Vorgang dauert länger, und das Zusehen ermüdet, also:

Kaffee kochen!

Ähnliche Arbeit wie DEFrag leisten auch kommerzielle Dienstprogramme, wie z.B. SPEEDISK von Norton.

Tipp



Falls System- oder versteckte (*hidden*) Dateien vorhanden sind, werden diese von den meisten Programmen nicht verschoben, so dass man ihre Attribute vor dem Defragmentieren ändern muss:

```
attrib -S -H C:\DATEI.XYZ /S
```

Die Windows-Swap-Datei ist eine dieser Systemdateien, die nicht ohne weiteres verschoben werden kann. Hier ist die Abhilfe einfacher: In der *Systemsteuerung* muss diese Auslagerungsdatei auf *temporär* gestellt werden. Danach kann die Swap-Datei gelöscht werden: Windows legt sie beim nächsten Start wieder an.

FIPS wird mit einer kompletten Dokumentation geliefert, die an dieser Stelle nicht wiedergegeben werden soll, aber dennoch unbedingt gelesen werden sollte.

Wir verweisen auf die Dateien *Readme.1st* und *Fips.doc* im Verzeichnis *DOSUTILS\FIPS\FIPS20*.

Außerdem sollte die Notfall-Win/DOS-Bootdiskette griffbereit sein, da FIPS die Sekturen, die verändert werden, auf Diskette sichert, so dass die Festplatte zumindest wieder in den alten Zustand versetzt werden kann, wenn der Daten-Gau eingetreten sein sollte. Wenn man die Dokumentation gelesen hat, sind die Menüs von FIPS selbsterklärend und verständlich.

Auch wenn oben erklärt wurde, dass Linux neben der eigentlichen Partition mindestens noch eine Swap-Partition benötigt, so legen Sie mit FIPS nur eine Partition für Linux mit dem errechneten freien Raum an – den Rest erledigen Sie später mit dem SUSE-Installationswerkzeug YaST.

Hinweis



Werden, aus welchen Gründen auch immer, mehrere Betriebssysteme neu auf der Platte installiert, dann sollte Linux konsequenterweise zuletzt installiert werden. Es stellt bezüglich der Lage auf der Platte die geringsten Ansprüche. Außerdem respektiert es die bereits vorhandenen Systeme und überschreibt nicht gnadenlos deren Informationen im Master Boot Record. Als Alternative raten wir Ihnen dringend, während der Linux-Installation eine Bootdiskette anzufertigen, wie weiter unten



Installation der mitgelieferten SuSE-Distribution



beschrieben. Damit lässt sich im Notfall Linux starten und eine Neukonfiguration von *LILO* (dem **L**inux **L**oader, der weiter unten in diesem Kapitel vorgestellt wird) durchführen.

Installation der mit dem Buch mitgelieferten SuSE-Distribution – YaST1 oder YaST2?

Die dem Buch auf der CD beiliegende SuSE-Distribution ist bootfähig. Falls Sie ein bootfähiges CD-ROM-Laufwerk besitzen, so muss sichergestellt sein, dass es auch per BIOS-Voreinstellung angesprochen wird. Da dies von dem jeweiligen BIOS abhängig ist, muss im Handbuch des Rechners nachgesehen werden, wie man ihn überzeugt, dass das erste Bootlaufwerk das CD-ROM-Laufwerk ist. In die BIOS-Einstellungen gelangt man in den meisten Fällen durch Drücken der Taste **[Entf]** (auf Tastaturen mit englischer Beschriftung: „Del“) während des Hochfahrens des Rechners. Unter *Bios-Einstellungen*, *Bios-Setup* oder ähnlichen Menüpunkten befindet sich die Option *Boot Sequence* (oder ähnlich). Hier sollte eine Option gewählt werden, bei der das erste (Boot-) Laufwerk das CD-ROM-Laufwerk ist. Beim Bestätigen findet sich gleich der erste Fallstrick für Anfänger: Da noch kein deutscher Tastaturreiber geladen wurde, muss statt des gewünschten **[Y]** ein **[Z]** eingeben werden – auf der amerikanischen Tastatur sind diese beiden Zeichen im Vergleich zum deutschen Tastaturlayout vertauscht angeordnet.

Wenn Ihr PC dieses Bootverfahren unterstützt, können Sie eine grafisch basierte Installation mit dem am Ende dieses Kapitels beschriebenen YaST2 durchführen.

Wer kein bootfähiges CD-ROM-Laufwerk besitzt, muss sich für die Installation eine Bootdiskette erstellen.

Dies ist auch dann notwendig, wenn Sie die zunächst in diesem Kapitel dargestellte Form der Installation mit YaST1 durchführen möchten (was wir Ihnen nahe legen, da die am Ende dieses Kapitels beschriebene grafisch orientierte Installation noch über einige Einschränkungen bezüglich der Konfigurierbarkeit verfügt und zum anderen in bestimmten Ausnahmesituationen aufgrund einiger ihrer ungewohnten Fehler – neu hochdeutsch: Bugs – nicht unbedingt zum gewünschten Ergebnis führt ...).



Hinweis für die Besitzer einer SuSE-Vollversion: Wenn Sie mit YaST1 von CD aus installieren möchten, booten Sie Ihren PC zunächst mit der zweiten CD!

Erstellen einer Linux-Bootdiskette

Da die Installationsdisketten nicht Bestandteil der mit diesem Buch ausgelieferten Distribution sind, müssen Sie diese selber erzeugen.

Die Schritte, die Sie hierfür durchführen müssen, hängen von dem Betriebssystem ab (entweder Sie haben bereits Zugriff auf ein Linux-System – und zumindest grundlegende Linux-Kenntnisse – oder auf einen DOS/Windows-Rechner), welches Sie zurzeit verwenden oder auf das Sie zum Zweck der Diskettenerstellung Zugriff haben.



Erstinstallation

In beiden Fällen müssen Sie zumindest eine Installationsdiskette mit einem so genannten „Kernel-Image“ erstellen, der die von Ihnen verwendete Hardware unterstützt. Folgende „Kernel-Varianten“ stehen Ihnen hierzu zur Verfügung:

Kernel von SuSE-Linux

Kernel	Einsatzbereich
<i>bootdisk</i>	Der Standard-Kernel. Sie sollten diesen verwenden, wenn Sie ein Standard-(E)IDE-Computer-System (kein SCSI-System) Ihr eigen nennen.
<i>eide</i>	Wie <i>bootdisk</i> , jedoch mit Unterstützung für spezielle (ältere) EIDE-Chipsätze
<i>laptop</i>	Für Laptops mit einem Diskettenlaufwerk am USB-Port
<i>yast2</i>	Bootdisk für den Start mit YaST2

Tabelle 1.1

Mittlerweile haben nicht mehr alle Module auf der Bootdiskette Platz. Daher besteht die Möglichkeit, eine spezielle Moduldiskette zu erzeugen, auf der sich dann (hoffentlich) die von Ihrer speziellen Hardware benötigten Treiber befinden. Sie benötigen diese Moduldiskette beispielsweise dann, wenn Sie mit einem Notebook arbeiten, in dem ein PCMCIA-Netzwerkadapter eingesetzt wird.

Tipp



Erstellen Sie in jedem Fall auch eine Moduldiskette, um für alle Fälle gerüstet zu sein

Wie Sie im weiteren Verlauf dieses Kapitels erfahren werden, führt Sie ein Programm namens Linuxrc durch die Installation; dieses fordert Sie zum Einlegen der Moduldiskette auf, wenn es erforderlich ist.

Erstellen der Installationsdisketten unter DOS/Windows

Um auf einem Win/DOS-Rechner die Boot- und die Moduldiskette zu erzeugen, verfahren Sie wie folgt:

- Starten Sie den Rechner im DOS-Modus oder öffnen Sie die MS-DOS-Eingabeaufforderung.
- Legen Sie die SuSE-CD in Ihr CD-Laufwerk (wir gehen an dieser Stelle davon aus, dass es sich hierbei um Laufwerk E: Ihres PCs handelt; passen Sie Ihre Eingaben ggf. an die von Ihnen verwendete Hardwarekonfiguration an).



Installation der mitgelieferten SuSE-Distribution

- Geben Sie am DOS-Prompt Folgendes ein:

```
e:\dosutils\rawrite\rawrite
```

Geben Sie als Quelldateinamen ein:

```
e:\disks\NAME_DES KERNELS
```

Wenn Sie mit dem Standard-Kernel arbeiten möchten, lautet die entsprechende Eingabe also:

```
e:\disks\bootdisk
```

Legen Sie eine erste Diskette in das Diskettenlaufwerk Ihres PCs und drücken Sie [Enter].

Erzeugen der Moduldiskette unter DOS/Windows

Um die Moduldiskette zu erzeugen, verfahren Sie analog, nur dass Sie hier natürlich als Namen der Quelldatei angeben müssen:

```
e:\disks\modules
```

Erstellen der Installationsdisketten unter Linux

Stellen Sie zunächst sicher, dass ein Verzeichnis `/cdrom` in Ihrem Dateisystem vorhanden ist. Sollte dies nicht der Fall sein, legen Sie als Systemverwalter `root` zunächst dieses Verzeichnis durch Eingabe von

```
mkdir /cdrom
```

an.

Mounten Sie nun die CD-ROM (zum „Mounten“ beachten Sie den Abschnitt „Mounten von Dateisystemen“ in Kapitel 3 *Linux-Dateisystem*), indem Sie beispielsweise eingeben:

```
mount -t iso9660 /dev/cdrom /cdrom
```

Legen Sie eine Diskette in das Diskettenlaufwerk Ihres Rechners und erzeugen Sie nun die Bootdiskette durch eine Eingabe gemäß folgender Syntax:

```
dd if=/cdrom/disks/NAME_DES_KERNELS of=/dev/fd0 bs=1024
```

Wenn Sie mit dem Standard-Kernel arbeiten möchten, lautet die entsprechende Befehlszeile also:

```
dd if=/cdrom/disks/bootdisk of=/dev/fd0 bs=1024
```



Erstinstallation



Erzeugen der Moduldiskette unter Linux

Wenn Sie die Moduldiskette erzeugen möchten (müssen), legen Sie eine neue Diskette in das Diskettenlaufwerk Ihres Rechners und geben Sie Folgendes ein:

```
dd if=/cdrom/disks/modules of=/dev/fd0 bs=1024
```

Vergessen Sie nicht, das CD-ROM-Laufwerk durch folgenden Befehl wieder aus Ihrem Dateisystem „auszuhängen“:

```
umount /cdrom
```

Start der Installation

Falls Sie ältere IDE-Hardware, SCSI-Laufwerke oder exotische Komponenten in Ihrem Rechner verwenden, sollten vorsorglich schon einmal die Handbücher hergeholt werden.

Hinweis



Die im Folgenden vorgestellte Installationsvariante verwendet YaST1. Um sie nachzuvollziehen, müssen Sie die zweite CD der SuSE-Vollversion einlegen. Wenn Sie dazu aufgefordert werden, die CD 2 gegen die CD 1 zu wechseln, tun Sie dies *nicht!* Bestätigen Sie aber den Dialog, der Sie dazu auffordert, und ebenso die folgende Warnmeldung

Nachdem die zuvor beschriebenen Vorbereitungen getroffen worden sind, sollte nun die Installationsdiskette eingelegt werden. Durch Druck auf den Reset-Knopf sollte Linux jetzt mit den SuSE-Installationsroutinen starten.

Die eigentliche Installation

Nach dem Neustart des Rechners wird kurz der in Abbildung 1.1 dargestellte Begrüßungsbildschirm angezeigt. Sie haben an dieser Stelle die Möglichkeit, Linux beim Start so genannte „Kernel-Parameter“ zu übergeben, um damit weitere Anpassungen an die von Ihnen verwendete Hardware vorzunehmen. Im Regelfall sollten Sie die Installation ohne solche zusätzlichen Parameter durchführen können und an dieser Stelle einfach drücken oder warten, bis Linux von alleine hochfährt.

Sollte sich das System jedoch bei diesem Erststart aufhängen, beachten Sie bitte im Kapitel 7 *Systemverwaltung* den Abschnitt *Kernel-Parameter* und den Anhang B zu beachten, wo Sie weitere Informationen zu Kernel-Parametern finden.



Start der Installation



Abb. 1.1

Der Begrüßungsbildschirm von SuSE-Linux

Wir gehen an dieser Stelle jedoch davon aus, dass die von Ihnen verwendete Hardware ohne Probleme unterstützt wird, so dass nach dem Start des Linux-„Rumpfsystems“ das Installations-Tool *Linuxrc* geladen wird.

Wählen Sie aus dem nun folgenden Dialog zunächst die Installationssprache aus. Wie Sie Abbildung 1.2 entnehmen können, steht Ihnen eine ganze Reihe von Sprachen zur Verfügung: vom Deutschen über das Englische, Italienische, Ungarische usw. Bestätigen Sie hier die Voreinstellung *Deutsch* mit .

Abb. 1.2

Die Sprachauswahl



In Abhängigkeit der von Ihnen verwendeten Tastatur können Sie nun das entsprechende Tastaturlayout einstellen:



Erstinstallation



Abb. 1.3

Auswahl des Tastaturlayouts



Das Hauptmenü von Linuxrc bietet die in Abbildung 1.4 dargestellten Konfigurationsmöglichkeiten:

Abb. 1.4

Hauptmenü von Linuxrc



Wenn Sie besondere Hardware, wie SCSI-, PCMCIA- oder Netzwerkadapter verwenden, können Sie diese über den Menüpunkt *Kernel-Module (Hardware-Treiber)* konfigurieren. Wir stellen dies beispielhaft anhand eines Netzwerkadapters dar.

Hinweis



Wenn Sie mit einem Notebook mit PCMCIA-Netzwerkadapter arbeiten, wählen Sie nicht den Menüpunkt *Netzwerktreiber laden*, sondern *Lade PCMCIA Module* und verfahren analog zu den beschriebenen Schritten.



Start der Installation



Führen Sie hierzu den Menübalken mithilfe der Cursortasten auf den Menüpunkt *Kernel-Module* und drücken Sie **[Enter]**.

Wählen Sie nun den Menüpunkt *Netzwerktreiber laden*.



○ Abb. 1.5

Linux-Kernel-Module

Wählen Sie die in Ihrem System vorhandene Netzwerkkarte (oder eine Karte, zu der Ihr Netzwerkadapter kompatibel ist) und drücken Sie **[Enter]**.

Abb. 1.6 ○

Auswahl des Netzwerk-
Kernel-Moduls



Im nun folgenden Dialog haben Sie die Möglichkeit, etwaige Parameter für die von Ihnen gewählte Netzwerkkarte zu übergeben. Dies geschieht nach dem Muster



Erstinstallation



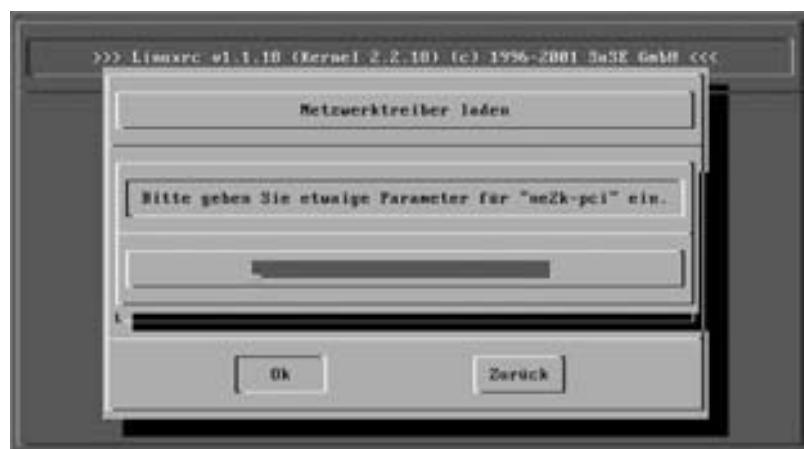
io=0xYYYY, irq=z

mit der für Ihre Karte gültigen IO-Adresse YYYY und dem eingestellten Interrupt Z. Beachten Sie zu diesen Werten ggf. die Dokumentation Ihrer Netzwerkkarte.

Wenn Sie mit einer PCI-Karte arbeiten, sollten Sie dieses Fenster durch Betätigung der Taste übergehen können (dies gilt auch für PCMCIA-Netzwerkadapter).

Abb. 1.7

Eingabe von Kernel-Parametern



Das erfolgreiche Laden Ihres Netzwerkadapters wird mit folgendem Dialog kommentiert:

Abb. 1.8

Das erfolgreiche Laden eines Kernel-Moduls



Kehren Sie in das Hauptmenü zurück und wählen Sie den Menüpunkt *Installation starten*.



Start der Installation



Abb. 1.9

Starten der Installation

Der nun folgende Dialog bietet eine Auswahl verschiedener Installationsquellen:

Abb. 1.10

Auswahl der Installationsquelle



Installation von CD-ROM

Hierbei handelt es sich um die wohl gängigste Art der Installation, die im weiteren Zuge der Darstellungen dieses Buches auch verwendet werden. Für diese Art der Installation benötigen Sie ein lokales CD-ROM-Laufwerk.



Installation via Netzwerk (NFS)

Um diese Art der Installation durchführen zu können, muss die Installations-CD in einem innerhalb des Netzwerks vorhandenen Server eingelegt und freigegeben worden sein. Sie benötigen außerdem Informationen darüber,



Erstinstallation



welche IP-Adresse der Server hat, über welchen Mount-Point das Installationsmedium zur Verfügung steht usw. Beachten Sie hierzu *Kapitel 13 Linux im Netzwerk*.

—○ Installation via Netzwerk (FTP)

Ähnlich einer Installation via NFS, nur unter Verwendung des FTP-Protokolls (siehe hierzu auch die *Kapitel 13 Linux im Netzwerk* und *Kapitel 14 Mit Linux ins Internet*).

—○ Installation von Festplatte

Sie haben die Dateien der Installations-CD auf eine Festplatte kopiert, auf die Sie aus der Installationsroutine heraus Zugriff haben (beispielsweise eine zweite Partition auf der aktuellen Festplatte oder eine weitere Festplatte in Ihrem System). Beachten Sie zu dieser Form der Installation auch *Kapitel 3 Linux-Dateisystem*.

Im Regelfall werden Sie auf die erste Installationsoption zurückgreifen, so dass wir diese hier detaillierter beschreiben möchten. Stellen Sie zunächst sicher, dass die CD (bei einer Vollversion muss es die erste CD sein, die Sie jetzt auch tatsächlich einlegen sollten) im Laufwerk liegt, notfalls werden Sie aufgefordert, dies nachzuholen. Bevor es nun weitergeht, haben Sie die Wahl zwischen textorientierter (YaST1) und grafischer (YaST2) Installation.

Abb. 1.11 ○

Textorientierte oder
grafische Installation



Wir folgen hier nun der textorientierten Variante, die grafische wird weiter unten besprochen.

Im nun folgenden Dialog *Typ der Installation* können Sie grob gesagt zwischen einer Neuinstallation und dem Update einer bereits bestehenden Installation wählen, wie die folgende Abbildung 1.12 zeigt:



Start der Installation



Abb. 1.12

Auswahl des Typs der Installation

Wir gehen an dieser Stelle von einer Neuinstallation aus und ignorieren auch die Möglichkeit, im Experten-Modus zu installieren.

Der nächste Dialog stellt Sie vor die größte Hürde innerhalb des gesamten Installationsvorgangs mit YaST1: Soll die Festplatte partitioniert werden, und wenn, wie? Andererseits steht Ihnen hiermit ein Werkzeug zur Verfügung, die Festplatte wirklich nach Ihren eigenen Wünschen einzurichten.

Im folgenden Dialog haben Sie die Möglichkeit, festzulegen, ob eine Partitionierung erfolgen soll (was bei einer Erstinstallation in jedem Fall erforderlich ist). Wenn Ihr PC über mehr als eine Festplatte verfügt, müssen Sie auch noch angeben, welche dieser Platten partitioniert werden soll.

Hier ein kurzer Überblick, welche Laufwerke Ihres Systems wie angesprochen werden (an dieser Stelle wird der Begriff „Laufwerk“ statt „Festplatte“ verwendet, da es sich bei dem entsprechenden Gerät auch um ein CD-ROM-Laufwerk handeln kann):

- /dev/hda: das primäre Laufwerk am primären IDE-Controller
- /dev/hdb: das sekundäre Laufwerk am primären IDE-Controller
- /dev/hdc: das primäre Laufwerk am sekundären IDE-Controller
- /dev/hdd: das sekundäre Laufwerk am sekundären IDE-Controller

usw.

Wenn Sie mit einem SCSI-basierten System arbeiten, werden die Laufwerke analog als /dev/sda, /dev/sdb usw. angesprochen.

Abb. 1.13

Soll die Festplatte partitioniert werden?



Erstinstallation



Wählen Sie an dieser Stelle in jedem Fall *Partitionieren*, außer Sie haben diese Aufgabe schon anderweitig erledigt. Wir geben allerdings zu bedenken, dass die entsprechenden Hilfsprogramme (wie Fdisk) unter WinDO(w)S bei weitem weniger leisten als die Linux-Tools. Auf die Beschreibung der Einrichtung eines *Logical Volume Managers (LVM)* wird hier verzichtet.

In dieser Phase müssen wir unsere Darstellungen wieder entsprechend der zwei möglichen Fälle aufteilen: Ist bereits ein anderes Betriebssystem auf dem Rechner vorhanden, neben dem Linux installiert werden soll, oder wird Linux das einzige System auf Ihrem PC sein, wobei Sie auf eine manuelle Partitionierung der Festplatte(n) verzichten möchten?

Wenn Letzteres der Fall ist, lesen Sie folgenden Abschnitt, ansonsten können Sie auch direkt zum Abschnitt „*Linux als Installation neben (einem) anderen System(en)*“ vorblättern.

Linux als einziges System

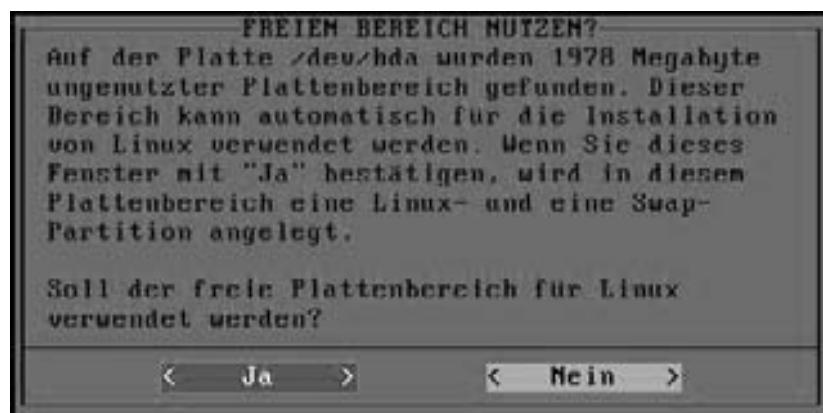
Hinweis

Wenn Sie Linux als einziges System auf Ihrem PC verwenden wollen und über eine freie Platte verfügen, dann können Sie den folgenden Dialog einfach mit bestätigen.

Diese Vorgehensweise ist im Übrigen auch dann möglich, wenn bereits ein Betriebssystem auf der Platte installiert, aber noch genügend Platz für Linux vorhanden ist.

Abb. 1.14

Freien Bereich nutzen



Wundern Sie sich nicht, wenn Sie einen anderen Dialog als den in Abbildung 1.14 dargestellten erhalten. Dies liegt dann daran, dass Ihre Festplatte zurzeit von einem anderen Betriebssystem belegt ist. Dieser Fall wird etwas später in diesem Abschnitt diskutiert.



Start der Installation



Wenn Sie Linux als einziges System auf Ihrem Rechner verwenden, aber dennoch Ihre Festplatte(n) manuell partitionieren möchten, weil Sie beispielsweise mit mehreren Swap-Partitionen arbeiten wollen, beachten Sie hierzu die Darstellungen zur Partitionierung im folgenden Abschnitt *Linux als Installation neben (einem) anderen System(en)*.

Sie möchten an dieser Stelle den gesamten freien Platz verwenden, bestätigen also mit *Ja*, und das Anlegen von Linux- und Swap-Partitionen beginnt. Dies schließt die Vorbereitung der Partitionen zur Datenaufnahme ein. Dieser letzte Vorgang wird auch als „Erzeugen eines Dateisystems“ oder „Formatieren“ bezeichnet.

Je nach Größe und Geschwindigkeit der von Ihnen verwendeten Festplatte kann dieser Vorgang mehr oder weniger Zeit in Anspruch nehmen, was bei der SuSE-Distribution leider nicht durch einen Fortschrittsbalken angezeigt wird, wodurch schon mal etwas Nervosität auftreten lassen kann, ob das System möglicherweise abgestürzt ist. Seien Sie jedoch beruhigt: Ein solcher Fall ist uns bisher noch nicht bekannt geworden – wenn Sie die Installation bis zu dieser Phase überstanden haben, sollte eigentlich nichts mehr schief gehen!

Abbildung 1.15 zeigt die Bildschirmausgabe während des Anlegens des Dateisystems auf einer Partitionen.

Abb. 1.15

Das Einrichten der Partitionen



Findet YaST keinen freien Platz auf der Platte, geht es nach Wahl von *Partitionieren* im Dialog *Festplatten Partitionieren* (vgl. Abbildung 1.13) nicht mit der Abfrage weiter, ob der freie Bereich genutzt werden soll. Sie erhalten stattdessen den Dialog wie in Abbildung 1.16.



Erstinstallation



Abb. 1.16

Gesamte Platte verwenden?



Die Wahl von *Partitionieren* führt auch hier zur manuellen Partitionierung, die wir im nächsten Abschnitt besprochen wird. Gemäß der Annahme, dass Linux das einzige Betriebssystem auf der Platte sein soll, ist hier also *Gesamte Platte* zu wählen.

Dies ist natürlich nicht unproblematisch, da sich ja noch das alte System und die damit erstellten Daten auf der Platte befinden. Abbildung 1.17 zeigt die nun erscheinende Sicherheitsabfrage, die es Ihnen ein letztes Mal ermöglicht, es sich noch einmal anders zu überlegen ...

Abb. 1.17

Noch ist Zeit,
umzukehren ...



In der Situation einer Neuinstallation einer bisher ungenutzten Platte sagen Sie sich: „Augen zu und durch!“ und bestätigen mit *Ja*.

Nach dem Anlegen der Partitionen (und Dateisysteme, die im folgenden Abschnitt *Linux als Installation neben (einem) anderen System(en)* und *Kapitel 3 Linux-Dateisystem* detailliert erläutert werden) gelangen Sie ins Installations-Hauptmenü.

Dabei kann folgender Sonderfall eintreten:

Der zur Verfügung stehende Speicherplatz Ihres Systems sinkt nach der Installation des eigentlichen Linux-Systems, seiner Hilfsprogramme und weiteren Applikationen unter 5%:



Start der Installation



Abb. 1.18

Warnung bei kleinen Platten

Diese Warnung schließt jedoch nicht aus, dass Sie Linux dennoch wie geplant installieren können, da die Vorauswahl der zu installierenden Softwarepakete (beachten Sie hierzu auch den Abschnitt „*Die SuSE-Pakete*“ weiter unten in diesem Kapitel) in der SuSE-Distribution recht großzügig angelegt ist. Wie Sie gegebenenfalls „abgespeckt“ werden kann, erfahren Sie später.

An dieser Stelle werden Sie mit dem unter SuSE-Linux wohl wichtigsten Konfigurationswerkzeug konfrontiert, dem YaST (Yet another Setup Tool).

Dieses unterstützt Sie im weiteren Verlauf der Installation interaktiv, dient aber auch später im „laufenden System“ immer wieder dazu, Konfigurationsänderungen vereinfacht ausführen zu können.

Über Nutzen und Schaden eines zentralen Konfigurations-Tools wie YaST werden mittlerweile fast schon Glaubenskriege zu nennende Diskussionen geführt, an denen wir uns an dieser Stelle nicht beteiligen möchten. Bilden Sie sich nach einer intensiveren Beschäftigung mit SuSE-Linux ein eigenes Urteil.

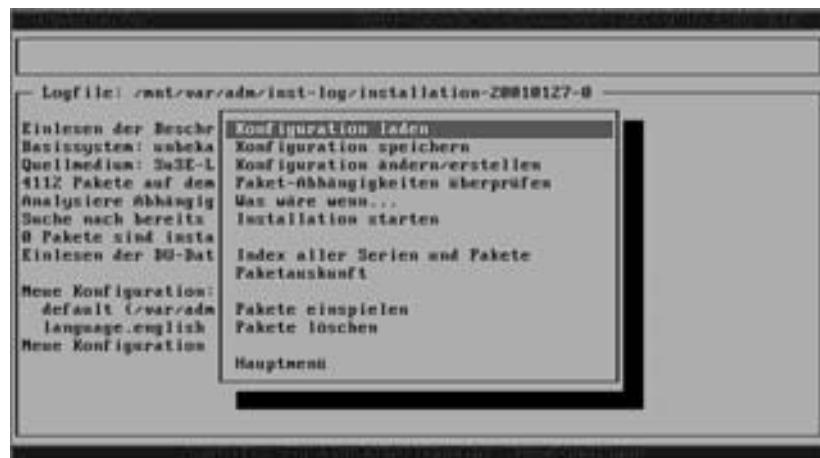


Hinweis

Wählen Sie nun den Menüpunkt *Konfiguration ändern/erstellen*:

Abb. 1.19

Das Installations-Hauptmenü



Erstinstallation



Dieser führt Sie in die so genannte „Serien- und Paketauswahl“, deren Diskussion nach der Darstellung der manuellen Partitionierung folgt, da in dieser Phase der Installation die beiden „Installationszweige“ wieder zusammenwachsen. Wenn Sie die manuelle Partitionierung an dieser Stelle (noch) nicht interessiert, können Sie natürlich folgenden Abschnitt überspringen und gleich beim Abschnitt „Die SuSE-Pakete“ weiterlesen.

Linux als Installation neben (einem) anderen System(en)

Wenn Sie Linux als weiteres Betriebssystem zu einem bereits auf der Festplatte existierenden installieren oder wenn Sie die Partitionierung der in Ihrem System vorhandenen Festplatte manuell vornehmen möchten, gehen Sie wie in diesem Abschnitt beschrieben vor. Wählen Sie zunächst im folgenden Dialog die Option *Partitionieren*:

Abb. 1.20



Auswahl der Partitionierungsoption

Abb. 1.21

Partitionieren der Platte



In diesem Beispiel liegt auf einer 2 GByte großen Festplatte bereits eine DOS-Partition mit 20 MByte vor (ein bisschen Nostalgie darf doch sein, oder ...?). Bezuglich der Partitionierung Ihrer Festplatte haben Sie folgende Überlegungen angestellt:



Start der Installation

- Sie möchten eine 900 MByte große Root-Partition anlegen (beachten Sie hierzu auch *Kapitel 3 Linux-Dateisystem* und hier speziell den Abschnitt „*Ein typischer Linux-Verzeichnisbaum*“).
- Sie möchten eine 64 MByte große so genannte „Swap-Partition“ erstellen – wozu diese dient, werden Sie nach der grundsätzlichen Diskussion der Partitionierung erfahren.
- Der restliche Speicherplatz soll für die „Heimatverzeichnisse“ (beachten Sie auch hierzu *Kapitel 3 Linux-Dateisystem*) der Benutzer zur Verfügung stehen. Ebenfalls zum Thema Heimatverzeichnisse, engl. *home directories*, finden Sie im *Kapitel 13 Linux im Netzwerk* einen Beitrag, wie man diese netzwerkweit zur Verfügung stellen kann.

Die hier geplanten Partitionierungsdaten sind sicherlich für ein vernünftiges Arbeiten mit dem System nicht gerade exemplarisch, reichen aber aus, um die für eine Partitionierung notwendigen Kenntnisse zu vermitteln.

Zum Erstellen einer geeigneten Partitionierung können Sie im nächsten Dialog einige Funktionstasten verwenden, deren Bedeutung in der folgenden Tabelle zusammengefasst sind:

Die für das Editieren der Partitionstabelle zur Verfügung stehenden Funktionstasten

Funktionstaste	Bedeutung
[F1]	Zeigt einen kurzen Hilfetext an.
[F3]	Dient dazu, den Typ einer Partition festzulegen; der Standardtyp ist hier (aus nahe liegenden Gründen ...) <i>Linux Partition</i> . Neben der Festlegung als <i>Linux-Swap-Partition</i> können Sie diese auch als <i>DOS-Partition</i> , als eine Partition, die das neu integrierte <i>Logical Volume Management (LVM-Partition)</i> oder als eine von über 50 weiteren Partitionstypen festlegen. Der <i>Logical Volume Manager (LVM)</i> ermöglicht eine Art virtuelle Betrachtung und Prüfung des vorhandenen Plattenplatzes. Dabei können unterschiedliche Anpassungen und Erweiterungen des Filesystems entsprechend den eigenen Vorstellungen vorgenommen werden.
[F4]	Dient zum Löschen einer bereits vorhandenen Partition.
[F5]	Dient zum Anlegen einer neuen Partition.

Tabelle 1.2

Erstinstallation



Einige Grundlagen zur Partitionierung

Bevor Sie mit der Partitionierung wie beschrieben beginnen, müssen Sie noch einige Fragen zu diesem Thema klären.

Zunächst einmal wird Ihnen, wenn Sie die Taste **F5** zum Anlegen drücken, die Auswahlmöglichkeit zwischen einer so genannten „primären“ und einer so genannten „erweiterten Partition“ gegeben. Was verbirgt sich hinter diesen Begriffen?

Nun, eine Partitionierung ist eine Unterteilung der Festplatte in mehrere physikalisch voneinander abgegrenzte Bereiche, die jeweils als eigenes Gerät angesprochen werden können.

Linux kann jedes Laufwerk in bis zu vier primäre Partitionen oder in bis zu drei primäre Partitionen und eine erweiterte Partition teilen, die dann wiederum in bis zu vier so genannte „logische Partitionen“ unterteilt werden kann.

Eine Partitionierung hat den Vorteil, dass Zugriffe auf Dateien innerhalb des Dateisystems wesentlich schneller erfolgen können, wenn es auf unterschiedliche Partitionen verteilt ist. Außerdem wird durch eine Minimierung der Schreibzugriffe auf einzelne Partitionen eine größere Systemsicherheit erreicht.

Eine besondere Funktion hat die hier gewünschte Swap-Partition: Sie dient als virtueller Speicher zur „Vergrößerung“ des physikalisch vorhandenen Arbeitsspeichers Ihres PCs, in den Daten ausgelagert werden können. Andere Systeme verwenden hierzu eine Auslagerungsdatei, die jedoch wie bereits oben angedeutet mehrere Nachteile auf sich vereint:

- Der Zugriff auf eine Auslagerungsdatei ist wesentlich langsamer als der Zugriff auf eine Auslagerungspartition.
- Eine Auslagerungsdatei kann mit „Bordmitteln“ (einem Hex-Editor) bei vielen Systemen von jedem User eingesehen werden, so dass vom Standpunkt des Datenschutzes her wenig Sicherheit geboten ist.
- Das irrtümliche Löschen einer Auslagerungsdatei ist bei entsprechenden Rechten im System relativ einfach.

Auch Linux erlaubt die Verwendung einer Auslagerungsdatei, eine eigene Partition ist jedoch immer vorzuziehen. Ihre Größe ist nicht länger auf 128 MByte beschränkt und sollte bei Systemen bis zu 64 MByte immer doppelt so groß wie der physikalisch vorhandene Arbeitsspeicher sein.

Bei Bedarf können Sie auch mehrere Swap-Partitionen anlegen.

Dass im vorliegenden Beispiel eine Partitionierung zwingend erfolgen muss, sollte einleuchten: Sie haben bereits eine DOS-Partition auf Ihrer Festplatte, die Sie weiter nutzen möchten, es ist also mindestens das Anlegen einer Linux-Partition erforderlich.

Start der Installation



Abbildung 1.22 zeigt die zuvor geforderten Partitionierungseinstellungen.

Ein Zylinder entspricht in diesem Fall ca. 2 MByte. Beachten Sie, dass Sie beim Festlegen der Partitionsgröße Zylinderangaben oder Kapazitäten (MByte oder KByte) angeben können.



Hinweis

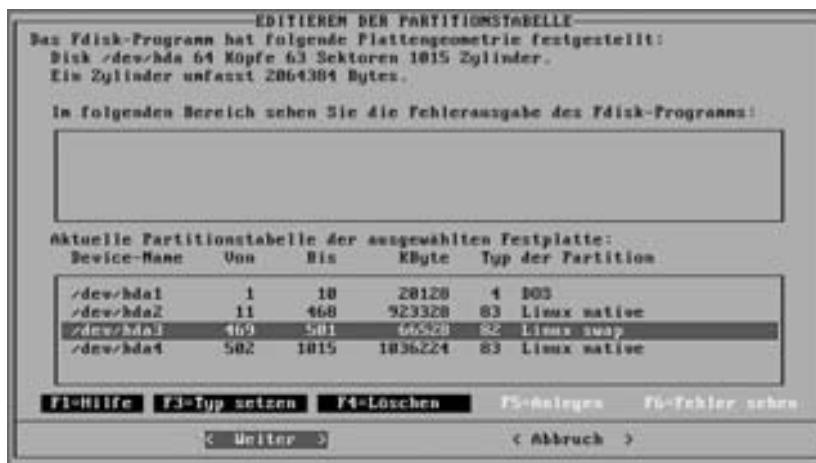


Abb. 1.22

Partitionseinstellungen
der Beispielinstallation

Das Setzen des Partitionstyps „Swap“ der Partition /dev/hda3 erfolgte mithilfe der Funktionstaste [F3] und durch Auswahl des entsprechenden Eintrags.

Wenn Sie diesen Dialog mit *Weiter* verlassen, müssen Sie bestätigen, dass die Partitionstabelle auf Ihre Platte geschrieben werden soll. Nach kurzer Wartezeit erscheint der Dialog aus Abbildung 1.23.

Abb. 1.23

Auswahl der für die Installation zu verwendenden Swap-Partition



Erstinstallation



Sie können in diesem Fenster festlegen, welche der von Ihnen definierten Swap-Partitionen während der Installation verwendet werden soll. In vorliegendem Beispiel ist die Auswahl nicht sehr groß, da nur eine Swap-Partition definiert wurde – bestätigen Sie also wieder mit *Weiter*.

Im nun folgenden Dialog müssen Sie die Struktur des zukünftig in Ihrem System zu verwendenden Dateisystems festlegen (eine detaillierte Betrachtung hierzu finden Sie im Kapitel 3 *Linux-Dateisystem*). Einzelne Partitionen werden hierbei über Verzeichniseinträge (so genannte „Mount-Points“) angesprochen.

Hinweis



Eine Partition muss als Root-Partition (*Mount-Point* */*) festgelegt werden!

Abbildung 1.24 zeigt die drei für vorliegendes Beispiel zur Verfügung stehenden Partitionen (*/dev/hda1*, */dev/hda2* und */dev/hda4*).

Abb. 1.24

Festlegen der Dateisysteme



Wundern Sie sich nicht, dass die Partition */dev/hda3* hier fehlt: Diese dient als Swap-Partition, die Sie natürlich nicht innerhalb Ihres Dateisystems ansprechen können (siehe die oben beschriebenen Vorteile einer Swap-Partition).

Ihre DOS-Partition soll von Linux aus ansprechbar sein: Drücken Sie die Taste **[F4]**, woraufhin folgender Dialog erscheint:

Wählen Sie *Anderer Eintrag* und geben Sie an dieser Stelle beispielsweise ein:

/dos

Anschließend bestätigen Sie mit *Weiter*.



Start der Installation

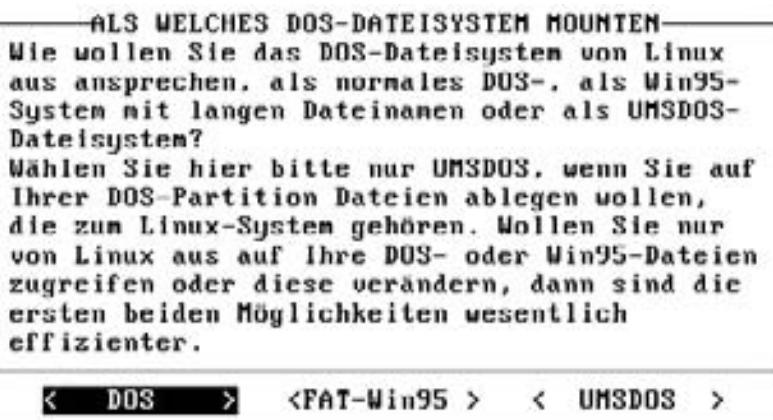


Abb. 1.25

Angabe des *Mount*-Verzeichnisses

Abb. 1.26

Festlegen des Dateisystemtyps



Jetzt legen Sie für die beiden weiteren Partitionen die Mount-Points / und /home fest (diese werden automatisch als Linux-Dateisysteme erkannt):



Erstinstallation

Abb. 1.27

Festlegen der weiteren
Mount-Points



Noch kurz einige Bemerkungen zu den Funktionstasten [F3] und [F5] bis [F7]:

Mit [F3] können Sie den Dateisystemtyp festlegen, entweder das standardmäßige *ext2* oder *ReiserFs* (vergleichen Sie dazu auch *Kapitel 3 Linux-Dateisystem*). Letzteres ist ab SuSE 7.0 verfügbar und erlaubt einen Systemstart ohne die lästige Plattenüberprüfung, auch nach einem Systemcrash. Für die *Root-Partition (Mount-Point /)* kann es jedoch nicht gewählt werden.

Hinweis



YaST2 erlaubte es in der uns vorliegenden Version, die *Root-Partition* als vom Typ *ReiserFs* festzulegen, was aber regelmäßig damit endete, dass diese Installation unvollständig blieb und Linux nicht gestartet werden konnte.

Mit [F5] können Sie zunächst die so genannte „inode-Dichte“ eines Dateisystems festlegen. Hierbei handelt es sich um die Größe der Blöcke, in denen die von Ihnen gespeicherten Dateien abgelegt werden (detailliertere Informationen hierzu finden Sie ebenfalls wieder in *Kapitel 3 Linux-Dateisystem*).

Auch wenn eine von Ihnen abzuspeichernde Datei nur 278 Byte (Zeichen) lang ist, nimmt sie einen gesamten Block (die Standardgröße hierfür ist 4.096) für sich in Anspruch. Im Regelfall sollte man mit der Standardeinstellung arbeiten, wenn allerdings Zweifel bestehen, lesen Sie zunächst das erwähnte *Kapitel 3 Linux-Dateisystem*.

Ohne auf alle Leistungsmerkmale dieses seit SuSE 6.3 als *Expertenmenü* bezeichneten Dialogs eingehen zu wollen, sei jedoch die Eingabezeile *Anzahl Mounts zwischen fsck-Aufrufen* noch kurz erwähnt: Um Inkonsistenzen vorzubeugen, checkt ein UNIX-System in regelmäßigen Abständen das Dateisystem. An dieser Stelle steht Ihnen eine einfache Methode zur Verfügung, die Frequenz dieser Prüfläufe zu beeinflussen, was besonders bei Notebooks, die ja doch relativ häufig herab- und wieder herauffahren werden, sehr angenehm ist, da sie natürlich Zeit in Anspruch nehmen.

Mit [F6] legen Sie die Art der Formatierung des entsprechenden Dateisystems fest. Eine Formatierung mit Prüfung ist nur dann notwendig, wenn die Platte nicht selbstständig defekte Sektoren als solche kennzeichnet. Sollten hier Zweifel bestehen, wählen Sie diese Option, die den Formatierungsvorgang allerdings verlängert.



Start der Installation



Wenn Sie eine Neuinstallation über ein bereits vorhandenes älteres SuSE-Linux vornehmen möchten, müssen Sie die Dateisysteme auf jeden Fall (zumindest) normal formatieren, da YaST sonst Pakete als bereits installiert identifiziert und Sie hierdurch nicht in den Genuss aktueller Pakete kommen würden.



Hinweis

Mit **F7** könnten Sie einen bereits vorhandenen *File System Table* (Tabelle mit Informationen zu den bestehenden Dateisystemen) einlesen, was bei einer Erstinstallation natürlich nicht möglich ist.

Nach Bestätigung mit *Weiter* erscheint eine Sicherheitsabfrage, die Sie ein letztes Mal vor dem Verlust von Daten warnt.

Sie bestätigen mit *Ja* und müssen nun in Abhängigkeit der Größe und Geschwindigkeit Ihrer Festplatte(n) und der Formatierungsart ausharren, bis Sie mit der Installation fortfahren können. Wenn die Dateisysteme erfolgreich angelegt werden konnten, treffen Sie sich wieder mit Ihren Kollegen/Kolleginnen, die die zuvor beschriebenen Schritte automatisiert durchgeführt haben und können mit der Auswahl der zu installierenden Softwarepakete beginnen.

Im Installations-Hauptmenü des YaST (vergleichen Sie dazu Abbildung 1.19) wählen Sie dazu *Konfiguration ändern/erstellen*.

Ach ja: SuSE spricht von Paketen, nicht von Packets – tun wir das also auch.



Hinweis

Die SuSE-Pakete

Wie jeder Linux-Distributor so hat auch die Firma SuSE die mit ihrer Distribution ausgelieferte Software thematisch geordnet.

Eine „Grobunterteilung“ erfolgt in so genannte „Serien“, die sich, wie Abbildung 1.28 zeigt, wie folgt gliedert:

Serien					
▲	Linux Grundsystem				(124.5 M)
ap	Anwendungen, die kein X11 benötigen				(42.4 M)
aplx	Applikäre Office				(8 M)
bco	Extreme Linux (Beowulf)				(8 M)
d	Programmentwicklung (C, C++, Lisp, etc.)				(99.1 M)
doc	Dokumentation				(114.7 M)
e	Entwickl.				(33.2 M)
emu	Emulatoren				(8 M)
fun	Spiele und mehr				(1.9 M)
gnome	GNOME - GNU Network Object Model Environment				(8 M)
gra	Alles rund um Grafik				(67.0 M)
ham	Amateurradio (AX.25, CW, Logbuch, etc.)				(8 M)

cF3>Zoom					
Device-Name	Partition	Gesamt	Belegt	Frei	Mount-Point
/dev/hda3	Linux	1.76 G	1.27 G	585.8 M	27x /
/dev/hda1	Linux	5.5 M	407.0 K	5.1 M	92x /boot

Abb. 1.28

Die Serienauswahl



Erstinstallation



Die wichtigsten Serien werden hier vorgestellt:

Serie a: Linux-Grundsystem

Dieses Paket enthält grundlegende Komponenten zum Betrieb eines Linux-Systems, und die Voreinstellungen sollten von Anfängern nicht ohne Grund verändert werden.

Falls Linux allerdings auf einem Laptop mit PCMCIA-Karten laufen soll, so muss zwingend das Paket *pcmcia* hinzugefügt werden.

Serie ap: Anwendungen, die kein X11 benötigen

Hier finden sich einige Dienst- und Hilfsprogramme, die ohne grafische Oberfläche auskommen. Wen es interessiert: Unter anderem findet sich hier ein Programm zum Beschreiben von CDs.

Serie d: Programmentwicklung

Auch wenn nicht programmiert werden soll, so sollte man die Grundeinstellungen nicht verändern: Diese Programme werden an den verschiedensten Stellen benötigt, auch wenn der Benutzer dies nicht unbedingt mitbekommt. Programmierer haben hier ihr Eldorado: Assembler, Fortran, Java; und natürlich C und C++ .

Serie doc: Dokumentation

Eines der sehr hilfreichen Merkmale von Linux ist seine Dokumentation. Hier findet sich nahezu zu jedem Thema Unterstützung in Form von Texten. Es sei aber nicht verschwiegen, dass noch vieles nur in englischer Sprache vorliegt.

Dem Anfänger ist empfohlen, hier etwas zu stöbern und das zu aktivieren, was er für interessant hält. Aber Vorsicht bei knappem Festplattenspeicher: hier kann viel Platz in Anspruch genommen werden!

Serie e: Emacs

Emacs ist die „Eier legende Wollmilchsau“, startete früher einmal als Editor, kann inzwischen fast alles und erschlägt so den Anfänger. Raus damit, und schon wieder wurden 22 MByte gespart!

Serie emu: Emulatoren

Die dem Buch beiliegende SuSE-Evaluations-Distribution bietet hier den *dosemu* und *ibcs2* an, um DOS-basierte Abwendungen bzw. Anwendungen, die zum Beispiel für SCO UNIX geschrieben wurden, unter Linux ausführen zu können. In der Vollversion finden Sie weitere Emulatoren bis hin zum C64- oder Amiga-Emulator.



Start der Installation



Serie fun: Spiele und mehr

Nun, jeder wie er möchte! Wir empfehlen *xskat*, und dann aber auch später die Option *Bock & Ramsch* (die ganz harten spielen noch mit *Spitze*) zu aktivieren!

Serie gnm: GNOME – GNU Network Object Model Environment

Das GNOME-Projekt setzt (ähnlich wie das KDE-Projekt) dabei an, eine einheitliche Benutzeroberfläche für Linux (und andere UNIX-Systeme) zur Verfügung zu stellen. Die dem Buch beiliegende SuSE-Evaluations-Distribution stellt nur das Basispaket (Bibliotheken) zur Verfügung. In der Vollversion haben Sie Zugriff auf Dutzende Applikationen.

Serie gra: Alles rund um Grafik

Hier finden sich in der Hauptsache grundlegende Grafik-Bibliotheken sowie das Bildbearbeitungsprogramm *gimp* als Voreinstellung. Wer z.B. Scannerunterstützung benötigt, muss diese manuell aktivieren.

Serie kde: K Desktop Environment

KDE ist schlichtweg DIE grafische Oberfläche für Linux. Die Grundpakete sind in der Voreinstellung aktiviert; wer mehr möchte, muss diese selbst aktivieren. Aber Vorsicht: Viele Programme befinden sich noch im Alpha-Teststadium!

Serie n: Netzwerk-Support

Diese Serie enthält wichtige Pakete, um Linux innerhalb eines Netzwerks betreiben zu können. Als Anfänger sollten hier nicht ohne zwingenden Grund Änderungen der Voreinstellungen vorgenommen werden.

Serie pay: Kommerzielle Software

Interessant ist hier der Acrobat Reader, da zusehends mehr Handbücher im Acrobat-eigenen PDF-Format ausgeliefert werden. Auch wenn es für (oder sollte man besser sagen: gegen?) Linux bislang kaum Viren gibt, so ist dennoch auch das hier zur Verfügung stehende Antiviren-Paket vielleicht nicht uninteressant, wenn Sie Ihren Linux-Rechner als Server betreiben, an dem diverse hilflose Win/DOS-Rechner hängen, die geschützt werden müssen.

Serie sec: Sicherheitsrelevante Software

Enthält mehr für den Systemadministrator interessante Pakete, um das hausinterne Intranet beispielsweise mithilfe einer so genannten „Firewall“ gegen Angriffe aus dem Internet abzuschirmen.

Serie snd: Sound und Software

Noch mehr Krach? Nun gut, wer's braucht, findet hier das Richtige.

Erstinstallation



Serie spl: Tools und Datenbanken zur Rechtschreibprüfung

Mehrere Datenbanken in den verschiedensten Sprachen. Wir haben dafür die Sybex-Textredaktion.

Serie tcl: Tcl/Tk/TclX, Tcl-Sprache und T-Toolkit für X

Tcl und seine Verwandten versehen an vielen Stellen des Systems als Basis für einzelne Anwendungsprogramme klaglos ihren Dienst. Nichts deaktivieren, was per Voreinstellung angewählt ist!

Serie tex: TeX/LaTeX und Applikationen

Diese Serie enthält Satzprogramme, die aus dem universitären Bereich entstammen und schon früh Einzug in die UNIX-Welt hielten. Viele Dokumentationen befinden sich in diesen Formaten.

Serie x: Grundpaket Xfree86 3.3: X Window System

Diese Serie stellt die grundlegenden Funktionen der grafischen Benutzeroberfläche zur Verfügung. Befehlszeilen-Fanatiker können sie deaktivieren ... Sie können dann aber auch nicht das weiter oben erwähnten KDE nutzen.

Serie x3d: 3D unter X11 und an der Konsole

Bibliotheken und Treiber, um 3D-Anwendungen unter Linux zu fahren.

Serie xap: X-Anwendungen

Hier finden sich viele Programme für die grafische Benutzeroberfläche, aber Obacht: Vieles ist aktuell durch KDE-Programme bereits ersetzt worden.

Serie xdev: Development unter X11

Für Programmierer, die Software für die grafische Oberfläche entwickeln wollen.

Serie xsrv: Verschiedene X-Server

Die verschiedenen X-Server sind zuständig für die Ausgabe der grafischen Daten, wobei für unterschiedliche Grafikchips unterschiedliche Server zur Verfügung stehen. Mehr dazu im Kapitel 12 X-Konfiguration und KDE über die X-Oberfläche.

Serie xv: Xview (OpenLook, Anwendungen)

Ein weiterer Versuch, Standards für grafische Benutzeroberflächen unter UNIX zu setzen.

Serie xwm: Window Manager und Desktop

Wer statt KDE eine grafische Oberfläche mit einem anderen Look & Feel starten möchte, findet hier Alternativen.



Start der Installation



Programmpakete

Innerhalb der einzelnen Serien finden Sie so genannte „Pakete“, die ein oder mehrere Programme enthalten.

Abb. 1.29



Über Sinn und Unsinn der Vorauswahl durch SuSE mag man sich wieder streiten – folgende „Tastaturtechniken“ erlauben es, diese an Ihre individuellen Bedürfnisse anzupassen:

Mit + oder – werden Pakete an- oder abgewählt, mit kann man sich jeweils eine Ebene weiter im Status des Pakets vortasten. Die verschiedenen Status werden durch folgende Buchstaben vor dem Paket angezeigt:

Statusanzeige in der Paketauswahl



Anzeige Bedeutung

Anzeige	Bedeutung
[]	Das Paket ist nicht installiert und auch nicht selektiert.
[X]	Das Paket ist zur Installation selektiert.
[i]	Das Paket ist installiert.
[R]	Das Paket ist installiert, soll aber durch die Version der CD ersetzt werden.
[D]	Das Paket ist installiert und soll deinstalliert werden.

Tabelle 1.3



Erstinstallation

Hinweis



Um die Paketauswahl einer Serie zu beenden und zur Serienauswahl zurückzukehren, müssen Sie die Taste **F10** drücken (beachten Sie hierzu die Hinweise zur Tastaturbelegung am unteren Bildschirmrand) – das Drücken der Taste **Esc** führt zum Abbruch der aktuellen Paketauswahl und zum Verlust der von Ihnen vorgenommenen Änderungen (seit SuSE 6.3 endlich durch eine entsprechende Fehlermeldung abgefangen ...).

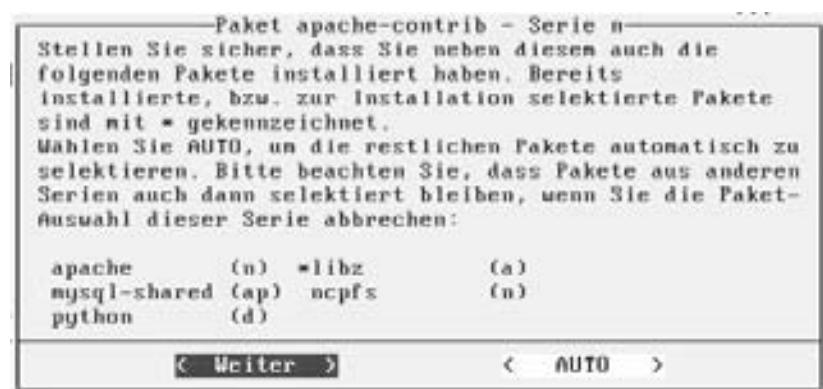
Dieser Abschnitt der Installation ist sicherlich der zeitaufwändigste – zumal, wenn man sich als Neuling das erste Mal durch die riesige Zahl der Auswahlmöglichkeiten durcharbeiten muss (Kurzbeschreibungen der einzelnen Pakete finden Sie für fast alle Pakete im unteren Bildschirmsdrittel, wobei Sie sich häufig weiterführende Informationen durch Drücken der Taste **F2** anzeigen lassen können). Es lohnt sich aber mit Sicherheit, da sich hier die eine oder andere „Programmperle“ versteckt, die nicht in der Vorauswahl berücksichtigt wurde.

Auf der anderen Seite ermöglicht Ihnen das Durchforsten der Vorauswahl natürlich auch, Pakete, die für Sie nicht von Interesse sind, abzuwählen, um hierdurch eine entsprechend schlankere Installation zu erhalten.

In diesem Zusammenhang sei auf ein sehr praktisches und nützliches Leistungsmerkmal von SuSE-Linux aufmerksam gemacht: Oftmals ist es so, dass die Installation eines Pakets das Vorhandensein eines oder mehrerer weiterer Pakete voraussetzt. Solche „Paketabhängigkeiten“ manuell aufzulösen setzt natürlich recht detaillierte Kenntnisse der zur Verfügung stehenden Pakete voraus, die im Regelfall (gerade bei Neueinsteigern natürlich) nicht gegeben sind; YaST weist auf solche Paketabhängigkeiten hin und bietet an, diese automatisch aufzulösen, wie Abbildung 1.30 zeigt:

Abb. 1.30

Automatische
Paketauswahl



Diese Prozedur kann sich wiederholen, da ja durch die automatische Paketauswahl weitere Pakete in die Installation übernommen werden, die ihrerseits zusätzliche Pakete erfordern.



Start der Installation



Mit Hilfe dieser Informationen sollten Sie nun in der Lage sein, die von Ihnen gewünschte Paketauswahl durchzuführen, nach deren Beendigung Sie wieder zum Installations-Hauptmenü von YaST zurückkehren und dort den Menüpunkt *Installation starten* wählen.

In Abhängigkeit der von Ihnen vorgenommenen Paketauswahl, der Geschwindigkeit Ihres CD-ROM-Laufwerks, gegebenenfalls Ihres Netzwerks, falls Sie eine Installation via NFS oder FTP durchführen, und Ihrer Festplatte kann die nun erfolgende Installation der Pakete durchaus bis zu eineinhalb Stunden Zeit in Anspruch nehmen; wenn Sie die mit dem Buch ausgelieferte Evaluations-Distribution verwenden, können Sie diese Zeit vielleicht zu einem Spaziergang mit Ihren Lieben nutzen. Wenn Sie jedoch mit einer SuSE-Vollversion arbeiten, auf der in der aktuellen Version 7.1 immerhin über 1.700 Anwendungen vorhanden sind, und Sie eine größere Zahl davon ausgewählt haben, werden Sie sich (sollten Sie sich schon etwas länger mit PCs befassen) an Disketten-Installationsorgien vergangener Tage erinnert fühlen, nur dass Sie jetzt bis zu siebenmal die CD wechseln müssen.

Die Installation der einzelnen Pakete wird über pseudo-grafische Fortschrittsbalken dokumentiert, wie Abbildung 1.31 zeigt:

Abb. 1.31

Der Fortschritt der Installation

```
Installiere 11: compat - 9.2 M - 326 weitere Pakete folgen...
Logfile: /var/adm/install-log/installation-20010127-0
Postinstall aaa_base...
    Updating etc/rc.conf...
    Updating etc/passwd... unchanged
    Updating etc/group... unchanged
    Updating etc/shadow... modified
    Updating etc/gshadow... modified
aaa_dir #####
aaa_skel #####
ash #####
at #####
Postinstall at...
    Updating etc/rc.conf...
base #####
bash #####
bc #####
bdfflush #####
bztp #####
compat #####
```

Weitere Installationsschritte

Nach Abschluss der Installation der einzelnen Pakete gelangen Sie zunächst zurück ins Installations-Hauptmenü. Wählen Sie hier den Punkt *Hauptmenü*, um die Installation fortzusetzen. Sie werden im folgenden Dialog aufgefordert, den für Ihr System passenden Kernel auszuwählen.



Erstinstallation

Abb. 1.32

Die Auswahl des Kernels



Der Kernel sollte natürlich Ihre Hardware so weit wie möglich unterstützen und mit Bedacht gewählt werden. Beachten Sie zur Auswahl des richtigen Kernels die weiter oben im Abschnitt *Erstellen einer Linux-Bootdiskette* gemachten Ausführungen.

Hinweis



Ist der Kernel, der nun gemäß Ihrer Wahl installiert wird, nicht für Ihre konkrete Hardwaresituation optimiert, etwa weil durch ihn besondere Peripherie nicht angesprochen wird, müssen Sie ihn im Anschluss an die eigentliche Installation neu erstellen (kompilieren). Dies wird ausführlich in *Kapitel 7 Systemverwaltung* beschrieben.

Beachten Sie in diesem Fall, dass die Grundeinstellungen für ein Neukompilieren nicht den Einstellungen entsprechen müssen, die beim Kompilieren des Kernels, den Sie soeben gewählt haben, vorlagen.

Sie sollten gerade in dem Fall, dass Sie den Kernel, den Sie bei der Installation gewählt haben, durch einen neuen ersetzen wollen, die Einstellungen für das Neukompilieren sorgfältig prüfen. Ansonsten kann es Ihnen passieren, dass nun zwar Ihre ISDN-Karte angesprochen wird, aber nicht mehr Ihre zweite Festplatte.

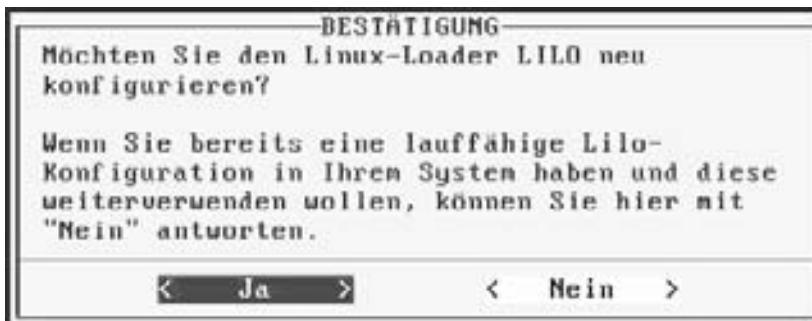
Nun folgt die bereits angesprochene Konfiguration des Linux-Loaders *LILO*.

Konfiguration von LILO

Verzichten kann darauf nur, wer bereits über eine solche Konfiguration verfügt (aber wer tut das schon bei der ersten Installation) oder wer ausschließlich über Diskette starten will. Der Systemstart via Diskette bietet eine gewisse Sicherheit – Linux kann nur noch von denjenigen gestartet werden, die Zugang zum Rechner und einer Bootdiskette haben (aber wer hat die nicht) – ist aber natürlich deutlich langsamer.



Start der Installation



○ Abb. 1.33

Soll LILO konfiguriert werden?

Sie wollen also eine Konfiguration des LILO durchführen und bestätigen die Abfrage, wie Sie in Abbildung 1.33 sehen, mit .

Im folgenden Dialog, den die Abbildung 1.34 zeigt, können Sie nun festlegen,

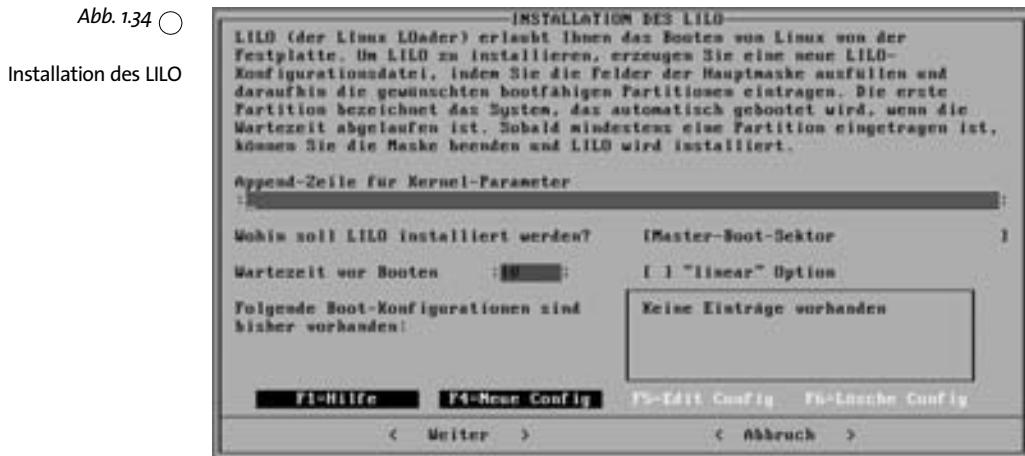
- welche Parameter dem Kernel beim Booten übergeben werden sollen (Einzelheiten zu Bootparametern entnehmen Sie Kapitel 5 *Systemadministration – der Bootvorgang*),
- wo LILO abgelegt wird: im Master Boot Record der Platte, im Bootsektor der aktiven Partition der Platte oder auf Diskette (Einzelheiten werden im Folgenden beschrieben),
- wie lange LILO nach dem Start auf Eingaben warten soll, bevor die Standardkonfiguration zum Booten benutzt wird (beachten Sie die folgenden Ausführungen),
- ob die *linear option* benutzt werden soll oder nicht (ist nur in seltenen Fällen notwendig, und zwar bei Festplattencontrollern, bei denen das BIOS beim Systemstart eine andere Festplattengeometrie erkennt als Linux – in diesem Fall werden dann Adressen im Zusammenhang mit dem Laden des Kernels als logische und nicht als physikalische Adressen gespeichert)
- und welche Konfigurationen zum Booten benutzt werden können (wird im Folgenden näher beschrieben).

Innerhalb des Dialogs können Sie sich am besten mithilfe der Taste bewegen.

LILO ist ein kleines Programm, das nach dem *POST (Power On Self Test)* aktiviert wird. Seine wesentliche Aufgabe besteht darin, das Betriebssystem zu starten und dem Benutzer gegebenenfalls vorab die Möglichkeit zu geben, zwischen verschiedenen Systemen zu wählen. Dies geschieht dann über eine Eingabe an einem Boot-Prompt.



Erstinstallation



Hinweis

Die Wahl ist durch die auf der Platte vorhandenen Betriebssysteme und die Konfiguration des LILO eingeschränkt. Jede wünschenswerte Startkonfiguration ist im Dialog *Installation des LILO* mithilfe der Taste [F4] anzulegen. Eine nachträgliche Änderung der Konfiguration lässt sich mit der Taste [F5] einleiten, eine nicht mehr gewünschte Konfiguration mithilfe von [F6] löschen. In dem nach Betätigung von [F4] aufgerufenen Dialog *LILO Boot Konfiguration*, den die Abbildung 1.35 zeigt, geben Sie Folgendes an:

- Einen Namen für die Konfiguration. Diesen muss man am Boot-Prompt angeben, wenn die entsprechende Konfiguration zum Systemstart benutzt werden soll. Er darf maximal 15 Zeichen lang sein und sollte nur aus Buchstaben (keine Umlaute oder ß), Ziffern und dem Unterstrich _ bestehen.
- Das Betriebssystem, das zum Booten benutzt werden soll (zurzeit werden Linux, Win/DOS und OS/2 angeboten).
- Die Partition, auf der das Betriebssystem installiert ist. Hier lässt sich gegebenenfalls mithilfe der Taste [F3] eine Liste der Partitionen einblenden, auf denen das im Punkt zuvor gewählte Betriebssystem zu finden ist.
- Ob LILO keinen Fehler melden soll ([x] Kernel optional), wenn der angegebene Kernel nicht existiert, was nur für Linux-Konfigurationen möglich und nur in einigen Sonderfällen – Starten eines Test-Kernels – interessant ist. Und schließlich
- den Kernel, der zum Booten benutzt werden soll. Auch hier lässt sich gegebenenfalls mithilfe der Taste [F3] eine Auswahlliste anzeigen. Es besteht also die Möglichkeit, das System mit unterschiedlichen Linux-Kernels zu starten, eine Option, die zum Beispiel im Zusammenhang mit dem Neukompilieren des Kernels interessant ist; bei entsprechender LILO-Konfiguration und Sicherung des alten Kernels lassen sich nun sowohl der neue als auch der alte Kernel zum Booten benutzen. Diese Angabe wird für Nicht-Linux-Konfigurationen ignoriert.



Start der Installation



Abb. 1.35

LILO Boot Konfiguration

Bei mehreren Bootkonfigurationen empfiehlt es sich natürlich, die Zeit, die dem Benutzer zur Wahl der gewünschten Konfiguration zur Verfügung steht, nicht zu knapp anzusetzen. Sie ist im Dialog *Installation des LILO* unter *Wartezeit vor Booten in Sekunden* anzugeben. Eine sinnvolle Einstellung sollte zwischen 10 und 30 Sekunden liegen. Ist nach dieser Zeit keine Eingabe erfolgt, wird die Standardkonfiguration gestartet, also die erste der aufgeführten Konfigurationen.

Mit Hilfe der **[↑]**-Taste kann der Benutzer am Boot-Prompt die verbleibende Wartezeit wieder auf den eingestellten Wert hochsetzen. Mit Hilfe der Taste **[Esc]** kann er sich die Namen der Bootkonfigurationen anzeigen lassen. Diese sollten also schon aussagekräftig sein (z.B. *linux*, *os-halbe*, *dos*, *windos*).



Hinweis

Bleibt noch zu klären, wo LILO installiert werden soll. Dies geschieht normalerweise im *Master Boot Record*. Das hat den Vorteil, dass Linux auch dann ohne Diskette gestartet werden kann, wenn alle Linux-Partitionen auf der zweiten (oder dritten) Festplatte liegen. Außerdem muss keine Partition aktiviert werden, in der dann LILO zu installieren wäre.

Die Installation in einem Bootsektor setzt voraus, dass dies der Bootsektor einer primären Linux-Partition oder der erweiterten Partition ist. Damit ist LILO natürlich sicher untergebracht (denken Sie daran, dass Win/DOS-Systeme bei der Installation keine Rücksicht auf vorhandene Eintragungen im Master Boot Record nehmen). Andererseits muss darauf geachtet werden, dass die fragliche Partition auch aktiv ist, damit LILO überhaupt beim Systemstart aktiviert werden kann.

Wer nun weder Bootsektoren noch den Master Boot Record verändern will, der kann LILO notfalls auf der Diskette installieren. Diese wird dann natürlich für den Bootvorgang gebraucht, was ihn unnötig verlangsamt. Im Regelfall empfiehlt sich also die Installation im Master Boot Record.

Nachdem Sie alle Angaben sorgfältig gemacht haben, bestätigen Sie den Dialog *Installation des LILO* mit *Weiter*. Auch die folgende Sicherheitsabfrage können Sie getrost mit *Weiter* bestätigen.

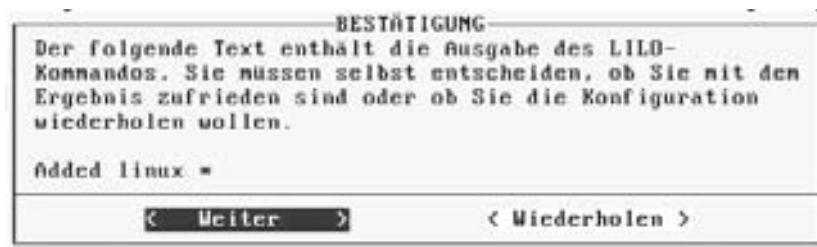


Erstinstallation



Abb. 1.36

Sicherheitsabfrage zur Installation des LILO



Wahl der Zeitzone

Nach Bestätigung der Sicherheitsabfrage gelangen Sie zur Einrichtung der Zeitzone. Üblicherweise wählt man für unsere Breiten *Europe/Berlin*, wie in Abbildung 1.37 zu sehen ist ...

Abb. 1.37

Wahl der Zeitzone



Zusätzlich will YaST nun noch wissen, ob die Systemzeit Ihres Rechners nach der Greenwich Mean Time oder nach lokaler Zeit (für Deutschland MEZ) eingestellt ist.



Start der Installation

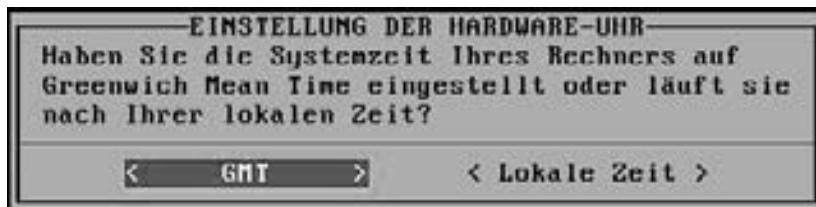


Abb. 1.38

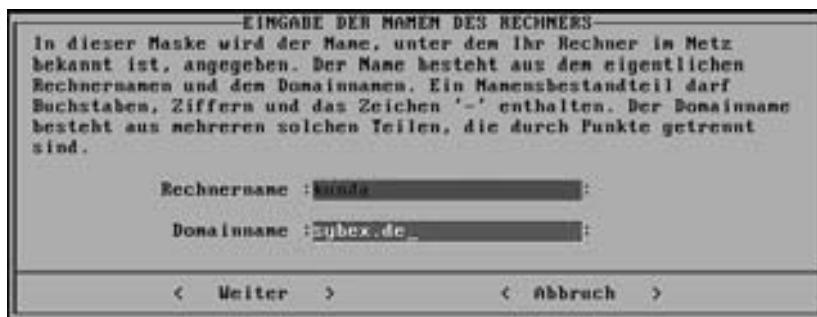
Einstellung der Hardware-Uhr

Netzwerkkonfiguration

Damit verlassen Sie die Zeitzonenkonfiguration und werden als Nächstes aufgefordert, dem Rechner einen Namen zu geben.

Abb. 1.39

Vergabe des Rechner-namens



Der Name des Rechners ist frei wählbar; es sollten aber keine Sonderzeichen wie Umlaute oder das (aussterbende) β vorkommen; auch das Leerzeichen ist nicht gestattet. Und nun kommen wir zu einer kleinen Merkwürdigkeit der SuSE-Installation: Dort kann man sich wohl nicht vorstellen, dass es noch Rechner gibt, die sich nicht in einem Netzwerk befinden. Daraus ergibt sich die Tatsache, dass es zwingend notwendig ist, einen Domainnamen zu vergeben.

Genaueres über Domains wird im *Kapitel 13 Linux im Netzwerk* erklärt. Jedenfalls kann dieser Eintrag später jederzeit geändert werden. Falls man eine eigene Domain besitzt, sollte diese hier eingeben werden. Andernfalls muss man aber das SuSE-Installationsprogramm zufrieden stellen; geben Sie z.B. *sybex.de* ein. Ein beherztes bringt Sie zum nächsten Dialog.

Falls der Rechner nicht in einem echten Netzwerk betrieben werden soll, genügt es, das interne Simulieren eines Netzes per *Loopback* einzuschalten.



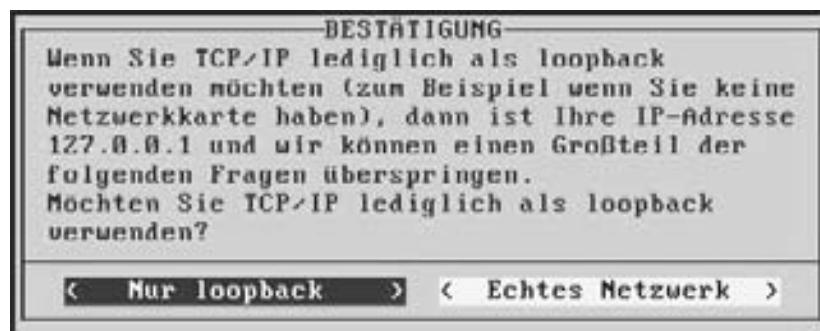
Erstinstallation



Wird nur das Loopback aktiviert, dann erspart man sich einige der folgenden Fragen betreffs der Netzwerkkonfiguration. Die Netzwerkfunktionen können jederzeit zu einem späteren Zeitpunkt per YaST aktiviert oder geändert werden.

Abb. 1.40

Installation des Netzwerks



Falls soeben die Option *Echtes Netzwerk* gewählt wurde, werden nun die Details des Netzwerks abgefragt. Dies ist, wie der folgende Dialog zeigt, automatisierbar, setzt aber einen DHCP-Server voraus. Da wir außerdem anhand der weiteren Dialoge zur Netzwerkkonfiguration einige Einzelheiten erläutern wollen, bestätigen Sie diesen Dialog mit *Nein*.

Abb. 1.41

IP-Konfiguration automatisieren



Es folgt die Eingabe der *IP-Adressen*. Wenn man ein lokales Netzwerk besitzt, sollte man eine Netzwerknummer aus dem Bereich der so genannten „privaten Nummern“, wie z.B. 192.168.o.x, wählen, wobei x hier einen Wert zwischen 1 und 255 annehmen kann.



Start der Installation

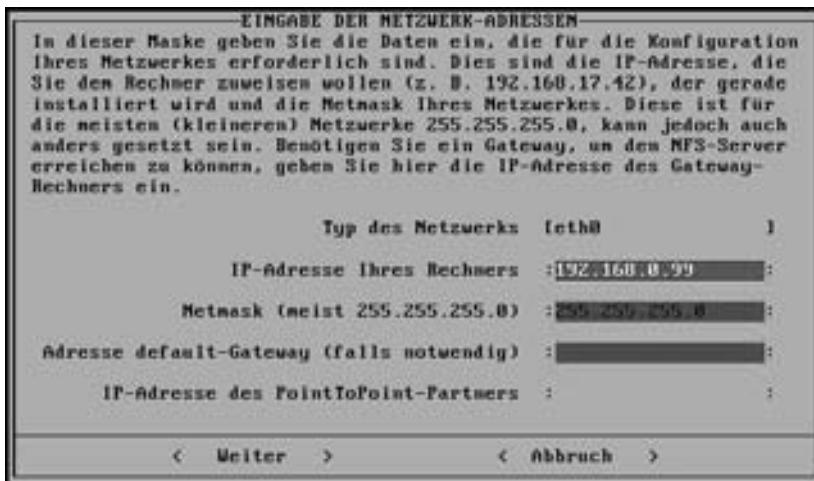


Abb. 1.42

Eingabe der Netzwerk-adressen

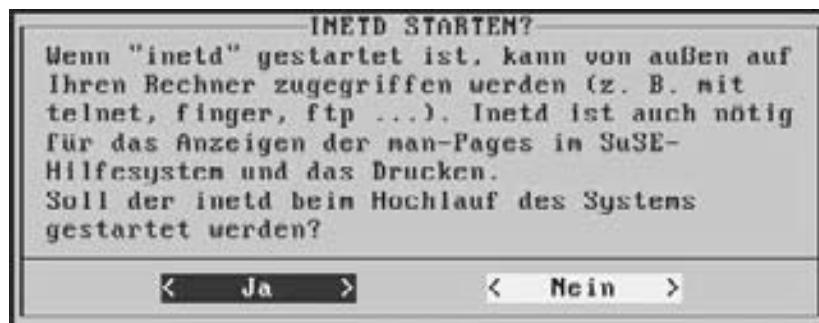
Die Netzmaske ist in diesem Fall 255.255.255.0. Genaueres findet sich wiederum in *Kapitel 13 Linux im Netzwerk*. Falls man sich mit seinem Rechner in einem Firmennetzwerk befindet, sollte man sich die Adresse, die Netzmaske und – falls benötigt – die Adresse des Standard-Gateways vom Systemadministrator besorgen.

Das vierte Editierfeld *IP-Adresse des PointToPoint-Partners* wird nur dann benötigt, wenn man eine Wählverbindung ins Internet aufbauen möchte (beachten Sie auch hierzu *Kapitel 13 Linux im Netzwerk*). Dazu muss die Adresse des Providers eingegeben werden, die dieser Ihnen normalerweise mit seinen Unterlagen zur Verfügung stellt.

Als Nächstes wird gefragt, ob der *inetd* gestartet werden soll. Wie SuSE hier schon warnt, stellt dieses eine potenzielle Sicherheitslücke dar, auf der anderen Seite ist *inetd* notwendig, um sich die Seiten des Hilfesystems anzeigen zu lassen sowie um drucken zu können. Der Anfänger sollte hier mit *Ja* antworten.

Abb. 1.43

Starten des inedt



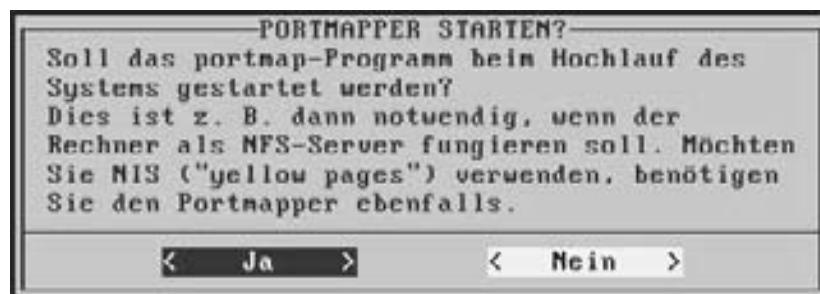
Erstinstallation



Nun wird der Dienst des *Portmappers* angeboten, der benötigt wird, damit der Rechner als Netzwerk-Fileserver dienen kann. Dies kann sich als praktisch erweisen, wenn man z.B. einen weiteren Rechner mit Linux ausstatten möchte und dieser kein CD-ROM-Laufwerk besitzt, man also eine Installation über das Netzwerk vornehmen muss. Hier kann dann der erste Linux-Rechner als Server für den zweiten dienen.

Abb. 1.44

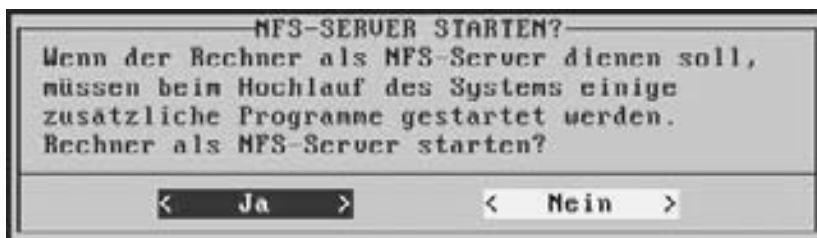
Starten des Portmap-
pers



Wer sich entschlossen hat, seinen Rechner als Fileserver – und sei es auch nur temporär – arbeiten zu lassen, der muss im folgenden Dialog mit *Ja* antworten. Beachten Sie zu NFS-Diensten auch *Kapitel 13 Linux im Netzwerk*.

Abb. 1.45

Starten des NFS-Servers



Falls Sie sich an Newsgroups beteiligen, so zeigt YaST hier an, wie die bei Postings an solche Gruppen von Ihrem System automatisch versendete Absenderzeile aussieht. Diese wird aus den bisherigen Einträgen des Rechner- und des Domänenamens gebildet:

Falls Sie an ein Netzwerk angeschlossen sind oder einen Wählzugang zum Internet besitzen, können die symbolischen Namen von Rechnern (z.B. www.sybex.de), mit denen Sie kommunizieren, durch einen Nameserver in ihre jeweiligen nummerischen Adressen (im Falle des zuvor erwähnten SYBEX-WWW-Servers: 149.221.231.163) aufgelöst werden – ein solcher Dienst sollte von Ihnen genutzt werden, da Sie sich die symbolischen sicherlich besser als die nummerischen Adressen merken können.



Start der Installation

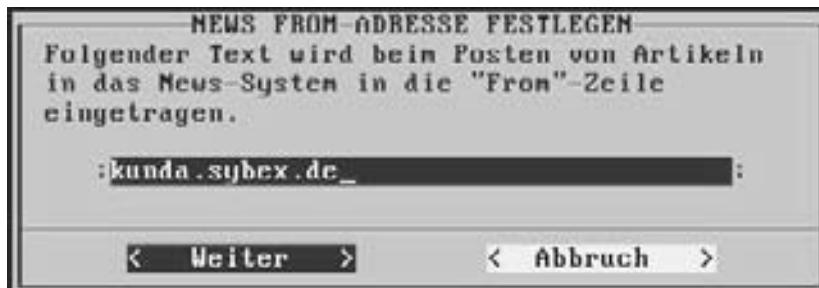


Abb. 1.46

Festlegen der „News from“-Zeile

Abb. 1.47

Soll auf einen Nameserver zugegriffen werden?



Bestätigen Sie also mit *Ja*.

Verfügen Sie über einen eigenen Internetzugang, liefert der Provider die Adressen des Nameservers oder der Nameserver. Diese können Sie im folgenden Dialog durch Leerzeichen getrennt eingeben.

In der zweiten Eingabezeile können Sie eine Liste von Domains angeben, deren Namen durch den Namensdienst nicht aufgelöst zu werden brauchen; dies ist dann sinnvoll, wenn sich Ihr Rechner in einem lokalen Netzwerk befindet, dessen Rechnernamen bekannt sind.

Abb. 1.48

Konfiguration der Nameservers



Gegebenenfalls müssen Sie noch die Netzwerkkarte angeben. Dies geschieht im folgenden Dialog.

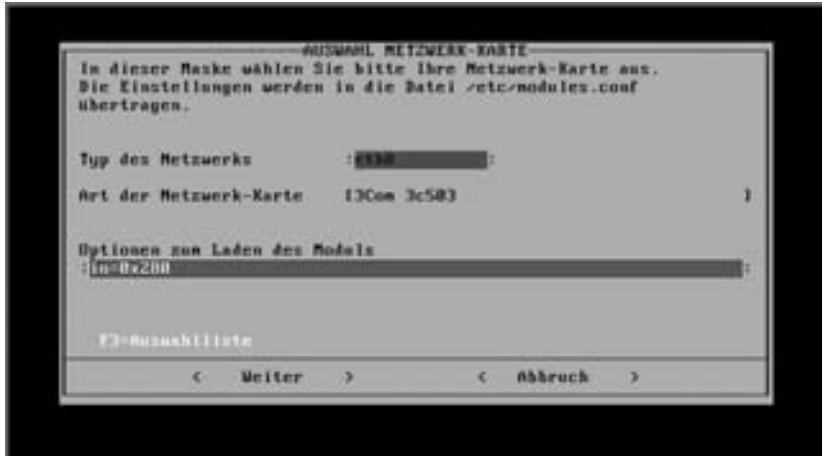


Erstinstallation



Abb. 1.49

Auswahl der Netzwerk-karte

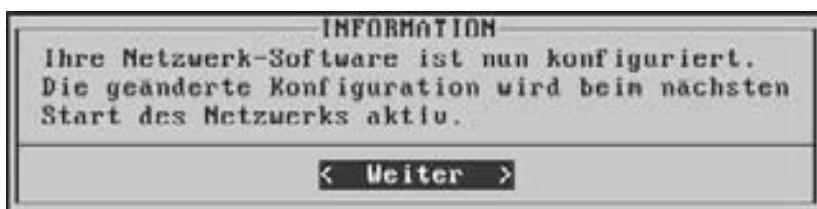


Sie legen den Typ (z.B. *eth0* für Ethernet), die Kartenbezeichnung und mögliche Optionen fest.(z.B. Portadressen – lesen Sie notfalls in der Beschreibung der Karte nach). Haben Sie die Netzwerkkarte auch unter Windows eingebunden, können Sie die erforderlichen Werte auch dort in Erfahrung bringen. Bezuglich Netzwerktyp und Kartenbezeichnung lassen Sie sich helfen und öffnen mit [F3] die Auswahllisten. Bestätigen Sie schließlich Ihre Wahl.

YaST bestätigt nun die erfolgreiche Installation des Netzwerks und weist darauf hin, dass alle Einstellungen erst nach dem nächsten Start des Rechners aktiviert werden. Sämtliche Einstellungen können später mit YaST umkonfiguriert werden, falls notwendig.

Abb. 1.50

Ende der Netzwerk-konfiguration



Da Sie Ihren unserer Erstinstallation nicht so weit gehen wollen, Ihren Rechner als Mailserver zu konfigurieren, wählen Sie im nun folgenden Dialog die Option *Keine Modifikation von /etc/sendmail.cf*.



Start der Installation

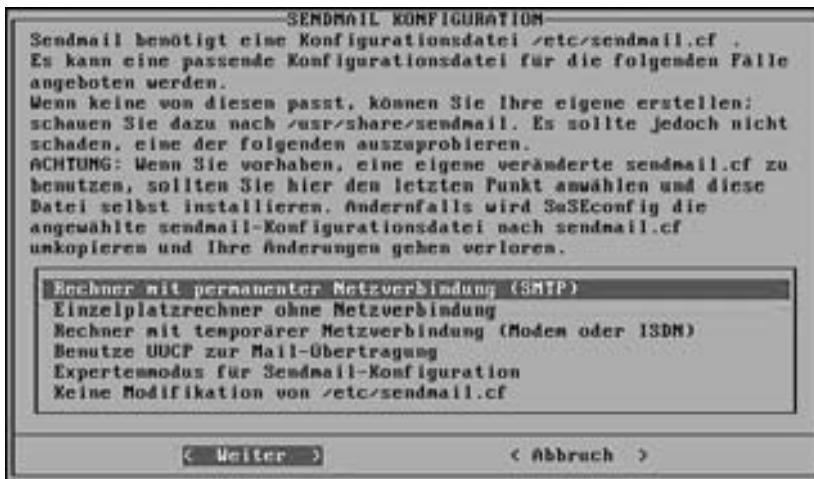


Abb. 1.51

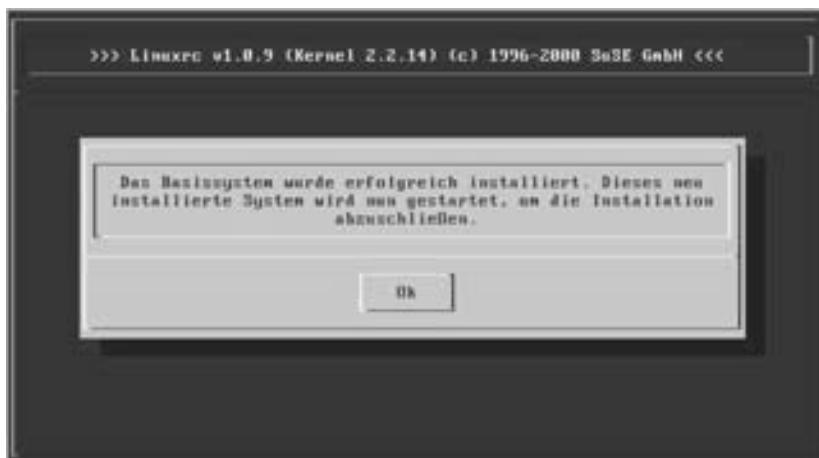
Sendmail-Konfiguration

Nach Abschluss dieser Einstellungen beginnt YaST nun mit der Konfiguration, deren Fortschritt in einem Dialog angezeigt wird. Der geneigte Anfänger kann nun betrachten, wie die bisher getätigten Eingaben jetzt von YaST abgearbeitet werden; der Profi wendet sich an dieser Stelle genussvoll dem Koffein oder dem Nikotin zu.

Der erfolgreiche Abschluss der Installation/Konfiguration wird wie im folgenden Dialog gezeigt kommentiert.

Abb. 1.52

Erfolgreicher Abschluss



Erstinstallation



Erster Start des Systems

Der Erfolg Ihrer Arbeit resultiert in einem Neustart des Systems: Die Initialisierungsskripte von Linux werden nun Schritt für Schritt abgearbeitet und im Erfolgsfall mit einem *done* kommentiert. Der Rechner fährt Linux mit all seinen Diensten hoch.

Abb. 1.53

Neustart des Systems
mit Abarbeitung der Initialisierungsskripte

```
INIT: version 2.70 booting
Running /etc/init.d/boot
Mounting /proc device
Boot logging started at Sat Jan 27 12:00:00 2001
Activating swap-devices in /etc/fstab...
Checking file systems...
Parallelizing fsck version 1.19a (13-Jul-2000)
/dev/hda3: clean, 84273/477120 files, 271327/476704 blocks
/dev/hda1: clean, 25/3000 files, 2657/6016 blocks
Setting up the CMOS clock
Setting up /lib/modules/2.2.10
Mounting local file systems...
proc on /proc type proc (rw)
devpts on /dev/pts type devpts (rw,mode=0620,gid=5)
/dev/hda1 on /boot type ext2 (rw)
Starting ldmconfig (setting up /etc/ld.so.cache)
Setting up timezone data
Setting up loopback device
Setting up hostname
Configuring serial ports...
Configuring serial ports
Running /etc/init.d/boot.local
Creating /var/log/boot.msg
```

Benutzereinrichtung

Danach heißt Sie SuSE willkommen und konfrontiert einen mit den Sicherheitsmechanismen eines UNIX-Systems. Im Gegensatz zu Win3.x/95/98, das nach dem Start für jedermann wie ein offenes Buch vor einem liegt, gibt es in einem Mehrbenutzersystem wie Linux klar abgegrenzte Bereiche: Der Systemverwalter heißt immer *root* und darf jegliche Systemeinstellungen eines Rechners ändern. Die normalen Benutzer werden von diesen Möglichkeiten ferngehalten und können so – auch nicht versehentlich – keine Einstellungen ändern oder Systemdateien löschen. Genaueres über Mehrbenutzerumgebungen finden Sie im *Kapitel 6 Userverwaltung*.

Auf jeden Fall muss der Systemverwalter ein Passwort erhalten; SuSE empfiehlt hier eine Mischung aus Buchstaben und Zahlen zwischen fünf und acht Zeichen. Außerdem wird zwischen Groß- und Kleinschreibung unterschieden.

Dieses Passwort sollte man sich sorgfältigst merken!

Hinweis



Ein Punkt in einem Passwort verursacht an dieser Stelle Probleme derart, dass man beim Neustart nicht mehr in sein System gelangt. Unbedingt vermeiden!



Erster Start des Systems

1

Die Eingabe des Passworts erfolgt ohne Bildschirmecho, d.h., man muss es „blind“ eintippen. Um zu gewährleisten, dass das eingegebene Passwort auch tatsächlich das gewünschte ist (schließlich kann man sich ja bei der Blindeingabe vertippen), muss es ein zweites Mal eingegeben werden.

Abb. 1.54

Eingabe des *root*-Passworts

```
deproto:~# /etc/init.d/deproto start
[ ok ] Starting deproto: deproto (rw,mode=0600,gid=5).
[ ok ] Starting devfsd on /boot type ext2 (rw)
[ ok ] Starting liloconfig (setting up /etc/lilo.conf.cache)
[ ok ] Setting up timezone data
[ ok ] Setting up loopback device
[ ok ] Setting up hostname
[ ok ] Configuring serial ports...
[ ok ] Configuring serial ports
[ ok ] Running /etc/rc.d/rc.boot.local
[ ok ] Creating /var/empty/boot.msg
[ ok ] Enabling syn flood protection
[ ok ] Disabling IP forwarding
[ ok ] Starting syslog services
[ ok ]
```

Da man seine normalen Aufgaben am Rechner nicht als Systemverwalter erledigen sollte, ist es sinnvoll, einen einfachen Benutzer ohne besondere Privilegien anzulegen. Dies bot YaST in den Versionen vor 1.09 (SuSE 7.1) in einem Dialog unmittelbar nach Vergabe des Superuser-Passworts an.

Es ist empfehlenswert, unmittelbar nach Abschluss der Installation als User **root** einzuloggen, YaST aufzurufen und im Menü **Systemadministration** unter **Benutzerverwaltung** genau dies nachzuholen. Sie erhalten folgende Maske:

Abb. 1.55

Anlegen eines Beispielbenutzers



Erstinstallation



Die Eingabefelder haben folgende Bedeutungen:

- *Login-Name*: Der Name, unter dem der Benutzer sich beim System anmeldet.
- *Passwort*: Das gewünschte Passwort des Benutzers; für unerwünschte Späher wird statt der eingegebenen Zeichen jeweils ein Stern ausgegeben.
- *Passwort wiederholen*: Da man wegen der ausgegebenen Sterne keine Kontrollmöglichkeit besitzt, muss zur Überprüfung das Passwort wiederholt werden.
- *Ausführliche Beschreibung des Benutzers*: In größeren Systemen für den Systemverwalter zur Unterscheidung der Benutzer hilfreich – z.B. bei der Frage: Welche der vielen Marias (s.u.) im System braucht nun mehr Festplattenplatz?

Das Programm fährt nun mit der Installation des Modems fort. Dies sollten Sie jetzt allerdings noch nicht tun – es kann jederzeit später nachgeholt werden. Außerdem entwickelt sich ISDN zumindest in Deutschland zum allgemeinen Standard – mehr dazu später in diesem Buch im *Kapitel 13 Linux im Netzwerk*. Wählen Sie hier also die Option *Nein*.

Konfiguration der Maus

Auch wenn Sie kein Mausfanatiker sind, ist die Aktivierung der Maus hilfreich, so dass Sie folgenden Dialog bejahen sollten.

Es folgt nun die Auswahl des Maustyps, wobei es in fast allen Fällen nur um zwei verschiedene Typen geht: Durchgesetzt haben sich zum einen die serielle Microsoft-Maus (von Hardware verstehen die etwas!) und die PS/2-Maus.

Faustformel für Anfänger: runder Stecker = PS/2, 9-poliger Stecker in zwei Reihen = serielle Maus. Freunde von exotischen Mäusen haben aber auch eine größere Auswahl zur Verfügung.

Sollten Sie eine serielle Maus gewählt haben, wird die Schnittstelle abgefragt, an der die Maus am Rechner angeschlossen ist. Für DOS-Umsteiger sind zusätzlich die dort üblichen Bezeichnungen aufgeführt.



Erster Start des Systems

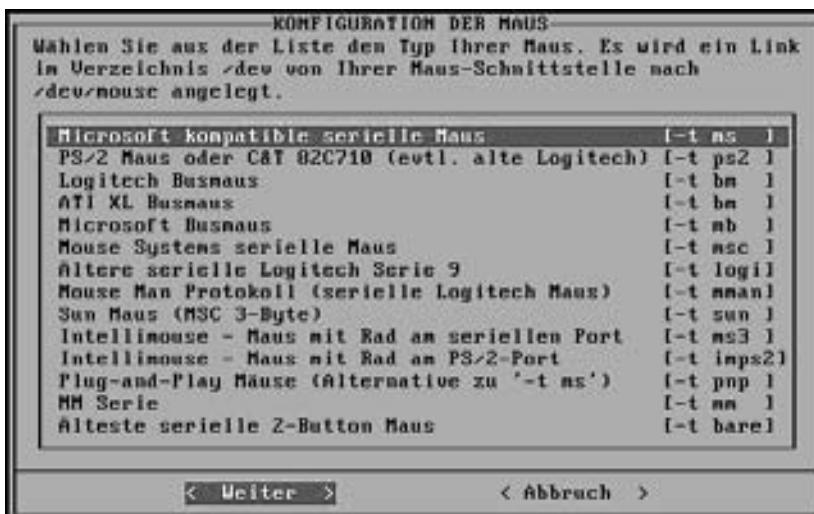
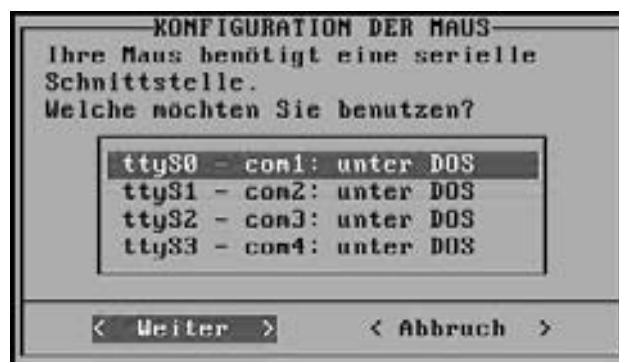


Abb. 1.56

Auswahl einer Maus

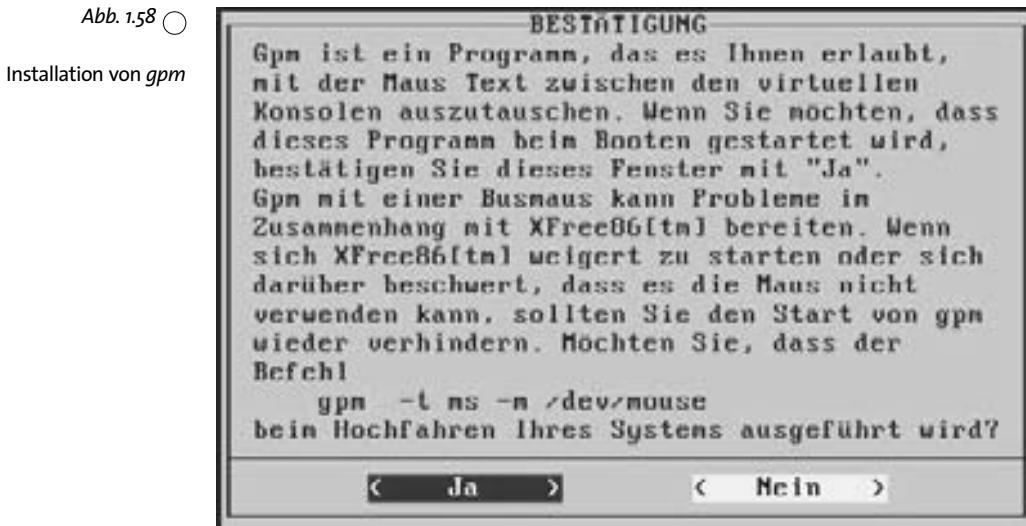
Abb. 1.57
Auswahl der seriellen Schnittstelle



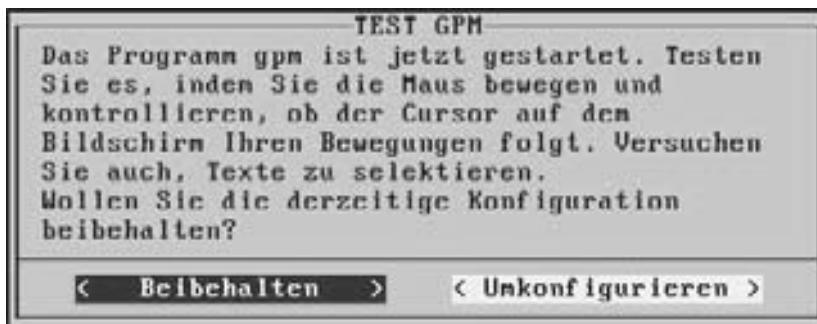
Das Programm *gpm* ermöglicht die Verwendung der Maus in der Befehlszeile-Umgebung (Kopieren und Einfügen von Textelementen). Wir empfehlen die Installation von *gpm* ausdrücklich, so dass Sie den folgenden Dialog mit *Ja* beantworten sollten:



Erstinstallation



Im nächsten Schritt kann man interaktiv testen, ob die Mauseinstellungen richtig sind.



Falls der Cursor wider Erwarten nicht analog den Mausbewegungen folgen sollte, kann hier jederzeit mit *Umkonfigurieren* zurückgegangen werden. Die Dialoge *Auswahl des Maustyps* und *Auswahl der Schnittstelle* (s.o.) stehen so lange zur Verfügung, bis man bei Erfolg hier *beibehalten* antwortet. Mit gpm kann man nun jederzeit Text aus den verschiedensten Anwendungen oder auch Text von der Konsole hin- und herkopian.

Zu guter Letzt meldet YaST seine erfolgreiche Arbeit – der Installationsvorgang ist beendet.



Erster Start des Systems



Um nun das Linux-System in Betrieb nehmen zu können, wird es neu gestartet (nicht jedoch der Rechner). Sie werden aufgefordert, als Benutzer *root* einzuloggen, was sicher dann Sinn macht, wenn es noch weitere Installationsarbeiten zu erledigen gilt. Ansonsten verweisen wir hier noch einmal darauf, nur dann als *root* einzuloggen, wenn es zwingend erforderlich ist. Ansonsten erinnern Sie sich daran, dass Sie während der Installation einen Standard-User eingerichtet haben. Nutzen Sie dessen Account (Benutzername und Passwort), wenn Sie das System nun erforschen wollen. Notfalls richten Sie über YaST einen weiteren „Normal“-User ein.

```
22:55:38:ca:7e:20:95:f1:48:6f:1e:a8:90:4e:00:03 root@honda
Starting SSH daemon                                done
Starting syslog services                           done
Re-Starting ipd                                    done
Creating /etc/hosts/ghost ...
USB determination method: ehhwispcl
No USB controller found
No USB Host found
Starting service at daemons:
Loading kmpnp quertz/de-latt-ad.map.gz          skipped
Loading compose table winkeys shiftctrl lattbl.add   done
Starting Name Service Cache Daemon                done
Starting CRON daemon                             done
Starting console mouse support (qpm)            done
Starting personal-firewall (final) init action    unused
Master Resource Control: runlevel 3 has been      reached
Please enter "root" to login as user root...
Welcome to SuSE Linux 7.1 (i386) - Kernel 2.2.10 (ttgt).
[unode login] ...
```

Abb. 1.60

Nach dem Neustart des Systems



Abb. 1.61

Grafisches Login



Erstinstallation



Diese Art der Systemkonfiguration werden Sie jedoch erst im *Kapitel 12 X-Konfiguration und KDE* kennen lernen.

Dem Anfänger bzw. Umsteiger sei an dieser Stelle zunächst die Lektüre von *Kapitel 2 Bootstrapping the new user* empfohlen.

Ein Hinweis noch zum Ende dieses Abschnitts: Wer am Anfang der Installation seine Bootsequenz auf das CD-ROM-Laufwerk umgestellt hat, sollte nicht vergessen, diese wieder auf die Festplatte C: zurückstellen oder bei einer Bootsequenz CD-ROM/C zumindest die CD-ROM aus dem Laufwerk entnehmen.

Die Installation mit YaST2

Wie bereits eingangs erwähnt, empfehlen wir Ihnen die Installation mit YaST1. Auch wenn Puristen hier vielleicht die Nase rümpfen: Mit Sicherheit wäre eine vollständige grafische Installation wünschenswert, doch sollte Sie diese nicht allzu sehr in Ihrer Flexibilität bezüglich der Konfigurationsmöglichkeiten einschränken oder gar in bestimmten (zugegeben: besonderen) Situationen die Installation und Konfiguration nahezu unmöglich machen.

Ein Problem von YaST2 ist es, Standardkomponenten zu erkennen. Die Installation auf Laptops bereitete uns auch immer wieder Schwierigkeiten (PCMCIA-Karten waren dem YaST2 bisweilen suspekt).

Die angebotene Möglichkeit, YaST2 vorzeitig zu beenden, funktioniert nicht immer. Eigentlich soll in dieser Situation ein Wechsel in die Installation mit YaST1 erfolgen, wir mussten jedoch mehrfach feststellen, dass sich YaST2 dann schlüssig aufhängte.

Der Umgang mit dem Reiser Dateisystem ist nicht einwandfrei. Will man die Bootpartition als *ReiserFs* definieren, so erlaubt YaST2 dies, installiert dann aber (wegen Platzmangels!) die Softwarepakete nicht.

Nochmals: Sicherlich wäre es wünschenswert, eine rein grafisch basierte Installation zu Wege zu bringen, doch sind die Autoren der Meinung, dass der YaST1 diesem Installationswerkzeug vorzuziehen ist. Gerade der Anfänger wird bei Problemen vor nahezu unlösbarer Aufgaben gestellt. Wir wollen jedoch insofern fair gegenüber *SuSE* sein, indem wir darauf hinweisen, dass auch andere Linux-Distributoren (etwa *RedHat*) zwar bemüht sind, die Fenster bereits zu Beginn der Installation aufzustößen, sich diesbezüglich aber auch nicht gerade mit Lorbeeren schmücken dürfen.

Positiv anzumerken ist, dass sich die grafische Installationsroutine seit ihrem ersten Erscheinen deutlich verbessert hat. Angenehm ist es zum Beispiel, dass man bei einer Minimal-Installation nicht mehr die CD wechseln muss.

Wer ein bisher noch „unberührtes“ Computersystem sein Eigen nennt, mag sich dennoch an dieser Form der Installation versuchen, die aber aus den dargestellten Gründen nur im Schnelldurchgang „abgehakt“ sei.



Die Installation mit YaST2

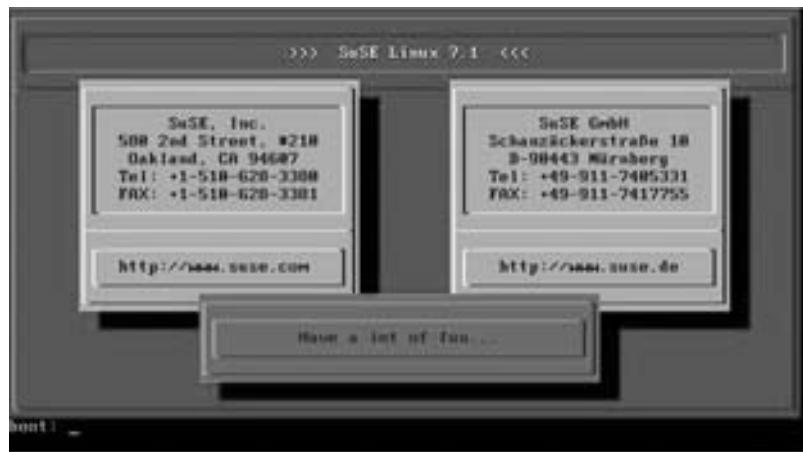


Start mit der dem Buch beiliegenden CD

Wenn Ihr PC die am Anfang des Kapitels dargestellten Voraussetzungen für den Start von CD-ROM erfüllt, legen Sie diese in das CD-ROM Laufwerk Ihres Computers und schalten Sie ihn ein. Der „Begrüßungsbildschirm“ unterscheidet sich nicht von dem, der sich beim Start von Diskette (oder beim Start mit CD2, wenn Sie mit der Vollversion von SUSE-Linux arbeiten) präsentiert, wie Abbildung 1.62 zeigt.

Abb. 1.62

Der Begrüßungsbildschirm von SuSE-Linux



Nach dem Booten des Linux-Systems erfolgt zunächst eine Systemüberprüfung.

Abb. 1.63

Die Systemüberprüfung



Erstinstallation



Schließlich erhalten Sie die Möglichkeit, die Standardsprache festzulegen. Diese wird auch für die Installation genutzt.



Abb. 1.64

Wahl der Sprache

Sollte Ihre Grafikkarte nicht mit dem SVGA-X-Server zusammenarbeiten, dann kann YaST2 nicht in den Grafikmodus wechseln. In diesem Fall sehen Sie statt der hier dargestellten Oberfläche schlichtere Klötzchengrafik, der Ablauf bleibt aber derselbe.

Im nächsten Dialog können Sie mithilfe der Maus die von Ihnen gewünschten Einstellungen bezüglich des zu verwendenden Tastaturlayouts und der zu verwendenden Zeitzone festlegen. (Hinweis an alle Besitzer eines Acer Travelmate 7100: Ihre PS/2-Maus können Sie leider vergessen – kramen Sie Ihre alte serielle Maus wieder hervor, wenn Sie mit der Installation Erfolg haben wollen ...) In einem kleinen Eingabefeld können Sie die Einstellung der Tastatur überprüfen. Machen Sie Gebrauch davon.

Im nun folgenden Dialog haben Sie die Wahl zwischen *Neuinstallation* und *Update*, wobei hier nur die erste Variante vorgestellt wird, da ja von einer Erstinstallation ausgegangen wird!



Die Installation mit YaST2



Abb. 1.65

Einstellen des Tastaturlayouts und der Zeitzone

Nun müssen Sie festlegen, wie Ihre Festplatte genutzt werden soll. In dieser heiklen Installationsfrage hat SuSE YaST2 seit der Distribution 6.4 erheblich überarbeitet. Zunächst kann man wählen, ob man SuSE die Einteilung der Platte überlässt (Option 1) oder, was dem erfahreneren Anwender empfohlen sei, man kann manuell partitionieren.



Abb. 1.66

Vorbereiten der Festplatte



Erstinstallation



Hier wurde dieser zweite Weg gewählt, der es Ihnen nun ermöglicht, Partitionen anzulegen, vorhandene zu löschen oder gerade angelegte zu bearbeiten.



Abb. 1.67

Manuelles
Partitionieren

Falls Sie vorhandene Partitionen löschen wollen/müssen, dann tun Sie dies mit Bedacht. In diesem Beispiel liegt eine jungfräuliche Platte vor, die Sie nun ähnlich dem obigen Beispiel (Partitionieren mit YaST1) einrichten wollen. Der vorhandene Platz von knapp 2GByte wird wie folgt eingeteilt:

- Eine root-Partition mit ca. 1,5 GByte. (Bei weniger als 1 GByte wird die folgende Wahl der Softwareausstattung problematisch. Die Standardausstattung benötigt in etwa diesen Platz!)
- Eine Partition /home für die Heimatbereiche der Benutzer (ca. 390 MByte).
- Eine Swap-Partition von ca. 68 MByte Umfang.
- Über die Hintergründe einer solchen Wahl haben Sie bereits weiter oben (Installation mit YaST1) gelesen.

Der bereits vorhandene Eintrag (*Gerät /dev/hda*) bezeichnet die zur Verfügung stehende Festplatte, die nun eingeteilt werden soll. Dazu markieren Sie diesen Eintrag und klicken dann auf die Schaltfläche *Anlegen*. Die Abfrage, ob Sie eine *Primäre* oder *Erweiterte Partition* anlegen wollen, sollten Sie, sofern das möglich ist, mit *Primäre Partition* beantworten. Es öffnet sich der Dialog, den Sie in der folgenden Abbildung 1.68 sehen.



Die Installation mit YaST2



Abb. 1.68

Anlegen einer primären Partition. Im Bezug auf den Endzylinder beachten Sie unbedingt den Text.

Hier sollten Sie nun in jedem Fall daran denken, dass Sie

- eine Swap-Partition benötigen und
- für eine der Partitionen der *Mount-Point* / zu wählen ist (zu den Hintergründen verweisen wir erneut auf die Installation mit YaST1).

Leider schlägt YaST2 an dieser Stelle für die erste anzulegende Partition (wie auch für alle folgenden) indirekt vor, den gesamten noch zur Verfügung stehenden Platz zu benutzen. Der Endzylinder (Option *Ende*) ist so voreingestellt. Da es nicht jedem gegeben ist, die Größe einer Platte in Zylindern zu bemessen, fällt dieser Umstand vielen sicher nicht sofort auf.

Nutzen Sie die Möglichkeit, die Größe in MB oder GB festzulegen und denken Sie daran für die weiteren Partitionen Platz aufzusparen. Seien Sie beruhigt: Über die Schaltfläche *Bearbeiten* im Dialog *Festplatte vorbereiten: Experten-Modus* (Abb. 1.67) können Sie nachträglich Änderungen vornehmen.

Neben der Größe sind noch festzulegen:

- Der *Typ* der Partition (*Ext2* ist gegenwärtig der Standard für Datenbereiche). Für Partitionen außer / ist aber auch *ReiserFs* zu empfehlen, was Festplattenaktionen protokolliert und damit lästige Plattenprüfungen nach einem Systemcrash vermeidet.



Erstinstallation



- Ob sie formatiert werden soll (natürlich).
- Der *Mount-Point* (/ für die *root*-Partition – ein Muss! –, /*home* für die Heimatbereiche, /*opt* und/oder /*usr* für Softwarepakete, um nur die wichtigsten Möglichkeiten zu nennen).
- Der *Startzyylinder* (ergibt sich in der Regel automatisch aus der bisherigen Partitionierung).
- Den *Endzyylinder* (glücklicherweise kann man hier durch Angabe von Kapazitäten – MByte oder GByte – einer umständlichen Berechnung aus dem Weg gehen).

Haben Sie keine Möglichkeit mehr, primäre Partitionen anzulegen, dann nutzen Sie die Variante der erweiterten Partition.

In einem Dialog legen Sie zunächst deren Gesamtgröße fest. Danach lassen sich auf der erweiterten Partition *logische Partitionen* einrichten. Dies geschieht analog zur Einrichtung der primären Partitionen auf der Platte (Selektieren Sie zuvor im Dialog *Festplatte vorbereiten: Expertenmodus* den Eintrag der erweiterten Partition).

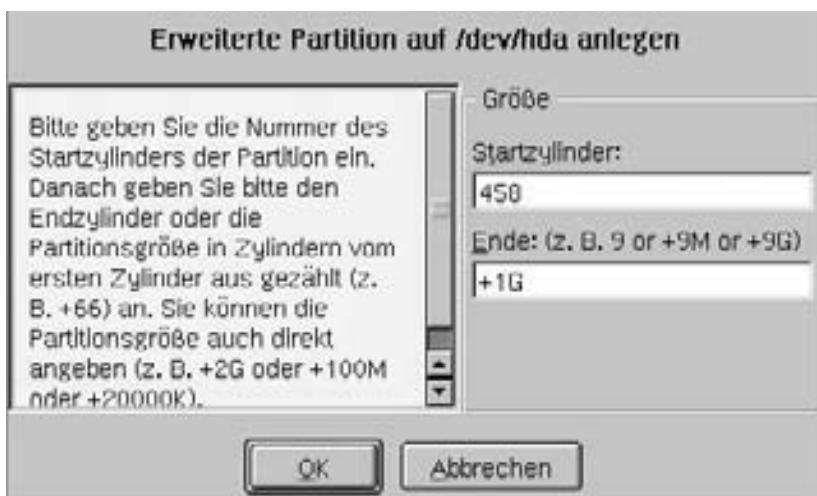


Abb. 1.69

Anlegen einer erweiterten Partition

Alle angelegten Bereiche werden mit den gewählten Einstellungen im Dialog *Festplatte vorbereiten: Expertenmodus* angezeigt. Dies sieht in vorliegendem Beispiel dann wie folgt aus:

Zum Abschluss dieses wichtigen (und bei vorhandenen Partitionen nicht ungefährlichen) Installationsschrittes klicken Sie wie nun schon gewohnt auf *Weiter*. Als Nächstes legen Sie dann fest, welche Software installiert werden soll.



Die Installation mit YaST2



Abb. 1.70

Die Einteilung der Festplatte

Sie können zwischen unterschiedlich umfangreichen Softwareausstattungen Ihres zukünftigen Linux-Systems (die Sie natürlich später beliebig ergänzen bzw. wieder entfernen können) wählen. Wir empfehlen an dieser Stelle zumindest die Installation des *Standard-Systems*, ergänzt durch Multimedia-, Netzwerk- und kommerzielle Komponenten.



Abb. 1.71

Grundlegende Software-Auswahl



Erstinstallation

Darüber hinaus bietet dieser Dialog bei der Distribution SuSE 7.1 die Möglichkeit, die Kernel-Version zu bestimmen:

- den erprobten Kernel 2.2.18 (voreingestellt) oder
- den aktuellsten Kernel 2.4 mit erweiterten Features.

Über diese grundlegende Auswahl hinaus können Sie ähnlich wie bei YaST1 auch einzelne Pakete für die Installation selektieren. Klicken Sie dazu zunächst die Schaltfläche *Erweiterte Auswahl* an. Sie öffnet einen Dialog (vgl. Abb. 1.72), in dem Sie bestimmte Software-Gruppen auswählen können.



Abb. 1.72

Erweiterte Software-Auswahl

Die Auswahl der Gruppe *Spiele* erfordert ca. 400 MByte zusätzlichen Platz!

Wenn Sie sich schon so weit vorgewagt haben, empfiehlt es sich, noch einen Blick in die Paketauswahl zu werfen, was durch Anklicken der Schaltfläche *Einzelne Anwendungen auswählen* ermöglicht wird.

Wir raten Ihnen, im folgenden Dialog (vgl. Abb. 1.73) in der Serie *pay* nachzuschauen und dort gegebenenfalls den *Acrobat Reader* auszuwählen, der die Anzeige von Dateien im so genannten „PDF-Format“ erlaubt, das ein Quasi-Standard für Onlinedokumente (Handbücher, Datenblätter usw.) ist. Wenn Sie ein wenig Zeit haben, können Sie an dieser Stelle schon einmal einen Blick auf die Vielfalt der mitgelieferten Software werfen. Wir weisen aber mit Nachdruck darauf hin, dass diese auch später noch problemlos nachinstalliert werden kann. Nutzen Sie die Zeit nach der Erstinstallation erst einmal zum Kennenlernen des Systems, bevor Sie sich an die Erforschung einzelner Pakete machen. Das können Sie im Anschluss daran mit dem nötigen Basiswissen wesentlich gezielter tun.



Die Installation mit YaST2

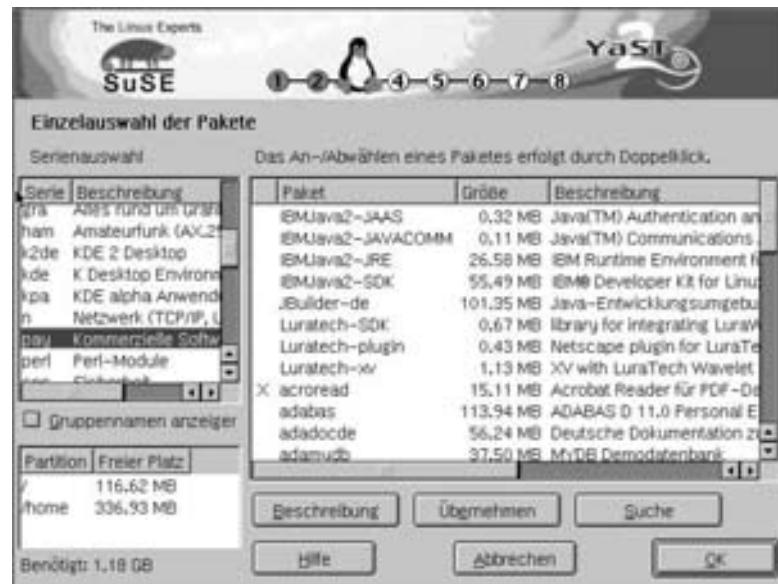


Abb. 1.73

Einzelauswahl der Pakete

Nach der Auswahl der Software gelangen Sie an einen neuralgischen Punkt: die Auswahl des Installationsortes für den *Linux Loader*, mit dessen Hilfe Sie beim Starten des Rechners zwischen verschiedenen Systemkonfigurationen wählen können. Empfehlenswert ist im Allgemeinen der *Master Boot Record (MBR)*.



Abb. 1.74

Installation des LILO



Erstinstallation



Für die bisher unbenutzte Platte schlägt YaST2 genau das vor (nennt hier den Master Boot Record allerdings Startsektor). Wer will, kann an dieser Stelle über die Schaltfläche *Andere Konfiguration* in eine weiteren Dialog gelangen, der andere Möglichkeiten der Installation erlaubt. Abgesehen von der Variante *Bootsdiskette erstellen* (dann müssen Sie Ihr System eben immer mithilfe dieser Diskette starten) kann eine Änderung der Standardeinstellung nur dann empfohlen werden, wenn Sie sich mit den anderen Möglichkeiten entsprechend gut auskennen.

Zu *Bootparametern* vergleichen Sie die Ausführungen in *Kapitel 7 Systemverwaltung*, zur *linear option* die Beschreibung der LILO-Konfiguration mit YaST1 weiter oben in diesem Kapitel.



Abb. 1.75

Konfiguration des LILO

Definieren Sie im folgenden Dialog einen „Normal-User“, unter dessen Benutzerkennung Sie am System arbeiten sollten, wenn Sie keine systemadministratiorischen Aufgaben erfüllen wollen.

Es kann gar nicht oft genug darauf hingewiesen werden, dass Sie sich als User *root* (als der Systemadministrator) nur dann einloggen sollten, wenn Sie auch wirklich nur ihm zugedachte Aufgaben erledigen wollen. Auch wir, die Autoren, haben in unseren Anfängen als Benutzer eines UNIX-Systems mit vollen Zugriffsrechten unbedacht Aktionen vorgenommen, die wir hinterher bereut haben ...

Beachten Sie bei der Bearbeitung dieses Dialogs, dass Sie die Eingabe in ein einzelnes Feld (z.B. die Eingabe des Passworts) nicht mit der Taste bestätigen. Klicken Sie ins nächste Feld oder benutzen Sie die -Taste.



Die Installation mit YaST2

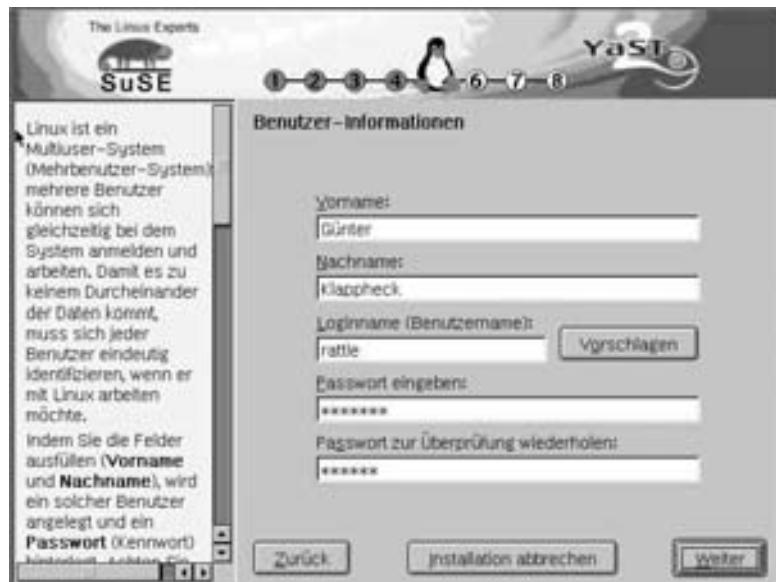


Abb. 1.76

Definition eines Beispielenutzers

Da Sie nun aber der Verwalter oder die Verwalterin dieses Systems sind, sollten Sie auch ein Passwort für den Superuser *root* vergeben – auch wenn dieses in Ihrer praktischen Arbeit lediglich ein Denkanstoß für Sie ist, noch einmal genau zu überlegen, ob Sie wirklich unter dieser Benutzerkennung arbeiten möchten.



Abb. 1.77

Passwortvergabe für den Systemverwalter *root*



Erstinstallation



Im nun folgenden Dialog zeigt Ihnen YaST2 noch einmal die von Ihnen getroffenen Einstellungen zur Installation an, speziell, was mit Ihrem System bezüglich der Partitionierung geschehen wird. Sie können gegebenenfalls durch Anklicken der Schaltfläche **Zurück** zu vorhergehenden Dialogen zurückkehren, um eventuelle Änderungen vorzunehmen, die getroffenen Einstellungen auf einer Diskette speichern und die Installation später fortsetzen oder aber notfalls die Installation abbrechen.

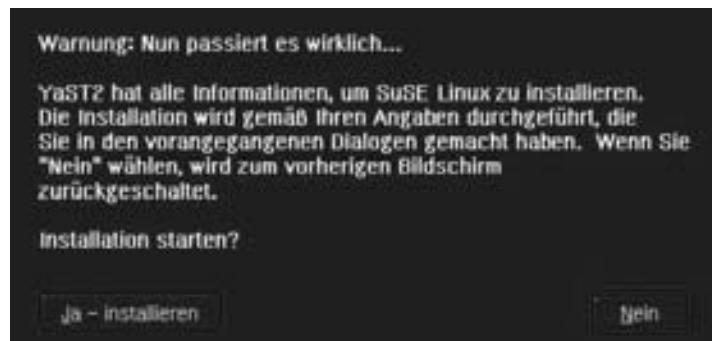


Abb. 1.78

Zusammenfassung der von Ihnen gewählten Installationsparameter

Wenn Sie jedoch die Schaltfläche **Weiter** anklicken, erscheint folgender letzter Dialog, der Sie nur noch einen Mausklick davon abhält, Linux auf Ihrem System einzurichten und gegebenenfalls vorhandene Daten zu überschreiben.

Abb. 1.79



Die Installation mit YaST2



Der nun folgende Installationsvorgang nimmt nun in Abhängigkeit der von Ihnen getroffenen Softwareauswahl (insbesondere Besitzer der Vollversion seien davor gewarnt, eine vollständige Installation vorzunehmen, da die anschließende Deinstallation von Softwarepaketen, die man doch nicht verwenden möchte, länger dauert als der Installationsvorgang) und Hardwareausstattung unterschiedlich lange Zeit in Anspruch (die Sie als Besitzer der Eval-Version vielleicht für einen Spaziergang mit Ihren Lieben nutzen könnten, da hier kein CD-Wechsel erforderlich ist ...).

Vielleicht sind Sie aber hinreichend masochistisch veranlagt oder auch nur zu faul zum spazieren gehen (oder zu neugierig), um sich im folgenden Dialog permanent über der Fortgang der Installation informieren zu lassen.

Abb. 1.8o

Fortschritt der Installation (hier Kopieren des Pakets *ncurses*)



Nach dem Kopieren der Pakete (nur die erste CD wird bis hierher benötigt) bleiben zwei Punkte der Installation offen: eine X-Window-Installation analog zu SaX (vergleichen Sie dazu unsere Ausführungen in *Kapitel 12 X-Konfiguration und KDE*) und die Möglichkeit, Drucker, Sound und Netzwerk zu konfigurieren. (Sie haben sicherlich die Bedeutung des Penguins in der Kopfzeile des Dialogs erkannt!).

Zunächst meldet Ihnen aber das Tool, dass das Basissystem erfolgreich installiert wurde und ein Reboot des Systems notwendig ist.

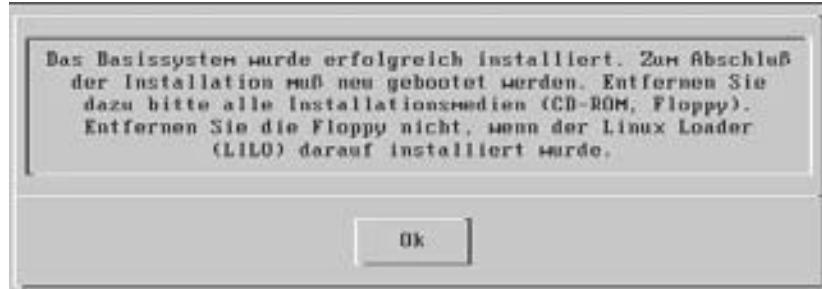


Erstinstallation



Abb. 1.81

Abschlussmeldung nach dem Kopieren



Folgen Sie den Anweisungen des Dialogs und bestätigen Sie sie durch Drücken der Taste .

Nach dem Neustart meldet sich LILO. Drücken Sie erneut oder warten Sie einfach ab, bis Linux automatisch startet.

Hinweis



Neben der Möglichkeit Linux oder, falls vorhanden, ein anderes Betriebssystem zu starten, können Sie auch einen Speicherzugriffstest durchzuführen (*memtest86*).

Abb. 1.82

LILO nach dem Neustart



Danach übernimmt wieder YaST2 die Regie und fordert Sie auf, CD 2 einzulegen, allerdings nur, wenn Sie bei der Auswahl der Pakete auch solche gewählt haben, die nicht auf der ersten CD sind.



Die Installation mit YaST2

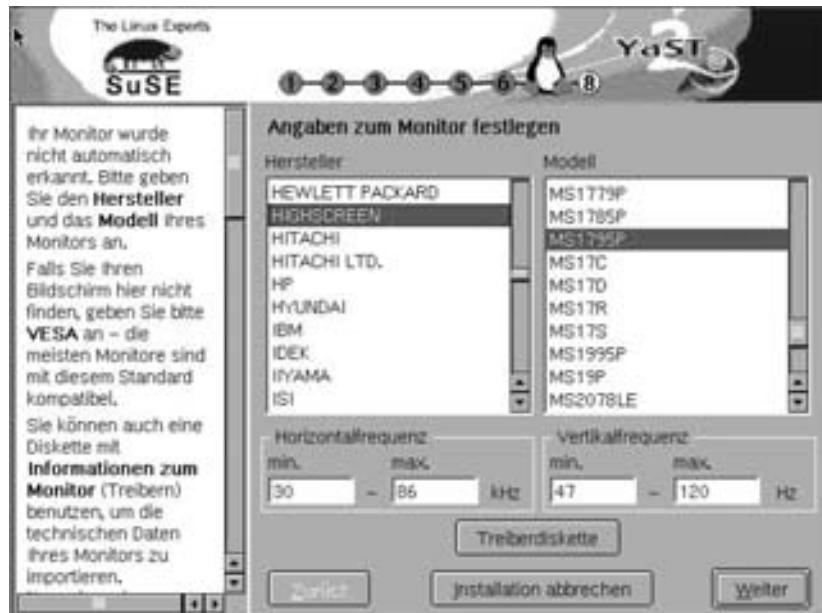


Abb. 1.83

Einlegen einer neuen CD

Nach Kopieren der restlichen Dateien erfolgt nun die X-Window-Installation. Zunächst ist der Monitor auszuwählen. Dies geschieht auf Basis einer umfangreichen Liste, eventuell kann aber auch eine Treiberdiskette benutzt werden.

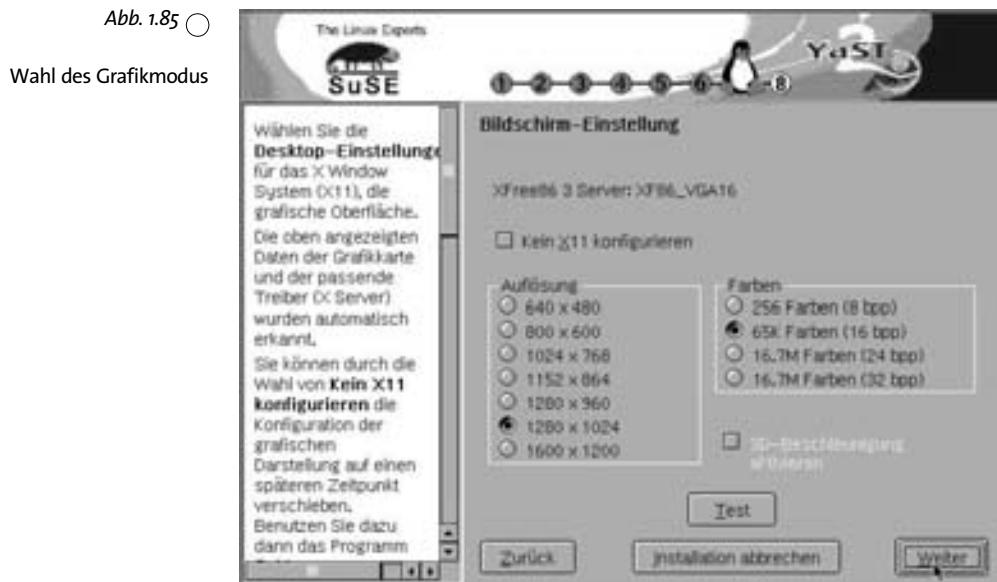
Abb. 1.84
Monitorauswahl



Hier sind Auflösung und Farbtiefe festzulegen und, falls verfügbar, die 3-D-Beschleunigung zu aktivieren.

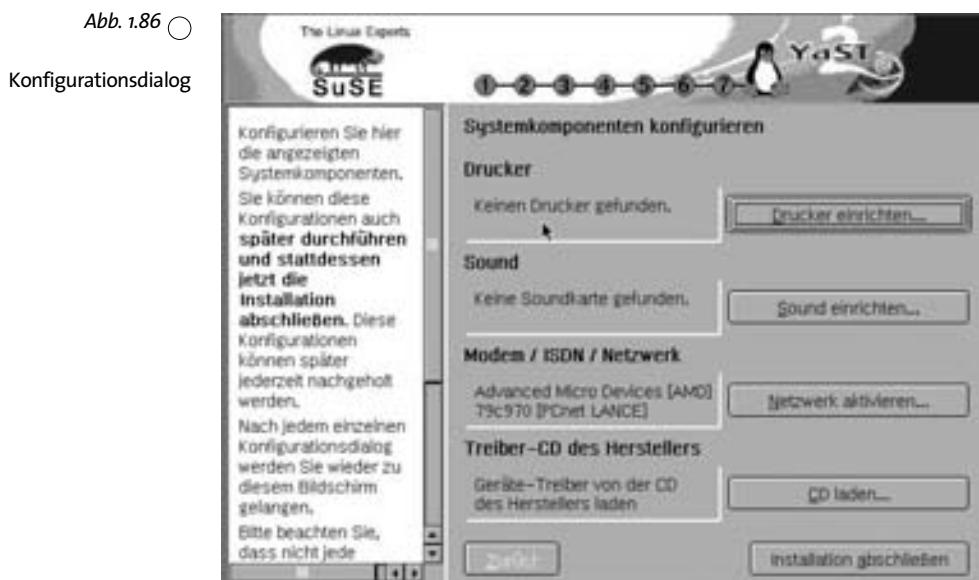


Erstinstallation



Die eigentliche Installation ist abgeschlossen. Sie können sich nun noch das Installationsprotokoll anschauen, was dann interessant ist, wenn Pakete nicht installiert werden konnten.

Schließlich gelangen Sie zum Konfigurationsdialog.



Die Installation mit YaST2



Von hier aus können Sie wahlweise Drucker, Sound oder Netzwerk einrichten oder eine Treiber-CD laden. Als Beispiel wird die Installation des Druckers gezeigt.

Zunächst legen Sie fest, wie der Drucker angesprochen werden soll (parallele, serielle Schnittstelle, Netz).

Abb. 1.87

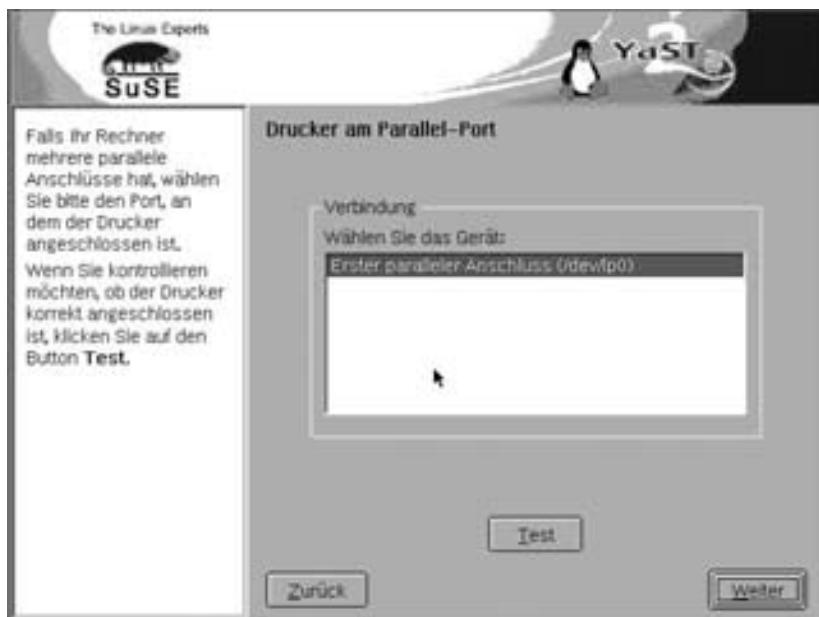
Festlegen des Druckeranschlusses



Im Falle der parallelen Schnittstelle sollte dann der folgende Dialog die Nummer der Schnittstelle abfragen sowie einen Test anbieten.

Abb. 1.88

Festlegen der parallelen Schnittstelle



Erstinstallation



Wie schon beim Monitor können Sie dann, falls YaST2 den angeschlossenen Drucker nicht schon erkannt hat, den Druckertyp wählen.

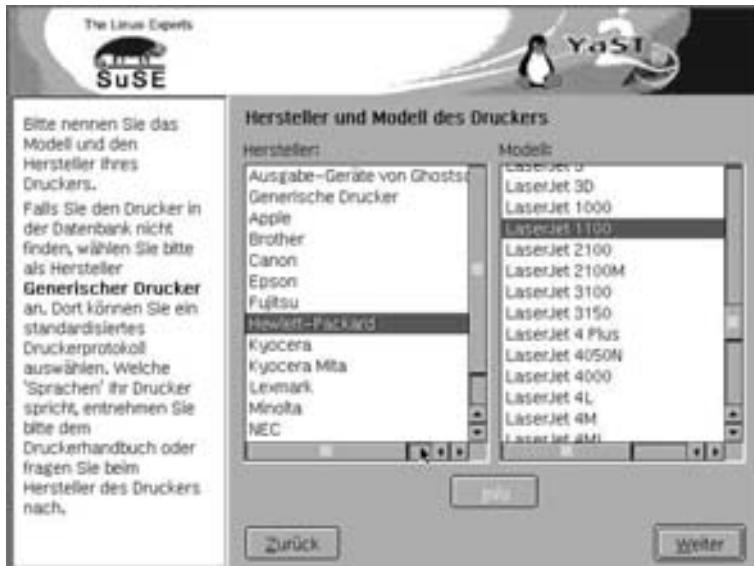


Abb. 1.89

Wahl des Druckers

Zu guter Letzt wird die gewählte Konfiguration angezeigt und kann übernommen, als Standard definiert oder aber noch einmal bearbeitet werden.

Abb. 1.90

Abschluss der Druckerkonfiguration



Die Installation mit YaST2

1

Wählen Sie nach dieser Konfigurationstour *Installation abschließen*, dann schlägt Ihnen YaST2 vor, als Standardbenutzer einzuloggen. Nutzen Sie die Möglichkeit. Wir raten Ihnen noch einmal davon ab, als Benutzer *root* einzuloggen und „alltägliche“ Arbeiten auszuführen, in diesem Falle also das System kennen zu lernen.

Abb. 1.91

```
[  0.000000] USB determination method: zshbin/scpi
[  0.000000] No USB controller found
[  0.000000] No USB Host found
[  0.000000] Starting service at daemon:
[  0.000000] Loading keymap qzscr/de-latin1-nodeadkeys.map.gz      skipped
[  0.000000] Loading compose table winkeys shiftctrl latin1.add    done
[  0.000000] Loading console font lat1-16.psfs                  done
[  0.000000] Loading unimap latin.unsi                      done
[  0.000000] Setting up console ttys                         done
[  0.000000] Starting Name Service Cache Daemon                done
[  0.000000] Starting CRON daemon                          done
[  0.000000] Starting service xdm                          done
[  0.000000] Starting inetsd                           done
[  0.000000] Starting personal-firewall (final) (not active)   unmet
[  0.000000] Master Resource Control: runlevel 5 has been reached
[  0.000000] Failed services in runlevel 5:                   lpd

[  0.000000] Please enter "root" to login as user root...  


```

Zusammenfassung

Wie auch immer Sie Ihr SuSE-Linux installiert haben mögen (mit YaST1 oder mit YaST2) – wir wünschen Ihnen viel Spaß und viel Erfolg beim Erlernen und im Umgang mit Ihrem neuen System, mit dem Sie folgendes *Kapitel 2 Bootstrapping the new user* in ersten Schritten vertraut macht.



Bootstrapping the new user



Die Shell	74
Dateinamen	75
Multiuser	76
Benutzerrechte	76
Multitasking	77
Das Filesystem	77
Linux und andere Filesysteme	79
Editoren	80
Einfache Dateioperationen	80
Onlinehilfe	82
Aliase	83
Hilfe – die Statusmeldungen laufen während des Bootens zu schnell vorbei!	85
Dienstprogramme	86
Oh Wunder ...	91
Midnight Commander	92
Herunterfahren des Systems	92



2

Bootstrapping the new user



Dieses Kapitel soll einen kurzen Einblick in die Handhabung von Linux geben. Es ist in erster Linie für Umsteiger von MS-DOS-Systemen gedacht, für die Begriffe wie Dateien, Pfad und Verzeichnisbaum keine Fremdwörter sind, damit sie nach der Installation ihres Linux-Systems gleich mit einigen Experimenten loslegen können.

Wenn diese Begriffe für Sie neu sind, empfehlen wir Ihnen, dieses Kapitel zu überschlagen und gleich mit den *Kapitel 3 Linux-Dateisystem* und *Kapitel 4 Benutzерumgebung* fortzufahren, in denen Sie mit den (zugegebenermaßen manchmal etwas, in Norddeutschland sagt man „drögen“, also „trockenen“) Konzepten eines UNIX-Systems und seiner Handhabung vertraut gemacht werden.

Dem erfahrenen DOS-User empfehlen wir: Entspannen Sie sich zunächst, und begeben Sie sich auf eine faszinierende erste Reise durch ein UNIX-System; doch seien Sie vorgewarnt – auch für Sie beginnt der Ernst des Lebens in ca. 25 Seiten, wo Sie sich dann intensiv mit (für Sie zum Teil) neuen Konzepten herumschlagen müssen, die gerade vom „DOS-Power-USER“ häufig zu sehr auf die leichte Schulter genommen werden. Machen Sie nicht den Fehler, sich auf Ihren DOS-Lorbeeren auszuruhen, denn dann würden Ihnen viele Leistungsmerkmale entgehen, die Linux zu bieten hat, so dass Sie gleich bei Ihrer rostigen alten DOSe hätten bleiben können.

Die Shell

Als Bill Gates IBM die Version 1.0 von MS-DOS verkaufte, war dies kaum mehr als eine etwas aufgepeppte Version des damals allgegenwärtigen Standard-Betriebssystems für Kleincomputer, CP/M. Bereits zu diesem Zeitpunkt hatte DOS mehrere grundlegende Prinzipien von UNIX kopiert: Z.B. wartet auf dem Schirm eine Benutzeroberfläche („Prompt“), an der die Befehle des Betriebssystems oder auch Programme ausgeführt bzw. gestartet werden können.

Die eingegebenen Befehle werden von einer Shell ausgeführt, den DOS-Usern als *command.com* bekannt. Auch diese Shell ist nichts weiter als ein Programm, das durch ein anderes mit ähnlicher Funktion ersetzt werden kann (ein beliebtes Substitut ist *4DOS*). Das passiert in DOS allerdings selten, so dass sich Millionen von Benutzern mit den Nickeligkeiten der *command.com* herumschlagen mussten, wobei der Fortschritt der Weiterentwicklung darin bestand, die Fehlermeldungen in der jeweiligen Landessprache auszugeben.

Der Benutzer von einst UNIX und jetzt Linux hat es da einfacher: Nicht nur, dass mehrere Shells vorhanden sind, wie z.B. die C-Shell, die Korn-Shell, die Bourne-Shell und die *bash*. Diese Shells sind wesentlich mächtiger als es die *command.com* jemals sein wird; sie werden weiterentwickelt und können jederzeit ausgetauscht werden.





Dateinamen

In der Version 1.0 von MS-DOS hatte eine Diskette ein Inhaltsverzeichnis ohne jegliche Schachtelung, wie es von CP/M kopiert wurde. Spätestens mit dem Eintreten der Festplatte in die PC-Welt erwies sich dies als äußerst unpraktisch, da mit der zunehmenden Größe der Speichermedien die Anzahl der Dateien unweigerlich wuchs. Auch hier wurde wieder eine Anleihe bei UNIX gemacht: Die Unter-Inhaltsverzeichnisse (*subdirectories*) wurden eingeführt. Was Microsoft bewogen hat, den in UNIX üblichen Slash / zur Schachtelung der Inhaltsverzeichnisse in einem Pfad durch den Backslash \ zu ersetzen, wird ein ewiges Rätsel bleiben. Auf jeden Fall ist dieses eine kleine, aber immer wieder störende Falle für den Umsteiger.

Auch wurde in MS-DOS die Unterscheidung zwischen Groß- und Kleinschreibung der Dateinamen (und damit der Befehle) abgeschafft. Was auf den ersten Blick relativ praktisch aussieht, verringert aber im Alltag die Möglichkeiten zur Benennung einer Datei. In UNIX und also auch in Linux sind *LIESMICH*, *LiesMich* und *Liesmich* drei verschiedene Dateinamen, in MS-DOS werden sie gleichwertig behandelt.

Um die Inhaltsverzeichnisse der Disketten klein zu halten und die Programmierung des Systems zu vereinfachen, hat MS-DOS der Namensgebung der Dateien weitere Gewalt angetan: Die Länge wurde auf das bekannte Schema 8 + 3 Zeichen reduziert, der Punkt zwischen diesen beiden Blöcken wurde zu einem festen Bestandteil des Namens.

Dies führte zum Krieg der Abkürzungen und teilweise zur Verbannung der Vokale aus den Dateinamen (z.B. *chkdsk* für den Befehl *check disk*).

UNIX erlaubte schon in seinen frühesten Zeiten 14 Zeichen, das Linux-Dateisystem *ext2* (*extended filesystem, version 2*) erlaubt Dateinamen bis zu 255 Zeichen Länge. Weiterhin ist der Punkt ein Zeichen wie jedes andere auch, also darf er mehrfach oder auch gar nicht in Dateinamen auftreten, zum Beispiel:

`Dies.ist.ein.langer.Dateiname`

Ist also unter Linux völlig legal.

Auch müssen Dateien nicht bestimmte Suffixe besitzen, damit sie ausgeführt werden können, wie dies in DOS mit den Endungen *.com*, *.exe* oder *.bat* der Fall ist. Suffixe können genutzt werden, um die Zugehörigkeit zu einem bestimmten Dateityp kenntlich zu machen, z.B. *.ps* für PostScript-Dateien. Dies dient aber im Gegensatz zu DOS der Übersichtlichkeit und ist keineswegs Pflicht. So besitzen ausführbare Dateien in Linux in aller Regel kein Suffix.





Multiuser

Linux ist ein Multiuser- und Multitasking-Betriebssystem. Seine Mehrbenutzerfähigkeit zeigt Linux im Alltag dadurch, dass man zwischen den „Benutzern“ bzw. so genannten „virtuellen Konsolen“ mittels **Alt+F1-F6** hin- und herschalten kann. Man kann also gleichzeitig sowohl als Systemverwalter (*root*) als auch als „Karl“ und als „Napp“ auf dem System arbeiten.

Unter UNIX stand ursprünglich ein Rechner in einer Abteilung, an den per serieller Schnittstellen mehrere ASCII-Terminals angeschlossen waren, so dass mehrere „echte“ Benutzer gleichzeitig an dem Rechner arbeiten konnten. Dies kann auch unter Linux realisiert werden.

Am Anfang der Installation – und in den meisten Fällen für den Rest der Zeit – sind die sechs „Benutzer“ von Linux über eine Tastatur und einen Bildschirm erreichbar, nämlich die des entsprechenden PCs.

Benutzerrechte

Ein Mehrbenutzersystem bringt Sicherheitsprobleme mit sich. Unter DOS gehören sämtliche Systemressourcen demjenigen, der vor Tastatur und Bildschirm sitzt. Selbst die seinerzeit unter CP/M üblichen *USER*-Bereiche zur Unterteilung des Dateisystems entfielen unter DOS. Ein DOS-Rechner ist wie ein offenes Buch.

Dass dieses Konzept für die Arbeitswelt nicht tragbar ist, leuchtet schon bei einem System für einen Benutzer ein, bei einem Mehrbenutzersystem ist es umso gefährlicher. Folgerichtig fordert Linux nach dem Start des Systems erst einmal den Benutzernamen und das Passwort, mit denen bestimmte Rechte verbunden sind. So kann z.B. nur der Systemverwalter bestimmte Konfigurationen ändern, neue Benutzer einrichten und bestimmte Programmpakete installieren.

Auch besitzt jede Datei *Benutzerrechte*, die den Zugriff regeln. Die Linux-Welt ist dreigeteilt: Für den Benutzer, die Benutzergruppe und „den Rest der Welt“ gibt es Zugriffsrechte, so dass man Dateien entweder nur sich selbst zugänglich machen kann (Spiele und Liebesbriefe) oder der gemeinsamen Arbeitsgruppe (Adressdatenbank) oder jedem Benutzer auf dem System (Hilfsprogramme, allgemeine Texte).

Für jede dieser drei Gruppen können wiederum drei verschiedene Arten von Zugriffsrechten vergeben werden: Eine Datei kann nur lesbar („r“, *read only*), veränderbar („w“, *writable*) oder ausführbar („x“, *executable*) sein – wie solche Rechte vergeben werden können, erfahren Sie im *Kapitel 3 Linux-Dateisystem*.

Eine globale Kennung weist dann noch eine Datei – wie unter DOS – als Inhaltsverzeichnis aus („d“, *directory*).

UNIX hat schon sehr früh Links eingeführt, die jetzt auch von Windows 95 kopiert wurden. Ein Link tritt nach außen hin wie eine Datei oder ein Verzeichnis auf, ist aber nur ein Verweis darauf. Das kann praktisch sein, wenn eine Datei oder ein Ver-



Das Filesystem



zeichnis tief im Dateisystem vorhanden ist und vielleicht auch noch einen langen Namen besitzt, man aber auf einfache Art Zugriff erhalten möchte. Die Datei bleibt also an dem angestammten Platz weiterhin existent, über einen Link kann aber der Zugriff vereinfacht werden. Bei der Anzeige über `ls -l` erkennt man Links an einem `l`.

Die unter DOS vorhandenen Attribute *hidden* (versteckt), *archive* und *system* entfallen. In Linux wird allerdings das Verstecken einer Datei durch einen schlichten Vorgang bewirkt: Der Dateiname muss als erstes Zeichen einen Punkt besitzen, und schon ist die entsprechende Datei für Standardoperationen nicht mehr sichtbar. Genutzt wird dies häufig für Konfigurationsdateien – entsprechend der *autoexec.bat* und der *config.sys* in DOS –, von denen es unter Linux eine große Anzahl gibt und die nicht zufällig oder von jedermann verändert werden sollten. Anzeigen lassen kann man sich diese Dateien mit:

```
ls -a
```

Multitasking

Es gibt aber nicht nur die sechs virtuellen Konsolen unter Linux; jeder Benutzer kann mehrere Jobs oder Tasks gleichzeitig abarbeiten. Das Schlagwort heißt hier: „Multitasking“.

Dies ist sehr sinnvoll für Aufgaben, bei denen der Benutzer ohne Eingriffsmöglichkeit gebannt auf den Schirm starrt und bestenfalls noch über den Fortgang der Arbeit informiert wird. Typische Beispiele für Jobs dieser Art sind das Formatieren von Disketten und Bändern, das Sortieren oder Kopieren größerer Datenmengen, das Drucken und das Berechnen von Pi auf die tausendste Stelle hinter dem Komma.

Während der DOS-Benutzer gewohnt ist, seine Pausen nach solchen Jobs einzurichten, gibt der Linux-Benutzer seiner Shell den Befehl, diese im Hintergrund abzuarbeiten, um selbst im Vordergrund wie gewohnt weiterzuarbeiten. Hierbei können auch mehrere Jobs in den Hintergrund verschoben werden. Dieses geschieht durch ein & am Ende der Befehlszeile oder nachträglich durch die Eingabe des Befehls *bg*.

Das Filesystem

Das Dateisystem von DOS ist geräteorientiert, die Laufwerke A: und B: sind typischerweise Diskettenlaufwerke, C: und D: die Festplatten, E: das CD-ROM-Laufwerk, und ab F: befinden sich die Laufwerke des Fileservers, wenn ein Netzwerk angegeschlossen ist.

Auf jedem dieser Laufwerke befindet sich ein Dateibaum mit seinen Unter-Inhaltsverzeichnissen und Dateien. Linux hingegen ist dateiorientiert. Es besitzt einen einzigen Dateibaum, so dass hier Datebaum und -system als austauschbare Begriffe gehandelt werden. Das bedeutet, dass die Gerätestruktur für den Benutzer nicht erkennbar ist.

Bootstrapping the new user



Dies ist eine logische Entwicklung von UNIX als Mehrbenutzersystem: Falls z.B. der Platz auf der Festplatte knapp wird, wird vom Systemverwalter über Nacht eine zweite eingebaut, ein Teil der Verzeichnisse wird auf die neue Platte kopiert. Am nächsten Morgen merkt keiner der möglicherweise großen Anzahl von Benutzern, dass sich an der Hardware etwas verändert hat, nachdem der Systemverwalter die entsprechenden Konfigurationsdateien geändert hat. Der Datebaum ist unverändert geblieben, und niemand muss seine Dateien auf neuen Laufwerken suchen, wie unter DOS oder NOVELL.

Allerdings müssen die Laufwerke an- und abgemeldet werden, was mit den Festplatten unmerklich und automatisch während des Hochfahrens des Systems geschieht, wenn sie einen entsprechenden Eintrag in der Datei `/etc/fstab` haben.

Basierend auf dem Befehl `mount`, spricht man beim Einbinden auch im Deutschen vom „Mounten“.

Bei Laufwerken mit austauschbaren Medien (Disketten, Iomega-Zip-Disketten, CD-ROMs, magneto-optische Platten) muss das Mounten jeweils manuell vor dem Wechsel geschehen. Hierzu dienen die Befehle `mount` und `umount`, wobei der Linux-Laufwerksname und der „Mount-Point“ angegeben werden müssen.

Interessant ist in diesem Zusammenhang die Funktion des Automounters, der im *Kapitel 13 Linux im Netzwerk* vorgestellt werden wird.

Der Mount-Point ist nichts anderes als ein Unter-Inhaltsverzeichnis, in dem dann die Dateien des entsprechenden Geräts zu finden sind.

Tabelle 2.1 stellt die Linux-Laufwerksbezeichner dar.

Bezeichner für Linux-Laufwerke



Bezeichnung	Erläuterung
<code>fdo</code>	Erstes Diskettenlaufwerk
<code>fd1</code>	Zweites Diskettenlaufwerk
<code>hda</code>	Erste Festplatte, gesamtes Laufwerk
<code>hda1</code>	Erste Festplatte, erste Partition
<code>hdb1</code>	Zweite Festplatte, erste Partition
<code>hdb2</code>	Zweite Festplatte, zweite Partition
<code>hdb3</code>	Zweite Festplatte, dritte Partition
<code>hdc4</code> oder <code>hdd4</code> oder <code>sdc4</code> oder <code>sdd4</code>	Je nachdem, an welchem Controller das Zip-Laufwerk angeschlossen ist
<code>cdrom</code>	CD-ROM-Laufwerk



Linux und andere Filesysteme



Bezeichner für Linux-Laufwerke (Forts.)

Bezeichnung	Erläuterung
<i>sda</i>	Erstes SCSI-Laufwerk
<i>sda1</i>	Erstes SCSI-Laufwerk, erste Partition

Tabelle 2.1



In der *Tabelle 2.1* sind die Massenspeicher eines typischen PCs mit ihren Mount-Points und – so vorhanden – DOS-Bezeichnern dargestellt. Dabei fällt auf, dass es eine Linux-Swap-Partition gibt, die nicht in das Filesystem eingebunden ist.

Im Gegensatz zu z.B. Windows wird Linux nicht eine Datei für den virtuellen Speicher, sondern eine ganze Festplattenpartition zugewiesen.

Mit Standardwerkzeugen zur Dateimanipulation kann dieser Bereich unter Linux also nicht ausgelesen oder verändert werden, da er nicht ins Dateisystem eingebunden ist.

Linux und andere Filesysteme

Wie in *Tabelle 2.1* zu sehen ist, kann unter Linux die DOS-Festplatte – in diesem Beispiel *C*: bzw. *hda1* – in den Datebaum integriert werden, so dass die Daten aus der DOS-Welt auch unter Linux zur Verfügung stehen. Dies ist einer der großen Vorteile von Linux: Intern arbeitet es mit einem virtuellen Dateisystem, so dass für jedes real vorhandene Dateisystem „nur“ ein Treiber geschrieben werden muss.

Derzeit können u.a. Filesysteme von DOS, Win95, mehreren UNIX-Derivaten, ISO-Standard-CD-ROMs und OS/2 gemountet werden – Letzteres leider derzeit aus Sicherheitsgründen nur zum Lesen (*read only*), da IBM die Spezifikationen nicht offen legt.

Aber nicht nur die bisherigen Daten können weiter genutzt werden; das bisherige Betriebssystem Ihrer Wahl (DOS, Windows 3.1x, Win95/98/2000, OS/2) kann ebenfalls weiterhin auf dem Rechner verbleiben und abwechselnd mit Linux genutzt werden. Außerdem: Unter Linux können Emulatoren betrieben werden, die es erlauben, andere Betriebssysteme quasi als eigenständige Tasks laufen zu lassen. Dies wird im *Kapitel 9 Windows für Linux? – Emulatoren* genauer beschrieben.

Wer nicht auf Windows oder OS/2 besteht, aber dennoch eine grafische Oberfläche nutzen möchte, sei auf *X Window* verwiesen, der De-Fakto-Standard auf UNIX- und daher auch Linux-Rechnern. Für diese Umgebung etabliert sich die grafische Benutzeroberfläche KDE (siehe *Kapitel 12 X-Konfiguration und KDE*) immer mehr als Standard.





Editorien

Editorien gibt es unter Linux wie Sand am Meer. Die UNIX-Standardeditoren *vi* und *emacs* laufen hier natürlich auch, wobei Letzter die Eier legende Wollmilchsau ist, also derart mächtig und komplex, dass er dem Anfänger nicht unbedingt zu empfehlen ist. Wir empfehlen *joe*, der klein und handlich ist, sich an einen der ersten Quasi-Standards der Computergeschichte hält, nämlich die WordStar-Befehle, und der die Erklärung der Tastaturkürzel jederzeit in der oberen Hälfte des Bildschirms einblenden kann (**Shift**+**K**, **H**).

Einfache Dateioperationen

Wie die Erfahrung zeigt, werden später im Buch lapidar Sätze wie „Benennen Sie aus Sicherheitsgründen die bisherige Datei *xyz.conf* um, ...“ stehen, und der neue Benutzer sitzt vor seinem Prompt und verzweifelt: Bekannte Standardbefehle von DOS, wie *copy*, *del* oder *ren(ame)* werden mit einem trockenen „command not found“ beantwortet. Deshalb hier in aller Kürze die wichtigsten Befehle in aller Schlichtheit.

Die – unter Linux oft komplexe – Syntax befindet sich weiter hinten in diesem Buch. Für den, der zu faul zum Blättern ist, bringt der Parameter *-help* oft mehr Details.

Listen von Dateien

Der Befehl zum Listen der Dateien eines Verzeichnisses hat folgende Syntax:

```
ls [Optionen] [Verzeichnis]
```

ls ohne Parameter listet die Dateien des aktuellen Verzeichnisses in mehreren Spalten auf, also ähnlich dem DOS-Befehl *dir /w*. Wer Genaueres wissen möchte, muss die Option *-l* anwenden:

```
ls -l /dos
```

zeigt die Dateien des Verzeichnisses *dos* mit Größe, Benutzerrechten etc. an. Wer auch die versteckten Dateien, also die, die mit einem Punkt beginnen, sehen will, muss das Argument *-a* für *all* hinzufügen:

```
ls -a
```

Kopieren von Dateien

Das Kopieren von Dateien geschieht mit dem Befehl *cp*. Die Syntax für diesen Befehl lautet:

```
cp [Optionen] Quelle Ziel
```



Einfache Dateioperationen



Beispiel: Die Datei *German-HOWTO* aus dem Verzeichnis */usr/doc/howto* soll in das Verzeichnis */dos/linuxdoc* kopiert werden.

```
cp /usr/doc/faq/howto/German-HOWTO /dos/linuxdoc
```

Funktioniert nicht? Funktioniert doch, aber Sie sind noch in alten DOS-Gewohnheiten gefangen: Achten Sie auf Groß- und Kleinschreibung und verwenden Sie den Slash / und nicht den Backslash \! Außerdem werden Namen von DOS-Verzeichnissen immer klein geschrieben!

Das Ziel darf übrigens nur ein Verzeichnisname sein; Wildcards im Ziel wie unter DOS, mit denen die Dateien gleichzeitig umbenannt werden können, sind nicht erlaubt.

Verschieben von Dateien

Der Befehl zum Verschieben von Dateien heißt *mv* (für *move*). Die Syntax dieses Befehls lautet:

```
mv [Optionen] Quelle Ziel
```

Die Datei *entwurf.txt* aus dem Verzeichnis */dos/mgstrarb* soll in das Verzeichnis */home/jutta* verschoben werden:

```
mv /dos/mgstrarb/entwurf.txt /home/jutta
```

Umbenennen von Dateien

DOS-Benutzer kennen den Befehl *rename*. Ein Pendant hierzu gibt es unter UNIX nicht, denn das wäre überflüssig:

```
mv [Optionen] Quelle Ziel
```

Wie sofort ersichtlich, ist das Umbenennen einer Datei nur ein Sonderfall des Verschiebens. Die jetzt im Verzeichnis */home/jutta* befindliche Datei *entwurf.txt* soll in *Magister.Arbeit* umbenannt werden:

```
mv /home/jutta/entwurf.txt Magister.Arbeit
```

Löschen von Dateien

Der Befehl zum Löschen einer Datei oder einer Gruppe von Dateien lautet *rm*. Seine Syntax sieht folgendermaßen aus:

```
rm [Optionen] [Pfad/Dateiname]
```

Was dann unser Beispiel ins Daten-Nirvana schickt:

```
rm /home/jutta/Magister.Arbeit
```



Bootstrapping the new user



Tipp



Nach Verwendung von `rm` helfen keine Programme `unerase` oder `undelete` wie unter DOS.

Erzeugen bzw. Löschen von Inhaltsverzeichnissen

Die Befehle zum Kreieren oder Entfernen von Unter-Inhaltsverzeichnissen heißen `mkdir` bzw. `rmdir`; sie lauten also wie ihre DOS-Pendants, von denen Sie vielleicht bisher nur deren Kurzschreibweise `md` und `rd` kennen, die aber unter DOS ebenfalls eingegeben werden können.

Onlinehilfe

Onlinehilfe gibt es in verschiedenen Abstufungen: Die Befehle `whatis` bzw. `apropos` geben eine einzeilige Antwort, welche die Funktion minimal erklärt. So wird auf die Eingabe

```
whatis rm
```

Folgendes ausgegeben:

```
rm (1)      -- remove files or directories
```

Beide Befehle geben dabei die Kopfzeile der *man*-Seiten wieder, die weiter unten erläutert werden. Die Zahl in Klammern gibt Auskunft über die Anzahl der gefundenen Einträge.

Nach der Eingabe von

```
whatis whatis
```

erkärt sich `whatis` selbst mit:

```
whatis (1) -- display manual page descriptions
```

Fast alle Befehle zeigen eine kurze Erklärung ihrer Optionen und Argumente, wenn man als Argument `-help`, manchmal auch `-h`, `-h` oder `/?` übergibt.

Das gleiche Ergebnis kann man oft auch dadurch erzielen, dass man den Befehl ohne Argumente und Optionen eingibt. Manchmal wird hier auch die Anweisung gezeigt, wie die Option lautet, um Hilfestellung zu erhalten.

Die Ausgabe/
Anzeige kann über
die Taste `Q`
beendet werden

Wer weitere Hilfe zu allen gängigen Befehlen benötigt, dem sind die *man pages* (*manual pages* – dt. Handbuchseiten) empfohlen, eine ausführliche Onlinehilfe, die jederzeit von der Shell aus aufgerufen werden kann, indem man den Befehl als Parameter übergibt:

```
man joe
```



Aliase



erläutert also den Gebrauch des Editors *joe* in allen seinen Einzelheiten.

Wer erst einmal Genaueres über die *man pages* selbst wissen möchte – sie geben auch über sich selbst Auskunft – gibt Folgendes ein:

```
man man
```

Aliase

Die UNIX-Welt besitzt schon lange eine Unterstützung des Benutzers, die man bei DOS gerne sähe: Aliase. Wenn unter DOS ein Befehl eingegeben wird, muss die Shell einen gleichnamigen Befehl mit einer der Endungen *.com*, *.bat* oder *.exe* finden.

Unter Linux bzw. UNIX muss eine gleich lautende Datei gefunden werden, es gibt aber als Ergänzung auch eine Alias-Tabelle, die dann durchsucht wird und die einem viel Tipparbeit abnehmen kann: Ein Alias ist ein selbstdefiniertes Kürzel, das eine lange oder komplexe Befehlszeile ausführt.

Zum Beispiel arbeitet der geplagte Neuling in seinem persönlichen *Home*-Verzeichnis mit seinen privaten Daten und muss aber immer wieder in den Dokumentationen nachlesen, die tief unten im Datebaum liegen.

Statt mit viel Tipparbeit von einem zum anderen Verzeichnis zu wechseln, erstellt er sich zwei Aliase, die ihn hin- und herspringen lassen. Diese beiden lauten z.B.:

```
alias doc="cd /usr/src/linux/Documentation"  
alias home="cd /home/glinski"
```

Wenn man diese beiden Zeilen eintippt, ist der Befehlsumfang um die beiden Kommandos *doc* und *home* erweitert worden: Durch die Eingabe von *doc* wechselt man tief in den Datebaum, um die Dokumentationen zu lesen, und mit *home* befindet man sich wieder in seinem *Home*-Verzeichnis.

Wenn man diese Art der Eingabe von Aliasen wählt, dauert die Freude genau eine Sitzung lang. Sobald man sich ausloggt oder den Rechner zum Rebooten zwingt, sind die Eingaben verschwunden.

Dies kann für bestimmte Sitzungen praktisch sein, weil man sich am nächsten Tag mit ganz anderen Themen beschäftigt. Andere Befehlsersetzung möchte man aber für ewig oder einen längeren Zeitraum greifbar halten.

Kein Problem, die Alias- (und auch andere) Informationen werden im *Home*-Verzeichnis des Benutzers in der Datei *.profile* gehalten. Je nach Distribution und Shell kann diese auch *.bash_profile* heißen. Das *Home*-Verzeichnis des Systemadministrators *root* lautet schlicht */root*, die Verzeichnisse der anderen Benutzer sind normalerweise Unterverzeichnisse von */home*, also z.B. */home/glinski*.

Bei der Arbeit mit der C-Shell werden die Alias-Informationen in der Datei *.cshrc* gehalten



Bootstrapping the new user



Dies bedeutet auch, dass jeder Benutzer seine eigenen Aliase besitzt und man sich so für verschiedene Benutzernamen verschiedene Umgebungen schaffen kann.

Tipp



Da `.profile` mit einem Punkt beginnt, ist die Datei für den Befehl `ls` unsichtbar. Erst mit `ls -a` kann man sich vergewissern, ob sie bereits existiert.

Die Syntax von `alias` innerhalb der Datei ist gleichlautend, wie oben gezeigt:

```
alias Name="Befehlszeile"
```

Es wird also mit dem Linux-Editor der Wahl eine Datei `.profile` angelegt, wenn sie nicht schon vorhanden ist, und man gibt nun Zeile für Zeile seine Aliase ein, um sie dann abzuspeichern. Je nach Speicherung im entsprechenden Verzeichnis sind all diese Aliase für den entsprechenden Benutzer nach dem nächsten Start der Shell gültig, was man mit `logout` und durch erneutes Einloggen schnell überprüfen kann.

Wenn sich auch die verschiedenen Linux-Distributionen teilweise gravierend unterscheiden, mindestens eines haben sie alle gemeinsam: Der Alias `dir` befindet sich bereits vorgefertigt auf der Festplatte. Wer wissen will, welche Aliase bereits auf dem System vorhanden sind, muss einfach nur `alias` ohne Argument eingeben.

Der Anwendungsbereich von Aliasen ist unbegrenzt. Der Linux-Neuling hat sich bestimmt schon darüber geärgert, dass die `bash` bei `cd ..` unbedingt auf einem Leerzeichen besteht, das in den späteren DOS-Versionen seit MS-DOS 3.2 durchaus weggelassen werden kann. Kein Problem, der entsprechende Alias lautet:

```
alias cd..="cd .."
```

Man achte bei der Eingabe auf das Leerzeichen innerhalb der Zuweisung in Anführungsstrichen – ein letztes Mal.

Noch nicht an die Linux-Syntax gewöhnt und immer wieder `copy` statt `cp` eingegeben? Kein Problem, die entsprechende Zeile lautet:

```
alias copy="cp"
```

Hier wird ein weiterer Vorteil von Aliasen deutlich: Man muss sich nicht um Befehlszeilenargumente kümmern, wie z.B. in der DOS-Batch-Programmierung, sondern die Argumente werden einfach an den ersetzen Befehl weitergereicht.

Der Phantasie sind keine Grenzen gesetzt. Wenn man an einem alten WYSE-Terminal mit amerikanischem Zeichensatz arbeitet, auf dem also das `Y` und das `Z` vertauscht sind, und sich dadurch des Öfteren vertippt, schafft Folgendes Abhilfe (courtesy of Peter):

```
alias copz="cp"
```

Der Lieblingseditor unter DOS hieß bisher `TED`, `EDIT` oder `QED`, und unter Linux gibt man das immer noch aus Gewohnheit ein?



Statusmeldungen laufen zu schnell vorbei



Vorbei und aus, unter Linux ist jetzt *joe* an der Reihe, der aber auch indirekt gerufen werden kann:

```
alias ted="joe"
```

Wer will, kann sich so zum einen seine Welt der DOS-Befehle nachbauen. Es lassen sich zum anderen aber auch viele weitere sinnvolle und arbeitssparende Aliase programmieren.

Das Löschen der *alias*-Eingaben geschieht, indem sie aus der Datei *.profile* herausgenommen werden oder mit dem Befehl *unalias*:

```
unalias copy
```

Hilfe – die Statusmeldungen laufen während des Bootens zu schnell vorbei!

An den verschiedensten Stellen dieses Buches wird auf die Statusmeldungen während des Hochfahrens von Linux verwiesen, wie z.B. „... der Standardwert ist ca. die Hälfte der beim Start gemeldeten Bogo Mips ...“.

Diese Statusmeldungen füllen allerdings recht schnell mehr als eine Bildschirmseite und – so denkt der DOS-Benutzer – sind für immer verschwunden.

Unter DOS wäre das auch wahr; hier gilt: „Aus dem Video-RAM, aus dem Sinn“.

Nicht so unter Linux. Die magischen Tastenkombinationen lauten **[Shift]**+**[Bild ↑]** bzw. **[Shift]**+**[Bild ↓]**, um sich die letzten Bildschirminhalte anzusehen. Es wird dann jeweils eine halbe Bildschirmseite nach oben bzw. unten gescrollt.

Es werden jeweils die letzten sechs bis sieben Bildschirme zwischengespeichert, so dass man sich nach dem Hochfahren von Linux in aller Ruhe ansehen kann, was alles an Hardware überprüft wurde und was an Hard- und Software initialisiert worden ist.

Hilfreich ist dies auch z.B. dafür, wenn man sich Inhaltsverzeichnisse mit vielen Dateien anzeigen lässt, erst recht, wenn man die Informationen des langen Formats (*ls -l*) benötigt.

Auch die Hilfsinformationen einiger Programme, die durch Eingabe des Arguments *-h* ausgegeben werden, umfassen mehr als eine Bildschirmseite.

Das Zurückscrollen der Bildschirmseiten funktioniert übrigens auch, wenn man sich in einem Linux-Programm befindet. Man kann also z.B. mit *joe* Texte editieren, um mitten in der Sitzung zwei Bildschirmseiten zurückzuscrollen, um sich dort wichtige Informationen anzusehen. Wenn es das doch unter DOS gäbe ...





Dienstprogramme

Linux wird – wie jedes bessere UNIX-System – mit einer Unzahl von Dienstprogrammen geliefert, die nicht Gegenstand dieses Buches sein können, für die es aber UNIX-Referenzen gibt.

Tipp



Das Kapitel 4 Benutzerumgebung liefert außerdem Genaueres zur Benutzerumgebung. Trotzdem werden hier kurz einige grundlegende Dienstprogramme beschrieben, ohne die der Anfänger „wie der Ochs“ vorm Berg stünde“.

unix2dos

Auf eine weitere Spezialität von Linux bzw. UNIX sei hier hingewiesen: Der Drucker will noch nicht unter Linux laufen, und die nach DOS kopierten Hilfstexte spielen beim Ausdrucken oder auch im DOS-Editor verrückt?

DOS besitzt zur Kennzeichnung des Zeilenendes eine andere Kennung (die dezimale ASCII-Zeichen 10 und 13 bzw. hexadezimal oAoD) als UNIX bzw. Linux (die nur das Zeichen 10 bzw. oA verwenden). Diese wird mit dem Programm unix2dos in das DOS-Format überführt:

```
unix2dos [Optionen] Dateiname
```

Hierbei wird die alte Datei (Linux-Format) mit der neuen Datei (DOS-Format) überschrieben; also fertigen Sie vorher Kopien an. Dieses funktioniert auch in umgekehrter Richtung mit dos2unix.

df

Wie oben erklärt, ist das Linux-Filesystem datei-, und nicht geräteorientiert. Nun möchte man aber als Benutzer trotzdem wissen, wie viel Platz auf den einzelnen Festplatten noch frei ist, welches der Mount-Point der einzelnen Partitionen ist und welche Laufwerke überhaupt vorhanden sind. Die Eingabe von

```
df
```

liefert diese Informationen.

Das Argument *-T* erzwingt außerdem die Ausgabe des Namens des Dateisystems, das Argument *-h* zeigt wiederum weitere Hilfe-Informationen an.

gzip

Viele Dokumentationen sind auch nach der Installation der Slackware 7.1 noch in komprimiertem Zustand auf der Festplatte, z.B. alle HOWTO-Dateien in dem Verzeichnis */usr/doc/faq/howto*.





Bevor man sie lesen kann, müssen sie dekomprimiert werden. Dazu dient das Dienstprogramm *gzip*.

Das Programm gzip hat eine große Anzahl von möglichen Argumenten, *gzip -h* gibt weitere Auskunft. Wir beschränken uns hier auf das Entpacken von Dateien, was folgendermaßen geschieht:

```
gzip -d Dateiname
```

Nach dem Auspacken löscht gzip das Original, damit Informationen nicht doppelt auf der Festplatte vorhanden sind und unnötigen Platz belegen.

gpm

gpm unterstützt die Maus unter Linux, wenn man sich im normalen Konsolenmodus – also nicht im Grafikmodus – befindet.

Das Programm *gpm* kann schon mittels des Slackware-Setup-Programms installiert worden sein. Falls nicht, muss einfach Folgendes eingegeben werden:

```
gpm
```

Danach können auf dem Bildschirm befindliche Zeilen oder Textblöcke mit der Maus markiert und kopiert werden. Für das Markieren gibt es zwei Möglichkeiten: Die eine funktioniert nach Windows-Art, also linke Maustaste drücken und gedrückt halten, bis man die Maus ans Ende des Textblocks bewegt hat, dann loslassen. Die zweite Möglichkeit ist, den Beginn des Textblocks mit der linken Maustaste zu markieren und das Ende mit der rechten. In beiden Fällen wird dann der Textblock mit der mittleren oder rechten Maustaste (dies hängt vom verwendeten Maustyp ab) in die aktuelle Cursorposition kopiert.

Das Praktische an *gpm* ist, dass es nicht nur mit der Shell arbeitet, sondern mit allen Linux-Anwendungsprogrammen, die nicht im Grafikmodus arbeiten. Das bedeutet, dass man für einen simplen Texteditor wie *joe* jetzt eine Maus zur Verfügung hat.

Mehr noch: *gpm* erlaubt das Kopieren von Bildschirmdateien zwischen verschiedenen virtuellen Konsolen. Wenn man sich z.B. auf der einen Konsole in der fünften Ebene der Unterverzeichnisse befindet und diesen langen Pfad in seinen Text aufnehmen möchte, markiert man diesen Pfad auf der ersten Konsole, wechselt zu seinem Editor auf der nächsten Konsole und klickt mit der mittleren (oder rechten) Maustaste. Schon ist die lange und komplexe Zeile ohne Tippfehler in den Text übernommen. Ein sehr praktisches Werkzeug.

gpm bietet eine große Anzahl von Einstellmöglichkeiten. Die Eingabe mit dem Argument *-h* zeigt alle Möglichkeiten an. Kurz erwähnt sei nur die Option für Linksänder. Mit



Bootstrapping the new user



`gpm -B321`

werden die linke und rechte Maustaste vertauscht.

whoami

Bei sechs virtuellen Konsolen kann schon der Überblick verloren gehen, wo man sich wie eingeloggt hat. Wenn man sich als normaler Benutzer eingeloggt hat und nicht als Systemadministrator *root*, verweigert die Shell bei einigen Befehlen den Dienst bzw. behauptet, diesen Befehle nicht zu kennen. Zum Beispiel kann man das System mit *reboot* nur erneut starten, wenn man als Systemadministrator eingeloggt ist.

Durch die Eingabe von *whoami* ohne Argumente erhält man sofort den Namen zurück, unter dem man sich eingeloggt hat.

history

history ist kein Dienstprogramm, sondern eine eingebaute Funktion der Shell, sei aber trotzdem an dieser Stelle erwähnt. *history* bietet ähnliche Funktionen wie *DOSKEY* unter DOS, das – wie so vieles – auch von UNIX abgekupfert worden ist. Während sich der Cursor am Eingabe-Prompt der Shell befindet, kann man sich mit \uparrow bzw. \downarrow durch die Liste der bisher eingegebenen Befehlszeilen bewegen. *history* kann aber noch mehr: Wie schnell auffällt, sind auch die Befehle der letzten Sitzungen vorhanden, d.h., alles bisher Eingegebene ist nicht wie unter *DOSKEY* beim Ausschalten des Rechners im Daten-Nirvana verschwunden, sondern es wird in einer Datei gespeichert.

Die Eingabe von *history* ohne weiteres Argument gibt die gesamte Liste aller bisher eingegebenen Befehlszeilen aus, die natürlich schnell unübersichtlich wird. Deshalb kann die Anzahl der letzten Befehle spezifiziert werden:

`history 20`

gibt die letzten 20 Befehle mit ihren laufenden Nummern aus. Da man jetzt die Nummer der gesuchten Befehlszeile kennt, die wiederholt werden soll, kann man z.B. die Zeile 169 mit

`!169`

wieder aufrufen. Der letzte Befehl kann durch die Eingabe von `!!` wiederholt werden. Die Shell besitzt mehrere Umgebungsvariablen für die History. Wer wissen will, welche Nummer die aktuelle Befehlszeile besitzt, gibt ein:

`echo $HISTCMD`

Die History wird übrigens für jeden Benutzer einzeln gespeichert, so dass jeder Benutzer sinnvollerweise seine eigene History besitzt. Der Name der Datei, in der die aktuelle History gespeichert ist, wird ausgegeben durch





```
echo $HISTFILE
```

und ergibt für *root* und den Systemverwalter als Antwort */root/.bash_history*.

Die Anzahl der gespeicherten Befehlszeilen in der *History*-Datei ist natürlich nicht unendlich. Ab einer bestimmten Anzahl werden die ältesten Zeilen verworfen, damit die Datei nicht permanent an Größe zunimmt. Die Anzahl der gespeicherten Zeilen kann wie folgt abgefragt werden:

```
echo $HISTSIZE
```

Der voreingestellte Wert von *HISTSIZE* beträgt 500.

HISTCMD, *HISTFILE* und *HISTSIZE* sind Umgebungsvariablen der Shell und werden durch das vorangestellte Dollarzeichen \$ referenziert.

find

find ist ein sehr leistungsfähiger Suchbefehl, der in diesem Buch an anderer Stelle ausführlich besprochen wird. *find* kann nicht nur nach Namen, sondern auch nach dem Datum, dem Dateityp und vielen anderen Kriterien suchen und an gefundenen Dateien sogar bestimmte Befehle ausführen. Der Anfänger bzw. Umsteiger wird aber nach bestimmten Dateinamen suchen, entsprechend den DOS-Dienstprogrammen *WHEREIS* oder *WO*. Für unsere Zwecke lautet die vereinfachte Syntax also

```
find Verzeichnis -name Dateiname -print
```

wobei *Verzeichnis* das Startverzeichnis ist, ab dem rekursiv gesucht werden soll. In dem *Dateinamen* darf eine komplexe Wildcard-Syntax angewendet werden. Hier erwähnen wir nur das Zeichen *, dessen Verwendung wie in DOS vonstatten geht. Wer also alle *README*-Dateien seines Linux-Systems finden will, fängt mit der Suche bei seinem Wurzelverzeichnis / an und benötigt folgende Zeile:

```
find / -name README* -print
```

Durch diese Befehlszeile erhält man eine Liste von ca. vier Bildschirmseiten mit *README*-Dateien, die dem Umsteiger genügend Lesestoff bieten sollten.

Entsprechend den UNIX- bzw. Linux-Konventionen unterscheidet *find* zwischen Groß- und Kleinschreibung.

Ein komfortabler interaktiver Suchbefehl wird in *Kapitel 17 Programmierung in Perl* im Abschnitt „*Optionsauswahl*“ vorgestellt, in dem ein interaktiver Suchdialog entwickelt wird.



Tipp



Bootstrapping the new user



mount und umount

Niemand weiß, wo und wann in der Geschichte von UNIX der Buchstabe „n“ des Begriffs *umount* für den Befehl verloren gegangen ist ...

Wie bereits oben erwähnt, müssen Laufwerke im Linux-Dateisystem an- und abgemeldet werden. Dieses geschieht mit den Befehlen *mount* und *umount*. *mount* ohne weiteres Argument zeigt die Liste der derzeitig gemounteten Laufwerke, die etwa so aussehen könnte:

```
/dev/hdb1 on / type ext2 (rw)
/dev/hdb2 on /usr type ext2 (rw)
/dev/hda1 on /dos type msdos (rw)
none on /proc type proc (rw)
/dev/mcd on /cdrom type iso9660 (ro)
```

mount kann mit einer komplexen Syntax aufgerufen werden, wir beschränken uns hier auf die einfachste Möglichkeit: Als Argument wird der Mount-Point übergeben, also das Verzeichnis, in dem Sie die Daten des Laufwerks standardmäßig wiederfinden wollen, die während des Setups eingegeben worden sind.

Wenn die DOS-Partition abgemeldet werden soll, z.B. um die Suchzeiten von *find* zu verringern, lautet der Befehl also schlicht

```
umount /dos
```

bzw.

```
mount /dos
```

um die DOS-Partition wieder anzumelden. Die Informationen über die Laufwerke und ihr jeweiliges Dateisystem sowie den entsprechenden Mount-Point speichert Linux in der Datei */etc/fstab*.

Diese Informationen ermöglichen auch die vereinfachte Syntax von *mount* bzw. *umount*.

Nun vergisst das Setup-Programm von Slackware schlicht, das CD-ROM-Laufwerk in der Datei *fstab* anzumelden, obwohl die Installation ja von CD geschieht und demnach alle notwendigen Informationen vorgelegen haben.

Wenn man die *fstab* editiert, findet man folgende bzw. eine ähnliche Liste vor:

```
/dev/hdb3 swap swap defaults 1 1
/dev/hdb1 / ext2 defaults 1 1
/dev/hdb2 /usr ext2 defaults 1 1
/dev/hda1 /dos msdos defaults 1 1
none /proc proc defaults 1 1
```

Hier muss nun eine Zeile eingegeben werden, damit das CD-ROM-Laufwerk auch mit der einfachen Syntax gemountet werden kann. Benötigt wird der Name des Treibers, der für das CD-ROM-Laufwerk zuständig ist und dem LILO oder LOADLIN als Parameter mitgegeben wurde.



Oh Wunder ...



Für das Goldstar-Laufwerk ist der Treiber *gscdo* zuständig, für das Sony-Laufwerk CDU 535 der Treiber *cdu535*, für ein IDE-CD-ROM-Laufwerk am ersten Controller der Treiber *hdb* und für die alten Mitsumi-Laufwerke der Treiber *mcd*. Die Zeile sollte also ähnlich wie die folgende aussehen:

```
/dev/mcd /cdrom iso9660 ro 0 0
```

Da auf Standard-CDs nicht geschrieben werden kann, muss statt der Standardeinstellungen *defaults* der Wert *ro* für *read only* eingegeben werden. Nun sollte dem Mounten der CD nichts mehr im Wege stehen.

clear

Wenn der Bildschirm voller Meldungen ist und man Klarheit haben möchte, heißt der entsprechende Befehl *clear*, was *cls* in DOS entspricht.

cal

Um nur kurz zu dokumentieren, welche Vielzahl an Dienstprogrammen Linux mitliefert, tippe man ein:

```
cal
```

Es wird sofort das Kalenderblatt des aktuellen Monats ausgegeben.

Wenn man den Jahreskalender von 1999 sehen möchte, muss Folgendes eingegeben werden:

```
cal 1999
```

Wer wissen will, auf welchen Werktag die Sylvesternacht des nächsten Jahres fällt, muss eingeben:

```
cal 12 2002
```

Oh Wunder ...

Wer sich jetzt noch ansehen möchte, wozu sein einfacher PC unter Linux fähig ist, obwohl er unter DOS oder Windows 3.1 immer nur ein Programm zur Zeit erledigen konnte, der gebe nur *top* ein und sehe sich die Liste der ca. 20 bis 30 Tasks und ihrer Zustände an, die unter Linux schon während des „Ruhezustands“ arbeiten. Außerdem wird hier die momentane Auslastung der CPU in Prozenten dargestellt.

Auch hier zeigt sich wieder, dass die eigene Hardwareausstattung allemal ausreicht; man benötigt nur ein intelligentes Betriebssystem.

Hilfe gibt es hier übrigens mit **H**, und mit **Q** kann man das Programm verlassen.

Bootstrapping the new user



Midnight Commander

Wem schon in DOS die Befehlszeilenakrobatik ein Gräuel war und wer schon dort lieber den Norton Commander oder einen seiner unzähligen Clones benutzte, dem kann geholfen werden: Es gibt den *Midnight Commander*, der die gleichen Fähigkeiten und Tastaturbelegungen wie der Norton Commander unter DOS hat, darüber hinaus aber diverse Erweiterungen besitzt. Nach der Eingabe von

`mc`

meldet sich der Midnight Commander mit dem Begrüßungsbildschirm.

Der Midnight Commander wird im Detail in *Kapitel 4 Benutzerumgebung* im Abschnitt *Wichtige Tools* vorgestellt.

Tipp



Herunterfahren des Systems

Während unter DOS noch der Griff zum 220-Volt-Schalter genügt, um Feierabend zu machen, muss bereits unter Windows 3.x das System geordnet verlassen werden, um die Initialisierungsdateien zu aktualisieren, offene Dateien zu schließen etc. Unter Windows 95 und NT muss das System bereits explizit heruntergefahren werden.

Dies gilt natürlich erst recht für Linux, bei dem im Hintergrund viele fleißige Dämonen (*daemons*) arbeiten.

Falls man Linux unvorbereitet den Strom abdreht, ist eine Inkonsistenz des Dateibaums wahrscheinlich. Allen sichtbar und unsichtbar laufenden Programmen muss also mitgeteilt werden, dass jetzt Schluss ist.

Wenn Sie den Rechner danach automatisch gleich wieder hochfahren möchten, geben Sie Folgendes ein:

`reboot`

Wenn Sie danach wirklich den Rechner ausschalten möchten, so muss eingegeben werden:

`shutdown -h now`

Eine weitere Möglichkeit ist die Eingabe von *halt*, um das System herunterzufahren!

Beides ist für normale Benutzer nicht möglich; die Shell behauptet dann, dass sie diesen Befehl nicht kennt. Man muss hierfür als Systemverwalter, also als *root* eingeloggt sein.

In beiden Fällen erscheinen eine Menge Meldungen, die bekannt geben, welche Prozesse heruntergefahren wurden. Im zweiten Fall erscheint als Letztes die Meldung „System halted“, wonach man den Rechner beruhigt ausschalten kann.





Linux-Dateisystem



Allgemeines	94
Überblick über das	
Linux-Dateisystem	95
Ein typischer Linux-Verzeichnisbaum	105
Arbeiten im Dateisystem	110
Dateien	129
Besondere Dateibefehle	144
Datensicherung	166
Rechte	166
Befehle für die Rechtevergabe	172
Einrichten des Drucksystems	179
Aufbau eines Dateisystems	197
Arbeiten mit Dateisystemen	202
Erstellen eines Dateisystems	202
Mounten von Dateisystemen	205
Prüfen von Dateisystemen	209



3

Linux-Dateisystem



Wie bereits im *Kapitel 2 Bootstrapping the new user* angedroht, werden Sie sich in diesem Abschnitt des Buches mit den Ideen und Konzepten der Dateiverwaltung unter UNIX befassen.

Von einer „Drohung“ ist aus dem Grunde die Rede, weil für das vollständige Verständnis der Prinzipien der Dateiverwaltung und -manipulation zum Teil tiefer gehende Betrachtungen auch physikalischer Hintergründe unumgänglich sind.

Als Nutzer eines Linux-Systems, der dieses für sich selbst einrichtet, sind Sie selbst der Systemadministrator, der für die Lauffähigkeit des Systems verantwortlich ist.

Aufgrund der Komplexität eines UNIX-Systems haben Sie hierdurch die Möglichkeit, durch eine vorschnelle Handhabung der Ihnen zur Verfügung stehenden Befehle das gesamte System lahm zu legen, was im schlimmsten Fall eine Neueinrichtung erforderlich macht.

Haben Sie aber die in diesem Kapitel vorgestellten Konzepte und Befehle, die für deren Umsetzung bereitgestellt werden, ausreichend verstanden, sind Sie auf dem richtigen Weg zum „Superuser“.

Stürzen Sie sich also voller Elan in die Tiefen des Dateisystems und lernen Sie grundlegende Befehle zur Verwaltung dieses Systems und zur Manipulation der in ihm enthaltenen Daten kennen.

Nach Abschluss dieses Kapitels können Sie sich dann gestärkt den Feinheiten der Benutzerumgebung widmen.

Allgemeines

Einer der Vorteile der EDV ist die Möglichkeit, Daten, die bei der Arbeit entstehen, zu speichern, also längerfristig zur Bearbeitung zur Verfügung zu stellen. Was für den normalen EDV-Anwender ein einfacher Schritt – nämlich in der Regel die Wahl des richtigen Befehls im richtigen Menü – ist, erfordert auf der Seite des Betriebssystems eine sorgfältige Organisation. Daneben muss das Betriebssystem Mittel zur Verfügung stellen, die eine Verwaltung der gespeicherten Daten durch den Anwender ermöglichen. So müssen wichtige Daten etwa aus Sicherheitsgründen kopiert werden können, bedeutungslos gewordene Daten müssen sich löschen lassen, die Daten müssen sich so ablegen lassen, dass sie leicht wieder aufzufinden sind. Nicht zuletzt sind Programme Daten. Ein Betriebssystem muss auch die Möglichkeit bieten, Programme zu starten.

Dienstleistungen dieser Art sind es, die für den gewöhnlichen Anwender ein Betriebssystem ausmachen. Man lernt sie je nach System als Befehle oder als Bestandteil besonderer Hilfsprogramme kennen. Im folgenden *Kapitel 4 Benutzerumgebung* werden zunächst die Mittel vorgestellt, die über die *bash* zur Verfügung stehen, um Dateien verwalten zu können. Um diese Mittel zu testen, müssen Sie an der Eingabeaufforderung (dem Prompt) der *bash* arbeiten. Notfalls ist daher eine Benutzeroberfläche wie etwa X Window zu verlassen.



Überblick über das Linux-Dateisystem



Zunächst erhalten Sie eine allgemeine Übersicht über die Linux-Dateiverwaltung. Begriffe wie Datei, Verzeichnis, Gerätedatei werden erklärt, die Dateinamensregeln erläutert und ein Verzeichnisbaum dargestellt, wie er in fast jedem Linux-System vorhanden ist. Zum Navigieren im Dateisystem und zum Handling von Dateien werden die nötigen Verzeichnis- und Dateibefehle bereitgestellt.

UNIX, das Betriebssystem, das bei der Entwicklung von Linux Pate gestanden hat (oder sollte man sagen Pate steht?), ist unter anderem deshalb entstanden, um Programmierern eine geeignete Programmierumgebung zu schaffen. Daher nimmt es nicht Wunder, dass es ein ganzes Bündel von Befehlen, die den Inhalt von Dateien anzeigen, in beiden Systemen gibt. Auch aus diesem Bereich sollen die wichtigsten Befehle gezeigt werden. Nicht zuletzt der Druck von Daten wird in diesem Zusammenhang erläutert werden.

In herkömmlichen Einplatzsystemen wie DOS, OS/2 oder Windows 95 spielt die Diskette als Datenträger eine wichtige Rolle. Es wird gezeigt, wie Disketten in das Linux-Dateisystem eingebunden und benutzt werden können. Diese Technik des „Mountens“ ist natürlich auch für CD-ROMs und (weitere) Festplatten einsetzbar. Das Einbinden zusätzlicher Festplatten ist ein eher ursprünglicher Einsatzbereich dieser Technik. Anhand von Disketten lässt sie sich recht einfach demonstrieren.

Datensicherung ist natürlich auch in einer Linux-Umgebung notwendig. Wer bereits intensiveren Kontakt zur EDV hat, wird schon einmal versehentlich eine Festplatte formatiert, wichtige Dateien unwiderruflich gelöscht oder auf eine andere Weise wesentliche Daten verloren haben. Im Folgenden finden Sie Befehle, die es Ihnen unter Linux ermöglichen, Daten zu sichern. Wir möchten aber darauf verweisen, dass auch ein so „intelligentes“ Betriebssystem wie Linux nur die Mittel zur Datensicherung bereitstellt – in der täglichen Praxis ist auch in Linux der Anwender gefordert.

Nachdem Sie die praktische Seite der Dateiverwaltung kennen gelernt haben, soll darauf aufbauend ein Einblick in die eigentliche Dateiverwaltung gegeben werden. Anhand des Standard-UNIX-Dateisystems werden Begriffe wie „I-Node“ oder „Link“ erläutert und kurz einige andere Dateisysteme angesprochen. In diesem Zusammenhang wird auch erklärt, wie Linux den Zugriff auf DOS-Daten realisiert.

Überblick über das Linux-Dateisystem

„Unter UNIX ist nahezu alles eine Datei.“ Dieser Satz aus dem Buch von Kernighan/Pike macht auf drastische Weise klar, wie UNIX-Systeme – und somit auch Linux – Daten verwalten. Es gibt nur und ausschließlich Dateien, wenn es um das Bearbeiten und Verwalten von Daten geht. Dabei ist eine Datei aus der Sicht des Betriebssystems nichts anderes als eine Folge von Bytes, Information in digitalisierter Form also.

Kernighan und Pike sind zwei an der Entwicklung von UNIX beteiligte Programmierer, die mit ihrem auf Deutsch unter dem Titel *Der UNIX Werkzeugkasten*



Linux-Dateisystem



erschienenen Buch
ein Standardwerk
geschaffen haben

Dateitypen

Für Linux ist der Inhalt einer Datei – die Bedeutung und/oder die Struktur der Daten – zunächst ohne Belang. Linux sieht nur die Informationen als solche. Informationen, die gespeichert werden sollen, wird in einer Datei zusammengefasst. Die Nutzdaten – der Dateiinhalt – werden auf dem Datenträger abgelegt, zur Pflege der Daten werden an anderer Stelle Verwaltungsinformationen gespeichert. Das ist Aufgabe von Linux. Der Inhalt der Datei wird durch den Benutzer festgelegt, das jeweilige Anwendungsprogramm wird diese Daten noch dahingehend verändern, dass es die Daten strukturiert – Datenbankprogramme werden die Datensätze in bestimmter Form ablegen, Textverarbeitungen werden zusätzlich zum eigentlichen Text Formatierungen so in die Datei einbringen, dass sie diese beim Wiederverwenden als Formatierungsdaten erkennen.

Kurzum, Linux speichert Daten als Bytefolgen, d.h., ohne den Inhalt zu beachten. Um die Wiederverwendung und sonstige Verwaltungstätigkeiten seitens des Anwenders ermöglichen zu können, werden zusätzlich Verwaltungsinformationen auf dem Datenträger abgelegt. Diese grundsätzliche Art, Dateien als zu verwaltende Bytefolgen zu betrachten, ist Vorbild gewesen für eine Reihe moderner Dateiverwaltungssysteme. So sollten Leser mit EDV-Erfahrung speziell im PC-Bereich viele konzeptionelle Ansätze des Linux-Dateisystems bereits kennen! Auch im EDV-Bereich gilt der Satz, dass das Rad nicht zweimal erfunden werden muss, um erfolgreich zu sein. Vielen Lesern wird also der Rahmen, in dem sich dieses Kapitel entwickelt, bekannt sein. Richten Sie dann aber Ihr Augenmerk auf die Details. Linux ist ein recht logisches und somit kompromissloses Betriebssystem. Dies wird zu Beginn hin und wieder Verständnisschwierigkeiten bedeuten. Je länger man sich jedoch mit Linux beschäftigt, desto klarer werden die Strukturen hervortreten und ein Erlernen neuer Konzepte erleichtern, denn Linux hält sich an das „kiss“-Prinzip (**keep it simple, stupid**).

Linux kennt nun sehr wohl unterschiedliche Dateitypen. Diese Unterscheidung ist aber verwaltungstechnischer Natur. Die wichtigsten Dateitypen sind:

- gewöhnliche Dateien (*ordinary files*)
- Verzeichnisse (*directories*)
- Gerätedateien (*devices*)
- Named Pipes (*FIFOs*)
- Sockets

Verweise (Links)

Gewöhnliche Dateien sind die Dateien, die mit einer mehr oder weniger gewöhnlichen Anwendung erzeugt werden – der Text, der mit einer Textverarbeitung erstellt wird, die Daten einer Datenbank, die Tabelle eines Kalkulationsprogramms oder auch die Zeilen eines Shell-Skripts, Tabellen, die von Linux zu diversen Verwal-



Überblick über das Linux-Dateisystem



tungszwecken benutzt werden, wie etwa die Passworddatei `/etc/passwd`, und nicht zuletzt Programmdateien. Gewöhnliche Dateien sind solche Dateien, deren Inhalte für den einzelnen Anwender direkt oder indirekt von Interesse sind. (Am Text, den man geschrieben hat, ist man direkt interessiert, der Inhalt der Datei `/etc/passwd` ist für jeden Anwender deshalb von Belang, weil über ihn der Zugang zum System ermöglicht sowie ein Teil der persönlichen Arbeitsumgebung bereitgestellt wird).

Verzeichnisse ermöglichen eine strukturierte Dateiverwaltung. Die Organisation der Daten wird durch sie wesentlich vereinfacht. Verzeichnisse sind Tabellen mit den Namen von Dateien sowie einem Vermerk, über den das System den Zugriff auf die Daten in den Dateien steuert.

Die Anzahl der gewöhnlichen Dateien ist schon in einem Einplatzsystem wie DOS oder Windows 95 so groß, dass ein Ordnungsprinzip erforderlich wird. So war ein wesentlicher Grund für die erste Überarbeitung des Betriebssystems DOS (Einführung der Version 2.0) die Verwendung von Festplatten für den IBM PC-XT. Diese ermöglichen die Speicherung großer Datenmengen. Die Hierarchie der DOS-Dateiverwaltung kannte bis dahin in Anlehnung an das System CP/M Laufwerke (Disketten!) und Dateien. Da nun die Anzahl der Dateien sehr hoch sein konnte, führten die Microsoft-Programmierer in DOS die Verzeichnisse als Verwaltungsinstrument hinzu. Somit kennt DOS drei Hierarchieebenen in der Dateiverwaltung: Laufwerke (aus CP/M geerbt), Verzeichnisse und – natürlich – gewöhnliche Dateien. Dass dabei die Verzeichnisse und ihr Handling im Wesentlichen aus UNIX übernommen wurden, erleichtert es den heutigen Linux-Programmierern, den Zugriff auf DOS-Daten zu realisieren; denn Dateien und Verzeichnisse sind in DOS und Linux prinzipiell sehr ähnlich.

Gerätedateien vereinfachen den Zugriff des Anwenders auf die Hardware. Sie sind im Prinzip Verweise auf die entsprechenden Treiber im Linux-Kernel. Sollen etwa Daten auf dem Bildschirm ausgegeben werden, so kann der Prozess, der diese Aufgabe zu erledigen hat, die Daten einfach in die zugehörige Gerätedatei, dies könnte z.B. `tty01` sein, kopieren. Dadurch wird der zuständige Gerätetreiber im Kernel zur Ausgabe auf dem Bildschirm veranlasst.

Named Pipes oder FIFOs sind eines der vielen Mittel, die Linux für die Kommunikation zwischen Prozessen (kurz **IPC** für **I**nter **Process **C**ommunication) bereitstellt. Vereinfacht gesagt, ist ein Prozess ein laufendes Programm. Solch ein Prozess kann Daten in eine FIFO schreiben, die dann von einem anderen Prozess gelesen werden können.**

FIFO ist die Abkürzung für **F**irst **i**n **F**irst **o**ut. Damit ist das Funktionsprinzip solcher *Named Pipes* kurz beschrieben. Daten werden in derselben Reihenfolge der Pipe entnommen, in der sie in die Pipe eingeleitet werden.



Hinweis

Eine Variante dieser Möglichkeit des Datenaustauschs zwischen Programmen kennen auch die etwas fortgeschrittenen DOS-Benutzer in Form des Pipelinings. Eine Kommandozeile wie

```
tree /f | more
```



Linux-Dateisystem



dürfte solchen Lesern in ihrer Bedeutung klar sein. Der Befehl *tree* leitet seine Daten statt auf den Bildschirm an den Filter *more* weiter, der sie so bearbeitet, dass er sie bildschirmseitenweise anzeigt.

Diese Form des Pipelining hat DOS von UNIX übernommen. Es handelt sich dabei unter Linux um temporäre Objekte, die von Prozessen benutzt werden können, welche durch ein und dieselbe Befehlszeile gestartet wurden.

Hinweis



Pipelining ist in DOS anders implementiert. Während unter UNIX/Linux für die Pipeline ein Bereich des RAMs benutzt wird, werden in DOS die Daten über temporäre Dateien weitergeleitet. Dies macht sich unangenehm in Netzwerkumgebungen bemerkbar, wo die Rechtesituation ein Erstellen von Dateien nicht überall erlaubt.

Named Pipes dagegen werden als besondere Dateien angelegt und können von Prozessen benutzt werden, die unabhängig voneinander sind. Erzeugen kann man sie im Übrigen mit dem Befehl *mkfifo* (*make fifo*).

Die Befehlszeile

```
mkfifo testpipe
```

würde z.B. eine Named Pipe mit Namen *testpipe* erzeugen. Genutzt werden könnte sie dann wie folgt:

```
ls /dev > testpipe & more < testpipe
```

ls /dev zeigt die Namen aller Gerätedateien (genauer: aller Dateien des Verzeichnisses */dev*). In diesem Fall werden die Daten in die Named Pipe umgeleitet statt auf dem Bildschirm ausgegeben. Der Befehl *more* liest die Daten aus *testpipe* und gibt sie bildschirmseitenweise aus. Das & zwischen den Befehlen sorgt dafür, dass beide Teilbefehle nacheinander als Prozesse gestartet werden, allerdings ohne dass auf ihr Ende gewartet werden müsste. In diesem Fall handelt es sich allerdings nicht um IPC, da DOS als Einplatzsystem keine Prozesse kennt.

Insgesamt werden in diesem Beispiel also die Namen aller Gerätedateien bildschirmseitenweise angezeigt. Es muss allerdings angemerkt werden, dass dieses Beispiel nur die Named Pipes prinzipiell vorstellen sollte, als Benutzer wird man in der gegebenen Situation eine gewöhnliche Pipe vorziehen:

```
ls /dev | more
```

Sockets dienen, wenn man so will, ebenfalls der IPC. Dabei geht es aber nicht um Kommunikation zwischen Prozessen in einem Linux/UNIX-Rechner. Sockets ermöglichen Kommunikation über ein Netzwerk hinweg. Dabei kann durchaus einer der beteiligten Kommunikationsteilnehmer ein Nicht-UNIX-Programm sein, d.h., Sockets ermöglichen durchaus Kommunikation zwischen einem Prozess, den ein Benutzer auf einem Linux-Rechner startet, und einer DOS-Anwendung auf einem weit entfernten PC. Da nun DOS als Einplatzsystem keine Prozesse kennt, lassen sich Sockets nicht eindeutig der IPC zuordnen.



Überblick über das Linux-Dateisystem



Im Wesentlichen steckt hinter den Sockets jedoch ein dem Pipelining verwandtes Prinzip des verbindungsorientierten Datenaustauschs. Sie ermöglichen also die Weitergabe von Daten, in diesem Fall sogar via Netzwerk.

Links sind keine eigentlichen Dateien, sondern lediglich Verweise darauf. Im Zusammenhang mit den I-Nodes werden Links später noch genauer untersucht.

Die Dateinamensregel

In der täglichen Arbeit hat der Benutzer vornehmlich mit gewöhnlichen Dateien, Verzeichnissen und Links zu tun. Er erzeugt Dateien, Verzeichnisse oder Links, benennt sie um, verschiebt oder löscht sie. Dabei kommuniziert er mit Linux, indem er den jeweiligen Befehl unter Angabe des Dateinamens eingibt. Der Name der Datei ist das Bindeglied zwischen dem Anwender, der die Daten benutzen will, und Linux, das die Daten verwalten muss.

Die Dateinamensregel ist recht einfach, beinhaltet aber eben deswegen für den Anfänger einige Tücken, die sich aus dem Zusammenspiel Benutzer, Shell und Linux-Kernel ergeben.

Drei Aspekte des Dateinamens sind für den Anwender von Belang:

- die Länge
- die erlaubten Zeichen
- die Unterscheidung von Groß- und Kleinbuchstaben

Die Länge des Dateinamens hängt vom konkret gewählten Dateisystem ab. Die Dateisysteme *ext*, *ext2* und *xiafs* unterstützen Dateinamen bis zu 255 Zeichen.

Einige aus dem Umfeld von BSD-UNIX stammende Dateisysteme erlauben Dateinamen bis zu 64 Zeichen. Häufig sind dann allerdings nur die ersten 14 Zeichen signifikant, d.h., Dateinamen werden nur bis zum 14. Zeichen unterschieden. Standard-UNIX erlaubt ohnehin maximal 14 Zeichen lange Dateinamen. Diese Beschränkung sollten vor allem solche Benutzer berücksichtigen, die in Netzwerkumgebungen arbeiten, in denen Standard-UNIX-Systeme vorhanden sind.

Erlaubte Zeichen sind im Prinzip alle Zeichen außer dem Schrägstrich (/, engl. Slash). Er dient zur Trennung der einzelnen Komponenten eines Pfadnamens, die ihrerseits Dateinamen sind. Um Unklarheiten in Pfadnamen zu vermeiden, kann der / also nicht in Dateinamen benutzt werden. Alle anderen Zeichen sind vom Grundsatz her erlaubt!

Als Anfänger sollte man jedoch eine ganze Reihe von Zeichen vermeiden, da sie in irgendeiner Weise Probleme bereiten werden! Dies betrifft insbesondere die ASCII-Zeichen 32 (Leerzeichen), 13 (Carriage Return/ Wagenrücklauf) und 12 (Form Feed/



Linux-Dateisystem



Seitenvorschub). Neben den nicht druckbaren (ASCII-) Zeichen 0 bis 32 sind das u.a. die folgenden Zeichen, die für die Shell – Ihre Benutzeroberfläche – eine besondere Bedeutung besitzen:

< > | * ? [] { } () \$ ` " ! ; \ # ^

Wie gesagt, diese Zeichen sind erlaubt, bereiten aber u.U. Schwierigkeiten. Eine Episode aus der Unterrichtstätigkeit eines der Autoren mag dies beleuchten:

Während eines UNIX-Kurses meldet einer der Teilnehmer plötzlich ein merkwürdiges Verhalten des Befehls *ls*, der gewöhnlich die Dateien eines Verzeichnisses auflistet, also dem DOS-Befehl *dir* entspricht. Nun, so wurde gemeldet, lösche dieser Befehl den Bildschirm. (Ob da nicht ein Link den Befehl *ls* außer Kraft und durch *clear* ersetzt, wurde vermutet.) Die Erklärung für das Phänomen war banal, aber nichtsdestotrotz erst nach einer längeren (Fehler-)Suche gefunden worden.

Im Namen der (ausgerechnet) letzten Datei der Liste hatte sich (ausgerechnet) als letztes Zeichen das ASCII-Zeichen mit dem dezimalen Wert 12 eingeschlichen!

Wer nun weiß, dass dieses Zeichen (auch am Bildschirm) einen Seitenvorschub bewirkt, der kann sich leicht vorstellen, wie das Unmögliche möglich wurde: Ausgabe der Dateiliste = Ausgabe des Seitenvorschubzeichens (ausgerechnet) am Schluss der Liste. Also: Ausgabe einer neuen (= leeren) Seite nach Ausgabe der Liste. Gehört der Rechner, auf dem dies geschieht, nicht gerade zu den langsamsten, dürfte der Effekt tatsächlich der sein, dass man meint, der Bildschirm würde gelöscht!

Solche problematischen Zeichen sollten zunächst vermieden werden. Sie werden später – speziell im Abschnitt über die bash – Mittel und Wege kennen lernen, wie man mit unliebsamen ASCII-Zeichen im Dateinamen umgeht.

Einen gewissen Sonderfall stellt das Minuszeichen (-) am Anfang eines Dateinamens dar. Die allgemeine Befehlssyntax (der Shell) ist

`befehlsname[-option] [objekt]`

wobei *objekt* der durch den Befehl zu bearbeitende Gegenstand ist; sei es eine Datei, ein Benutzer, ein Prozess oder einfach ein Text. Dies hängt vom jeweiligen Befehl ab. Eine *option* ist eine zusätzlich angeforderte Dienstleistung. So gibt der Befehl *who* eine Liste der eingeloggten Benutzer aus. Mit der Option *-H* werden zusätzlich Spaltenüberschriften ausgegeben. Die folgenden Zeilen zeigen eine Beispieldatenausgabe:

```
$ who -H
USER      LINE      LOGIN-TIME   FROM
root      tty1      Oct  4 08:43
rumpel    tty2      Oct  4 08:46
stern     tty6      Oct  4 08:46
rattle    tty4      Oct  4 08:47
```

Überblick über das Linux-Dateisystem



Die erste Zeile enthält den Befehl samt Option, die folgenden Zeilen sind Ausgabezeilen. Davon enthält die erste die zusätzlich angeforderten Überschriften.

Verweilen wir noch kurz bei der Befehlssyntax und halten wir fest, dass die eckigen Klammern nur der Syntaxbeschreibung dienen und keinesfalls mitgeschrieben werden dürfen! Sie dienen im Allgemeinen dazu, kenntlich zu machen, dass der eingeschlossene Teil der Syntax optional ist, d.h. in der konkreten Eingabesituation entfallen kann.

Aus dem soeben Gesagten ergibt sich nun das Problem des Minuszeichens als erstes Zeichen eines Dateinamens. Optionen beginnen *immer* mit Minus, Dateien *können* damit beginnen. Entfällt nun jede Option – was bei vielen Datei-/Verzeichnisbefehlen in der Praxis der Fall ist – so hat der benutzte Befehl die Schwierigkeit, das zweite Wort der Befehlszeile korrekt zu interpretieren: Ist es eine Option oder ist es ein Dateiname?

Die Situation wird dadurch für den Befehl noch unangenehmer, dass Linux eine Zusammenfassung mehrerer Optionen zu einem Wort nach dem Minuszeichen zulässt. Der Befehl *who* kennt etwa noch die Optionen *i* und *w*. Die obige Befehlszeile lässt sich also wie folgt erweitern:

```
who -Hiw
```

Die Reihenfolge der Optionen spielt natürlich keine Rolle, wohl aber unterscheidet Linux Groß- und Kleinschreibung!

Am auffälligsten tritt dieses Problem zusammen mit dem Befehl *ls* auf. Dieser soll eine Liste von Dateien anzeigen, kennt dabei aber eine Vielzahl von Optionen. So ist z.B. *-fault* eine (zulässige) Kombination von solchen Optionen. Wie aber wird nun die Zeile

```
ls -fault
```

interpretiert? Ist *fault* eine Kombination von Optionen oder ist *-fault* ein Dateiname, vielleicht gar einer aus der auszugebenden Liste? Um es kurz zu machen: *ls* interpretiert *-fault* nicht als Dateiname. Dieser ist somit zunächst gar nicht verfügbar!

Sie vermeiden sicherlich Fehler, wenn Sie Ihre Datei nicht *-fault* nennen

Der ungeübte Benutzer sei also noch einmal vor Dateinamen gewarnt, die Sonderzeichen enthalten oder mit einem Minuszeichen beginnen.

Sie werden bei der Besprechung der bash Mittel kennen lernen, diese Probleme zu umgehen.

Linux unterscheidet Groß-/Kleinschreibung. Dies gilt insbesondere für Dateinamen. *PRIMA*, *Prima* und *prima* sind drei verschiedene Dateinamen!



Tipp

101



Linux-Dateisystem



Die grundlegende Regel sagt also insgesamt aus:

Dateinamen sind 255 (bzw. 64 bzw. 14) Zeichen lang und können jedes Zeichen außer / enthalten. Groß-/Kleinschreibung wird unterschieden. Außergewöhnliche Zeichen im Dateinamen verursachen in der Regel Probleme bei der Bearbeitung dieser Dateien.

Dateinamenskonventionen

Neben der nackten Dateinamensregel sollten Sie in der täglichen Praxis einige Konventionen beachten.

Dateinamen, die mit . (Punkt) beginnen

Aus der DOS-Welt kennen viele unserer Leser Dateiattribute als Mechanismus, um Dateien zu schützen oder für besondere Zwecke bereitzustellen. Attribute oder etwas Verwandtes kennt Linux nicht. Linux schützt Dateien stattdessen durch Rechte.

Eine der Konventionen bezüglich Dateinamen hat aber einen ähnlichen Effekt für den Benutzer wie das DOS-Attribut *h* (*hidden*, dt. versteckt). Gibt man einer Datei einen Namen, der mit einem Punkt beginnt, so wird diese Datei besonders behandelt – man könnte sagen, diese Datei ist versteckt.

Mit dem Kommando *ls-a* können die versteckten Dateien angezeigt werden.

Eine .-Datei wird von den Befehlen zur Dateiverwaltung nur dann berücksichtigt, wenn ihr Name in der Liste der zu bearbeitenden Dateien ausdrücklich auftaucht.

Nehmen wir als Beispiel den Löschbefehl *rm*. Mit der Befehlszeile

```
rm *
```

löscht man alle Dateien – außer denjenigen, deren Namen mit einem Punkt beginnen! Der * hat in dieser Zeile eine ähnliche Bedeutung wie unter DOS: In diesem Fall meint er – eigentlich – jede Datei!

Hinweis



Einzelheiten zur Benutzung von Jokerzeichen wie * erfahren Sie bei der Besprechung der bash im Kapitel 16 *Shell-Programmierung*. Dort dürfte auch deutlich werden, warum oben davon gesprochen wurde, dass .-Dateien nur gelöscht werden, wenn „ihr Name ausdrücklich in der Befehlszeile auftaucht“.

Will man aber eine Datei namens *.help* löschen, steht dem kein Attribut im Wege, man muss lediglich den Namen der besagten Datei ausdrücklich in die Befehlszeile aufnehmen! Ein Löschen erzwingt man z.B. mit:

```
rm .help
```



Überblick über das Linux-Dateisystem



Es würde aber auch mit

```
rm .*
```

gelingen; denn nun meint das Argument `.`* nicht alle Dateien, sondern durchaus analog zu DOS alle Dateien, die mit `.` (Punkt) beginnen! Darunter befindet sich aber auch `.help`.

An dieser Stelle sei vermerkt, dass das Löschen *aller* Dateien des aktuellen Verzeichnisses mit der Befehlszeile

```
rm * .*
```

erfolgt. Man beachte das Leerzeichen vor dem Punkt!

Dateien, deren Name mit einem Punkt beginnt, gelten als „versteckt“, d.h., sie entziehen sich dem Zugriff durch Befehle, sofern sich ihr Name nicht mehr oder weniger direkt in der Befehlszeile befindet. Wollen Sie Dateien effektiv vor irgendwelchen Manipulationen schützen, sollten Sie ihnen entsprechende Rechte zuweisen, ein Dateiname, der mit Punkt beginnt, schützt nur oberflächlich!

Die Bedeutung des Punktes am Beginn eines Dateinamens ist aber umfassender und nicht einfach als „Schutzfunktion“ zu sehen. Auch Dateien, die man nicht unnötig kopieren möchte, erhalten einen Dateinamen, der mit einem Punkt beginnt.

Reservierte Dateinamen

Wie weiter oben gezeigt wurde, ist nahezu jeder Dateiname erlaubt. Sie sollten aber berücksichtigen, dass es Dateinamen gibt, die Sie nicht für neu zu erstellende Dateien benutzen dürfen – es sind allerdings nur die zwei Namen `.` (Punkt) und `..` (Punkt Punkt).

Dateien werden natürlich in Verzeichnissen gespeichert. Diese werden ihrerseits angelegt, um (gewöhnliche) Dateien sinnvoll verwalten zu können. Erstellt man aber ein Verzeichnis zur Dateiverwaltung oder wird es bei der Systeminstallation automatisch angelegt, so erhält es automatisch zwei Einträge, nämlich `.` und `..`! Mit `.` ist im Übrigen das aktuelle Verzeichnis und mit `..` das übergeordnete Elternverzeichnis gemeint. Da *alle* Verzeichnisse diese beiden Einträge enthalten, können beide als Namen nicht mehr benutzt werden.

Natürlich sollte man beachten, dass Namen von existierenden Dateien als Namen für neue Dateien insofern zu vermeiden sind, als dass dann die existierenden Dateien überschrieben werden könnten. Besonderen Schutz genießen aber nur `.` und `..`!

Böse Zungen behaupten, dass dies die DOS-Entwickler glatt überlesen und so die Dateinamenserweiterung erfunden haben

Linux-Dateisystem



Dateinamenserweiterungen

DOS und seine Verwandten (Windows 95, NT und OS/2) kennzeichnen bestimmte Dateitypen mit besonderen Dateinamenserweiterungen. Diese werden durch einen Punkt vom eigentlichen Dateinamen abgetrennt (ein Grund dafür, dass der Punkt als Zeichen im Dateinamen bei DOS nicht erlaubt ist!).

So kennzeichnen die Endungen *.com*, *.exe*, *.bat* und *.sys* unter MS-DOS Programmdateien unterschiedlichen Zuschnitts. Unter Windows sind die *.ini*-Dateien von erheblicher Bedeutung: Sie werden von Windows-Programmen benutzt, um Informationen für den Programmstart – die Initialisierungsphase also – bereitzustellen. OS/2 und Windows kennen *.dll*-Dateien, **Dynamic Link Libraries**, d.h. Programmteile, die erst während des Programmlaufs benötigt und in den Arbeitsspeicher geladen werden. Ferner können diese Programmteile von unterschiedlichen Programmen genutzt werden, wenn sie einmal im Speicher sind.

Neben diesen Endungen für systemrelevante Dateien unterstützen diverse Anwendungsprogramme spezielle Endungen. Damit erleichtern sie dem Benutzer den Zugriff auf Dateien, die mit dieser Anwendung erstellt werden.

Mit dem DOS-Editor *EDIT* wird man beispielsweise die Datei *autoexec.bat* erstellen wollen. Dann muss man diesen Namen einschließlich der Endung beim Speichern eingeben.

Nehmen wir einen einfachen Editor. Unterstützt er die Endung *.txt*, so wird er alle Texte, die man mit ihm erstellt, so abspeichern, dass er ihnen automatisch die Endung *.txt* anhängt; es sei denn, man vergibt im Dateinamen ausdrücklich eine andere.

Beim Öffnen von Dateien wird er eine Dateiliste bereitstellen, in der zunächst nur die *.txt*-Dateien zu sehen sind. Will man auch andere Dateien angezeigt bekommen, muss man dies ausdrücklich verlangen (etwa, indem man die Voreinstellung **.txt* durch **** ersetzt).

Diese nützliche Unterstützung der Benutzer kennen selbstverständlich auch Linux-Programme. Das Betriebssystem als solches ist darauf aber in keiner Weise angewiesen. Linux kennt keine Dateinamenserweiterungen als Element der Dateiverwaltung!

Im Dateinamen ist (bis auf \wedge) jedes Zeichen erlaubt. Wie oft es im Dateinamen auftritt, ist ebenfalls nicht eingeschränkt. Somit kann der Punkt im Dateinamen prinzipiell beliebig oft auftreten. Als Dateiname zulässig ist also durchaus *kapitel.1.6.12.4*, sofern Ihr Dateisystem eine Dateinamenslänge von mehr als 14 Zeichen erlaubt. Was soll aber in diesem Fall als Dateinamenserweiterung angesehen werden? *1.6.12.4*, weil es nach dem ersten Punkt steht, oder *4*, weil es nach dem letzten steht?



Ein typischer Linux-Verzeichnisbaum



Versuchen Sie doch einmal mit

```
echo hallo > kapitel.1.6.12.4
```

eine solche Datei im Heimatverzeichnis zu erstellen!

Um Programmdateien zu erkennen, benutzt Linux einen anderen Mechanismus als den, die letzten Zeichen eines Dateinamens zu interpretieren. Zunächst prüft Linux den Dateityp (gewöhnliche Datei, Verzeichnis, Gerätedatei etc.) anhand der Verwaltungsinformationen, die über die Datei gespeichert sind, sowie die Rechte (Linux kennt ein „Ausführungsrecht“, das Dateien als Programmdateien kenntlich macht, doch davon später mehr). Falls nötig, liest Linux zusätzlich den Inhalt der Datei, genauer den Anfang, und vergleicht ihn mit den Daten aus einer eigens dafür angelegten Datei namens */etc/magic*, um zu entscheiden, ob die Datei ein Programm, ein Shell-Skript, eine Datendatei oder eine Datei anderer Herkunft ist. Dateinamenserweiterungen sind für Linux also ohne zwingende Bedeutung.

Wie unter DOS und seinen Verwandten kennen aber auch Linux-Programme die Unterstützung spezieller Dateinamenserweiterungen. Schon in den frühen UNIX-Varianten bevorzugte der C-Compiler Quellcode dateien, welche die Endung *.c* und Objektdateien, welche die Endung *.o* hatten. Zwingend sind solche Endungen jedoch nicht!



Unter Linux sollte man besser von Dateinamensendung sprechen, wenn überhaupt

Ein typischer Linux-Verzeichnisbaum

Verzeichnisse dienen der organisierten Dateiverwaltung. Üblicherweise werden Dateien, die zusammengehören, z.B. weil sie Daten ein und desselben Programms sind, in einem gemeinsamen Verzeichnis abgespeichert.

In einer Einplatzumgebung wie DOS, OS/2 oder Windows 95 ist der Benutzer im Wesentlichen selbst für den Aufbau der Verzeichnisstruktur verantwortlich. Durch die Installation diverser Programme wird allerdings ein Teil dieser Struktur vorgegeben sein.

In einer Mehrplatzumgebung wie Linux wird man erwarten können, dass wesentliche Teile des Dateisystems vorgegeben sind, schon aus dem Grund, dass man gewisse Informationen im System einfacher wiederfindet. In der Tat wird ein Linux-Benutzer bestimmte Verzeichnisse und auch gewöhnliche Dateien in einem Standard-Linux-System unter speziellem Namen an spezieller Stelle wiederfinden.

Abbildung 3.1 zeigt ein typisches Dateisystem. Aufgelistet sind Verzeichnisse, die Sie in einer normalen Linux-Umgebung in der angegebenen Position und mit entsprechendem Namen wiederfinden werden. Der Systemverwalter kann natürlich selbst oder über von ihm beauftragte Benutzer beliebige Änderungen an dieser Struktur vornehmen. Dies hat in der Regel für die Verwaltung unangenehme Konsequenzen, so lassen sich Voreinstellungen bei einigen allerdings sehr nützlichen Befehlen nicht mehr benutzen.



Linux-Dateisystem

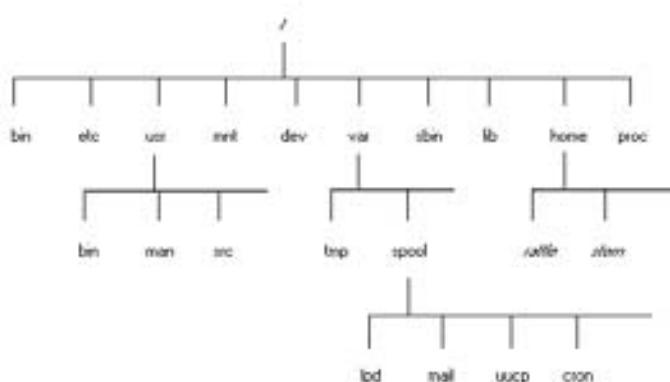


Außerdem hängen die Namen der Verzeichnisse und ihre Positionen im Dateisystem von der jeweiligen Linux-Distribution ab. Da sich die meisten Linux-Anbieter (aus Kompatibilitätsgründen) und auch die meisten Systemverwalter (weil sie unnötige Arbeit vermeiden wollen) scheuen, vollkommen eigenständige Dateihierarchien anzulegen, werden Sie in Ihrem Linux-System einen Verzeichnisbaum finden, der dem in Abbildung 3.1 gezeigten Baum entspricht oder auf ihm aufbaut.

Die Bedeutung der wichtigsten Verzeichnisse wird nun kurz beschrieben. Leser, die mit Standard-UNIX vertraut sind, werden die meisten der genannten Verzeichnisse und ihren Sinn kennen. Linux ist in der Tat kein komplett neues Betriebssystem. Wesentliche, auch für den normalen Benutzer wichtige Komponenten sind dem Vorbild nachgebildet.

Abb. 3.1

Ein typischer
Linux-Dateibaum



Pfade

Eine Vorbemerkung zur folgenden Tabelle: Im Gegensatz zur Abbildung werden hier nicht die eigentlichen Verzeichnisnamen, sondern Pfadnamen verwendet.

Eine Besonderheit von Verzeichnissen ist es, dass sie neben Dateien auch (Unter-)Verzeichnisse aufnehmen können. So ergibt sich eine baumartige Verwaltungsstruktur der Daten. Der komplette Name einer Datei innerhalb dieser Struktur enthält neben dem tatsächlichen Dateinamen auch sämtliche Namen der Verzeichnisse, die auf dem Weg (Pfad!) von der Wurzel zur Datei liegen. Die einzelnen Namensbestandteile werden dabei durch / getrennt (beachten Sie noch einmal die Dateinamensregel!). Ist etwa die Datei *erstaunlich* in einem Verzeichnis namens *ja*, das sich seinerseits im Verzeichnis *ist* befindet, welches ein Unterverzeichnis von *das* ist, das wiederum ein Unterverzeichnis der Wurzel ist, dann lautet der komplette Name der Datei

/das/ist/ja/erstaunlich



Ein typischer Linux-Verzeichnisbaum



Der / zu Beginn des Namens steht für die Wurzel, alle restlichen / im Namen sind Trenner zwischen den einzelnen Namensbestandteilen.

Während der aktuellen Arbeit ist stets eines der Verzeichnisse des gesamten Dateibaus aktiv. Dieses Verzeichnis wird auch „aktuelles Verzeichnis“ genannt. Einmal angenommen, das Verzeichnis *ist* im obigen Beispiel sei das aktive, dann kann man, um die Datei *erstaunlich* zu benennen, die Bestandteile des absoluten Pfades bis zum aktuellen Verzeichnis, also *ist*, weglassen. Der so gebildete Name

ja/erstaunlich

heißt „relativer Pfadname“. Beachten Sie, dass er nicht mit / beginnt!

Absolute Pfade haben den Vorteil, dass man sie immer benutzen kann, während relative Pfade nur in Abhängigkeit vom aktuellen Verzeichnis benutzt werden können.

Relative Pfade werden wegen ihrer Kürze bei der Eingabe von Befehlen vorgezogen.

In der folgenden Übersicht werden absolute Pfade benutzt. Das Verzeichnis *spool* aus Abbildung 3.1 heißt dann also */var/spool*.

Den Vorgang des Verfügbarmachens, dessen Durchführung weiter unten genauer beschrieben werden soll, nennt sich „Mounten“. Eine geeignete deutsche Übersetzung dieses Begriffes ist den Autoren nicht bekannt. Es sind Begriffe wie montieren, einbinden, einhängen oder einkleben vorgeschlagen worden, in diesem Buch wird aber der Originalbegriff verwendet.



Mit / beginnende Dateinamen werden auch „absolute Pfade“ genannt

Die wichtigsten Linux-Verzeichnisse im Überblick



Verzeichnis	enthält
/	alle anderen Verzeichnisse und Dateien des System. Es ist die Wurzel des Dateisystems und wird auch „Wurzelverzeichnis“ genannt.
/bin	die wichtigsten, für alle Benutzer verfügbaren Linux-Befehle
/etc	Konfigurations- und Informationsdateien, darunter <i>passwd</i> , in der die Benutzerdaten verwaltet werden



Linux-Dateisystem

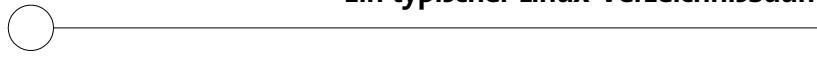


Die wichtigsten Linux-Verzeichnisse im Überblick (Forts.)

Verzeichnis	enthält
/usr	enthält alle wichtigen Programme und Daten, wie etwa das Online-Manual, die nicht zum Booten des Systems benötigt werden. Es kann ein Verzeichnis sein, das sich physisch auf einem anderen Datenträger befindet als das Bootmedium, auf einer CD-ROM etwa oder an einem (entfernten) NFS-Server. In dieser Situation müsste dieser Teil des Dateisystems nach dem Booten erst verfügbar gemacht werden. Für den Fall, dass es dabei zu Problemen kommen sollte, stehen die allernötigsten Befehle in verschiedenen anderen Verzeichnissen, z.B. /bin, zur Verfügung.
/mnt	Hilfsverzeichnis zur Aufnahme von Dateisystemen, die in den Linux-Verzeichnisbaum gemountet werden sollen
/dev	die Gerätedateien
/var	Informationsdateien für das System. Dies sind Tabellen, die von Linux laufend verändert werden, z.B. die Dateien <i>utmp</i> und <i>wtmp</i> , die Informationen über die eingeloggten Benutzer enthalten. (In Standard-UNIX sind diese Tabellen im Wesentlichen in /etc untergebracht.)
/sbin	Befehle zur Systemverwaltung wie <i>adduser</i> , mit dessen Hilfe neue Benutzer definiert werden, <i>fsck</i> zur Überprüfung von Dateisystemen oder <i>shutdown</i> , mit dem das System heruntergefahren werden kann. Programme für den Systemstart wie <i>init</i> , mit dem diverse Systemdienste (z.B. Drucker- und Zeitdienste) initiiert werden, darunter auch <i>getty</i> (ebenfalls in /sbin zu finden). Über <i>getty</i> erfolgt das Anmelden an der Konsole oder an Terminals, die via serieller Leitung mit dem Linux-Rechner verbunden sind.
/lib	die Bibliothek des C-Compilers sowie so genannte „Shared Libraries“, Programmbibliotheken (vereinfacht gesagt Programmteile), die von verschiedenen Anwendungen benutzt werden können. Diese Programmteile werden erst zur Laufzeit in den Arbeitsspeicher geladen, wenn das betreffende Programm sie benötigt. Dadurch und durch die Tatsache, dass verschiedene Programme darauf zugreifen können (<i>shared</i>), wird der Arbeitsspeicher wesentlich entlastet. Sind Sie Windows-Benutzer, kennen Sie sicherlich <i>.dll</i> -Dateien. Diese werden unter Windows entsprechend genutzt.



Ein typischer Linux-Verzeichnisbaum



Die wichtigsten Linux-Verzeichnisse im Überblick (Forts.)

Verzeichnis	enthält
/home	die Heimatverzeichnisse, in denen normalerweise keine Rechte die Arbeit des Benutzers einschränken. Sie sind somit für den normalen Benutzer von existentieller Bedeutung. In größeren Linux-Systemen sollte für ein solches Verzeichnis entsprechender Platzbedarf einkalkuliert werden, da hier die Benutzerdaten abgelegt werden. Häufig wird es dann auf einen Extra-Datenträger ausgelagert, der beim Systemstart separat gemountet wird.
/proc	Informationen über die laufenden Prozesse. Der Befehl <i>ps (process status)</i> liefert seine Angaben auf der Basis des Inhalts dieses Verzeichnisses. Jeder Prozess erhält ein eigenes Unterverzeichnis, in dem die Informationen über ihn gespeichert werden.
/usr/bin	die Befehle, die allen Teilnehmern zur Verfügung stehen sollen. Ein Teil dieser „Benutzerbefehle“ befindet sich aus den oben genannten Gründen (vgl. Eintrag bei /usr) in /bin. Analog existiert in /usr auch ein Verzeichnis /usr/sbin.
/usr/man	das Online-Manual, eine unerschöpfliche Informationsquelle für den User. Viele Befehle und ihre Optionen können aus Platzgründen in diesem Buch nicht oder nicht ausführlich dargestellt werden. Das Online-Manual liefert in solchen Fällen nützliche Zusatzinformationen. Seien Sie aber darauf vorbereitet: Es ist in Englisch abgefasst.
/usr/src	die Quellcodes für Systemprogramme. Im Unterverzeichnis /usr/src/linux findet man den Quellcode des Systemkerns.
/var/tmp	temporäre Dateien, die von Anwendungen aus verschiedenen Gründen während des Programmablaufs angelegt und verändert, zum Programmende aber wieder gelöscht werden. Editoren oder Textverarbeitungen speichern unter Umständen den vom Anwender geschriebenen Text zur Sicherheit in einer temporären Datei. Temporäre Dateien sind kurzlebig und ständiger Änderung unterworfen. Sie sind ein sprechendes Beispiel für die allgemeine Bedeutung des Verzeichnisses /var.

Linux-Dateisystem



Die wichtigsten Linux-Verzeichnisse im Überblick (Forts.)



Verzeichnis	enthält
/var/spool	die Warteschlangen (<i>queues</i>) diverser Spool-Programme. Dies sind Programme, die im Hintergrund arbeitend Aufträge der verschiedenen Benutzer entgegennehmen und in einer vorausseitigen Reihenfolge abwickeln. Da mehrere Benutzer (beinahe) gleichzeitig z.B. einen Druckauftrag an den Druck-Spooler geben können, sprich drucken wollen, muss dieser dafür sorgen, dass sich die einzelnen Aufträge nicht vermischen und etwa am Drucker zwei Texte zu einem – bunt gemischten – werden. Als Hilfsmittel dienen ihm Warteschlangen, in welche die einzelnen Aufträge nacheinander eingereiht und (nach dem Wirkungsprinzip einer FIFO) nach und nach abgearbeitet werden. Beispiele: /var/spool/lpd enthält die Warteschlangen des Druck-Spoolers, /var/spool/mail dient der elektronischen Post, /var/spool/uucp dem Kommunikationsdienst uucp (unix-to-unix-copy), /var/spool/cron der Abwicklung von Zeitdiensten über den cron-Dämonprozess, der in regelmäßigen Abständen Tabellen in /var/spool/cron/crontabs daraufhin überprüft, ob sie Anweisungen enthalten, die zum gegenwärtigen Zeitpunkt auszuführen sind.

Tabelle 3.1



Dies sind einige der Standardverzeichnisse, die Sie in Ihrem Linux-System erwarten dürfen, die Benutzerverzeichnisse unter /home werden natürlich individuelle Namen haben.

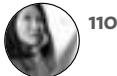
Nachdem Sie nun einen Überblick über das Linux-Dateisystem erhalten haben, wird es Zeit für einige praktische Arbeiten.

Arbeiten im Dateisystem

Im letzten Abschnitt wurde die Bedeutung einiger wichtiger Verzeichnisse beschrieben, und als Nächstes sollen Ihnen nun die Befehle zur Verfügung gestellt werden, mit denen man durch das Dateisystem navigiert und sich Dateien und Verzeichnisse anzeigen lässt.

Verzeichnisse

Nach dem Einloggen befindet man sich gewöhnlich in seinem Heimatverzeichnis. Loggt man als Benutzer *stern* ein, ist dies also /home/stern. DOS-Anwender sind es gewohnt, dass ihnen der Prompt das aktuelle Verzeichnis anzeigt. Dabei ist das aktuelle Verzeichnis dasjenige, dessen Inhalt einem aktuell ohne Aufwand zur Verfügung steht. Befehle, die man eingibt, beziehen sich auf dieses Verzeichnis, es sei denn, man nennt ausdrücklich ein anderes.



Arbeiten im Dateisystem



pwd

Nicht jede Linux-Distribution zeigt im Prompt das aktuelle Verzeichnis. Falls Sie trotzdem wissen möchten, welches Verzeichnis das aktuelle ist, steht Ihnen dafür der Befehl **pwd** (kurz für **p**rint **w**orking **d**irectory) zur Verfügung. Seine Syntax ist einfach

```
pwd
```

In einem weit verzweigten Verzeichnisbaum, wie er unter Linux existiert, sollte man sich immer darüber im Klaren sein, welches das aktuelle Verzeichnis ist. Datenverlust entsteht meist dadurch, dass Dateien in „falschen“ Verzeichnissen gelöscht werden. Insbesondere der Superuser (der Verwalter des Systems) sollte sich dieser Gefahr bewusst sein. Löschen von Dateien in der Wurzel des Dateisystems bedeutet den Verlust des Linux-Kernels!

cd

Zum Navigieren im Verzeichnisbaum kennt Linux den Befehl **cd** (*change directory*). Auch seine Syntax ist denkbar einfach:

```
cd [zielverzeichnis]
```

Beachten Sie bitte, dass, wenn Sie ein Zielverzeichnis angeben, zwischen Befehl und Verzeichnis mindestens ein Trennzeichen (ein Tabulator oder Leerzeichen) stehen muss. Die aus DOS bekannte Befehlssequenz

```
cd..
```

(ohne Leerzeichen) führt zu folgender Fehlermeldung:

```
bash: cd..: command not found
```

Sie besagt, dass Linux, genauer die bash, einen Befehl namens *cd.* (!) nicht finden konnte. Die bash wertet rigoros das erste Wort der Befehlszeile als den Befehl selbst! Sie sollten sich also an das Leerzeichen nach dem Befehlsnamen gewöhnen.

Das Zielverzeichnis kann durch einen gültigen relativen oder absoluten Pfad beschrieben werden. Gültig heißt, das so beschriebene Verzeichnis muss existieren!

Nehmen Sie den Beispiel-Verzeichnisbaum aus *Abbildung 3.1*. Ist das Verzeichnis */usr* das aktive, dann sind gültige Dateinamen *bin*, *man*, *scr*, aber auch */dev*, */var/tmp* oder */var/spool/lpd*. Ein ungültiger Dateiname wäre dagegen *mnt*, obwohl ein Verzeichnis dieses Namens existiert. Beachten Sie jedoch, dass hier ein relativer Pfad (beginnt nicht mit */*) benutzt wurde. Dieser setzt voraus, dass das aktuelle Verzeichnis auf dem Weg zwischen Wurzel (*/*) und *mnt* zu finden sein muss! Das ist aber nicht der Fall. Dieser Sachverhalt lässt sich auch wie folgt beschreiben: *mnt* ist kein Unterverzeichnis von */usr*!



Linux-Dateisystem



Warum sind dann aber sowohl *bin* als auch */bin* gültige Pfade? Nun, es gibt ein Verzeichnis *bin*, das Unterverzeichnis von */usr* ist, und ein Verzeichnis mit dem absoluten Pfadnamen */bin*!

Dass im Dateisystem Verzeichnisse gleichen Namens auftauchen können, stellt für Linux natürlich kein Problem dar. Ausschlaggebend ist, dass der absolute Pfadname eindeutig ist. In diesem Sinne gibt es keine Probleme mit */bin* und */usr/bin*: Die absoluten Pfade sind eindeutig.

Umgekehrt dürfen in einem Verzeichnis keine gewöhnliche Datei und ein Verzeichnis (oder eine Gerätedatei oder eine FIFO ...) mit dem gleichen Namen vorhanden sein. Schließlich sind FIFOs, Gerätedateien, Verzeichnisse, gewöhnliche Dateien, Sockets oder Links nur verschiedene Varianten der „Gattung“ Datei. Es darf in einem Verzeichnis wegen der Eindeutigkeit der absoluten Pfadnamen nie zwei oder mehr Dateien, egal, welcher Gattung, mit demselben Namen geben.

Bisher wurde immer nur der Verzeichniswechsel weg von der Wurzel betrachtet. Jetzt sollen zwei Wege in die andere Richtung aufgezeigt werden. Angenommen, man befindet sich im Verzeichnis */var*, dann gibt es zwei Möglichkeiten, um ins Verzeichnis *uucp* zu wechseln:

`cd spool/uucp`

und

`cd /var/spool/uucp`

Es gibt aber auch zwei Wege, um aus diesem aktuellen Verzeichnis zur Wurzel zu gelangen, nämlich (wie nicht anders zu erwarten):

`cd /`

aber auch

`cd ..`

In jedem Verzeichnis gibt es einen Eintrag `..`! Dieser meint das übergeordnete Verzeichnis, bisweilen „Eltern-“, „Vater-“ oder „Mutterverzeichnis“ genannt. Es befindet sich einen Schritt in Richtung Wurzel vom aktuellen Verzeichnis entfernt. Für */var/spool* ist dies */var*, für */var/spool/uucp* ist es */var/spool* und für */var* eben die Wurzel.

Hinweis



Der Befehl *ls* entspricht dem DOS-Befehl *dir* und wird im nächsten Abschnitt ausführlich besprochen

Falls Sie noch wenig Übung mit relativen und absoluten Paden haben, sollten Sie ein wenig mit *cd* experimentieren. Prüfen Sie dabei mit *pwd*, wo Sie gelandet sind. Erforschen Sie mit *ls*, ob es weitere Ziele, sprich Verzeichnisse gibt. Diese werden in der durch *ls* ausgegebenen Liste durch einen nachgestellten */* angezeigt. Das folgende Beispiel zeigt die Ausgabe eines *ls*-Befehls:



Arbeiten im Dateisystem



Info@	dll/	Include/	sbin/
X11@	doc/	info/	share/
X11R6/	etc/	lib/	spool@
X386@	g++-include@	Local/	src/
adm@	games/	man/	tmp@
bin/	i486-linux/	Openwin/	
dict/	i486-linuxaout/	Preserve@	

Verzeichnisse in dieser Liste sind:

X11R6, bin, dict, dll, doc, etc, games, i486-linux, i486-linuxaout, include, info, lib, local, man, openwin, sbin, share und src.

Nach den obigen Erklärungen zur Verzeichnisstruktur sollten Sie erkennen können, dass die vorliegende Liste den Inhalt des Verzeichnisses */usr* abbildet. Die durch @ abgeschlossenen Namen stellen Links dar, auf deren Bedeutung im übernächsten Abschnitt eingegangen wird.

Eine Besonderheit des Befehls *cd* sei noch erwähnt: Lassen Sie ein Zielverzeichnis in der Befehlszeile weg, so wechselt *cd* automatisch in Ihr Heimatverzeichnis. Etwas genauer: *cd* ersetzt die fehlende Zielangabe durch den Inhalt der Systemvariablen *HOME*.

Über Systemvariablen wie z.B. *HOME* wird Sie das *Kapitel 4 Benutzerumgebung* informieren. Inhalt von *HOME* ist normalerweise das Heimatverzeichnis.



Tipps

mkdir

Das Instrument Verzeichnis für eine geordnete Dateiverwaltung steht natürlich jedem Benutzer zur Verfügung. Man kann, sofern man die nötigen Rechte besitzt, Verzeichnisse anlegen, um darin Daten abzulegen, und man kann auch, die nötigen Rechte vorausgesetzt, Verzeichnisse löschen. Das Erstellen erfolgt mit *mkdir* (*make directory*). Die Syntax ist

```
mkdir [-option] verzeichnis1 [verzeichnis2 ...]
```

Die Verzeichnisse können wieder als absolute oder relative Pfade angegeben werden. Da man sie anlegen will, können sie natürlich nicht existieren, zu beachten ist aber, dass das vorletzte Verzeichnis des (absoluten) Pfadnamens existieren muss. Will man etwa mit

```
mkdir /home/rattle/bin/help
```

das Verzeichnis *help* anlegen, so müssen *bin* und damit *rattle* sowie *home* bereits existieren. Gleiches gilt natürlich für relative Pfade. Angenommen, *ls* zeigt im aktuellen Verzeichnis folgende Dateiliste:

info	kermit	liste	shellscripts/
junk/	kubik	newslist	Start



Ein Hinweis für
DOS-Benutzer: Der
Befehl heißt *mkdir*,
nicht *md*



Linux-Dateisystem



Dann lässt sich ein Verzeichnis *test* mit folgenden korrekten relativen Pfaden einrichten:

```
mkdir junk/test  
mkdir shellscripts/test  
mkdir test
```

Voraussetzung ist natürlich, dass keine Dateien (z.B. Verzeichnisse) namens *test* in *junk* oder *shellscripts* existieren. Nicht zulässig wären folgende Möglichkeiten:

- *mkdir kermit/test*
- *mkdir egal/test*

Im ersten Fall ist *kermit* eine gewöhnliche Datei und kann keine Unterverzeichnisse enthalten, im zweiten Fall existiert *egal* nicht. Beachten Sie, dass das Verzeichnis *egal* nicht automatisch angelegt wird.

Mit der Option *-p* kann man *mkdir* allerdings zwingen, solche nicht existenten Unterverzeichnisse anzulegen.

```
mkdir -p egal/test
```

legt also sowohl *egal* als auch das Unterverzeichnis *test* an.

Beachten Sie noch einen anderen, allgemein interessanten Aspekt der Befehlssyntax:

Die Punkte in der Befehlssyntax deuten an, dass prinzipiell beliebig viele Verzeichnisse in der Befehlszeile angegeben werden können. Dies ist ein Merkmal fast aller Linux-Befehle. Sie können nicht nur *ein* Objekt bearbeiten, in vorliegendem Beispiel ein Verzeichnis erstellen, sondern in einem Arbeitsgang mehrere in einer Liste angegebene. In diesem Fall gilt es zu beachten, dass für *mkdir* (ohne die Option *-p*) die Reihenfolge von Bedeutung ist.

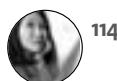
```
mkdir trouble trouble/shooting
```

legt problemlos *trouble* und anschließend *trouble/shooting* an. Vertauscht man bei der Eingabe jedoch die Reihenfolge

```
mkdir trouble/shooting trouble
```

so wird dies einen Fehler provozieren. *shooting* kann nicht angelegt werden, da *trouble* beim ersten Arbeitsgang noch nicht existiert. Das erste anzulegende Verzeichnis der Liste kann nicht erstellt werden! Im Anschluss daran wird aber *trouble* problemlos erzeugt.

Prüfen Sie dies doch einmal in Ihrem Heimatverzeichnis!



Arbeiten im Dateisystem



Eine weitere wichtige Option von *mkdir* ist

`-m mode`

mit deren Hilfe die Rechte, die für das neue Verzeichnis gelten sollen, festgelegt werden können. Im Normalfall werden dafür die durch *umask* (wird im Abschnitt über Rechte näher besprochen) definierten Rechte genommen. *mode* entspricht dabei der oktalen Syntax, die im Zusammenhang mit dem Befehl *chmod* vorgestellt wird. So legt etwa

`mkdir -m 777 newdir`

fest, dass alle Benutzer am neuen Verzeichnis *newdir* alle Rechte haben sollen!

rmdir

Verzeichnisse, die nicht mehr gebraucht werden, löscht man mit *rmdir* (*remove directory*).

Dabei ist zu beachten, dass

- die Verzeichnisse, die man löschen will, leer sind. Ein nicht zu unterschätzender Sicherheitsaspekt. Der Befehl zum Löschen von Dateien (die keine Verzeichnisse sind) bietet weiter reichende Möglichkeiten, deren Gefährlichkeit nicht verkannt werden sollte.
- das zu lösrende Verzeichnis nicht das aktuelle ist. Den Ast, auf dem man sitzt, sollte man nicht absägen. *rmdir*. ist ein Fall höheren Blödsinns!

Die Syntax des Befehls lautet:

`rmdir [-p] verzeichnis1 [verzeichnis2 ...]`

Es können also mehrere Unterverzeichnisse gleichzeitig gelöscht werden, wobei analog Überlegungen wie zu *mkdir* berücksichtigt werden müssen.

So löscht

`rmdir trouble/shooting trouble`

zunächst das (hoffentlich vorhandene und leere) *trouble/shooting* und danach *trouble*. In der Annahme, dass keine sonstigen Dateien in *trouble* und *shooting* existieren, verläuft diese Operation ohne Probleme.

Betrachten Sie dagegen die Zeile

`rmdir trouble trouble/shooting`



Linux-Dateisystem



(beachten Sie genau den Unterschied in der Reihenfolge), so kann in diesem Fall *rmdir* das nicht leere Verzeichnis *trouble* nicht löschen (es enthält ja *shooting*). Den zweiten Auftrag kann *rmdir* allerdings lösen (sofern sich in *shooting* nicht irgendwelche Dateien befinden).

rmdir wird eine Fehlermeldung wegen des nicht gelöschten Verzeichnisses *trouble* ausgeben. *trouble/shooting* wird unter den gerade genannten Bedingungen ohne Probleme gelöscht.

Die Option *-p* hat ähnliche Wirkung wie bei *mkdir*. Allerdings muss auch hier umgekehrt „gedacht“ werden.

Angenommen, es existieren die Verzeichnisse *check* und *check/fd* im aktuellen Verzeichnis und beide sind ansonsten leer.

`rmdir -p check`

erzeugt dann eine Fehlermeldung, die besagt, dass *check* nicht leer ist. Dagegen löscht

`rmdir -p check/fd`

zunächst *fd* und dann *check*.

Beachten Sie zu dem Thema Löschen von Verzeichnissen mit Inhalt auch den Befehl *rm* mit seiner Option *-r*.

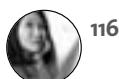
Der Befehl *ls*

Sollten Sie die obigen Beispiele durchgetestet haben, werden Sie auch den *ls*-Befehl eingesetzt haben, um sich im System ein wenig zu orientieren.

Einige der Autoren sind mit dem Betriebssystem MS-DOS „groß geworden“. Sie haben als Dozenten versucht, eine große Zahl ihrer Kursteilnehmer mit mehr oder weniger intelligenten Aufgaben in die Welt der Dateien und Verzeichnisse einzuführen. Dabei haben sie immer und immer wieder auf die elementare Bedeutung des Informationsbefehls *dir* hingewiesen. *ls* kommt eine ähnlich überragende Bedeutung zu.

Wenn weiter oben angemerkt wurde, dass man beim Navigieren durch das System den Befehl *pwd* als Orientierungshilfe einsetzen sollte, so muss dies nun dahingehend ergänzt werden, dass man sich neben dem Namen des Verzeichnisses auch dessen Inhalt ansehen sollte.

Der Befehl *ls* bietet dazu eine Reihe interessanter Möglichkeiten, mit deren Hilfe man auch Details des Dateisystems kennen lernen kann. *ls* zeigt in seiner einfachsten Variante lediglich die Dateinamen an. Mit den entsprechenden Optionen zeigt er dem erfahrenen User aber auch nahezu alle Einzelheiten, die für Linux-Dateiverwaltung von Belang sind.





Die Verwaltungsinformationen einer Datei

Als Benutzer einer Datei ist man meist nur an deren Inhalt interessiert. An diesen gelangt man in der Regel, wenn man den Dateinamen kennt sowie einen Linux-Befehl, der den Dateiinhalt anzeigt. Häufig wird dies nicht reichen. Dann benötigt man zusätzlich das Anwendungsprogramm, mit dem die Datei erstellt wurde, um sich ihren Inhalt ansehen zu können. Der Dateiname ist dabei nun nicht mehr integraler Bestandteil der Datei, sondern eine Kennzeichnung, die dem Anwender dient, dem Betriebssystem klar zu machen, welche Daten er einsehen, verändern oder beispielsweise löschen möchte.

Der Dateiname ist das Bindeglied zwischen Benutzer und Betriebssystem, um als Benutzer an den Dateiinhalt zu kommen. Das Betriebssystem verwaltet darüber hinaus weitere Informationen über eine Datei, um diese korrekt bearbeiten zu können. Eine Reihe dieser Informationen, im Folgenden „Verwaltungsinformationen“ genannt, sind dabei auch für den Benutzer interessant. Sie sollen an dieser Stelle kurz erläutert werden.

Überlegen Sie doch einmal, welche Verwaltungsinformationen außer dem Dateinamen Ihnen aus dem Betriebssystem bekannt sind, das Sie bisher benutztten.

Wollen Sie den Inhalt einer Datei vom Betriebssystem bereitstellen lassen, um ihn in irgendeiner Form zu bearbeiten, müssen Sie sich über den Dateinamen legitimieren, um überhaupt auf die Daten zugreifen zu können.

In Linux wird eine Datei aber nicht direkt über ihren Dateinamen identifiziert. Eine Datei kann mehrere Namen haben, sofern die entsprechenden absoluten Pfadnamen nicht identisch sind. Sie kann gleichzeitig `/home/muster/grumble` wie auch `/usr/bin/chgr` heißen. Einer der Vorteile dieser aus den gängigen Einplatzsystemen nicht bekannten Regel ist offenbar, dass man mehrfach Zugriff auf diese Datei hat.

Die interne Verwaltung einer Linux-Datei erfolgt über eine Nummer, die so genannte „I-Node-Nummer“. Diese bildet zusammen mit dem konkreten Dateinamen den Zugang zu den restlichen Verwaltungsinformationen einer Datei. Die Kombination konkreter Dateiname und I-Node-Nummer nennt man auch „Link“.

Zum Begriff „I-Node“: *Node* bedeutet *Knoten*, man könnte auch sagen „ein bestimmter Punkt“ im Dateisystem. *I* steht für *Information*.



Hinweis



Linux-Dateisystem



Weitere Verwaltungsinformationen über eine Datei sind:

- der Dateityp (gewöhnliche Datei, Verzeichnis, Gerätedatei ...)
- die Rechteabelle, die den Zugriff auf die Datei regelt
- die Anzahl der Links, man könnte auch vereinfacht sagen die Anzahl unterschiedlicher Dateinamen
- der Dateibesitzer
- die Besitzergruppe
- die Größe der Datei in Byte
- das Datum der letzten Änderung
- das Datum der letzten Benutzung
- das Datum der letzten Änderung eben dieser Verwaltungsinformationen. Dieses Datum wird häufig als „Datum der Erstellung“ bezeichnet. Dies ist objektiv falsch. Weiter unten wird eine Option des *ls*-Befehls gezeigt, mit der man dieses Datum abrufen kann. Sie können dann testen, dass dies kein Erstellungsdatum ist, indem Sie eine Datei editieren. Dabei sollte diese größer oder kleiner werden, was die Verwaltungsinformation „Größe der Datei in Byte“ ändert. Danach können Sie überprüfen, dass sich das so genannte „Erstellungsdatum“ geändert hat, ohne dass die Datei erstellt worden wäre. Die Annahme, es sei ein Erstellungsdatum, ist maßgeblich dadurch beeinflusst, dass die Option, mit der *ls* dieses Datum angeigt, *-c (create)* ist.
- Angaben über den physikalischen Ort auf dem Datenträger, an dem die Datei zu finden ist

Die Größe in Byte entfällt bei den Gerätedateien. Stattdessen werden zwei Nummern gespeichert: die Hauptgerätenummer (*major device number*) und die Hilfsgerätenummer (*minor device number*). Mit

```
ls -l /dev/hda1
```

erhalten Sie etwa folgende Informationen:

```
brw-rw---- 1 root disk 3, 1 Nov 12 1995  
/dev/hda1
```

Die Gerätenummern stehen vor dem Datum, dabei ist die Hauptgerätenummer die 3 und die Hilfsgerätenummer die 1.

Über sie erfolgt der Zugriff auf den Gerätetreiber im Kernel. Benutzt ein Prozess eine Gerätedatei, werden über diese Datei die Gerätenummern ermittelt und darüber wiederum der Treiber im Systemkern angesprochen. Dort wird eine Tabelle



Arbeiten im Dateisystem



gepflegt, in der die diversen Treiber verwaltet werden. Die Hauptgerätenummer verweist auf den Treiber selbst, die Hilfsgerätenummer gibt dem Treiber zusätzliche Informationen, etwa auf welchen I/O-Port und damit auf welches konkrete Gerät zugegriffen werden soll.

So wird in vorliegendem Beispiel mithilfe des Treibers 3 auf die Festplatten zugegriffen. Die Hilfsgerätenummer hilft diesem Treiber, zu unterscheiden, auf welche Platte bzw. auf welche Partition zugegriffen werden soll.

Ob nun gewöhnliche Datei, Verzeichnis, Gerätedatei oder FIFO, verwaltungstechnisch besteht eine Datei aus drei Teilen: dem Inhalt oder dem Verweis auf den Treiber, dem Link (Dateiname und I-Node-Nummer) und den restlichen Verwaltungsinformationen.

Den Inhalt einer Datei kann man sich mit dem Befehl *cat* anzeigen lassen. Die Datei *.profile* in Ihrem Heimatverzeichnis können Sie etwa mit

```
cat .profile
```

einsehen. Die oben genannten Verwaltungsinformationen lassen sich fast ausnahmslos mit *ls* unter Verwendung seiner diversen Optionen anzeigen.

***ls* und seine Optionen**

Die Befehlssyntax von *ls* ist

```
ls [ -optionen] [ pfade1[ pfade2 ... ]]
```

Es können, wie schon kennen gelernt, mehrere Pfade angegeben werden. Ohne Pfadangabe wird das aktuelle Verzeichnis als Pfad angenommen.

Benutzt man *ls* ohne *option*, zeigt der Befehl nur die Dateinamen sowie eine Kennung über die Art der Datei. Dies könnte etwa so aussehen (die Eingabe ist unterstrichen dargestellt):

```
$ ls  
FIFO|bin/f*listestuff/workdir@
```

liste ist eine gewöhnliche Datei und als solche nicht gekennzeichnet. *FIFO* ist eine Named Pipe, was am nachgestellten Pipezeichen | abzulesen ist. *bin* und *stuff* sind Unterverzeichnisse, wie man unschwer am nachgestellten / erkennen kann. An ausführbare Dateien hängt *ls* einen * an, also ist *f* eine ausführbare Datei (im konkreten Fall ein Shell-Skript, aber natürlich werden auch Programmdateien mit dem Asterisk * gekennzeichnet). *workdir* schließlich ist ein Link, was *ls* durch @ darstellt.

Daneben erhalten u.a. auch Gerätedateien und Archive eine besondere Kennung.



Linux-Dateisystem

Navigieren Sie durch das System und machen Sie sich dabei mit den Kennungen des Befehls *ls* vertraut!

Beachten Sie dabei, dass diese Kennungen nicht zum Dateinamen gehören! Aus diesem Grund sollten solche Zeichen am Ende eines Dateinamens vermieden werden. Wer allerdings über einen Farbmonitor verfügt, wird schon bemerkt haben, dass das Farbspiel des Befehls *ls* nicht Selbstzweck, sondern eine weitere Kennung der verschiedenen Dateitypen ist.

Eine ausführliche Information über Dateien erhält man mithilfe des Schalters *-l* (*long*).

Beispiel:

```
$ ls -l
total 4
prw-r--r-- 1 rattle  users      0 Nov 19 17:45 FIFO|
drwxr-xr-x 2 rattle  users  1024 Mar 22 1994 bin/
-rwxr--r-- 1 rattle  users     23 Dec  8 23:30 f*
-rw-r--r-- 1 rattle  users    184 Dec  8 07:38 liste
drwxr-xr-x 2 rattle  users  1024 Dec  8 23:28 stuff/
```

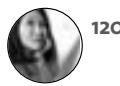
Zur Erklärung der einzelnen Informationen dient der vorletzte Eintrag, die Informationen über die Datei *liste*. Etwas unüblich erscheint zunächst, dass der Dateiname, hier *liste*, am Ende der Zeile erscheint.

Schauen Sie sich die übrigen Informationen von links nach rechts an. Das erste Zeichen kennzeichnet den Dateityp. Die Bedeutung der einzelnen Zeichen entnehmen Sie bitte der Tabelle 3.2.

Kennzeichen der Dateiarten bei *ls -l*

Kennzeichen	Bedeutung
-	Gewöhnliche Datei
d	Verzeichnis (<i>directory</i>)
c	Zeichenorientierte Gerätedatei (<i>character device</i>)
b	Blockorientierte Gerätedatei (<i>block device</i>)
p	FIFO (<i>named pipe</i>)
l	(symbolischer) Link
s	Socket

Tabelle 3.2



Arbeiten im Dateisystem



Zeichenorientierte Gerätedateien bedienen Geräte, die Daten Zeichen für Zeichen bearbeiten. Als Beispiel können Sie die Terminaldateien `/dev/ttyxx` oder `/dev/console` betrachten. Daten von Tastatur oder für den Bildschirm werden zeichenweise bearbeitet.

Blockorientierte Gerätedateien bedienen z.B. Festplatte und Floppy, deren Daten blockweise (in der Regel 512 Byte) gelesen und geschrieben werden, und unterstützen insofern wahlfreien Datenzugriff.

Die Ökonomie besteht halt darin, dass für die Information Dateityp beim Speichern kein Extra-Byte verbraucht wird. Im Zeitalter großer Massenspeicher sind solche Überlegungen zur Modellierung von Datenstrukturen bei Programmierern leider nicht mehr so häufig anzutreffen.

Dem Kennzeichen für die Dateiart folgt eine neunstellige Zeichenkombination, welche die zur Datei gehörenden Rechte beschreibt, die Rechtetabelle. Beachten Sie bitte, dass Rechtetabelle und Dateityp nichts miteinander gemein haben. Sie werden aus ökonomischen Gründen in einem gemeinsamen Datenfeld gespeichert. Ein inhaltlicher Zusammenhang zwischen beiden besteht nicht.

Die nächste Information (für die Beispieldatei `liste` ist das die Zahl 1) betrifft die Anzahl der Links einer Datei. Die Datei `liste` existiert in diesem Dateisystem also nur mit einem Namen.

Die folgenden beiden Namen nennen den Dateibesitzer (hier `rattle`) und die Besitzergruppe (hier `users`). Beide stehen in engem Zusammenhang mit der Rechtetabelle, im Gegensatz zum Dateityp.

Nach den beiden Namen wird die Größe (184) in Byte angezeigt. Diese gibt, wie auch in anderen Systemen, nicht den tatsächlichen Speicherbedarf wieder. Dateien, ausgenommen natürlich Gerätedateien, werden in Blöcken gespeichert, die immer ein Vielfaches von 512 Byte groß sind. Die tatsächliche Größe der Blöcke hängt dabei wesentlich vom gewählten Dateisystem ab. Besitzt das Dateisystem beispielsweise eine Blockgröße von 2.048 Byte, so wird auch eine nur 184 Byte große Datei 2 KByte Speicherplatz in Anspruch nehmen; der „Rest“ dieses Blocks bleibt ungenutzt.

○—————
Ermitteln Sie einmal mittels `ls -l` / die Anzahl der Links von `/usr`

Als Dateiinformationen folgen der Größe in Byte das Datum (8. Dezember des laufenden Jahres) und die Uhrzeit (7:38) der letzten Änderung. Bei Dateien, die älter als ein halbes Jahr sind (etwa beim Verzeichnis `bin`), wird nur das Datum, dann aber mit Jahresangabe ausgegeben.

Die letzte Angabe ist, wie schon erwähnt, der Dateiname.

`ls -l` zeigt also recht ausführliche Informationen, aber nicht alle. Die *I-Node-Nummer* zum Beispiel ist nur über die Option `-i` verfügbar.

`ls -il FIFO`

liefert etwa folgende Informationen:



Linux-Dateisystem



```
92784 prw-r--r-- 1 rattle users          0 Nov 19  
17:45 FIFO|
```

Beachten Sie dabei, dass die Option `-i` nicht die Option `-l` ersetzt.

```
ls -i FIFO
```

liefert lediglich

```
92784 FIFO|
```

Dieses Verhalten sollten Sie bei der Benutzung von Optionen stets beachten. Im Zusammenhang mit `-l` ist es natürlich besonders auffällig.

Weiter oben wurde bereits besprochen, dass Dateien, deren Namen mit einem Punkt beginnen, von den Dateibefehlen nur berücksichtigt werden, wenn der Name ausdrücklich in der Liste der zu bearbeitenden Dateien auftaucht. `ls` bildet keine Ausnahme von dieser Regel. Da sich der Name solcher „versteckten“ Dateien beim Durchsuchen eines Dateibaums aber erst mittels `ls` erschließt, kennt dieser Befehl zwei Optionen zu deren Anzeige, nämlich `-a` und `-A`.

`-a` zeigt alle `.`-Dateien mit an, `-A` alle außer den „selbstverständlichen“ Dateien `.` und `..`!

Im Heimatverzeichnis eines gewöhnlichen Benutzers sollte

```
ls -a
```

etwa folgende Ausgabe erzeugen:

```
./           .kermrc   .profile    bin/      stuff/  
../           .less      .term/     lhelp  
.bash_history .lessrc    FIFO|       liste
```

Bei gleicher Konstellation dagegen `ls -A`:

```
.bash_history .lessrc    FIFO|       liste  
.kermrc       .profile   bin/      stuff/  
.less         .term/     lhelp
```

Punkt-Dateien

Der Zweck der Dateien `.` und `..` wurde bereits zu Anfang des Kapitels erläutert. Aber auch die anderen Punkt-Dateien sollen in ihrer Bedeutung kurz erläutert werden, spielen sie doch eine nicht unerhebliche Rolle bei der täglichen Arbeit der meisten Benutzer.



Arbeiten im Dateisystem

- .profile oftmals auch *.bash_profile* ist eine Startdatei für die bash, analog zur Datei *autoexec.bat* unter MS-DOS. Sie ist ein Shell-Skript, das bei jedem Start der bash ausgeführt wird. Sie enthält Befehle zur Konfiguration der Systemumgebung, die der Benutzer nach dem Einloggen vorfinden soll. Im Wesentlichen sind dies die Bereitstellung eines Suchpfades und eines Prompts, der die Bereitschaft der bash zur Entgegennahme von Befehlen signalisiert.
- *.bash_history* dient der bash zur Speicherung der zuletzt eingegebenen Befehle. Während einer Sitzung werden die eingegebenen Befehle von der bash registriert und sind in einem Mechanismus analog zum *DOSKEY* unter MS-DOS mittels Cursor-tasten wiederhol- und editierbar. Darüber hinaus werden die Befehle einer Sitzung beim Ausloggen in der Datei *.bash_history* gespeichert. Gegebenenfalls werden die Daten an bereits vorhandene angehängt, so dass insgesamt die Befehle sämtlicher bash-Sitzungen reproduzierbar sind.
- *.less* und *.lessrc* dienen der Konfiguration des Dateiausgabe-Utilities *less*, das weiter unten besprochen werden soll.
- *.term* ist ein Verzeichnis und enthält Informationen (in der Regel ein Shell-Skript) namens *.termrc* zur Konfiguration von Terminals, die direkt oder mit seriellen Leitungen an der Konsole angeschlossen sind.
- Über *.kermrc* lässt sich ein Datenübertragungsprogramm einstellen.

—○ Daher wird der Prompt auch Bereitschaftszeichen genannt

Wie Sie sehen, sind all diese Dateien von erheblicher Bedeutung, werden aber einen „normalen Benutzer“ zunächst wenig interessieren. Der Punkt ist, wie hier andeutungsweise zu sehen ist, nicht nur als Schutzfunktion gedacht, er dient auch dazu, wichtige, aber den Benutzer nicht direkt interessierende Informationen zu unterdrücken.

Falls Sie praktisch versucht haben, sich die *I-Node-Nummer* der Dateien . und .. in der Wurzel anzeigen zu lassen, werden Sie möglicherweise folgenden Versuch unternommen haben:

```
ls -i /.*
```

Dies führt zu einer zunächst überraschenden Fülle an Informationen, was nicht daran liegt, dass in der Wurzel besonders viele Dateien des Musters .* zu finden sind. Ursache ist die Behandlung der Argumente durch den Befehl *ls*.

ls erwartet als Argument Pfadangaben. Endet der Pfad mit einem Verzeichnis, wird *ls* zunächst dessen Inhalt prüfen und Informationen über die Dateien in diesem Verzeichnis liefern. In allen anderen Fällen werden Informationen über die Datei selbst angezeigt.



Linux-Dateisystem

Das nebenstehende Beispiel können Sie in Ihrem Heimatverzeichnis leicht mithilfe von `mkdir` und `echo text > datei` nachvollziehen

Angenommen, im aktuellen Verzeichnis gebe es eine gewöhnliche Datei namens `testdatei` und ein Unterverzeichnis namens `testverz`. `testverz` enthalte seinerseits die Dateien `test1` und `test2`. Dann kann sich folgende Befehlsabfolge ergeben (Ihre Eingaben sind unterstrichen):

```
$ ls testdatei  
testdatei  
$ ls testverz  
test1 test2
```

Im ersten Fall – das Argument ist eine Datei – wird nur deren Name angezeigt (das ist nicht besonders sinnig, wenn man ihn bereits kennt, das Beispiel soll aber lediglich die Arbeitsweise von `ls` demonstrieren). Im zweiten Fall – das Argument ist nun ein Verzeichnis – erhält man die Namen der darin gespeicherten Dateien.

Vor diesem Hintergrund wird verständlich, warum

```
ls -i ./.*
```

sehr viele Information liefert. `ls` erhält den Auftrag, Informationen über alle Dateien in der Wurzel zu liefern, deren Name mit `.` beginnt. Es wird dabei treu und brav zunächst alle Informationen über entsprechende „Nichtverzeichnisse“ anzeigen (hier also *I-Node-Nummer* und Dateiname). Sodann werden alle entsprechenden Verzeichnisse auf ihren Inhalt hin überprüft, und die Informationen darüber werden aufgelistet.

Nun sind aber `.` und `..` zwei solche Verzeichnisse. `.` ist das aktuelle Verzeichnis in der Wurzel – also die Wurzel selbst. Sollten Sie die *I-Node-Nummern* von `.` und `..` bereits verglichen haben, so wissen Sie, dass beide Dateien identisch sind! Alles in allem: `ls -i ./.*` liefert unter anderem zweimal Informationen über alle Dateien in der Wurzel.

Unter Umständen will man jedoch nichts über den Inhalt eines Verzeichnisses wissen, sondern eine Angabe über das Verzeichnis selbst, z.B. dessen *I-Node-Nummer*. In diesen Fällen ist der Schalter `-d` sehr nützlich. Er unterdrückt die soeben geschilderte Behandlung von Verzeichnissen durch den Befehl `ls` und lässt ihn Verzeichnisse wie alle anderen Dateien bearbeiten, d.h., es wird nicht der Inhalt des Verzeichnisses untersucht, es werden nur die (gewünschten) Angaben über das Verzeichnis angezeigt.

Eine elegante Methode, die *I-Node-Nummern* von `.` und `..` in der Wurzel zu ermitteln, wäre

```
ls -id ./ ./..
```

Die Angabe von `.` und `..` in der Befehlszeile bewirkt in Verbindung mit dem Schalter `-d`, dass nur Informationen über diese beiden Dateien ausgegeben werden, `-i` veranlasst als Ausgabe neben dem Dateinamen die gewünschte *I-Node-Nummer*.

Arbeiten im Dateisystem



Das Ergebnis sollte dabei wie folgt aussehen:

```
$ ls -id ./ ../ ↵  
2 ./ 2 ../
```

Beide „Dateien“ sind Verzeichnisse (daher das nachgestellte Zeichen /), beide haben die *I-Node-Nummer* 2!

Jokerzeichen

Da nun schon häufiger über Joker gesprochen wurde, seien die beiden wichtigsten hier in ihrer Wirkungsweise kurz erklärt:

Allgemein verkürzen Joker die Eingabe von Befehlen. Will man mehrere Dateien mit einem Befehl gleichzeitig bearbeiten, kann man sie natürlich Name für Name in der Befehlszeile unterbringen, wie es etwa das letzte Beispiel zeigt. In vielen Fällen ist das zu mühselig. Sind die Namen „ähnlich“, können sie bisweilen mithilfe der Jokerzeichen gemeinsam angesprochen werden.

Will man sich z.B. alle Dateien des aktuellen Verzeichnisses anzeigen lassen, die mit der Buchstabenfolge *test* beginnen, dann geschieht dies mithilfe des Jokers * in der Form

```
ls test*
```

Der Joker * ist ein Platzhalter für eine beliebige Zeichenfolge innerhalb eines Dateinamens! Für DOS-Benutzer ist dies nichts Neues, dennoch soll auf zwei wichtige Unterschiede hingewiesen werden, die durch zwei Beispiele illustriert werden sollen:

Man beachte aber die Problematik mit Dateien, deren Name mit . beginnt!

```
ls a*
```



Hinweis

zeigt, falls sie existieren, auch Dateien mit Namen wie *abc.txt* oder *alpha.doc*.

```
ls a*z
```

zeigt dagegen nur Dateien, deren Namen mit a beginnen und mit z enden. Eine solche Konstruktion ist unter DOS nicht möglich (wohl aber unter Windows 95 und NT) – das z würde einfach ignoriert werden, da der * sämtliche Zeichen ab der Position, die er einnimmt, ersetzt.

Das ? als Joker hat unter Linux die gleiche Bedeutung wie unter DOS: Es ersetzt ein einzelnes Zeichen.

Ein weiterer nützlicher Schalter ist -R. Mit seiner Hilfe lässt sich die Verzeichnisstruktur eines Linux-Systems erkunden. ls zeigt nun nicht nur den Inhalt eines Verzeichnisses an, sondern auch den der Unterverzeichnisse und Unterunterverzeichnisse und so fort.

125



Linux-Dateisystem

Hinweis



R steht für rekursiv, was man annähernd mit „wiederholend zugreifen“ übersetzen könnte. Denken Sie etwa an die Puppe in der Puppe in der Puppe ..., deren Inhalt Sie kennen lernen wollen.

Eine Struktur wie im obigen Beispiel vorausgesetzt, ergibt dann folgende Ausgabe:

```
$ ls -R←
testdatei      testverz/
testverz:
test1          test2
```

In Tabelle 3.3 finden Sie die Optionen, bisweilen auch „Schalter“ genannt, von ls im Überblick.

Die Optionen von ls

Optionen	Bedeutung
-a	Anzeige aller Dateien, auch derjenigen, die mit . beginnen
-b	oktale Darstellung nicht druckbarer Zeichen im Dateinamen (wichtiger Notagel, wenn ls „merkwürdiges“ Verhalten zeigt)
-c	sortiert zusammen mit -t Dateien nach dem Datum der letzten Änderung der I-Node (Verwaltungsinformationen). Standard-sortierfolge ist nach ASCII-Zeichen. Die Anzeige des entsprechenden Datums erfolgt zusammen mit -l.
-d	Verzeichnisse in der Liste der Argumente werden wie andere Dateien auch behandelt. Unterdrückung der Durchsuchung des Inhalts von Verzeichnissen.
-e	zusammen mit -l Ausgabe von Datum und Uhrzeit
-f	unsortierte Ausgabe (Dateien werden in der Reihenfolge angezeigt, in der sie im Verzeichnis eingetragen sind). Beinhaltet automatisch -a.
-g	existiert nur aus Kompatibilitätsgründen, wird ignoriert
-i	Anzeige der I-Node-Nummer vor dem Dateinamen
-k	zusammen mit -l Anzeige der Dateigröße in Kilobyte (aufgerundet) vor dem Dateinamen (siehe auch -s)
-l	Anzeige ausführlicher Dateiinformationen. Viele Schalter haben nur in Zusammenhang mit -l eine Bedeutung, da sie nicht automatisch die Information anzeigen, sondern nur vorbereiten (z.B. -e oder -k).
-m	Ausgabe der Dateinamen als durch Komma getrennte Liste



Arbeiten im Dateisystem



Die Optionen von ls (Forts.)

Optionen	Bedeutung
-n	ersetzt (in Verbindung mit -l) die Namen von Benutzer und Gruppe durch die entsprechenden Nummern (UID und GID)
-o	keine farbliche Kennzeichnung der Dateitypen
-p	Kennzeichnung diverser Dateitypen durch Anhängen von Sonderzeichen. Ausführbare (gewöhnliche) Dateien werden im Gegensatz zu -F nicht gesondert gekennzeichnet.
-q	nicht druckbare Zeichen im Dateinamen werden als ? dargestellt (vgl. auch -b)
-r	Ausgabe in umgekehrter Sortierreihenfolge
-s	zusammen mit -l Angabe der Dateigröße in Blocks vor dem Dateinamen. Ist die Umgebungsvariable <i>POSIXLY_CORRECT</i> definiert, ist die Blockgröße 512 Byte, ansonsten ein Kilobyte (vgl. auch -k).
-t	sortiert Dateien nach dem Datum der letzten Änderung. Standardsortierfolge ist nach ASCII-Zeichen. Die Anzeige des Datums erfolgt zusammen mit -l.
-u	sortiert zusammen mit -t Dateien nach dem Datum der letzten Benutzung. Standardsortierfolge ist nach ASCII-Zeichen. Die Anzeige des entsprechenden Datums erfolgt zusammen mit -l.
-w <i>zahl</i>	legt die maximale Breite (<i>width</i>) der Ausgabezeile auf <i>zahl</i> fest. Wird durch -l aufgehoben.
-x	spaltenweise Ausgabe der Dateinamen, im Gegensatz zu -C jedoch waagerecht geordnet
-A	Anzeige aller Dateien außer . und ..
-B	keine Anzeige von Backup-Dateien (Enden mit ~)
-C	spaltenweise Ausgabe der Dateinamen, senkrecht geordnet. (Dies ist im Übrigen die Standardeinstellung. Sie wird durch -l aufgehoben.)
-D	erzeugt eine für einen bestimmten <i>emacs</i> -Modus geeignete Ausgabe
-F	Kennzeichnung diverser Dateitypen durch Anhängen von Sonderzeichen: / für Verzeichnisse, * für ausführbare Dateien oder @ für Links, für FIFOs und = für Sockets
-G	unterdrückt zusammen mit -l die Anzeige der Gruppe

Linux-Dateisystem



Die Optionen von *ls* (Forts.)



Optionen	Bedeutung
<i>-l Muster</i>	unterdrückt die Ausgabe von Informationen über Dateien, deren Name Muster entspricht. Muster kann Joker enthalten. Der Schalter kann mehrfach (mit unterschiedlichen Mustern) benutzt werden. Beachten Sie, dass auch hier Verzeichnisse besonders behandelt werden.
<i>-L</i>	zeigt im Zusammenhang mit <i>-l</i> und <i>-i</i> bei symbolischen Links die Informationen über die Originaldatei und nicht die Daten des symbolischen Links an
<i>-N</i>	Dateien mit Sonderzeichen im Namen werden nicht durch Anführungszeichen gekennzeichnet (Voreinstellung ")
<i>-Q</i>	Dateien mit Sonderzeichen im Namen werden durch Anführungszeichen gekennzeichnet. Die Sonderzeichen selbst werden in der aus der Programmiersprache C bekannten Form dargestellt.
<i>-R</i>	rekursive Anzeige, d.h., es werden nicht nur Verzeichnisse nach ihrem Inhalt durchsucht, sondern auch darin enthaltene (Unter-)Verzeichnisse. Nicht zusammen mit <i>-d</i> anwendbar. Eine mit der Anzeige des DOS-Befehls <i>tree</i> vergleichbare Ausgabe des Verzeichnisbaums ohne sonstige Dateien ist nicht verfügbar.
<i>-S</i>	Sortierung der Dateinamen nach Größe
<i>-T Zahl</i>	definiert den Abstand von Tabstopps (Tabulatoren)
<i>-U</i>	unsortierte Ausgabe (im Gegensatz zu <i>-f</i> wird nicht der Schalter <i>-a</i> aktiviert)
<i>-X</i>	sortiert die Ausgabe nach Dateiendung. Erstes Sortierkriterium ist der Namensteil nach dem letzten Punkt im Namen. Z.B. werden Dateien, die mit .c enden, vor Dateien angezeigt, die mit .o enden. Dateien „ohne Endung“ werden zuerst angezeigt.
<i>-1</i>	Ausgabe der Dateinamen in nur einer Spalte (war in älteren UNIX-Systemen voreingestellt)
<i>-help</i>	Ausgabe einer Kurzhilfe, welche die Form und die Schalter von <i>ls</i> erläutert (beachten Sie das doppelte Minuszeichen)
<i>-version</i>	gibt Informationen über die Version von <i>ls</i> (beachten Sie das doppelte Minuszeichen)

Tabelle 3.3





Daneben gibt es noch eine Reihe von Alternativschaltern, die anstelle der in *Tabelle 3.3* genannten benutzt werden können, wie etwa der Schalter `-tabsize=Zahl`. Solche Schalter werden stets mit doppeltem Minuszeichen eingeleitet. Der soeben genannte `-tabsize=Zahl` ist die Alternative zu `-T`.

Nähere Einzelheiten entnehmen Sie bitte der Onlinehilfe.

Der Befehl `ls` ist sehr informativ. Deshalb wurde er hier in aller Ausführlichkeit dargestellt. Im weiteren Verlauf werden bei einer Befehlsbeschreibung nur die wichtigsten Optionen aufgeführt.

Dateien

Zu Beginn dieses Kapitels wurde darauf hingewiesen, dass es sehr viele Befehle zur Bearbeitung von Textdateien gibt. Ein einfaches Beispiel dafür ist `cat`.

cat

Der Befehl `cat` ist ein vielfältig einsetzbarer Ausgabebefehl. Seine Syntax lautet:

```
cat [ -option] [ datei1[ datei2 ...]]
```

Als einfaches Beispiel für die Benutzung kann man sich den Inhalt der Datei `/etc/passwd` ansehen und somit Informationen über die Benutzer im System sammeln.

```
$ cat /etc/passwd
root::0:0:root:/root:/bin/bash
bin:*:1:1:bin:/bin:
daemon:*:2:2:daemon:/sbin:
adm:*:3:4:adm:/var/adm:
lp:*:4:7:lp:/var/spool/lpd:
sync:*:5:0:sync:/sbin:/bin/sync
shutdown:*:6:0:shutdown:/sbin:/sbin/shutdown
halt:*:7:0:halt:/sbin:/sbin/halt
mail:*:8:12:mail:/var/spool/mail:
news:*:9:13:news:/usr/lib/news:
uucp:*:10:14:uucp:/var/spool/uucppublic:
operator:*:11:0:operator:/root:/bin/bash
games:*:12:100:games:/usr/games:
man:*:13:15:man:/usr/man:
postmaster:*:14:12:postmaster:/var/spool/mail:/bin/bash
nobody:*:-1:100:nobody:/dev/null:
ftp:*:404:1::/home/ftp:/bin/bash
guest:epDR3g4iRX3m6:405:100:guest:/dev/null:/dev/null
rattle:l/vf19fumtNV2:501:100:Guy
Rattlestern:/home/rattle:/bin/bash
michi::502:100:M.A.:/home/michi:/bin/bash
steffi::503:100:M.A.:/home/steffi:/bin/bash
```



Linux-Dateisystem



```
norbert::504:100:M.A.:/home/michi:/bin/bash
dirk::505:100:M.A.:/home/dirk:/bin/bash
filizz::506:100:M.A.:/home/phyllis:/bin/bash
claudia::507:100:M.A.:/home/claudia:/bin/bash
doris::508:100:Dodo:/home/dodo:/bin/bash
```

cat liest den Inhalt der Datei und gibt ihn am Bildschirm (Standardausgabe) aus.

Bei der Syntaxbeschreibung fällt sofort eine Besonderheit auf, die Ihnen im Folgenden bei der Ausgabe von Dateien immer wieder begegnen wird: *cat* kann ohne Argument benutzt werden. All diese Befehle sind so konzipiert, dass sie innerhalb von Pipelines eingesetzt werden können. Dabei wird das Argument *datei* schlicht und einfach durch die Ausgabedaten eines Befehls ersetzt.

An dieser Stelle erweist sich das Linux-Dateikonzept als nützlich, das Geräte wie Dateien behandelt. Die Standardein-/ausgabe, Tastatur und Bildschirm also, können gewöhnliche Dateien ersetzen oder durch sie ersetzt werden. Dies gilt nicht nur für Pipelines, sondern auch für einfache Befehlszeilen.

Die konsequente Umsetzung dieser Idee kann an *cat* verdeutlicht werden. Gibt man es ohne Argument ein, verwendet es stattdessen die Standardeingabe, sprich die Tastatur. Nach Eingabe von

```
cat
```

wird also darauf gewartet, dass etwas über die Tastatur eingegeben wird. Wie lange kann/darf man eingeben? Gewöhnlich arbeitet *cat* eine Datei bis zu deren Ende ab. Dies gilt auch in vorliegendem Fall. Liest *cat* von der Tastatur das Dateiendezeichen, so beendet es die Ausgabe. Das Dateiendezeichen wird durch die Tastenkombination **Strg+D** erzeugt.

Das folgende Listing zeigt einen möglichen Ablauf (Eingaben sind unterstrichen):

```
$ cat
Dies ist ein Test
Dies ist ein Test
Strg+D
$
```

Natürlich lässt sich am Bildschirm kein Unterschied zwischen der Eingabe- und der unmittelbar folgenden Ausgabezeile feststellen.

Dies mag anfangs für Verwirrung sorgen, zumal die gesamte Aktion keinen Sinn zu ergeben scheint: Man gibt etwas ein, und der Computer gibt den gleichen Text noch einmal aus.

Leiten Sie jedoch die Ausgabe von *cat* in eine Datei um, so haben Sie eine einfache Möglichkeit kennen gelernt, kleine Textdateien zu erstellen.

Dateien



```
cat > dateiname
```

leitet die Erstellung einer Textdatei ein, und die Tastenkombination

Strg+D

schließt sie ab.

Dies entspricht im Wesentlichen *copy con dateiname* unter DOS und ist mit der gleichen Vorsicht zu genießen: Das Editieren einer Zeile ist nur bedingt möglich (mit der Taste können Korrekturen links vom Cursor in der aktiven Eingabezeile vorgenommen werden), wird die Eingabe der Zeile mit beendet, kann keine Korrektur mehr erfolgen!

Zur Erzeugung kleiner Shell-Skripte oder für das Erstellen einfacher Testdateien haben Sie aber nun ein geeignetes „Werkzeug“ zur Hand.

Darüber hinaus bietet *cat* einfache Möglichkeiten zum Kopieren von Textdateien.

```
cat quelldatei > zieldatei
```

erzeugt *zieldatei* als Kopie von *quelldatei*. Noch interessanter ist die Variante, aus mehreren Quelldateien eine Zieldatei zu erzeugen:

```
cat quelldatei1 quelldatei2 ... > zieldatei
```

Diesem „Zusammenfassen“ von Texten verdankt der Befehl wahrscheinlich seinen Namen. *cat* ist eine Verballhornung von *concatenate*, was man mit „verbinden“ oder „zusammenfassen“ übersetzen könnte.

Für das Kopieren von Dateien sollte man allerdings im Normalfall den Befehl *cp* (kurz für *copy*) benutzen.

cp

Der Befehl *cp* kopiert eine Datei oder eine Gruppe von Dateien.

Seine Syntax lautet:

```
cp [ -option] datei1 datei2
```

oder

```
cp [ -option] datei1[ datei2 ...] verzeichnis
```

Das letzte Argument des Befehls (*datei1* oder *verzeichnis*) ist das Ziel des Kopiervorgangs.

Es *muss* in jedem Fall angegeben werden.



Linux-Dateisystem



Die Kopie der Datei `/etc/passwd` ins aktuelle Verzeichnis würde demnach mit

```
cp /etc/passwd .
```

erfolgen. Diese Syntaxeigenschaft dürfte speziell DOS-Anwendern zunächst Schwierigkeiten bereiten, da sie die zwingende Angabe des Zielverzeichnisses (auch wenn es das aktive ist) nicht gewohnt sind.

Sie werden vermutlich häufiger mit der Fehlermeldung

```
cp: missing file argument
Try `cp --help' for more information.
```

konfrontiert werden.

Man beachte, dass auch mehrere Dateien gleichzeitig kopiert werden können, indem man ihre Namen (jeweils durch mindestens ein Leerzeichen getrennt) hintereinander in der Befehlszeile angibt.

Die Befehlssyntax verlangt dann aber, dass das Ziel ein Verzeichnis sein muss. Dies sollte man berücksichtigen, wenn man Joker benutzt!

```
cp ../rc.config ../rc.config.alt .
```

Diese Besonderheit von `cp` verhindert im Übrigen eine unliebsame Überraschung, die man mit dem DOS-Äquivalent `copy` erleben kann. Dieser Befehl erlaubt in jedem Fall als Zielangabe eine Datei. Während nun `cp` beim Kopieren mehrerer Dateien als Ziel ein (existierendes) Verzeichnis voraussetzt und gegebenenfalls das Kopieren verweigert, prüft `copy` lediglich, ob das Ziel existiert. Ist das nicht der Fall, wird eine neue Datei erzeugt, in die hinein nacheinander alle Quelldateien kopiert werden. Dies führt in allen Fällen zu ungewünschtem Datensalat!

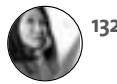
Rekursives Kopieren von Dateien

Als weitere Besonderheit sollte beachtet werden, dass `cp` zum Kopieren von Dateien konzipiert wurde. Als Quelle darf man also keine Verzeichnisse angeben, es sei denn, man verwendet die Optionen `-r` bzw. `-R`. Diese erlauben rekursives Kopieren von Verzeichnissen, d.h., es werden ganze Verzeichnisstrukturen kopiert.

Eine Verzeichnisstruktur im aktuellen Verzeichnis angenommen, wie sie in *Abbildung 3.2* dargestellt ist, lässt sich das Verzeichnis `help` samt der darin enthaltenen Dateien und Unterverzeichnisse wie folgt kopieren:

```
cp -r help daten
```

Gewöhnliche Dateien wurden in der Abbildung zur Vereinfachung weggelassen.



Dateien

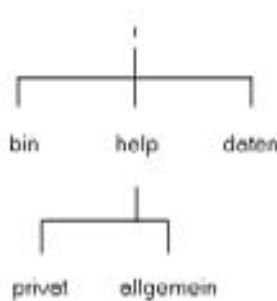


Abb. 3.2

Dateistruktur vor der Kopie

Als Ergebnis erhält man eine Struktur, wie sie in Abbildung 3.3 dargestellt ist.

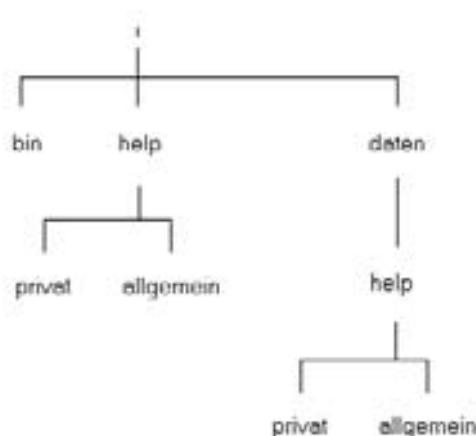


Abb. 3.3

Dateistruktur nach der Kopie

Unter Umständen werden beim Kopieren Dateien überschrieben, d.h., der Inhalt existierender Dateien wird durch die zu kopierenden Daten ersetzt. Dies ist dann der Fall, wenn die Zielfile bereits existiert. Ein einfaches Beispiel soll dies illustrieren:

```
$ cat > ueb<br/>Dies ist ein Test für das Kopieren<br/>[Strg]+D<br/>$ cat > nummer2<br/>Dieser Text zerstört den vorhergehenden<br/>[Strg]+D<br/>$ cp nummer2 ueb
```



Linux-Dateisystem



```
$ cat ueb←  
Dieser Text zerstört den vorhergehenden  
$
```

Dieses Beispiel zeigt bereits, warum *cat* so nützlich ist. Hier werden zunächst nach der oben beschriebenen Methode die beiden Dateien *ueb* und *nummer2* erzeugt. Dann wird *nummer2* auf *ueb* kopiert, womit der Inhalt dieser Datei durch den von *nummer2* ersetzt wird. Zuletzt wird zur Überprüfung der Inhalt von *ueb* ausgegeben.

Zu beachten ist, dass das Überschreiben ohne Rückmeldung erfolgt. Diese nicht ungefährliche Problematik lässt sich durch zwei in ihrer Wirkungsweise unterschiedliche Schalter entschärfen.

Die Option *-i* erzwingt ein interaktives Kopieren, d.h., vor dem Überschreiben wird nachgefragt, ob eine Ersetzung der Daten erfolgen soll.

Die Option *-b* veranlasst *cp*, vor dem Überschreiben eine Sicherungskopie der Zielfile zu erstellen. Solche Sicherungs- oder Backup-Dateien werden von *ls* durch eine nachgestellte Tilde (~) gekennzeichnet.

```
$ ls←  
originaltest  
$ cp -b test original←  
$ ls←  
originaloriginal~test
```

Zunächst wird der Inhalt des Verzeichnisses aufgelistet, dann wird kopiert. Zuletzt wird erneut der Inhalt aufgelistet. Die Datei *original* ist nun inhaltlich identisch mit *test* (sie ist eben eine Kopie!). Der Inhalt von *original~* entspricht dem der Originaldatei.

mv

Eng verwandt mit *cp* ist der Befehl *mv* (kurz für *move* – dt. bewegen). Mit ihm lassen sich Dateien umbenennen oder in andere Verzeichnisse verschieben. Das Dateisystem erlaubt dabei, wie Sie noch sehen werden, eine recht einfache Realisation des Verschiebens. Verschieben heißt im Allgemeinen nicht kopieren und dann löschen, hier werden nur Dateiinformationen manipuliert, die Daten bleiben von der Aktion unberührt.

Es gibt UNIX-Varianten, in denen *cp*, *mv* und *ln* in einer Datei zusammengefasst sind.

Die Syntax des Befehls lautet:

```
mv [ -option] datei1 datei2
```

oder



Dateien



`mv [-option] datei1[datei2 ...] verzeichnis`

Schon die Befehlssyntax zeigt die Nähe zu *cp*. Im ersten Fall erhält *datei1* einen anderen Namen (nämlich *datei2*). Existiert die Zielfile bereits, wird sie wie im Falle des *cp*-Befehls überschrieben.

Ist das Ziel ein Verzeichnis, werden die Quelldateien in dieses Verzeichnis aufgenommen und im Ursprungsverzeichnis entfernt. Auch hier gilt wieder die Regel, dass im schlimmsten Falle überschrieben wird. Die Optionen entsprechen im Wesentlichen denen des Befehls *cp*.

rm

Gelöscht werden Dateien mit *rm* (kurz für *remove* – dt. entfernen).

Dieser Befehl sollte nur mit absoluter Vorsicht benutzt werden, da eine Wiederherstellung gelöschter Dateien (wie Sie dies vielleicht von anderen Betriebssystemen her kennen) nicht möglich ist.

Die Syntax lautet:

`rm [-option] datei1[datei2 ...]`

Will man etwa die Datei */etc/passwd.old* löschen, weil diese Sicherungskopie nicht mehr benötigt wird, so erfolgt das Löschen mit:

`rm /etc/passwd.old`

Die Rechtesituation an einer Datei kann *rm* zu einer Rückfrage wie folgt veranlassen:

`rm: remove |`egal", overriding mode 0400?`

Die Datei ist, wenn man so will, schreibgeschützt (mode 0400), was *rm* aber nicht daran hindert, sie zu löschen, wenn die Frage mit **[Y]** beantwortet wird.

Einzelheiten zu Rechten werden weiter unten in diesem Kapitel besprochen.

Will man mehrere Dateien gleichzeitig löschen, empfiehlt sich aus Sicherheitsgründen die Option *-i* (insbesondere dann, wenn man mit Jokern arbeitet). Sie erzwingt eine Sicherheitsabfrage (*i* = interaktiver Modus) vor dem Löschen jeder einzelnen Datei.

Diese Sicherheitsabfrage erfolgt dann auch bei Dateien, bei denen die Rechte ein unmittelbares Löschen erlauben.

Will man umgekehrt, weil man seiner Sache absolut sicher ist, löschen, ohne Rückfragen beantworten zu müssen, so kann man den Schalter *-f* verwenden.

`rm -f *`



Es ist gängige Praxis, selbsterstellten Sicherungskopien Endungen wie *.old* anzuhängen



Hinweis



Linux-Dateisystem



löscht alle Dateien des aktuellen Verzeichnisses ohne Rückfrage bei entsprechender Rechtesituation.

Rekursives Löschen mit *rm*

Auch *rm* ist zum Löschen von gewöhnlichen Dateien bestimmt, deshalb werden Verzeichnisse als Befehlsargumente abgelehnt. Aber auch *rm* kennt eine Option *-r*, die rekursives Arbeiten – hier Löschen – erlaubt.

Angenommen, *help* ist ein Unterverzeichnis des aktuellen Verzeichnisses. Dann löscht

```
rm -r help
```

nacheinander alle Dateien, die in irgendwelchen Unterverzeichnissen von *help* existieren. Die dann leeren Unterverzeichnisse werden ebenfalls gelöscht.

Die potenzielle Gefahr des Schalters *-r* sollten Sie niemals unterschätzen, weshalb Sie ihn auch nur sehr gezielt einsetzen sollten. (Mit *rm -r /** hat sich schon manch ein Superuser sein Arbeitsverhältnis – ungewollt – verkürzt ...)

rm löscht allerdings keine Daten, sondern Links. Eine Datei, die über mehrere Namen, d.h. Links, verfügt, bleibt als solche erhalten, es wird lediglich einer der Namen der Datei für den Benutzer unbrauchbar.

Hinweis



Auch der Superuser kann nicht mehr über diesen gelöschten Link auf die Datei zugreifen.

Schauen Sie sich zunächst einmal an, wie man einer Datei einen neuen Namen gibt, d.h. für eine Datei einen zusätzlichen Link definiert. Dies leistet der Befehl *ln*.

ln

Die Syntax des Befehls *ln* lautet:

```
ln[ -option] datei1 datei2
```

oder

```
ln[ -option] datei1
```

oder

```
ln[ -option] datei1 [datei2 ...] verzeichnis
```

Im ersten Fall erhält die (gewöhnliche) Datei *datei1* den neuen Namen *datei2*. Betrachten Sie ein einfaches Beispiel:





```
$ cat > heute<  
Dies ist Inhalt der Datei heute<  
[Strg] + [D]  
$ ln heute morgen<  
$ cat morgen<  
Dies ist Inhalt der Datei heute
```

Zunächst wird eine Datei namens *heute* mit einem sie kennzeichnenden Inhalt erzeugt. Im zweiten Schritt erhält sie den neuen Namen *morgen*. Die Datei hat nun zwei Namen (Links), unter denen sie verfügbar ist, was der letzte Schritt demonstrieren soll. *cat morgen* gibt den Inhalt der Datei *heute* aus. *cat heute* hätte zum gleichen Ergebnis geführt, denn die beiden Namen bezeichnen dieselbe Datei.

Prüfen Sie die Gleichheit, indem Sie die obigen Schritte nachvollziehen und dann den Verzeichnisinhalt mit *ls -il* prüfen. Beachten Sie insbesondere die *I-Node-Nummern*.



Hinweis

Was würde nun mit der Datei *heute* geschehen, wenn man die Datei *morgen* editiert? In Linux stellt sich eine solche Frage nicht. Es gibt keine zwei Dateien, von denen eine *heute* und eine andere *morgen* heißt, es gibt *eine* Datei mit einer *I-Node-Nummer*, sagen wir *92788*, die unter den beiden genannten Namen im System bekannt ist. Unter welchem Namen der Benutzer auf die Datei zugreift, ist für Linux ohne Bedeutung. Es erkennt die Datei an der Nummer, der Benutzer am Namen.

Unter den herkömmlichen PC-Betriebssystemen sind diese Links unbekannt, und es besteht eine engere Beziehung zwischen Datei und Dateiname (eine Annäherung gibt es unter Windows 95 und NT). Daher werden Datei und Dateiname unbewusst gleich gesetzt.

Unter Linux führen die Befehle *cat abrakadabra*, *cat dreimal*, *cat schwarzer* und *cat Kater* stets zum selben Ergebnis, wenn die vier genannten Namen vier Links auf ein und dieselbe Datei sind.

Diese Bewertung und Benutzung von Namen könnte man mit Spitznamen vergleichen.

So wie mein Kumpel Karsten auch auf Zasi hört, hört aus der Sicht von Linux die Datei *92788* auf die Namen *heute* und *morgen*.

Ein Grund für die Verwendung von Links wurde weiter oben genannt – erhöhte Sicherheit beim Löschen. Hat eine Datei mehrere Links, so löscht *rm* nur jeweils einen Link. Erst beim Löschen des letzten Links wird die Datei insgesamt unzugänglich.



Linux-Dateisystem



Weitere Gründe, die für die Benutzung von Links sprechen:

- Dateien werden leichter verfügbar. Lange Pfadnamen lassen sich durch einen Link verkürzen. In `/usr/bin/test/cprogs/mnm probe` erlaubt unmittelbar nach Eingabe des Befehls den Zugriff auf `/usr/bin/test/cprogs/mnm` unter dem Namen `probe`.
- Links ermöglichen die Benutzung selbstgewählter Dateinamen und damit eine gewisse Anpassung des Systems an eigene Bedürfnisse. So erlaubt `/bin/cat ausgabe` den Aufruf des Befehls `cat` unter dem Namen `ausgabe`.
- Links sparen in all diesen Fällen Platz. Die Erzeugung eines Links bedeutet lediglich den zusätzlichen Eintrag eines Dateinamens in ein Verzeichnis. Sonstige neue Daten werden nicht angelegt.

Den Programmierern bietet sich hier ein interessantes Betätigungsfeld. Ein Programm, das verschiedene, aber dennoch verwandte Arbeitsgänge erledigen soll, kann so gestaltet werden, dass es beim Start prüft, unter welchem Namen es aufgerufen wurde, um dann in Abhängigkeit vom Aufruf seine Arbeit durchzuführen.

Betrachten Sie noch einmal die erste Variante der Syntax:

Das zweite Argument `dateiz` kann der Name einer existierenden gewöhnlichen Datei sein. In diesem Fall gibt `ln` eine Fehlermeldung aus, um die vorhandene Datei zu schützen.

Will man aber die existierende Dateien tatsächlich durch den neuen Link ersetzen, muss man die Option `-f` benutzen.

```
ln -f .profile egal
```

vergibt an `.profile` den zusätzlichen Namen `egal`. Sollte eine Datei des Namens `egal` bereits existieren, wird sie nun durch den Verweis (Link) auf `.profile` überschrieben.

Fehlt bei der Eingabe das zweite Argument (zweite Variante der Syntax), ersetzt `ln` es automatisch durch das aktuelle Verzeichnis. Die „ursprüngliche“ Datei erhält also einen zusätzlichen Namen im aktuellen Verzeichnis.

Somit ist diese Form ein Sonderfall der dritten Syntaxvariante, in der für eine oder mehrere Dateien gleichzeitig ein neuer Link in einem Verzeichnis erzeugt wird.

```
ln /bin/ls /bin/cp .
```

macht die Dateien `/bin/ls` und `/bin/cp` im aktuellen Verzeichnis verfügbar. Dies erlaubt den (zusätzlichen) Zugriff auf beide im derzeit aktuellen Verzeichnis, ohne dass dabei nennenswert Speicherplatz im System vergeudet wurde. Schließlich wurden die Daten, der Inhalt der Dateien, nicht kopiert.

Dateien



Das Löschen eines „Originals“, z.B. `/bin/ls`, löscht nun nicht die Datei als solches, sondern lediglich den Zugang zur Datei über den Namen `/bin/ls`. Im gewählten Beispiel wäre dies allerdings sehr unangenehm, da damit einer der am häufigsten benutzten Befehle nicht mehr über den üblichen Pfad zu erreichen wäre.

Der normale Benutzer würde nun nicht mehr über den Befehl `ls` verfügen, es sei denn, er weiß, wo der zusätzliche Link erzeugt wurde, und er hat dort hinreichend Rechte.

Die bisherigen Beispiele zeigen ausnahmslos so genannte „Hard-Links“. Hard-Links erlauben den direkten Zugriff auf die I-Node, d.h. die Verwaltungsinformationen und damit die Daten der betreffenden Datei (sofern die Rechte dies zulassen). So zeigt zum Beispiel `ls -l` sämtliche Informationen über die Originaldatei. Ein Beispiel soll dies veranschaulichen:

```
$ ln /etc/passwd .  
$ ls -l passwd  
-rw-r--r-- 2 root      root          889 Dec 29 05:34  
passwd  
$
```

Für `/etc/passwd` wird ein Hard-Link im aktuellen Verzeichnis erzeugt. Danach existiert hier ein Dateieintrag `passwd`. Informationen darüber liefert `ls`. Diese beschreiben (wie im Beispiel offensichtlich erkennbar) die Originaldatei.

Es gelten zwei wichtige Einschränkungen: Hard-Links können nicht

- für Verzeichnisse erzeugt werden,
- dateisystemübergreifend benutzt werden.

Beispiel: Eine Datei auf Diskette kann keinen zusätzlichen Namen (Hard-Link) in einem Verzeichnis erhalten, das sich auf der Festplatte befindet.

Theoretisch denkbar wäre ein systemübergreifender Hard-Link. Was aber geschieht, wenn die Diskette aus dem Laufwerk entfernt wird?

Da Hard-Links direkten Zugriff auf die Datei gewähren, ergibt sich hier ein offensichtliches Dilemma: Die Informationen über die Daten befinden sich auf der Festplatte, die Datei und damit die Daten selbst befinden sich auf der Diskette. Die Platte erlaubt dem Benutzer (genügend Rechte vorausgesetzt) den Zugriff auf die Daten, diese sind aber real nicht vorhanden (Diskette nicht eingelegt).

Zur Vermeidung von solchen Schwierigkeiten stellt Linux so genannte „symbolische Links“ (symbolic links) zur Verfügung. Sie sind sowohl auf Verzeichnisse anwendbar, als auch dateisystemübergreifend zu benutzen.

Linux-Dateisystem



Hinweis



Programmierer würden symbolic links als „pointer“ (Zeiger) bezeichnen.

Symbolische Links sind Verweise auf die Originaldatei. Man kann sie unter Zuhilfenahme des Schalters `-s` erstellen.

Die Eingabe des Befehls

```
ln -s /bin exec
```

erzeugt einen symbolischen Link namens `exec` auf das Verzeichnis `/bin`. Ein solcher symbolischer Link führt durchaus ein Eigenleben als Datei. Vergleicht man die Verzeichniseinträge eines Hard-Links mit denen eines symbolischen Links, fällt dies sofort ins Auge.

Im letzten Beispiel wurde ein Hard-Link auf die Passwortdatei `/etc/passwd` besprochen – dieses Beispiel wird nun wie folgt erweitert:

```
$ ln -s /etc/passwd userlist
$ ls -l passwd userlist
-rw-r--r-- 2 root      root          889 Dec 29 05:34
passwd
lrwxrwxrwx 1 rattle    users        11 Jan  2 08:19
userlist --> /etc/passwd
$
```

Zuerst wird der symbolische Link mittels `ln -s ...` erstellt, anschließend werden die Dateiinformationen über die im letzten Beispiel erstellte Datei `passwd` und die neue Datei `userlist` verglichen. Auf den ersten Blick erkennbar sind Unterschiede in folgenden Punkten:

- Dateityp (gewöhnliche Datei/Link)
- Rechtetabelle
- Besitzverhältnisse
- Darstellung des Dateinamens

Darüber hinaus finden sich weitere, nicht so auffällige Abweichungen:

- die Anzahl der Links
- die Größe der Datei in Byte

All diese Unterschiede ergeben sich aus der Tatsache, dass im Falle von `passwd` keine neue Datei erzeugt wurde, sondern lediglich ein Eintrag im aktuellen Verzeichnis, der einen direkten Zugriff auf die „Originaldatei“ ermöglicht. Somit werden auch mit `ls` alle Verwaltungsinformationen über das Original angezeigt:



Dateien



- als Dateityp gewöhnliche Datei
 - als Rechtetabelle `-rw-r--r-`
 - als Anzahl der Links 2
 - als Besitzer *root*
- etc.

ln -s dagegen erzeugt eine ganz neue Datei, nämlich *userlist*. Diese ist allerdings von bescheidenen Größe (hier 11 Byte!). Besitzer der Datei ist, wie nicht anders zu erwarten, der Erzeuger (im Beispiel *rattle*), und nicht der Besitzer des Originals.

Darauf, dass hinter dieser (bescheidenen) Datei real eine andere steckt, weist *ls* mit der Darstellung des Dateinamens hin, hinter *userlist* erscheint, durch `->` eingeleitet, der Name der Originaldatei.

Im Zusammenhang mit symbolischen Links ist es durchaus erlaubt, von „Originaldateien“ zu sprechen, bei Hard-Links lässt sich das Original ohne Zusatzinformationen nicht ermitteln.



Hinweis

Man beachte, dass der symbolische Link nichts anderes als ein Verweis auf die Originaldatei ist, d.h. mit

```
cat userlist
```

wird natürlich nicht der Inhalt des Links angezeigt, sondern der Inhalt der Datei */etc/passwd*, auf die *userlist* verweist.

Um die Unterschiede zu demonstrieren, haben wir für unser Beispiel die gewöhnliche Datei */etc/passwd* als Original gewählt. Symbolische Links sind also nicht nur für Verzeichnisse zulässig.

Ein anderes Beispiel zeigt auf recht drastische Weise den Unterschied auf.

Angenommen, für eine Datei *alpha* besteht ein symbolischer Link namens *beta*. Löscht man nun die Originaldatei *alpha*, bleibt *beta* als nutzlose Dateileiche im System zurück – sie verweist auf eine nicht mehr existierende Datei!

Wäre *beta* ein Hard-Link gewesen, würde die Datei noch weiter existieren und unter dem Namen *beta* verfügbar sein.

```
$ echo Hallo > alpha
$ ln -s alpha beta
$ ls -l
lrwxrwxrwx 1 rattle  users 3 Jan  3 09:06 beta ->
alpha
-rw-r--r-- 1 rattle  users 6 Jan  3 09:05 alpha
$ cat beta
```



Linux-Dateisystem



```
Hello
$ cat alpha<input type="button" value="..." />
Hello
$ rm alpha<input type="button" value="..." />
$ ls -l<input type="button" value="..." />
lrwxrwxrwx 1 rattle users 3 Jan 3 09:06 beta ->
alpha
$ cat beta<input type="button" value="..." />
cat: beta: No such file or directory
$
```

Zunächst wird *alpha* als Datei erzeugt, dann der symbolische Link namens *beta* angelegt. Anschließend wird mit *ls* die Existenz und der Zusammenhang zwischen den beiden Dateien geprüft, *beta* verweist in der Tat auf *alpha*.

Der Vollständigkeit halber lassen Sie sich den Inhalt beider Dateien ausgeben: Er ist, wie zu erwarten war, gleich. Dann wird *alpha* gelöscht.

Mit *ls* lässt sich sofort nachprüfen, dass *alpha* nicht mehr existiert, jedoch *beta*. Außerdem ist zu erkennen, dass *beta* immer noch auf (ein nicht mehr existierendes) *alpha* verweist.

beta existiert noch, nicht aber der Inhalt, auf den es verweist.

Hinweis



Zuletzt versuchen Sie, sich den Inhalt von *beta* anzeigen zu lassen. Dieser existiert für das System nicht mehr, da *alpha* gelöscht wurde. Eine irritierende Fehlermeldung weist auf dieses Problem hin.

An dieser Stelle muss andererseits eingeräumt werden, dass die obige Behauptung, Hard-Links könnten nicht für Verzeichnisse erzeugt werden, in dieser Allgemeinheit nicht stimmt. Sie ist dahingehend einzuschränken, dass der Befehl *ln* nicht dazu benutzt werden kann. Hard-Links für Verzeichnisse werden in bestimmten Fällen von Linux automatisch erzeugt.

Gehen Sie der Sache auf den Grund, indem Sie im Heimatverzeichnis ein Verzeichnis anlegen.

```
$ mkdir hltest<input type="button" value="..." />
$ ls -ld hltest<input type="button" value="..." />
drwxr-xr-x 2 rattle users 1024 Dec 29 05:38
hltest/
$
```

Das Ergebnis der Überprüfung der Dateiinformationen mit *ls* mag zunächst überraschen, schließlich zeigt es an, dass für das neue Verzeichnis *hltest* bereits zwei Namen (Hard-Links) existieren. In diesem Zusammenhang erinnern wir an die Dateien *.* und *..*, die in *jedem* Verzeichnis enthalten sind. In der Tat, hier findet sich des Rätsels Lösung.



Dateien



Beachten Sie, dass die Anzahl der symbolischen Links nicht in den Informationen von `ls` angezeigt wird, wie das Beispiel mit `userlist` zeigt.



Hinweis

```
ls -id htest htest/.
```

offenbart, dass das Verzeichnis selbst und der darin enthaltene Eintrag. (= aktuelles Verzeichnis!) dieselbe *I-Node-Nummer* besitzen! Dabei wurde der Schalter `-d` benutzt, um zu verhindern, dass der Inhalt der Verzeichnisse angezeigt wird, statt der Informationen über dieselben.

Weiter:

```
ls -id .. /home/rattle
```

(angewandt in einem Unterverzeichnis des Heimatverzeichnisses von Benutzer *rattle*) legt offen, dass .. ein Hard-Link auf das übergeordnete Verzeichnis ist (die *I-Node-Nummern* sind gleich).

Hard-Links auf Verzeichnisse entstehen also durch den Befehl `mkdir`. Durch ihn erhält das Verzeichnis, in dem ein Unterverzeichnis angelegt wird, einen zusätzlichen Namen, den Eintrag .. im neuen Verzeichnis.

Das neue Verzeichnis seinerseits erhält gleich zwei Benennungen:

- den vom Benutzer eingegebenen Namen dort, wo es als Unterverzeichnis angelegt wird,
- den Namen .. im neuen Verzeichnis selbst.

`mkdir` vergibt also (unter der Hand) zusätzlich Dateinamen. Dieser Vorgang lässt sich vom User aber nicht steuern, insofern lässt sich die obige Behauptung bezüglich Verzeichnissen und Hard-Links durchaus als korrekt bezeichnen.

Bewusstes Erzeugen eines Links (Hard-Link oder symbolisch) lässt sich statt mit `ln` auch mit `cp` erreichen. Dazu stehen die beiden Schalter `-s` und `-l` für `cp` zur Verfügung. Im Wesentlichen entsprechen sich die Anweisungen aus *Tabelle 3.4*.

Zusammenhang zwischen `ln` und `cp`

<code>ln</code>	<code>cp</code>
<code>ln</code>	<code>cp -l</code>
<code>ln -s</code>	<code>cp -s</code>

Tabelle 3.4





Besondere Dateibefehle

Die Informationen über das allgemeine Datei-Handling sollen durch die Beschreibung von zwei sehr unterschiedlichen Tools, die in jedem Fall die Leistungsfähigkeit von Linux demonstrieren, abgerundet werden.

Zum einen handelt es sich um ein einfaches Klassifizierungsprogramm mit dem sinnigen Namen `file`. `file` heißt einfach Datei! An ihm sollen einige nicht unwesentliche Interna von Linux verdeutlicht werden.

Zum anderen wird mit `find` eines der mächtigen Linux-Werkzeuge vorgestellt. Es hat eine etwas komplexe Syntax, kann aber in der täglichen Praxis ein nicht zu unterschätzender Helfer sein.

`file`

Der Befehl `file` analysiert Dateien und gibt dem Benutzer entsprechende Informationen über deren Typ. Dabei benutzt er drei Prüfverfahren, wie sie sehr ähnlich auch Linux selbst benutzt.

Der Befehl `file` wird vor allem dazu benutzt, festzustellen, ob eine Datei aus lesbaren Zeichen besteht. Dies bedeutet nämlich, dass sich die Datei mit einem Editor wie joe, vi, oder emacs bearbeiten lässt. Weiter oben wurde bereits festgehalten, dass Linux keinen wesentlichen Gebrauch von Dateinamenserweiterungen macht.

Für MS-DOS können nur solche Dateien Programme sein, die spezielle Endungen haben wie `.com`, `.exe` oder `.bat`. Diese kennzeichnen ausführbare Dateien.

Unter diesem Betriebssystem könnten Sie Ihren Lebenslauf schreiben und ihn `Leben.com` nennen. DOS würde diese Datei aufgrund des Namens als Programm akzeptieren.

Natürlich würde ein Aufruf von `Leben.com` zu einer Katastrophe führen (nicht wegen des Lebenslaufes!), denn der Inhalt der Datei ist alles andere als ein für DOS ausführbares Programm (obwohl man dies nicht mit absoluter Sicherheit ausschließen kann).

Linux – genauer die jeweilige Shell, etwa die bash – analysiert die Datei, bevor es sie als Programm startet. Dabei wird sie u.a. nach bestimmten Charakteristika durchsucht, die über die Art des Inhalts Auskunft geben.

Das Programm `file` prüft die Datei in drei Stufen. Zunächst werden die Verwaltungsinformationen untersucht. Anhand des eingetragenen Dateityps werden so bereits eine Reihe von Dateien klassifiziert: FIFOs, Sockets, Verzeichnisse oder Gerätedateien.



Besondere Dateibefehle



Einer genaueren Analyse werden dann die gewöhnlichen Dateien unterzogen. Der Anfang der Datei wird nach bestimmten Schlüsselwörtern durchsucht, die eine engere Bestimmung erlauben. Zum Beispiel wird `#include` für `file` ein Hinweis darauf sein, dass es sich bei der Datei um einen C-Quelltext handelt.

Eine letzte Prüfung vergleicht den Anfang der zu klassifizierenden Datei mit den Eintragungen in `/etc/magic`. Dies können Werte bestimmter Bytes einer Datei sein oder kurze, an bestimmten Stellen darin enthaltene Texte, die ihre Herkunft offen legen.

Die Form des Befehls lautet:

```
file[ -option] [ datei]
```

Eine Analyse mit `file` könnte etwa zu folgendem Ergebnis führen:

```
FIFO:      fifo (named pipe)
bin:       directory
hello.c:   c program text
help:      directory
liste:    ascii text
stuff:    directory
true:     Bourne Shell script text
```

Da nur der Anfang der Dateien untersucht wird und außerdem eine allgemeine Klassifizierung erfolgt und nicht etwa eine gezielte Suche nach Kennzeichen für einen bestimmten Dateityp, kann es vorkommen, dass einige Dateien unzureichend charakterisiert werden.

So wird das Shell-Skript `.profile` sehr oft als ASCII-Text erkannt, was es auch zweifellos ist (schließlich ist es ein druckbarer Text), dass es sich bei ihm aber um einen Linux-Befehlsstapel – ein Shell-Skript – handelt, erkennt `file` in diesem Falle nicht.

find

Mit dem Befehl `find` soll eines der für Linux so charakteristischen Werkzeuge vorgestellt werden. Viele Linux-Befehle wie etwa `banner`, `logname`, `head` oder `wc` sind in Handhabung und Wirkung sehr einfach. Eine der Ideen bei der Entwicklung von UNIX war es, Befehle mit klar beschriebenen, einfachen Aufgaben zur Verfügung zu stellen, wobei diese Befehle, wenn möglich, kombinierbar sein sollten, um komplexere Aufgaben zu erledigen. So ließ sich in der Praxis aus einfachen Einzelteilen ein umfassendes Ganzes erstellen. Dies ist einer der Gründe, warum die UNIX-Entwickler ihr Betriebssystem als Werkzeugkasten verstehen.

Ein anderer ist, dass sie es mit einer Reihe erstaunlich leistungsfähiger Programme ausgestattet haben, welche Funktionen wahrnehmen, die man sich in anderen PC-Betriebssystemen wünscht, die aber meist nur zur Verfügung stehen, wenn man zusätzlich zum Betriebssystem erweiternde Utilities besitzt.



Linux-Dateisystem



Hinweis



Eines dieser Werkzeuge, *find*, dient dazu, Dateien aufzufinden. Diese einfach anmutende Aufgabe wird natürlich auch von MS-DOS, Windows 3.x, Windows 95, NT oder OS/2 erledigt.

Immerhin: Die DOS-Versionen 1 und 2 kannten eine solche Möglichkeit nicht. Noch unter der Version 3 war das Suchen nach Dateien nur den Eingeweihten vorbehalten, da eine entsprechende Syntax nicht dokumentiert war.

Von *find* gelernt haben einige dieser Systeme inzwischen, dass man eine Datei nicht nur an ihrem Namen, sondern auch an anderen Merkmalen erkennen kann: Suchfilter können benutzt werden.

find unterscheidet seinerseits u.a. die Suche nach Dateien, die

- einen bestimmten Namen haben,
- eine bestimmte Größe, Mindest- oder Maximalgröße haben,
- einem bestimmten Benutzer gehören,
- zu einem bestimmten Datum (oder davor oder danach) verändert oder benutzt wurden.

Ein entscheidender Unterschied zu anderen Suchroutinen ist allerdings der, dass *find* nicht einfach die gefundenen Dateien anzeigt, sondern diverse Operationen mit den gefundenen Dateien gestattet. So erlaubt *find* z.B.:

- das Archivieren sämtlicher Dateien, die in den letzten 14 Tagen erzeugt wurden,
 - das Löschen aller Dateien, die einem bestimmten Benutzer gehören,
 - die Anzeige aller Dateien, die größer als 10 Kilobyte sind,
- oder einfach auch nur
- die Ausgabe aller Pfade der Dateien, die größer als 100 KByte sind und seit mehr als einem Jahr nicht mehr benutzt wurden!

Diese umfangreichen Möglichkeiten gehen zu Lasten einer einfachen Syntax, sprich Benutzerführung.

Die allgemeine Form lautet:

```
find pfad1 [pfad2 ...] ausdruck
```

Die Angaben *pfad1* und *pfad2* bezeichnen Verzeichnisse, in denen die Suche beginnen soll. Natürlich werden auch Unterverzeichnisse der Pfade einbezogen.



Besondere Dateibefehle



Eine Durchsuchung des gesamten Dateisystems wird mit

`find /`

eingeleitet.

Der folgende Ausdruck beschreibt einerseits, nach welchen Dateien gesucht werden soll (Suchkriterien), und andererseits, was mit den gefundenen Dateien geschehen soll (Aktion).

Die Argumente lassen sich dabei logisch verknüpfen, d.h., es können Suchkriterien aus verschiedenen Teilbedingungen zusammengesetzt werden. Nummerische Angaben als Bestandteil eines Suchkriteriums lassen sich auf drei Arten benutzen:

`n`, `+n` und `-n`, wobei `n` eine ganze Zahl ist.

Das Kriterium `-links` veranlasst z.B. eine Suche nach Dateien mit einer bestimmten Anzahl von Links und erfordert eine entsprechende nummerische Angabe. So bedeuten:

- `-links n` Suche nach Dateien mit `n` Links
- `-links +n` Suche nach Dateien mit mehr als `n` Links
- `-links -n` Suche nach Dateien mit weniger als `n` Links

In der folgenden *Tabelle 3.5* werden diese drei Fälle durch `n` zusammengefasst.

Die am häufigsten benutzte Aktion des `find`-Befehls ist `-print`. Sie bedeutet Ausgabe der Pfade der gefundenen Dateien. Zu beachten ist, dass mindestens eine Aktion angegeben werden muss, damit der Befehl ein sinnvolles Ergebnis liefert. Ansonsten durchsucht er den angegebenen Verzeichnisbaum gemäß der Suchkriterien, liefert aber keine Ergebnisse. So findet `find / -links 10` alle Dateien mit genau zehn Links, zeigt aber nicht deren Pfadnamen an!

Will man etwa alle Dateien namens `passwd` im Dateisystem finden, so muss man die Suche durchführen mit

`find / -name passwd -print`

Dabei veranlasst `-name` eine Suche nach bestimmten Dateinamen, hier `passwd`, im Gegensatz zu einer Suche nach Dateien, die einem bestimmten Benutzer gehören. Die Aktion `-print` darf, wie bereits angesprochen, nicht weggelassen werden, da sonst eine Angabe der Pfadnamen unterbleibt!

Dass für diese einfachste Form der Dateisuche eine doch recht komplexe Syntax vonnöten ist, mag auf den ersten Blick irritieren, sie hat ihren Grund aber in den vielfältigen Möglichkeiten, die `find` über die einfache Suche hinaus bietet.

Die wichtigsten Suchkriterien für `find` sind in *Tabelle 3.5* aufgeführt. Variable Bestandteile (z.B. nummerische Angaben) werden kursiv dargestellt. Alle anderen Angaben müssen in der genannten Schreibweise angegeben werden.



Linux-Dateisystem

Wichtige Suchkriterien des Befehls *find*

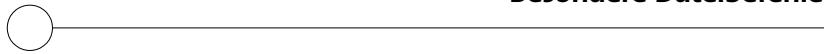
Kriterium	Bedeutung
<code>-amin n</code>	Dateien, auf die zuletzt vor <i>n</i> Minuten zugegriffen wurde
<code>-atime n</code>	Dateien, auf die zuletzt vor <i>n</i> Tagen zugegriffen wurde
<code>-ctime n</code>	Dateien, deren Verwaltungsinformationen zuletzt vor <i>n</i> Tagen geändert wurden
<code>-fstype typ</code>	Dateien, die sich in einem (Unter-)Dateisystem eines bestimmten Typs befinden. <i>typ</i> kann z.B. <i>ext2</i> , <i>msdos</i> oder <i>minix</i> sein.
<code>-group gruppe</code>	Dateien, die einer bestimmten Gruppe zugehören. <i>gruppe</i> kann entweder der Name der Gruppe oder deren <i>gid</i> sein.
<code>-inum I-Node</code>	Datei mit der entsprechenden I-Node-Nummer
<code>-links n</code>	Dateien mit <i>n</i> Links
<code>-mtime n</code>	Dateien, die inhaltlich zuletzt vor <i>n</i> Tagen geändert wurden
<code>-name muster</code>	Dateien, deren Name dem Suchmuster entsprechen (Joker im <i>muster</i> sind erlaubt)
<code>-newer datei</code>	Dateien, deren Änderungsdatum jünger ist als das der angegebenen Datei
<code>-perm oktalzahl</code>	Dateien, deren Rechtetabelle <i>oktalzahl</i> entspricht (nähre Einzelheiten dazu lernen Sie im Abschnitt <i>Rechte</i> dieses Kapitels kennen)
<code>-size n</code>	Dateien einer bestimmten Größe. Ohne nähere Angabe ist die Größe in Blocks (512 oder 1.024 Byte) gemeint. Ein nachgestelltes <i>c</i> (character) meint Größe in Byte, ein <i>k</i> Größe in KByte. <code>-size 1024c</code> entspricht also <code>-size 1k</code> .
<code>-type typ</code>	Dateien eines bestimmten Typs. Die Werte für <i>typ</i> entnehmen Sie der Tabelle 3.6.
<code>-user benutzer</code>	Dateien, die einem bestimmten Benutzer gehören. <i>benutzer</i> kann der Name des Benutzers oder dessen <i>uid</i> sein.

Tabelle 3.5

Ergänzend finden Sie in Tabelle 3.6 die Typkennungen des Suchkriteriums `-type`.



Besondere Dateibefehle



Die Typkennungen des Suchkriteriums -type

Typkennung	Dateityp
B	blockorientierte Gerätedatei
C	zeichenorientierte Gerätedatei
D	Verzeichnis
p	FIFO
l	symbolischer Link
s	Socket
f	gewöhnliche Datei

Tabelle 3.6

Die wichtigsten Aktionen von *find* sind in der *Tabelle 3.7* zusammengefasst.

Wichtige Aktionen des Befehls *find*

Aktion	Bedeutung
-exec <i>befehl</i>	Ausführen eines bestimmten Befehls
-ok <i>befehl</i>	Ausführen eines bestimmten Befehls; die Ausführung erfolgt erst nach Bestätigung einer Sicherheitsabfrage
-print	Ausgabe der Pfadnamen

Tabelle 3.7

Soll ein bestimmter Befehl ausgeführt werden, wenn die gesuchte Datei gefunden ist, dann soll sich dieser in der Regel auf die gefundene Datei beziehen. Die gefundene Datei wird innerhalb dieser Befehlssyntax mit *{}* beschrieben. Der Befehl selbst muss mit *;* abgeschlossen werden.

Diese etwas merkwürdige Regel hat ihre tiefere Begründung in den Besonderheiten der Shell (hier der bash). Sie verlangt ein Semikolon als Listentrennzeichen, wenn mehrere Befehle in einer Befehlszeile enthalten sind (*find* kann natürlich in einem Arbeitsgang mehrere Dateien finden, die dem entsprechenden Suchkriterium genügen, und muss dann mehrere Befehle zur Ausführung bringen).



Linux-Dateisystem



Der \ verhindert, dass die bash beim Lesen der Originalbefehlszeile das Folgezeichen (hier ;) in der eben geschilderten Bedeutung erkennt. Sie würde ansonsten einen zweiten Befehl in der Befehlszeile erwarten. Das Semikolon wird aber von *find* benötigt, damit es aus den Ergebnissen seiner Suche und den eingegebenen Aktionen eine geeignete Befehlszeile für die bash herleiten kann.

Einige einfache Beispiele sollen die Befehlssyntax und die Möglichkeiten von *find* verdeutlichen:

```
find /usr -type d -print
```

sucht nach allen Verzeichnissen, die unterhalb von */usr* existieren, und zeigt ihre Pfadnamen an.

```
find / -name tty* -print
```

liefert die Pfade aller Dateien des gesamten Dateisystems, die mit *tty* beginnen.

```
find / -user 406 -ok rm {} \;
```

sucht im gesamten Dateisystem nach Dateien des Benutzers mit der *uid* 406. Gefundene Dateien werden bei positiver Beantwortung einer Sicherheitsabfrage gelöscht.

```
find . -newer .version_time -print
```

zeigt alle Dateien des aktuellen Verzeichnisses, die jüngeren Datums sind als die Datei *.version_time*.

```
find / -size +10k -user rattle -print
```

zeigt alle Dateien, die größer als 10 KByte sind und dem Benutzer *rattle* gehören.

Im letzten Beispiel wurden zwei Suchkriterien angegeben:

—○ Größe über zehn Kilobyte

—○ Benutzername ist *rattle*

find wird beide Argumente akzeptieren und nach einfachen logischen Regeln miteinander verknüpfen. Eine Aneinanderreihung von Argumenten entspricht einer logischen UND-Verknüpfung. Bei Suchkriterien heißt dies, das Gesamtkriterium ist genau dann erfüllt, wenn *alle* einzelnen Kriterien erfüllt sind.

Auch die Aktionen werden in die logische Auswertung einbezogen. Dabei liefert *-print* stets den Wert WAHR, *-exec* und *-ok* nur, wenn der entsprechende Befehl erfolgreich ausgeführt werden konnte.

Die folgende Tabelle zeigt weitere Möglichkeiten logischer Verknüpfungen der Argumente von *find*.



Besondere Dateibefehle



Logische Verknüpfungen des Befehls *find*

Argument Bedeutung

-o	logische ODER-Verknüpfung
!	Negation. Kehrt den Wahrheitswert des folgenden Arguments um.
()	fasst Argumente zu Gruppen zusammen. Die Zeichen (und) sind Metazeichen der bash. Sie müssen durch \ „maskiert“ werden, damit die Shell sie ohne Interpretation passieren lässt.

Tabelle 3.8



Die gewöhnliche Reihenfolge der Auswertung eines Ausdrucks für den Befehl *find* ist von links nach rechts. Dabei gilt für eine ODER-Verknüpfung, dass sie bereits dann als WAHR erkannt ist, wenn der erste Teilausdruck WAHR ist. Eine UND-Verknüpfung wird als FALSCH bewertet, wenn der erste Teilausdruck FALSCH ist.

In solchen günstigen Fällen braucht also der zweite Teil der Verknüpfung nicht mehr geprüft zu werden. Durch geschickte Klammerung kann dieser Umstand zur Optimierung der Suche eingesetzt werden.

Ausgabe von Dateiinhalten, Druck

Mit *cat* haben Sie bereits ein Hilfsmittel zur Ausgabe von Dateiinhalten kennen gelernt. In vielen Fällen reicht dieser einfache Befehl nicht aus, die anstehende Aufgabe zu lösen. Beispielsweise könnte die Ausgabe länger als eine Seite sein, der Textanfang verschwindet allzu schnell vom Bildschirm.

more

Hier hilft der auch in DOS bekannte Befehl *more*.

Seine Syntax lautet:

```
more[ -option] [ datei1 [datei2 ...]]
```

Abgesehen von der Form der Ausgabe entspricht die Arbeitsweise von *more* der von *cat*. Das gilt insbesondere für den Fall, dass keine Dateinamen als Argument angegeben werden. Dies erlaubt, wie schon weiter oben ausgeführt, die Nutzung des Befehls in einer Pipeline.

In Ermangelung eines geeigneten Schalters lässt sich eine seitenweise Ausgabe von Dateiinformationen am besten mit

```
ls -l | more
```



more ist eine der vielen Erweiterungen, welche die Berkeley Universität der UNIX-Gemeinde geschenkt hat



Linux-Dateisystem



erreichen. Aber natürlich lässt sich jede Textdatei, z.B. `/etc/passwd`, mit *more* direkt ausgeben:

```
more /etc/passwd
```

Interessant ist nun die formatierte Ausgabe:

Ist die Datei länger als eine Bildschirmseite, erfolgt die Ausgabe interaktiv. Die Anzeige wird nach der ersten Seite unterbrochen. *more* wartet auf die Eingabe eines Befehls. Neben dem Weiterblättern (die Taste bewirkt ein zeilenweises, die Taste – die Leertaste also – ein seitenweises Blättern) stehen dem Anwender zahlreiche zusätzliche Möglichkeiten zur Verfügung wie:

- Suchen nach bestimmten Textpassagen,
- Wechsel zum Editor *vi* (man stelle sich vor, dass man während des Lesens einen Fehler im Text entdeckt, dann lässt sich dieser unmittelbar dadurch beheben, dass man direkt zum Editor wechselt und ohne Umgang den fraglichen Textteil angezeigt bekommt),
- Zurückblättern,
- Aufruf eines Befehls, ohne *more* zu verlassen
- oder einfach nur das vorzeitige Verlassen von *more*.

Die folgende Liste zeigt die wichtigsten Befehle, die während der Anzeige benutzt werden können.

In sehr vielen Fällen lässt sich eine Zahl voranstellen, welche die Anzahl möglicher Wiederholungen kennzeichnet. So kann man mit + statt einer Zeile fünf Zeilen vorblättern. Eine mögliche numerische Angabe wird in der Liste mit k angegeben.

Wichtige Befehle am *more*-Prompt



Befehl	Bedeutung
<i>H</i>	Hilfefunktion, Anzeige einer Befehlsübersicht
<i>?</i>	wie <i>H</i>
	Anzeige der nächsten k Zeilen (Voreinstellung: eine Bildschirmseite)
<i>z</i>	wie Leertaste. Wird k angegeben, ersetzt es die Voreinstellung.



Besondere Dateibefehle



Wichtige Befehle am *more*-Prompt (Forts.)

Befehl	Bedeutung
<code>[←]</code>	Anzeige der nächsten k Zeilen (Voreinstellung: eine Zeile), wird k angegeben, ersetzt es die Voreinstellung
<code>d</code>	Scrollen um k Zeilen (Voreinstellung: 11 Zeilen). Wird k angegeben, ersetzt es die Voreinstellung.
<code>[Strg]+[D]</code>	wie d
<code>b</code>	k Bildschirmseiten zurückblättern
<code>q</code>	<i>more</i> beenden
<code>Q</code>	wie q
<code>s</code>	k Zeilen vorwärtsspringen (Voreinstellung: 1)
<code>f</code>	k Bildschirme vorwärtsspringen (Voreinstellung: 1)
<code>'</code> (Apostroph)	zurück zu der Zeile, an der die letzte Suche begann
<code>=</code>	Anzeige der Zeilennummer
<code>/muster</code>	Suche nach dem k-ten Auftreten von <i>muster</i> (Voreinstellung: 1). Metazeichen in <i>muster</i> sind erlaubt (vergleichen Sie dazu auch <i>ed</i> und <i>vi</i>).
<code>n</code>	Wiederholung des letzten Suchvorgangs
<code>!befehl</code>	führt den Linux-Befehl <i>befehl</i> in einer Sub-Shell aus, anschließende Rückkehr zu <i>more</i>
<code>v</code>	startet den Editor <i>vi</i> mit Anzeige der aktuellen Zeile
<code>[Strg]+[L]</code>	frischt den Bildschirm auf
<code>:n</code>	wechselt zur nächsten, in der Befehlszeile angegebenen Datei. Es können k-1 Dateien übersprungen werden.
<code>:p</code>	wechselt zur vorherigen, in der Befehlszeile angegebenen Datei. Es können k-1 Dateien übersprungen werden.
<code>:f</code>	Anzeige von Dateinamen und Zeilennummer
<code>.</code>	Wiederholung des letzten Befehls

Tabelle 3.9



Linux-Dateisystem



less

more heißt mehr,
less heißt weniger

Eine interessante Erweiterung des *more*-Befehls stellt *less* dar. Sein Name mag als Beleg für den etwas skurrilen Humor der Linux/UNIX-Programmierer dienen. Auch seine Syntax und prinzipielle Handhabung entspricht der von *cat*:

```
less[ -option] [ datei1[ datei2]]
```

Die zusätzlichen Möglichkeiten ergeben sich während der Anzeige. So unterstützt *less* ein detailliertes Rückwärtsblättern und -suchen sowie spezielle Suchen nach Klammerpaaren, was insbesondere Programmierern nützt.

Der an Einzelheiten interessierte Leser sei auf die *man pages* verwiesen.

Analysieren Sie
doch einmal, ob
man auf *more* oder
auf *less* aufbaut!

Es wurde ja schon erwähnt, dass *more* (und natürlich auch *less*) sehr häufig in Pipelines eingesetzt werden. So darf es nicht verwundern, dass Linux-Befehle, die umfangreiche Texte ausgeben, diese Textbrowser ebenfalls zu ihrer Unterstützung einsetzen. Ein Beispiel dafür ist die Onlinehilfe *man*.

Neben diesen leistungsfähigen Tools kennt Linux auch eine Reihe recht simpler Programme zur Arbeit mit Dateiinhalten. Will man beispielsweise eine Art Vorschau auf eine Textdatei aufrufen, kann man dies mit *head* oder *tail* erreichen.

head* und *tail

Diese Befehle zeigen den Anfang (*head*) oder das Ende (*tail*) einer Datei, im Standardfall zehn Zeilen. Die Syntax ist:

```
head[ +n] [ -option] [ datei1 [datei2]]
```

bzw.

```
tail[ +n] [ -option] [ datei1 [datei2]]
```

Dabei überschreibt *+n* die Voreinstellung von zehn Zeilen.

```
tail +8 /etc/passwd
```

gibt die letzten acht Zeilen der Datei */etc/passwd* aus. Wer jemals programmiert hat, wird solche einfach gebauten Tools zu schätzen wissen.

Im Versionswirrwarr der Programme hilft häufig nicht mehr der Blick auf das Datum, hier nützt nur noch eine Vorschau, wie sie *head* oder *tail* bieten.



Besondere Dateibefehle



WC

Ein kleines Juwel stellt der Befehl `wc` dar. Er liefert einfache Statistiken über Texte. Seine Syntax lautet:

```
wc [ -option] [ datei1[ datei2]]
```

Die wenigen Optionen und ihre Bedeutungen sind der folgenden Tabelle zu entnehmen.

Die Optionen des Befehls `wc`

Option	Bedeutung
-c	zeigt an, wie viele Zeichen (characters) die Datei enthält
-l	zeigt an, wie viele Zeilen (lines) die Datei enthält
-w	zeigt an, wie viele Wörter (words) die Datei enthält

Tabelle 3.10



Ohne Argument benutzt, zeigt `wc` sowohl die Zeichen- als auch die Zeilen- und die Wörterzahl. Eine Analyse der Datei `.profile` könnte wie folgt aussehen:

```
$ ls -l .profile
-rw-r--r-- 1 rattle  users          53 Mar 22 1994
.profile
$ cat .profile
PS1=\$\n
PATH=$PATH:/home/rattle/bin
export PS1 PATH
$ wc .profile
      3      5     53 .profile
$
```

`ls` informiert über die Dateilänge (53 Byte).

Der Inhalt der Datei besteht offenbar aus drei Zeilen, wie `cat` ausgibt. Die Unterscheidung nach Wörtern fällt hier nicht ganz so leicht: Die beiden ersten Zeilen enthalten weder ein Leerzeichen noch einen Tabstop. Sie bestehen also aus einem einzigen Wort. In der letzten Zeile kann man allerdings drei Wörter zählen. Die Datei enthält also insgesamt fünf Wörter.

Damit erklärt sich natürlich die Ausgabe von `wc`: 3 Zeilen, 5 Wörter und 53 Zeichen sind in der Datei `.profile` enthalten.

Eine Untersuchung von `/etc/passwd` könnte zu folgendem Ergebnis führen:



Linux-Dateisystem



```
$ wc -l /etc/passwd  
22 /etc/passwd
```

Die Passworddatei enthält also 22 Zeilen, d.h. 22 Einträge. Anhand dieser Zahl lässt sich in etwa die Anzahl der real für das System zugelassenen Benutzer abschätzen. Jeder User ist durch einen Eintrag in */etc/passwd* vertreten. Allerdings enthält die Datei auch einige „Pseudo“-Benutzer.

Kennt man diese, so lässt sich auf diesem Wege die Anzahl der registrierten Benutzer ermitteln.

Hinweis



Insbesondere bei der Abarbeitung von Shell-Skripte wird gewöhnlich Wert darauf gelegt, dass nur die Daten am Bildschirm erscheinen, die der Benutzer vom System erwartet.

Aus der Syntax von *wc* ergibt sich analog zu *cat*, dass dieser Befehl innerhalb von Pipelines benutzt werden kann. Spätestens im Verbund mit anderen Befehlen wird *wc* seine Qualitäten enthüllen.

Mit *more* und *less* haben Sie zwei Befehle kennen gelernt, mit deren Hilfe Sie Text durchsuchen können. Während der Textanzeige lassen sich z.B. mit */muster* bestimmte Zeichenfolgen in der angezeigten Datei suchen. Natürlich gelingt dies nicht mit Dateien, deren Inhalte problemlos auf eine Bildschirmseite passen.

Beide Befehle bewirken in jedem Fall, dass der Anfang des Textes auf dem Bildschirm zu lesen ist. Häufig ist man als Benutzer aber daran interessiert festzustellen, ob bestimmte Daten in einer Datei enthalten sind, ohne sich ganze Dateiseiten am Bildschirm anzeigen zu lassen.

grep

Mit dem Befehl *grep* lassen sich Dateien nach bestimmten Inhalten durchsuchen. *grep* steht für **get regular expression**. Der Begriff „regulärer Ausdruck“ (regular expression) lässt sich etwa mit Suchmuster übersetzen.

grep kann also nicht nur nach bestimmten, fest vorgegebenen Texten in Dateien suchen, die Suche kann auch auf Muster ausgedehnt werden. Die Bildung der Muster entspricht dabei im Wesentlichen der Mustergestaltung in den diversen Editoren.

Die Syntax des Befehls lautet:

```
grep[ -option] suchmuster[ datei1[ datei2 ... ]]
```

Er durchsucht die angegebenen Dateien zeilenweise nach dem Vorhandensein von Suchmuster. (Ein Suchmuster kann prinzipiell nicht das newline-Zeichen enthalten, d.h., die Suche von *grep* endet immer am Ende einer Zeile.)



Besondere Dateibefehle



Bei Übereinstimmung einer Textpassage mit dem Suchmuster wird die gesamte Zeile, welche die Übereinstimmung beinhaltet, am Bildschirm angezeigt. Natürlich gilt auch für *grep*, dass es sich für den Einsatz in Pipelines eignet.

Bevor Sie sich im Folgenden der Bildung regulärer Ausdrücke zuwenden, ein einfaches Anwendungsbeispiel für die Arbeitsweise von *grep*. Will man den Eintrag des Superusers in der Datei */etc/passwd* ansehen, kann dies mit

```
grep root /etc/passwd
```

geschehen. Das Ergebnis am Bildschirm könnte etwa wie folgt aussehen:

```
$ grep root /etc/passwd
root::0:0:root:/root:/bin/bash
operator:*:11:0:operator:/root:/bin/bash
```

Offensichtlich genügt es im vorhergehendem Beispiel nicht, nach dem Muster *root* zu suchen, um den Eintrag des Superusers als einzige Zeile ausgeben zu lassen. Wenn man weiß, dass der Benutzername in */etc/passwd* stets am Anfang einer Zeile erscheinen muss, so bietet es sich an, den Zeilenanfang in die Suche mit einzubeziehen.

Mit dem Zeichen ^ im Suchmuster wird *grep* veranlasst, nach dem Zeilenanfang zu suchen. In vorliegendem Fall würde

```
grep "^root" /etc/passwd
```

das Gewünschte leisten. Es wird nach dem Text *root* am Anfang der jeweiligen Zeile gesucht, mit dem Ergebnis, dass jetzt nur noch eine Zeile gefunden wird.

```
$ grep "^root" /etc/passwd
root::0:0:root:/root:/bin/bash
```

Die Anführungszeichen müssen zusätzlich benutzt werden, da die bash sonst das Zeichen ^ falsch bearbeiten würde, denn es ist ein Metazeichen der Shell.

Metazeichen sind Zeichen, die ein Programm, das Texte bearbeitet/analysiert, nicht wie normalen Text behandelt. Ein Metazeichen signalisiert dem Programm, dass es an dieser Stelle etwas Besonderes zu tun hat.

Ein bekanntes Metazeichen der bash ist der Asterisk (*). Er dient als Joker in Dateinamen. Findet ihn die Shell in der Eingabezeile, stellt sie eine Liste aller Dateinamen zusammen, die dem mit * gebildeten Muster entsprechen.

Ein weiteres, häufig benutztes Metazeichen ist das Umleitungszeichen >. Hiermit wird bash veranlasst, eine Datei zu erstellen. Deren Namen entnimmt sie dem Teil der Befehlszeile nach dem >, was ja in den vorhergehenden Beispielen schon häufig erwähnt wurde.



Linux-Dateisystem



Neben der Shell (die bash dient immer als Beispiel, da sie die standardgemäß zur Verfügung stehende Shell ist) kennen auch Suchbefehle von Editoren wie *ed* oder Suchprogramme wie *grep* Metazeichen. Man beachte aber, dass diese in ihrer Bedeutung nicht immer miteinander übereinstimmen. Im Wesentlichen kann jedoch gesagt werden, dass Editoren und Suchprogramme die gleichen Metazeichen benutzen.

Eines dieser Zeichen haben Sie im letzten Beispiel kennen gelernt: das Caret-Zeichen (accent circonflexe) ^. Es veranlasst *grep*, den folgenden Text (hier *root*) nur am jeweiligen Zeilenanfang zu suchen. Dieses Beispiel veranschaulicht, was der Begriff „regulärer Ausdruck“ bedeutet. Ein regulärer Ausdruck ist ein Text, der Metazeichen enthalten kann.

Die folgende Tabelle zeigt die Metazeichen für *grep* und deren Bedeutung.

Metazeichen für *grep*



Metazeichen	steht für
.	ein beliebiges Zeichen
*	eine beliebige Wiederholung des vorhergehenden Zeichens – es ist auch „null“-malige Wiederholung gemeint. So steht .* für eine beliebige Zeichenfolge.
[...]	genau eines der Zeichen in der Klammer, z.B. [aA] bedeutet, dass der gesuchte Text an dieser Stelle ein a oder ein A enthalten muss
[...-...]	genau eines der Zeichen aus der Liste in der Klammer, z.B. bedeutet [a-z] alle Kleinbuchstaben
[^...]	genau eines der Zeichen, das nicht in der Klammer ist, z.B. steht [^123] für alle Zeichen außer 1, 2 und 3
[^...-...]	genau eines der Zeichen, das nicht in der Liste in der Klammer ist, z.B. steht [^o-9] für alle Zeichen, die keine Ziffern sind
^	Zeilenanfang
\$	Zeilenende
^\$	leere Zeile
^text\$	Zeile, die genau den angegebenen <i>text</i> enthält.
\<	Wortanfang
\>	Wortende



Besondere Dateibefehle



Metazeichen für *grep* (Forts.)



Metazeichen	steht für
\	hebt die Bedeutung des folgenden Zeichens als Metazeichen auf. Will man tatsächlich nach einem Text suchen, der einen Punkt enthält, muss verhindert werden, dass <i>grep</i> ihn als Ersatz für ein beliebiges Zeichen interpretiert. Mit \. hebt man die Bedeutung des Punktes als Metazeichen für diesen einen Fall auf.

\	hebt die Bedeutung des folgenden Zeichens als Metazeichen auf. Will man tatsächlich nach einem Text suchen, der einen Punkt enthält, muss verhindert werden, dass <i>grep</i> ihn als Ersatz für ein beliebiges Zeichen interpretiert. Mit \. hebt man die Bedeutung des Punktes als Metazeichen für diesen einen Fall auf.
---	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tabelle 3.11



Einige Beispiele sollen den Gebrauch regulärer Ausdrücke veranschaulichen.

```
grep "[0--9][0-9]*" testdatei
```

sucht nach beliebig langen Ziffernfolgen in einer *testdatei*. Der zu suchende Text muss mindestens eine Ziffer (erste eckige Klammer) enthalten, der beliebig viele weitere folgen können (zweite eckige Klammer, gefolgt vom *).

```
grep "Der \. ist das Satzende" testdatei
```

sucht nach dem Text *Der . ist das Satzende* in der *testdatei*. Die Benutzung des Backslashes \ ist erzwungen durch die Tatsache, dass der Punkt ein Metazeichen für *grep* ist.

```
grep "\<ein" testdatei
```

sucht nach dem Text *ein* in *testdatei*. Dabei muss *ein* am Wortanfang stehen, der Text *kein* entspricht dem Suchmuster also nicht.

```
grep "\\\" testdatei
```

sucht den Backslash in der *testdatei*. Die doppelte Benutzung ist dadurch begründet, dass der erste \ verhindern muss, dass *grep* den zweiten als Metazeichen interpretiert.

Alle oben genannten Beispiele benutzen die Anführungszeichen, um das Suchmuster zu kennzeichnen. Dies geschieht, um zu verhindern, dass die Shell das Suchmuster interpretiert. Schließlich ist sie es, welche die Eingabezeile untersucht, noch bevor *grep* den jeweiligen regulären Ausdruck zur Weiterverarbeitung erhält. Daher ist er vor der Interpretation durch die Shell zu schützen, was mithilfe der (Shell-Metazeichen!) „...“ geschieht.

Die Suche von *grep* kann durch eine Reihe von Optionen variiert werden. Einige davon können Sie der Tabelle 3.12 entnehmen.



Linux-Dateisystem

Einige wichtige Optionen des Befehls *grep*

Option	Bedeutung
-c	zeigt nur die Zeilennummern der Zeilen, in denen das Suchmuster gefunden wurde
-f <i>datei</i>	Die Suchmuster werden der Datei <i>datei</i> entnommen. Jede Zeile der Datei enthält ein Suchmuster.
-i	sucht ohne Unterscheidung von Groß-/Kleinschreibung
-n	zeigt zusätzlich die Nummern der gefundenen Zeilen
-v	sucht nach Zeilen, die das Suchmuster <i>nicht</i> enthalten

Tabelle 3.12

Neben *grep* kennt Linux noch die beiden damit verwandten Suchbefehle *fgrep* (*fast grep*) und *egrep* (*enhanced grep*).

fgrep verwendet einen schnelleren Suchalgorithmus (*fast*), erlaubt aber nur die Suche nach festen Texten, d.h., reguläre Ausdrücke als Suchmuster sind nicht erlaubt.

egrep lässt zusätzliche Sonderzeichen im regulären Ausdruck zu, die der folgenden Tabelle zu entnehmen sind.

Zusätzliche Metazeichen von *egrep*

Zeichen	steht für
+	beliebige Wiederholung des vorhergehenden Zeichens, es muss jedoch mindestens einmal auftreten
?	keine oder eine Wiederholung des vorangehenden Zeichens
	Verknüpfung zweier Suchmuster, nach denen dann alternativ gesucht wird
()	Gruppieren von Ausdrücken

Tabelle 3.13

So sucht

```
egrep "[0-9] +" testdatei
```

nach beliebig langen Ziffernfolgen in der *testdatei*.



Besondere Dateibefehle



Der Druck von Dateien ist nun nichts anderes als eine Variation der Ausgabe von Dateiinhalten. Das Ziel der Ausgabe ist allerdings nicht der Bildschirm, sondern der Drucker.

Druck von Dateien

Der Druck lässt sich auf einfachste Weise durch Umleitung der Daten erreichen.

```
cat .profile > /dev/lp0
```

leitet die Daten aus *.profile* an die Datei */dev/lp0*. Es ist eine Gerätedatei, deren Name (*lp* steht für **L**ine**p**rinter) signalisiert, für welches Gerät sie steht.

Dass die Nummerierung der Drucker anders als in MS-DOS mit 0 beginnt, kann nur anfänglich irritieren. Die Datei */dev/lpo* entspricht also dem unter MS-DOS bekannten LPT1:.

Ganz Hartgesottene werden

```
cp .profile /dev/lp0
```

versuchen, und es sei vermerkt, dass dies ebenfalls den Druck der Datei *.profile* bewirkt.

Benutzen Sie beide Möglichkeiten nicht! Beachten Sie, dass Sie normalerweise nicht der einzige Benutzer im System sind!



Hinweis

Es muss aber auf einen wichtigen Unterschied hingewiesen werden:

Linux ist ein Mehrbenutzersystem, d.h., es können gleichzeitig mehrere Benutzer auf die Ressourcen des Systems, etwa den Drucker, zugreifen.

Um möglichen Problemen beim gemeinsamen Zugriff auf den Drucker zu begegnen, ist unter Linux dafür ein so genanntes „Spool-System“ eingerichtet.

Für jeden angeschlossenen und installierten Drucker wird im Verzeichnis */var/spool/lpd* ein Unterverzeichnis eingerichtet. Dies ist Aufgabe des Systemverwalters. In das Verzeichnis hinein wird für jeden einzelnen Druckauftrag eine Datei angelegt. Ein solcher Druckauftrag wird auch „(Druck-)Job“ genannt. Er wird durch den Befehl *lprm* erzeugt.

Spool ist ein Akronym für **S**imultaneous **P**eripheral **O**peration **Off** **L**ine. Der Begriff stammt aus den Tagen der IBM-Großrechner und bezeichnete ursprünglich eine Technik, welche die Großrechner von so „unwichtigen“ Aufgaben wie dem Druck entlastete. Diese wurden an separaten Computern durchgeführt.



Hinweis



Linux-Dateisystem



lpr

Die Syntax des Befehls *lpr* lautet:

```
lpr[ -option] [ datei1[ datei2]]
```

Angenommen, Sie wollen die Datei *.profile* drucken, so geschieht dies mit der Befehlszeile

```
lpr .profile
```

lpr erstellt nun eine neue Datei im entsprechenden Unterverzeichnis von */var/spool/lpd*. Diese enthält die zu druckenden Daten bzw. einen Verweis darauf.

lpr kennt eine Reihe von Optionen, die sich auf die Formatierung des zu druckenden Textes beziehen. In Abhängigkeit von der Herkunft des Textes (reiner ASCII-Text, Ausgabe von TeX oder anderen Programmen, die Texte formatieren können) bereitet *lpr* diese unterschiedlich auf den Druck vor.

Weitere wichtige Optionen entnehmen Sie bitte der *Tabelle 3.14*.

Allgemeine Optionen von *lpr*



Option	Bedeutung
<i>-#n</i>	druckt die Daten n-mal
<i>-Ctext</i>	druckt <i>text</i> auf der Titelseite
<i>-h</i>	druckt keinen Header
<i>-job</i>	druckt den Namen des Jobs auf der Titelseite
<i>-m</i>	sendet dem Benutzer eine Mail, wenn der Job gedruckt worden ist
<i>-Pqueue</i>	veranlasst <i>lpr</i> , die angegebene Warteschlange zu benutzen. Ansonsten wird die Standardwarteschlange benutzt. Durch Setzen der Umgebungsvariablen <i>PRINTER</i> kann eine andere Queue als Standardwarteschlange festgelegt werden.
<i>-r</i>	löscht die Datei nach dem Druck (nur zusammen mit <i>-s</i>)
<i>-s</i>	erzeugt im Spool-Bereich einen Link auf die zu druckende Datei
<i>-Uname</i>	druckt den Benutzernamen auf der Titelseite

Tabelle 3.14



Wichtig ist insbesondere die Option *-Pqueue*, mit deren Hilfe unterschiedliche Queues angesteuert werden können.



Besondere Dateibefehle



Normalerweise wird eine Standard-Queue vorgegeben. Diese kann durch Setzen der Umgebungsvariablen *PRINTER* selbst bestimmt werden. Will man aber darüber hinaus eine andere Warteschlange benutzen, z.B. weil über sie ein besonderer Drucker angesprochen wird, so lässt sich die Voreinstellung einfach mit *-Pqueue* übersteuern.

lpd – der Druckdämon

Der eigentliche Druck erfolgt mithilfe des Druckdämons *lpd*.

Als „Dämon“ bezeichnet man in der UNIX-Welt einen Prozess (ein laufendes Programm), der nicht von einem Terminal kontrolliert wird. Normalerweise wird er beim Systemstart automatisch über die Datei */etc/rc.d/rc.local* gestartet, so auch *lpd*.

Dass er nicht von einem Terminal aus kontrolliert wird, bedeutet unter anderem, dass er nicht durch spezielle Tastendrücke beeinflussbar ist. In der Tat: Dämon-Prozesse sind im Allgemeinen solche Prozesse, die in regelmäßigen Abständen bestimmte Systemzustände überprüfen, um immer wiederkehrende Aufgaben zu erledigen. Das heißt, sie laufen so lange, bis das Linux-System heruntergefahren wird.

Ein gutes Beispiel für einen solchen Dämon-Prozess ist der Drucker-Spooler *lpd*. Er durchsucht in regelmäßigen Abständen den Spool-Bereich, das Verzeichnis */var/spool/lpd*, nach darin enthaltenen zu druckenden Dateien. Diese wird *lpd* an den Drucker senden, und zwar in der Reihenfolge, in der Sie im Spool-Bereich des jeweiligen Druckers angelegt wurden.

In diesem Sinne spricht man auch von „Druckerwarteschlangen“ oder „Queues“. Ein Druckjob wird in dem Sinne in eine Warteschlange eingereiht, dass eine Datei in einem bestimmten Verzeichnis angelegt wird. Der Druckdämon prüft nun das Vorhandensein von Dateien. Da ein Drucker immer nur je eine Datei drucken kann, wird *lpd* bei Vorhandensein mehrerer Dateien eine auswählen müssen. Dies ist diejenige mit dem ältesten „Erstellungsdatum“ (korrekt müsste es heißen das älteste Datum der letzten Änderung der I-Node). Es gilt auch hier das Prinzip „First In – First Out“.

Denken Sie in diesem Zusammenhang doch einfach an eine Warteschlange an einer britischen Busstation, eine Queue also!

lpq

`lpq [-option] [Druckauftragsnummer (n)]`

Über den aktuellen Inhalt der Warteschlangen kann man sich mit *lpq* informieren. Dies kann besonders dann nützlich sein, wenn man einen Druckauftrag löschen will. Durch das Anhängen von Druckauftragsnummern an den Befehl *lpq* lässt sich die Liste der Druckaufträge auf einen bestimmten Druckauftrag beschränken.



Linux-Dateisystem



lprm

lpq zeigt unter anderem die Nummern der einzelnen Jobs, über die diese mit dem Befehl *lprm* gelöscht werden können.

Seine Syntax lautet:

```
lprm[ -option] [ Jobnummer] [ benutzer]
```

Wird nur die *Jobnummer* angegeben, wird der entsprechende Job der Standard-Queue gelöscht. Mit der Option *-Pqueue* kann auf eine andere Queue zugegriffen werden.

Mit der Option – (ein einfaches Minuszeichen) werden alle Jobs, die man in eine Queue gestellt hat, gelöscht. Der Benutzer *root* löscht allerdings mit

```
lprm -Pnadel -
```

alle Jobs der Queue *nadel*. Ihm steht darüber hinaus noch eine weitere Möglichkeit des „tabula rasa“ zur Verfügung.

```
lprm -Pnadel rattle
```

löscht alle Jobs des Benutzers *rattle* in der Queue *nadel*.

Ein gewöhnlicher Benutzer kann selbstverständlich nur Druckjobs löschen, die er selbst veranlasst hat.

lpc

Einen allgemeinen Statusbericht über den Druckdämon und die Warteschlangen liefert der Befehl *lpc*. Mit ihm kann man u.a. feststellen, ob die Queue zurzeit überhaupt benutzt werden kann oder ob der Drucker bedient wird. Der Superuser ist darüber hinaus in der Lage, mit diesem Tool die Queues zu administrieren.

Seine allgemeine Form ist:

```
lpc[ Kommando]
```

Wird *lpc* ohne Angabe eines *Kommandos* aufgerufen, so geht es in einen interaktiven Modus über. Es zeigt einen Prompt (>) und wartet auf die Eingabe eines entsprechenden Befehls.

Die folgende Tabelle zeigt eine Liste der wichtigsten Kommandos von *lpc*:

Besondere Dateibefehle



Kommandos von *lpc*

Kommando	Bedeutung
<i>disable queue</i>	Die Queue nimmt keine Jobs mehr entgegen
<i>down queue</i>	Jobs in der Queue werden nicht mehr gedruckt
<i>enable queue</i>	Die Queue nimmt wieder Jobs entgegen
<i>start queue</i>	Jobs können wieder gedruckt werden
<i>status[queue]</i>	Statusbericht
<i>stop queue</i>	Stoppt den Druck
<i>topq jobnummer</i>	ändert die Reihenfolge der Abarbeitung der Queue. Der Job <i>jobnummer</i> wird als Nächster gedruckt (an die Spitze – engl. top – der Queue gestellt).

Tabelle 3.15



Mit dem Befehl *lpc restart all*, den nur der Superuser root ausführen darf, werden alle *lp*-Prozesse beendet und dann wieder gestartet.

Sollten Sie den interaktiven Modus von *lpc* benutzen wollen: Er wird mit *q* oder *exit* beendet.

Ein mit *lpc* erstellter Statusbericht könnte etwa wie folgt aussehen:

```
lp:  
    queuing is enabled  
    printing is enabled  
    no entries  
    no daemon present  
Nadel:  
    queuing is enabled  
    printing is disabled  
    2 entries  
    no daemon present
```

In die Queue *lp* (dies ist die Standard-Queue) können derzeit Jobs eingereiht (queuing is enabled) und gedruckt (printing is enabled) werden.





Datensicherung

Dass Daten stets von Verlust bedroht sind, ist eine Binsenweisheit für alle, die je mit EDV konfrontiert worden sind. Daten zu sichern ist eine der wichtigsten Aufgaben eines EDV-Anwenders. Wegen der Bedeutung dieses Themas werden die Einzelheiten in einem eigenen Kapitel besprochen. Hier sollen nur die Grundlagen erwähnt werden.

Prinzipiell werden Daten dadurch gesichert, dass man sie auf einen anderen Datenträger kopiert. Im einfachsten Fall ist dazu der Befehl `cp` geeignet. Im Allgemeinen wird aber die Menge der Daten andere Verfahren der Sicherung erfordern als die einfache Kopie.

Hinweis



Linux stellt für diesen Zweck eine ganze Reihe von Tools zur Verfügung, von denen die wichtigsten gesondert im *Kapitel 10 Sicherheit ist Trumpf: Datensicherung*.

tar

Eines der einfacheren Tools ist `tar` (tape archiver). Es wurde ursprünglich entwickelt, um Daten auf Bänder (tapes) zu speichern.

Ohne auf Einzelheiten einzugehen, seine Syntax lautet:

```
tar[ -option][ datei1 [datei2] ]
```

Die Form des Befehls sollte Ihnen inzwischen vertraut sein: `tar` kopiert beispielsweise Dateien in eine durch die Option `-f dateiname` benannte Archivdatei. Diese kann eine gewöhnliche Datei oder eine Gerätedatei sein.

```
tar -cf /dev/fd0 *
```

sichert z.B. alle Dateien des aktuellen Verzeichnisses auf einer Diskette (Floppy o würde unter DOS das Laufwerk A: bedeuten).

Dass die Angabe der Quelldateien nicht notwendig ist, sollte Ihnen signalisieren, dass anstelle von Dateien auch die Eingabe über Tastatur „archiviert“ werden kann. Ach ja: `tar` lässt sich, Konsequenz dieser Syntax, auch in Pipelines benutzen!

Auch die Datensicherung ist eine Arbeit in Zusammenhang mit dem Dateisystem.

Rechte

Bei allen bisherigen Arbeiten im Dateisystem wurde ein wichtiger Aspekt unberücksichtigt gelassen: die Rechte. In einem Mehrbenutzersystem gibt es in der Regel mehrere Verfahren, die Daten vor unberechtigtem Zugriff zu schützen.





An erster Stelle steht normalerweise die Login-Sicherheit. Durch Angabe eines Benutzernamens und meist auch eines Passworts muss sich ein Anwender beim System legitimieren – ohne einen Account (Benutzername + Passwort) kein Zugang zum System!

Eine zweite Sicherheitsstufe stellen – auch in Linux – die Rechte dar. Haben mehrere Benutzer zu den Daten eines Systems Zugriff, so ergibt sich schon aus recht einfachen Erwägungen heraus, dass dieser Zugriff reglementiert werden muss.

Als in den 80er Jahren die ersten Computerfreaks feststellten, dass auch ihre Kinder an dem Zauberkasten PC Interesse zeigten, wurde ihnen sehr schnell bewusst, welches Dilemma mit den seinerzeit für PCs verfügbaren Betriebssystemen verbunden war: Hatte man mit viel Mühe einen längeren Text geschrieben oder eine komplexe Kalkulation fertig gestellt, beanspruchten plötzlich Filia oder Filius den Rechner, um darauf die neueste Version eines Computerspiels testen zu können.

Dass ein anderer den Computer benutzen würde, bedeutete, dass dieser vollen Zugriff auf die Daten hatte! Statt des Starts des Lieblingsspiels ein versehentliches *format c:* war wohl der Schrecken vieler PC-Besitzer.

Während jedoch ein PC konzipiert wurde, um von *einem* Anwender benutzt zu werden (*Personal Computer*), steht ein Mehrbenutzersystem wie Linux von vornherein mehreren Anwendern zur Verfügung.

Diese Situation erzwingt eine Regelung für den Zugriff auf Dateien. Der Peer-to-Peer-Ansatz „Gibst du mir, so geb ich dir!“ in allen Ehren, aber er unterschätzt die Fehlerquelle Mensch.

Ein *rm -rf ** zur falschen Zeit am falschen Ort kann das Ergebnis vieler Arbeitsstunden unbrauchbar machen.

Sie brauchen überhaupt keine böse Absicht zu unterstellen. Aufgrund umfangreicher statistischer Erhebungen ist belegt, dass die meisten Fälle von Datenverlust durch unsachgemäße Benutzung der Software entstehen.

Man stelle sich nur einmal vor, wie häufig in einem Linux-System mit 100 Benutzern die Befehlszeile *rm ** eingegeben wird. Dies passiert ein einziges Mal in der Wurzel des Dateisystems, und der Linux-Kernel ist gelöscht.

Kurzum, ein System, in dem viele Benutzer gleichzeitig auf die Dateien zugreifen können, muss selbiges vor unberechtigtem Zugriff schützen, da jeder Zugriff – auch unbeabsichtigt – wichtige Daten zerstören kann.

Rechte und ihre Bedeutung

Basierend auf der Art, wie auf Dateien zugegriffen werden kann, unterscheidet Linux drei Rechte. Sie werden abkürzend mit r, w oder x bezeichnet. Die einzelnen Rechte sind an die jeweilige Datei gebunden. Sie werden in der I-Node der Datei gespeichert.



Linux-Dateisystem



Die folgende Tabelle beschreibt die Wirkung dieser Rechte für Dateien, sofern sie *keine* Verzeichnisse sind.

Rechte für (gewöhnliche) Dateien



Recht	Bedeutung
R	(read = lesen) der Benutzer kann den Inhalt der Datei einsehen, d.h. unter anderem, er kann sie am Bildschirm anzeigen lassen, drucken oder kopieren
W	(write = schreiben) der Benutzer kann die Datei verändert unter dem bisherigen Namen speichern
X	(execute = ausführen) die Datei kann als Programm gestartet werden. Dies setzt natürlich voraus, dass die Datei ein Programm ist (beachten Sie dazu aber auch die Ausführungen zum Befehl <i>file</i> in diesem Kapitel weiter oben)

Tabelle 3.16



Angenommen, Sie haben an der Datei *testdatei* kein *r*-Recht, wollen sie aber trotzdem auf dem Bildschirm sehen. Die Eingabe von

```
cat testdatei
```

führt dann zu einer Fehlermeldung der Art:

```
testdatei: permission denied
```

Viele Anwender, die aus einer Einplatzumgebung zu Linux stoßen, überschätzen allerdings die Bedeutung der Rechte, die man an einer (gewöhnlichen) Datei hat. Dies mag daran liegen, dass MS-DOS und viele seiner Verwandten den Zugriff auf Dateien durch Attribute zu regeln versuchen: Die vielzitierte „schreibgeschützte Datei“ gibt es unter Linux nicht.

Hinweis



„Versuch“ scheint der richtige Begriff für diese Art des Zugriffsschutzes zu sein. Wie will man denn über Attribute eine Datei schützen, wenn der Befehl zum Ändern von Dateiattributen von jedermann ohne Einschränkung benutzt werden kann?

Der Dateischutz wird ausschließlich über die Rechte gewährleistet. Hierbei gilt eine glasklare Logik, die sich im Wesentlichen aus dem Dateisystem ergibt: Wird in einer Befehlszeile ein Pfad angegeben, auf den der Befehl zugreifen soll, werden die einzelnen Bestandteile des Pfades daraufhin überprüft, ob hinreichende Rechte bestehen. Ist dies nicht der Fall, wird der Zugriff verweigert.





Verzeichnisse sind für Linux nur eine spezielle Art von Dateien. Dies macht sich hier dadurch bemerkbar, dass die Rechte *r* und *w* für Verzeichnisse genauso gehandhabt werden wie für Dateien. Daraus ergeben sich weit reichende Konsequenzen für die darin enthaltenen Dateien.

Das *r*-Recht an einem Verzeichnis zu besitzen, heißt, man kann den Inhalt des Verzeichnisses lesen – mit anderen Worten, man kann den Befehl *ls* für dieses Verzeichnis einsetzen.

Was aber ist nun der Inhalt eines Verzeichnisses? Ein Verzeichnis ist nichts anderes als eine kleine Tabelle. Jede Zeile dieser Tabelle hat zwei Einträge: den Dateinamen und die zugehörige I-Node-Nummer. Wie Sie bereits wissen, ist der Dateiname für Linux nicht das entscheidende Kriterium für die Bearbeitung einer Datei. Linux orientiert sich an den I-Node-Nummern.

Da die Benutzer in aller Regel besser mit Namen als mit Nummern arbeiten können, bietet Linux folgende einfache Schnittstelle zwischen Benutzer und Dateisystem an.

Der Benutzer verwendet in den Befehlen einen Dateinamen. Die benutzte Linux-Shell, z.B. die bash, ermittelt über den Verzeichniseintrag – den Link(!) – die zugehörige I-Node-Nummer.

Was bedeutet in diesem Zusammenhang das *r*-Recht für ein Verzeichnis?

Ganz einfach: Mit dem *r*-Recht am Verzeichnis können darin Dateien gefunden werden (der Befehl *ls* liefert die Namen der im Verzeichnis enthaltenen Dateien) und zusätzlich kann man die *I-Node-Nummern* dieser Dateien abfragen (*ls -i*).

Da in der I-Node selbst die Verwaltungsinformationen (darunter die Rechtetabelle für die Datei) gespeichert sind, ist ein Zugriff auf eine Datei also erst dann möglich, wenn man ihre I-Node-Nummer kennt! Das heißt also, wenn einem das *r*-Recht an einem Verzeichnis fehlt, hat man keinen Zugang zu den darin enthaltenen Dateien! Insbesondere sind auch Unterverzeichnisse dem Zugriff des Anwenders entzogen.

Das für das Verständnis der Linux-Rechte wichtigste Recht ist wohl das *w*-Recht für Verzeichnisse. Dieses Recht beinhaltet die Möglichkeit, ein Verzeichnis verändert abzuspeichern. Anders ausgedrückt: Ein Benutzer, der an einem Verzeichnis das *w*-Recht hat, kann es verändern.

Nun ist, wie bereits gesagt, ein Verzeichnis eine Liste von Dateinamen (plus zugehöriger I-Node-Nummer). Besitzt man das *w*-Recht an einem Verzeichnis, kann man einen Verzeichniseintrag löschen. Das Verzeichnis wird verändert gespeichert, der Dateiname und die I-Node-Nummer existieren nicht mehr. Einen Verzeichniseintrag löschen heißt also den Zugang (Link) zu einer Datei löschen. Dies geschieht allerdings nicht mit einem Editor, sondern mithilfe des Befehls *rm*.



Linux-Dateisystem



Verzeichnisse sind insofern „schreibgeschützt“, als Linux aus Konsistenzgründen schreibende Zugriffe auf Verzeichnisse nur mit dafür geeigneten Befehlen erlaubt. Das Bearbeiten mit einem Editor würde viele Benutzer überfordern, denn es muss Rücksicht auf die Struktur des Verzeichnisinhalts genommen werden. Wird diese aber zerstört, sind die Daten im Verzeichnis nicht mehr verfügbar!

Hinweis



Im Übrigen lässt Linux auch lesende Zugriffe auf Verzeichnisse nur mit geeigneten Befehlen zu. *ls verzeichnis* ist erlaubt, nicht jedoch *cat verzeichnis*.

Einen Extremfall angenommen: Eine Datei, an der Sie keinerlei Rechte besitzen, befindet sich in einem Verzeichnis, in dem Sie das *w*-Recht haben. Diese Datei können Sie, auch ohne Rechte an ihr selbst, löschen, denn Sie können das Verzeichnis manipulieren! Sie entfernen mit *rm* den Eintrag der Datei im Verzeichnis und löschen somit den Zugang zu den Daten.

Gleiches gilt für das Umbenennen (ändert den Eintrag in einem Verzeichnis), Verschieben (löscht den Eintrag in einem Verzeichnis) oder Erstellen (erzeugt einen neuen Eintrag in einem Verzeichnis) einer Datei. Auch hierzu ist das *w*-Recht am Verzeichnis von entscheidender Bedeutung.

Sie sehen, sehr viele Dateioperationen hängen von Verzeichnisrechten ab. Vereinfachend gesagt: Verwaltungsoperationen mit Dateien, wie z.B. Löschen, Verschieben oder Umbenennen, sind Eingriffe in die Verzeichnisstruktur und werden deshalb über Rechte an Verzeichnissen geregelt. Die inhaltliche Manipulation von Dateien ist dagegen nur möglich, wenn man Rechte direkt an der Datei hat.

Es kann sein, dass man eine Datei löschen, umbenennen oder verschieben kann, ohne ihren Inhalt auch nur sehen zu können. Zur Manipulation der Dateiinhalte benötigt man zumindest das *r*-Recht an ihr, alle anderen genannten Operationen erfordern das *w*-Recht am Verzeichnis.

Alles in allem kann man feststellen, dass Verzeichnisse bezüglich der Rechte *r* und *w* wie gewöhnliche Dateien behandelt werden. Da man aber mit einem Verzeichnis eine Liste von Dateinamen bearbeitet, bedeutet das Ändern von Verzeichnissen sehr häufig ein Löschen, Umbenennen oder Verschieben von Dateien.

Wurde bisher herausgestellt, dass Verzeichnisse eigentlich Dateien sind, müssen Sie bei der Besprechung des Rechtes *x* auf wichtige Unterschiede hingewiesen werden. *x* steht für das Recht, ein Programm ausführen zu können. In diesem Sinne kann also ein Verzeichnis nicht ausführbar sein.

Das *x*-Recht an einem Verzeichnis bedeutet nun, dass man

- mit *cd* in das Verzeichnis wechseln darf und
- Zugang zu den I-Nodes aller darin enthaltenen Dateien erhält.

Und wieder: Rechte an einem Verzeichnis betreffen die darin enthaltenen Dateien.



Rechte



Will man sich den Inhalt der Datei `/home/rattle/.profile` mit

```
cat /home/rattle/.profile
```

ansehen, so setzt dies folgende Rechte voraus:

- das *r*-Recht an der Datei selbst,
- das *r*-Recht am Verzeichnis `/home/rattle`,
- das *x*-Recht an diesem Verzeichnis.

`cat` muss zunächst den Inhalt von `/home/rattle` lesen, um zu prüfen, ob die Datei überhaupt existiert. Dies erlaubt das *r*-Recht am Verzeichnis.

Weiter benötigt `cat` die I-Node-Nummer der Datei, um über diese die *I-Node* selbst finden. Zugang dazu erhält `cat` über das *x*-Recht an `/home/rattle`!

In der I-Node befinden sich nun aber alle relevanten Informationen, die `cat` braucht, um den Dateiinhalt endlich lesen zu können. Hier ist die Rechtetabelle abgelegt, die über das *r*-Recht die Erlaubnis zum Lesen gewährt, und schließlich auch die Informationen, mit deren Hilfe die Daten auf der Platte gefunden werden können.

In der Praxis erhält man mit dem *r*-Recht an einem Verzeichnis stets auch das *x*-Recht. Natürlich kann man sich das *x*-Recht wieder nehmen (sofern man dazu die Erlaubnis hat), dann schafft man aber eine skurrile Situation: Das Leserecht am Verzeichnis erlaubt es dann zum Beispiel `ls`, Dateien zu finden, weitere Informationen werden aber verweigert. Man erhält in jedem Fall eine Fehlermeldung wie die folgende:

```
/bin/ls: egal/testdatei: Permission denied
```

`ls` meldet, dass man auf die I-Node von `testdatei` im Verzeichnis `egal` nicht zugreifen darf: Permission denied.

An den meisten Verzeichnissen im Dateisystem hat der gewöhnliche Benutzer die Rechte *r* und *x*, am Heimatverzeichnis aber alle Rechte.

Rechte und Gruppen

Die Rechte selbst werden in der I-Node der jeweiligen Datei abgelegt.

Ferner speichert Linux dort die User-Identifikationsnummer, kurz *UID*, des Dateibesitzers und die *GID* (Gruppenidentifikationsnummer) einer Gruppe. Beim Einloggen prüft Linux den Inhalt der Datei `/etc/passwd`. Dort befindet sich ein Eintrag wie

```
rattle:1/vf19fumtNV2:501:100:Guy
Rattlestern:/home/rattle:/bin/bash
```



Linux-Dateisystem



Die einzelnen Einträge sind durch einen Doppelpunkt getrennt.

- Der erste Eintrag ist der Benutzername.
- Der zweite Eintrag ist das (verschlüsselte) Passwort.
- Dann folgen die UID (hier *501*) und die GID (hier *100*). Mit dem Einloggen gehört der Benutzer einer Gruppe an, nämlich der, deren GID in der Passwortdatei eingetragen ist.

Die Gruppenzugehörigkeit spielt eine wichtige Rolle für die Ermittlung der Rechte, die man tatsächlich an einer Datei hat. Linux speichert je Datei neun Einträge für die Rechte. Die ersten drei sind die, die für den Dateibesitzer gelten. Die folgenden drei beschreiben die Rechte der Gruppe, deren GID in der I-Node gespeichert ist, und die letzten drei gelten für die übrigen Benutzer im System.

Is -In liefert statt der
Namens UID und GID

```
$ ls -ln liste
-rw-r----- 1 500 100 184 Dec 8 07:38 liste
```

Der erste Eintrag steht für den Dateityp; auf diesen wurde schon weiter oben in diesem Kapitel eingegangen. (Nur noch einmal so viel: Da es sich hier um eine „normale“ Datei handelt, wird dieser Eintrag nicht explizit hervorgehoben.)

Die weiteren drei Einträge sind *rw-*. Der Eintrag – besagt natürlich, dass dieses Recht fehlt. Dies sind die Rechte des Besitzers. Es handelt sich hierbei um den Benutzer *rattle*.

Demzufolge hat die Gruppe *users* das Recht *r*, andere Benutzer haben keine Rechte.

Greift ein Benutzer auf eine Datei zu, wird zunächst geprüft, ob er der Dateibesitzer ist. Falls ja, dann gelten für ihn die ersten drei der eingetragenen Rechte. Ist er nicht der Besitzer, wird weiter geprüft, ob er der Gruppe angehört, deren GID in der I-Node abgelegt ist. Wenn ja, gelten für ihn die Rechte vier bis sechs, wenn nein, die letzten drei Rechte.

Befehle für die Rechtevergabe

Es kann theoretisch vorkommen, dass ein Dateibesitzer weniger Rechte an der Datei hat, als die eingetragene Gruppe. Bei der Ermittlung der effektiv gewährten Rechte wird Linux die Gruppenrechte aber nicht zu denen des Besitzers addieren. Eine Prüfung auf Gruppenzugehörigkeit und damit das Feststellen der Gruppenrechte erfolgt nur für Benutzer, die nicht Dateibesitzer sind.

Andererseits ist natürlich der Besitzer einer Datei neben dem Superuser der einzige, der die Rechtetabelle verändern darf. Dies geschieht mit dem Befehl *chmod* (change mode).

Befehle für die Rechtevergabe



chmod

Die Syntax des Befehls *chmod* lautet:

`chmod WerWieWas [,WerWieWas, ...] dateiliste`

oder

`chmod Oktalzahl dateiliste`

Durch *WerWieWas* bzw. *Oktalzahl* wird die Rechtabellen definiert. Innerhalb von *WerWieWas* darf es kein Leerzeichen geben. *Wer* bezeichnet den Benutzerkreis, dem man Rechte gibt, *Wie* die Art, in der die Rechte gegeben werden, und *Was* die Rechte als solche. Im Einzelnen können die Kürzel aus *Tabelle 3.17* benutzt werden.

Für *Wer* kann eines der folgenden Kürzel oder eine Kombination davon stehen:

Kürzel des Befehls *chmod*

Kürzel	Bedeutung
<i>u</i>	(user) Rechte für den Dateibesitzer
<i>g</i>	(group) Rechte für die Gruppe
<i>o</i>	(others) Rechte für alle anderen Benutzer
<i>a</i>	(all) Rechte für alle Benutzer (entspricht <i>ugo</i>)

Für *Was* steht eines der folgenden Kürzel:

<i>+</i>	Die Rechte werden zusätzlich zu den vorhandenen vergeben
<i>-</i>	Die im Folgenden genannten Rechte werden entzogen
<i>=</i>	Die im Folgenden genannten Rechte ersetzen die bisherigen

Für *Wie* steht eines der zuvor beschriebenen Rechtekürzel, eine Kombination davon oder:

<i>X</i>	(großes X!) vergibt das x-Recht nur dann, wenn die Datei ein Verzeichnis ist oder es schon der Besitzer, die Gruppe oder andere Benutzer haben
<i>u</i>	vergibt die Rechte so, wie sie beim Besitzer (user) zurzeit sind
<i>g</i>	vergibt die Rechte so, wie sie bei der Gruppe (group) zurzeit sind



Linux-Dateisystem

Kürzel des Befehls `chmod` (Forts.)

Kürzel	Bedeutung
o	vergibt die Rechte so, wie sie bei anderen Benutzern (others) zurzeit sind

- | | |
| --- | --- |
| o | vergibt die Rechte so, wie sie bei anderen Benutzern (others) zurzeit sind |

Tabelle 3.17

Will sich der Dateibesitzer das *x*-Recht an der Datei `testdatei` geben, so geschieht dies mit

```
chmod u+x testdatei
```

Der Entzug des *w*-Rechts für alle Benutzer erfolgt mit

```
chmod a-x testdatei
```

Will man sich selbst *rwx*, der Gruppe *rx* und anderen nur *x* gewähren, lautet der Befehl

```
chmod u=rwx,g=rx,o=x testdatei
```

Das oktale Zahlen-
system basiert auf 8
Ziffern, nämlich 0
bis 7

Der letzte Fall lässt sich eleganter mit der zweiten Syntaxvariante lösen. *Oktalzahl* ist eine dreistellige oktale Zahl, bei der die erste Ziffer die Rechte des Besitzers beschreibt, die zweite die der Gruppe und die dritte wieder die der anderen Benutzer.

Dabei haben die Ziffern die Bedeutung aus *Tabelle 3.18*.

Der Oktalcode des Befehls `chmod`

Ziffer	Bedeutung
0	kein Recht
1	<i>x</i> -Recht
2	<i>w</i> -Recht
4	<i>r</i> -Recht

Tabelle 3.18

Rechtekombinationen werden durch Addition der Grundwerte beschrieben.

Die Rechtekombination *rx* entspricht der Ziffer 5, umgekehrt repräsentiert die Ziffer 7 alle Rechte, also *rwx*. Das letzte Befehlsbeispiel lässt sich in dieser Syntax so schreiben:



Befehle für die Rechtevergabe



```
chmod 751 testdatei
```

Beachten Sie, dass nur die erste Syntax das Entziehen einzelner Rechte oder deren zusätzliche Vergabe gestattet. Durch `chmod 751 testdatei` werden bisher vorhandene Rechte ersetzt. Im Einzelfall liegt es im Ermessen des Benutzers, welche Befehlsform er anwenden will.

umask

Mit `chmod` werden also Rechte an existierenden Dateien geändert. Der Befehl `umask` gestattet es, eine Maske zu definieren, welche die Vergabe von Rechten für neu zu erstellende Dateien automatisiert.

Beachten Sie dabei, dass Linux eine Kopie nicht als neue Datei ansieht. Beim Kopieren wird die Rechtetabelle vom Original übernommen – nur wirklich neue Dateien erhalten die Rechte, die durch die Maske festgelegt werden.

Die Syntax des Befehls lautet:

```
umask [ Oktalzahl ]
```

Die *Oktalzahl* wird nach den Regeln gemäß *Tabelle 3.18* gebildet. Ohne ihre Angabe zeigt `umask` die aktuelle Maske:

```
$ umask ↵  
022
```

Bei näherem Hinsehen stellt man fest, dass für den Besitzer in dieser Maske kein Recht eingetragen ist, die zweite Ziffer ist 0. Die Maske ist das Negativ der zu vergebenden Rechte. Die Maske `022` bedeutet also, dass für den Besitzer alle Rechte vergeben werden, für die Gruppe und andere Benutzer alle Rechte außer `w`.

Im Folgenden wird mit einem kleinen Beispiel eine weitere Besonderheit von `umask` untersucht.

```
$ umask ↵  
022  
$ mkdir neuverz ↵  
$ ls -ld neuverz ↵  
drwxr-xr-x  2 rattle  users   1024 Jan  6 03:28 neuverz/  
$ cat > neudat ↵  
qwertzui ↵  
Strg D  
$ ls -l neudat ↵  
-rw-r--r--  1 rattle  users      7 Jan  6 03:29 neudat  
$ cat > hello.c ↵  
main() ↵  
{ ↵  
    printf(" Hello world\n "); ↵
```



Linux-Dateisystem



```
} [←]  
Strg[D]  
$ ls hello[←]  
/bin/ls: hello: no such file  
$ cc hello.c -o hello[←]  
$ ls -l hello[←]  
-rwxr-xr-x 1 rattle users 12798 Jan 6 03:29  
hello*
```

Zunächst wird die aktuelle Maske überprüft. Wollen Sie das Beispiel nachvollziehen, müssen Sie gegebenenfalls mit

`umask 022`

den gleichen Zustand herstellen, wie er im Beispiel angenommen wird.

Erst wird ein neues Verzeichnis angelegt. Sie überzeugen sich mit `ls -ld`, dass die Rechte, wie durch `umask` festgelegt, vergeben wurden: `rwx` für den Besitzer, `rx` für alle anderen.

Als Gegenkontrolle wird mit `cat` eine neue Datei erzeugt. Und nun ergibt die Prüfung, dass die Rechte *nicht* gemäß `umask` vergeben wurden! Es fehlt in allen drei Fällen das `x`-Recht.

Sie prüfen weiter, indem Sie einen einfachen C-Quelltext erstellen. Dieser wird so kompiliert, dass als Ergebnis die Datei `hello` entsteht. Zuvor haben Sie sich noch vergewissert, dass eine Datei dieses Namens nicht existiert.

Die Rechtablage dieser neuen Datei `hello` enthält nun ihrerseits das `x`-Recht. Des Rätsels Lösung ist kein Bug des Systems, vielmehr nimmt Linux eine einfache Klassifizierung vor: Gewöhnliche Dateien benötigen das `x`-Recht nur, wenn sie wirklich ausführbar sind! Dies ist der Fall, wenn Sie mit einem Compiler ein Programm erzeugen.

Mit `cat > datei` wird „nur“ ein Text erstellt, für den Linux zunächst aus nahe liegenden Gründen das `x`-Recht verweigert. Ein Verzeichnis ohne das `x`-Recht ist wiederum nicht denkbar.

Unangenehm ist diese Situation nur dann, wenn man ein Shell-Skript anlegt. Dies ist auf der einen Seite zwar ein Text, auf der anderen aber auch prinzipiell eine ausführbare Datei. Ihr muss z.B. mit

`chmod u+x datei`

nachträglich das `x`-Recht zugewiesen werden.

Der Besitzer einer Datei zu sein, erlaubt neben `chmod` auch die Verwendung des Befehls `chown`.



Befehle für die Rechtevergabe



chown

Der Befehl *chown* (change owner – Besitzer ändern) hat folgende Syntax:

```
chown benutzer dateiliste
```

Achtung: Hat man einer Datei einen neuen Besitzer zugewiesen, kann man selbstverständlich nicht mehr mit *chmod* oder *chown* darauf zugreifen, man hat die Datei gewissermaßen verschenkt!



Hinweis

```
chown root testdatei
```

verschenkt also die *testdatei* an den Superuser! Befindet Sie sich im Heimatverzeichnis, kann man sie allerdings noch löschen, umbenennen oder verschieben.

Voraussetzung ist natürlich, dass der neue Besitzer in der Datei */etc/passwd* eingetragen ist.

Aus Sicherheitsgründen darf nur der Superuser eine Datei „verschenken“.



Hinweis

chgrp

Analog kann die Gruppe geändert werden. Aus leicht einzusehenden Gründen darf dies allerdings nur *root*. Der Befehl lautet:

```
chgrp gruppe dateiliste
```

Besonderheiten des x-Rechts

Vielleicht ist Ihnen beim Wandern durch das Dateisystem aufgefallen, dass einige der Dateien dort, wo normalerweise das x-Recht zu finden ist, einen Eintrag s oder t haben.

```
$ ls -l /usr/bin/lpr
-r-s--s--x 1 root lp 16388 Apr 29 1995 /usr/bin/lpr*
```

Wie im vorhergehenden Abschnitt beschrieben wurde, reiht *lpr* Druckjobs in eine Queue ein. Dies ist ein Verzeichnis, auf das ein gewöhnlicher Benutzer keinen schreibenden Zugriff hat. Ihm fehlt das w-Recht, denn sonst könnte er dort befindliche Dateien löschen, d.h., Druckaufträge anderer Benutzer entfernen.

Andererseits muss aber der Druckjob als neue Datei in das Verzeichnis aufgenommen werden.

Das s-Recht an einer ausführbaren Datei bedeutet, dass der Benutzer, der es startet, während des Programmlaufs die UID des Dateibesitzers erhält. Im Fall von *lpr* wird man, vereinfacht gesagt, zum Superuser!



Linux-Dateisystem



Mit den Rechten des Superusers ausgestattet, kann man natürlich auf den Spool-Bereich schreibend zugreifen: Eine neue Datei kann dort angelegt, ein Druckjob in die Queue eingereiht werden. Nach Beendigung des Befehls verliert man den vorübergehend verliehenen Superuser-Status.

Das *s*-Recht stattet Benutzer also vorübergehend mit neuen Rechten aus, indem es die Identität des Benutzers ändert. Etwas präziser: Beim Start erhält jeder Prozess zwei UIDs zugewiesen: Die eine ist immer die tatsächliche des Benutzers, die zweite ist normalerweise mit der ersten identisch. Nur wenn für die Programmdatei beim Dateibesitzer das *s*-Recht eingetragen ist, wird dessen UID zur zweiten UID für den Prozess. Über diese zweite Prozess-UID werden aber die Rechte ermittelt!

Hinweis



Weiter oben wurde darauf hingewiesen, dass nur der Superuser Dateien verschenken darf. Würde nämlich ein gewöhnlicher Benutzer Programm-Dateien an *root* verschenken, die für den Benutzer mit dem *s*-Recht ausgestattet sind, würde er sich für dieses Programm *root*-Rechte erschleichen können! (Alternativ ist es in einigen UNIX-Varianten dem gewöhnlichen Benutzer untersagt, Dateien mit dem *s*-Recht zu versehen.)

Die Kurzbezeichnung *s* steht abkürzend für set UID.

Wie Sie obigem Beispiel entnehmen können, ist das *s*-Recht aber nicht nur beim Dateibesitzer, sondern auch bei der Gruppe eingetragen. In diesem Fall gilt das soeben Gesagte mit der GID, anstelle der UID.

Der Programmcode
heißt in der Linux-
Terminologie **text**

Das *t*-Recht, auch „sticky bit“ oder „save text bit“ genannt, veranlasst Linux, den Programmcode eines gestarteten Programms nicht wieder aus dem Arbeitsspeicher zu entfernen. Dies hat den Vorteil, dass bei einem weiteren Start des Programms dieses nicht erst in den RAM geladen werden muss.

Dieses Merkmal ist besonders nützlich, wenn ein Programm häufig eingesetzt wird; denn der Programmcode muss nur einmal im Speicher sein, er kann aber von mehreren Benutzern gleichzeitig genutzt werden.

Da durch übermäßige Vergabe des *t*-Rechts sehr schnell der Speicher blockiert werden kann, lässt es sich nur vom Superuser vergeben.

Wie kann man nun diese Rechte vergeben?

Besondere Kürzel des Befehls *chmod*



Kürzel	Bedeutung
u+s	Vergabe von set UID
g+s	Vergabe von set GID
o+t	Vergabe des sticky bit (statt o können auch u, g oder a benutzt werden)

Tabelle 3.19



Einrichten des Drucksystems

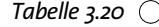


In der zweiten Syntaxvariante wird eine zusätzliche Ziffer vorangestellt, die sich wie in *Tabelle 3.20* berechnet.

Besondere Werte für den Oktalcode des Befehls `chmod`

Ziffer	Bedeutung
4	Vergabe von set UID
2	Vergabe von set GID (nur wenn für die Gruppe das x-Recht schon gegeben ist)
1	Vergabe des sticky bit

Tabelle 3.20



Somit kann der Superuser mit

```
chmod 5711 testdatei
```

für die *testdatei* gleichzeitig das sticky bit und das set UID vergeben.

Ein wesentliches Beispiel für den Einsatz dieser besonderen Rechte ist die Einrichtung des Drucksystems, die im folgenden Abschnitt beschrieben wird.

Einrichten des Drucksystems

Wir haben bereits weiter oben beschrieben, wie ein Benutzer mithilfe der Befehle *lpr*, *lpq* und *lprm* drucken bzw. seine Druckaufträge verwalten kann. Das Drucken setzt allerdings voraus, dass das Drucksystem vom Systemverwalter eingerichtet ist.

In einigen Distributionen ist dies bereits durch die Installation so weit geschehen, dass zumindest einfache Textdateien auf einem Standarddrucker ausgegeben werden können. Sollte dies bei Ihrem Linux-System der Fall sein, so ist der folgende Abschnitt für Sie wahrscheinlich auch interessant, da er einige wichtige Informationen über den Aufbau des Drucksystems enthält, die es Ihnen erlauben, die vorhandene Einrichtung Ihren Wünschen anzupassen.

Für das Einrichten notwendig sind dabei grundlegende Kenntnisse des Dateisystems einschließlich der Rechte. Da in der Regel auch Dateien zu verändern sind, ist es nützlich, wenn man einen Editor so weit beherrscht, dass man einfache Textmanipulationen mit ihm durchführen kann. Im folgenden *Kapitel 4 Benutzерumgebung* werden die wichtigsten Funktionen der Standardeditoren *ed* und *vi* sowie der in seiner Handhabung recht angenehme Befehl *joe* vorgestellt.



Linux-Dateisystem



In jedem Fall sollten Sie von den ange- sprochenen Dateien eine Sicherungs- kopie anlegen!

Sollten Sie die im Folgenden beschriebenen Schritte zur Installation des Drucksystems nachvollziehen wollen, empfiehlt es sich, einen der Editoren vorher kennen zu lernen. Zur allergrößten Not lassen sich natürlich die betreffenden Dateien jeweils mit

```
cat > datei  
neu erstellen oder durch  
cat >> datei  
ergänzen.
```

Grundlagen

Der Druck erfolgt, wie sollte es anders sein, über die Gerätedateien */dev/lp?*. Es wurde bereits darauf hingewiesen, dass es prinzipiell möglich ist, eine Datei, etwa *.profile*, mittels

```
cat .profile > /dev/lp0  
auszudrucken.
```

Auch für das Drucksystem ist die Gerätedatei die Schnittstelle zum Drucker. Aus Sicherheitsgründen – Linux ist ein Mehrbenutzersystem! – wird allerdings im Allgemeinen ein direkter schreibender Zugriff auf diese Gerätedateien nicht möglich sein.

Der Zugang zum Gerät Drucker wird normalerweise über Warteschlangen, so genannte „queues“ geregelt. Druckaufträge werden nicht unmittelbar an den Drucker gesandt (so wie im obigen Beispiel), sondern mit dem Befehl *lpr* in eine Warteschlange eingereiht.

Warteschlangen sind nun im Wesentlichen nichts anderes als Verzeichnisse im Linux-Dateisystem. Drucken mit *lpr* bedeutet zunächst einmal, dass *lpr* die zu druckende Datei in ein spezielles Verzeichnis des Dateisystems kopiert. Die Gesamtheit dieser speziellen Verzeichnisse nennt man auch den „Spool-Bereich“.

Dieser befindet sich üblicherweise im Verzeichnis */var/spool/lpd*. Es gibt im Übrigen eine ganze Reihe von Diensten des Systems, die in ähnlicher Weise wie der Druck abgewickelt werden. Diese haben ebenfalls ihren „Spool-Bereich“ – auch dies ist in der Regel ein Unterverzeichnis von */var/spool*.

Der Sinn einer Warteschlange ist es, zu verhindern, dass ein zeitgleicher Zugriff verschiedener Benutzer auf das (langsame) Gerät Drucker zu sinnlosen, weil vermischten, Druckergebnissen führt. Was aber geschieht nun mit den Dateien in der Warteschlange? Wie gelangen sie von dort zum Drucker?



Einrichten des Drucksystems



Für den eigentlichen Druck ist ein Programm namens *lpd* verantwortlich. Es überprüft permanent den Spool-Bereich auf neu eingetroffene Daten und schickt sie, falls möglich, an den jeweiligen Drucker.

Die Aufgabenstellung von *lpd* beinhaltet, dass dieses Programm *ständig* für *alle* Benutzer, die drucken wollen, aktiv sein muss. Es ist also sinnvoll, *lpd* bereits beim Hochfahren des Systems automatisch zu starten. Dies erfolgt dann der Einfachheit halber nicht durch einen Benutzer, sondern durch entsprechende Konfiguration von Startdateien

Prozesse sind, vereinfacht ausgedrückt, laufende Programme. Sofern sie nicht von einem Terminal aus, d.h. durch Eingabe des Programmnamens seitens eines Benutzers, gestartet wurden, nennt man sie in der UNIX-Welt „Dämonprozesse“. Konsequenterweise heißt *lpd* auch der „Druckdämon“.

Das Programm *lpd* kann natürlich auch von einem Terminal aus gestartet werden. Dies sollte allerdings die Ausnahme sein, etwa während der Installation des Drucksystems.



Hinweis

Bei seinem Start liest der Druckdämon *lpd* den Inhalt der Datei */etc/printcap*, der er Informationen darüber entnimmt, wo genau der Spool-Bereich zu finden ist, auf welchen Druckern die Daten auszugeben sind und ob die Daten vor dem Druck noch zu bearbeiten sind.

Wesentliche Aufgaben des Systemadministrators zur Einrichtung des Drucksystems sind daher:

- das Veranlassen des automatischen Starts von *lpd* beim Hochfahren von Linux
- das Einrichten des Spool-Bereichs,
- das Bereitstellen der notwendigen Informationen für *lpd* in der Datei */etc/printcap*.

Diese Arbeiten zur Einrichtung des Drucksystems sind natürlich Aufgaben der Systemadministration und somit nur als Benutzer *root* durchführbar. Loggen Sie sich gegebenenfalls also als *root* ein.

***lpd* automatisch starten**

Normalerweise befindet sich der Druckdämon im Verzeichnis */usr/sbin*. Sein Start erfolgt also durch

```
/usr/sbin/lpd
```

Um einen automatischen Start des Druckdämons beim Hochfahren von Linux zu veranlassen, genügt es, die Startdatei */etc/rc.d/rc.local* um diese Befehlszeile zu ergänzen. Dies geht auch ohne den Einsatz eines Editors, z.B. mit

181



Linux-Dateisystem



```
echo /usr/sbin/lpd >> /etc/rc.d/rc.local
```

Geben Sie diese Zeile aber mit der nötigen Vorsicht ein. Vergessen Sie insbesondere nicht das zweite >-Zeichen – ansonsten löschen Sie den Inhalt dieser nützlichen Datei!

Arbeiten Sie mit einer SuSE-Distribution, sollten Sie sich einmal die Datei */etc/rc.config* anschauen. Enthält sie eine Zeile wie

```
START_LPD="no"
```

dann sollten Sie diese mit einem Editor (und der nötigen Vorsicht) ändern in

```
START_LPD="yes"
```

Beim Booten des Systems führt dies dazu, dass das Skript */sbin/init.d/lpd* ausgeführt wird, was seinerseits den Druckdämon startet.

Einrichten des Spool-Bereichs

Eine Queue ist, wie bereits erwähnt, im Wesentlichen ein Verzeichnis. Also ist die erste Aufgabe bei der Einrichtung des Spool-Bereichs das Anlegen eines Verzeichnisses. In der Regel wird man ein Unterverzeichnis von */usr/spool/lpd* als Queue benutzen, und demzufolge könnte der erste Schritt lauten:

```
mkdir /usr/spool/lpd/laser
```

Der Name der Queue sollte dabei selbstverständlich so gewählt sein, dass er eine Beziehung zum realen Drucker erkennen lässt. Hier wird also eine Queue eingerichtet, über die ein Laserdrucker bedient wird.

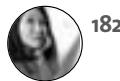
Innerhalb des Verzeichnisses werden, wenn das Drucksystem eingerichtet ist und benutzt wird, die einzelnen Druckjobs als temporäre Dateien angelegt, die *lpd* dann zum jeweiligen Drucker weiterleitet und nach erfolgter Arbeit wieder löscht.

Neben den Druckjobs sollte das Verzeichnis aber auch noch einige Dateien enthalten, die *lpd* bei der Verwaltung der Queue unterstützen. Diese sind:

Dateien in einem Spool-Verzeichnis



Datei	Bedeutung
.seq	enthält den Zähler, mit dessen Hilfe die Nummer des Druckjobs vergeben wird
status	enthält die Statusmeldung, die gegebenenfalls durch den Befehl <i>lpq status</i> ausgegeben wird



Einrichten des Drucksystems



Dateien in einem Spool-Verzeichnis (Forts.)



Datei	Bedeutung
<i>lock</i>	wird von <i>lpd</i> benutzt, um zu verhindern, dass mehrere Druckjobs gleichzeitig bearbeitet werden
<i>errs</i>	enthält Meldungen über eventuell aufgetretene Fehler

Tabelle 3.21



Hinzu kommen noch bei Bedarf Filterdateien, welche die Ausgabe der Daten vorbereiten können. Ein Beispiel für einen solchen Filter zeigt das folgende Listing:

```
#!/bin/bash
echo **** >> /tmp/lp.test
date >> /tmp/lp.test
logname >> /tmp/lp.test
echo **** >> /tmp/lp.test
cat >>/tmp/lp.test
```

Es handelt sich um ein einfaches Shell-Skript (auszuführen durch */bin/bash*), das Datum und Login-Name als Druckkopf ausgeben soll.

Die in *Tabelle 3.21* angegebenen Dateien werden vom Drucksystem verwaltet, ihr Inhalt ist für den Systemverwalter bei der Installation dieses Systems nicht weiter von Belang. Sie sollten jedoch sichergehen, dass sie vorhanden sind, bevor *lpd* gestartet wird. Mit Hilfe des Befehls *touch* lässt sich dies auf elegante Weise realisieren. Seine Syntax lautet

```
touch[ -option] [ zeit] datei1[ datei2 ...]
```

In einigen Linux-Distributionen werden die Dateien bei Bedarf von *lpr* angelegt.



Hinweis

Üblicherweise wird *touch* dazu benutzt, den „Zeitstempel“ einer Datei zu ändern, das heißt, die Zeit der letzten Änderung und des letzten Zugriffs auf die genannte *zeit* zu setzen. *zeit* wird dabei in der Form *MMTThhmmJJ* (Monat, Tag, Stunde, Minute, Jahr jeweils zweistellig) angegeben, ansonsten verwendet *touch* die aktuelle Systemzeit:

```
touch 1212040099 bubu
```

Diese Befehlszeile versieht die Datei *bubu* mit dem Zeitstempel 12. Dezember 1999, 04:00 Uhr.

Für diesen Fall besonders nützlich: Existieren die angegebenen Dateien nicht und wird nicht ausdrücklich die Option *-c* verwendet, legt *touch* die genannten Dateien neu an. Mit

```
cd /var/spool/lpd/laser
```



Linux-Dateisystem



und anschließendem

```
touch .seq errs lock status
```

legt man die gewünschten Dateien an.

Wesentlich sind nun die Zugriffsrechte für das neue Verzeichnis und die darin enthaltenen Dateien.

Besitzer sollte jeweils *root* sein. Hat man sie als *root* angelegt, ist dies bereits Fakt. Notfalls müssen die Verzeichnisse mit

```
chown root /var/spool/lpd /var/spool/lpd/laser
```

und die Dateien mit

○ chown root .seq errs lock status

dem Benutzer *root* zugewiesen werden.

Die Besitzergruppe sollte *lp* (in einigen Varianten *daemon*) sein. Mit

```
chgrp lp /var/spool/lpd /var/spool/lpd/laser
```

bzw.

```
chgrp lp .seq errs lock status
```

ist das einfach zu erledigen.

Schließlich müssen die Rechte so gesetzt werden, dass der Besitzer und die (Besitzer-)Gruppe *lp* an den Verzeichnissen alle Rechte hat, an den Dateien die Rechte *r* und *w*, alle anderen Benutzer *r* und *x* an den Verzeichnissen sowie *r* an den Dateien.

Zu Informationszwecken müssen natürlich alle Benutzer Zugriff auf die entsprechenden Daten haben, zum Beispiel muss der Status einer Queue (Datei *status* im Spool-Verzeichnis) durch *lp* *status* jedem Benutzer zugänglich sein. Mit

```
chmod 775 /var/spool/lpd /var/spool/lpd/laser  
chmod 664 .seq errs lock status
```

vergeben Sie also die geforderten Rechte. Eventuelle Filter müssen darüber hinaus mit dem *x*-Recht versehen sein. Ist etwa *testfilter* der Name einer solchen Datei, so sind die Rechte mit

```
chmod 775 testfilter
```

zu vergeben.



Einrichten des Drucksystems



Der gewöhnliche Benutzer kann also das Spool-Verzeichnis und die darin vorhandenen Dateien lesen, aber nicht manipulieren – ihm fehlt aus verständlichen Gründen das Recht *w*. Andererseits will er mit *lpr* einen Druckjob in die Queue einreihen bzw. einen eigenen Druckjob mit *lprm* löschen. Dies wird durch den Einsatz des *set UID*- und *set GID*-Bits erreicht.

Die für das Drucksystem wichtigen Programmdateien sollten dabei folgende Rechtabellen, Besitzer und Besitzergruppen aufweisen:

```
$ cd /usr/bin[←]  
$ ls -l lp*[←]  
-rwsr-sr-x 1 root lp 16388 Apr 29 1995 lpq*  
-rwsr-sr-x 1 root lp 16388 Apr 29 1995 lpr*  
-rwsr-sr-x 1 root lp 16388 Apr 29 1995 lprm*  
-rwsr-sr-x 1 root lp 20484 Apr 29 1995 lpc*  
$ cd /usr/sbin[←]  
$ ls -l lpd[←]  
-rwxr-s--- 1 root lp 36868 Apr 29 1995 lpd*  
$
```

Das *s*-Recht lässt sich hier beispielsweise für die Datei */usr/bin/lpr* mit

```
chmod u+s /usr/bin/lpr  
chmod g+s /usr/bin/lpr
```

vergeben.

Die in diesem Abschnitt beschriebenen Schritte beziehen sich bis auf die Rechtevergabe für die Programme auf *eine* Queue. Sollten mehrere Queues einzurichten sein, sind die entsprechenden Schritte für jede Queue zu wiederholen. Dabei ist zu beachten, dass prinzipiell ein Drucker durch verschiedene Queues bedient werden kann. Durch geschickten Einsatz von Queues wird dem Benutzer das Drucken erheblich erleichtert.

Bereitstellen der Informationen für *lpd* – die Datei */etc/printcap*

Sie haben nun zwar einen Spool-Bereich eingerichtet und können den Druckdämon automatisch starten. Zu klären bleibt aber noch die Frage, wie der Druckdämon *lpd* dem gewünschten Drucker die Daten aus den Warteschlangen zuordnet. Die Anbindung der Drucker an eine Queue erfolgt mithilfe der Datei */etc/printcap*.

Sie ist prinzipiell eine der Datei */etc/passwd* vergleichbare Liste. Werden in der Passwortdatei Benutzer beschrieben, so werden in */etc/printcap* Warteschlangen definiert und beschrieben.



Linux-Dateisystem



Die Einträge in beiden Dateien verstehen sich zeilenweise, d.h., jede Zeile definiert einen Benutzer bzw. eine Warteschlange. Sollten Sie in Ihrem System eine Datei */etc/printcap* vorfinden, die Sie sich schon einmal angeschaut haben, werden Sie sehr viele Zeilen gefunden haben, die mit dem Zeichen # beginnen. Dies sind Kommentarzeilen, die von *lpd* nicht genutzt werden.

Zeilen, die tatsächlich eine Warteschlange definieren, enthalten durch das Zeichen : getrennte Felder, die der Beschreibung der Warteschlange dienen. Hier besteht ein entscheidender Unterschied zu */etc/passwd*: Die insgesamt möglichen Felder sind bei */etc/passwd* recht überschaubar und daher in ihrer Reihenfolge fest vorgegeben.

Die Felder in */etc/printcap* werden dagegen normalerweise durch ein bezeichnendes Kürzel eingeleitet, dem der gewünschte Wert folgt. Dies gestattet es dem Systemverwalter, Angaben zu einer Queue auf das Notwendigste zu beschränken. Sollte trotzdem die Zeile nicht zur Aufnahme der nötigen Informationen ausreichen, kann man sie durch

\ \rightarrow

„künstlich“ verlängern.

Hinweis



Der \ maskiert den Zeilenumbruch. Einzelheiten zur Maskierung finden Sie im Kapitel 4 Benutzerumgebung

Für das erste Feld wird kein Kürzel benötigt, es enthält den Namen der Queue, unter dem sie dann von *lpr* angesprochen werden kann. Man kann im Übrigen mehrere Namen angeben, indem man sie durch das Pipeline-Zeichen | trennt.

Für die weiteren Felder gibt es, wie bereits gesagt, Kürzel, mit deren Hilfe die Eigenschaften der Queue festgelegt werden. Einige dieser Eigenschaften werden in Form von Texten festgelegt, wie etwa der Name der Gerätedatei, über die der Druck erfolgen soll, andere wiederum als numerischer Wert, z.B. die maximale Anzahl erlaubter Kopien eines Druckjobs. Schließlich gibt es auch noch einige Eigenschaften, die durch einen logischen Wert festgelegt werden, z.B. das Unterdrücken von Seitenvorschüben oder Header-Seiten.

Im ersten Fall folgt dem Kürzel ein Gleichheitszeichen und daran anschließend der beschreibende Text, im zweiten ein Nummernzeichen (#), gefolgt von dem gewünschten numerischen Wert, und im dritten Fall wird einfach das Kürzel (ohne Zusatz) geschrieben, um anzugeben, dass die Eigenschaft gesetzt ist. Tabelle 3.22 listet die wichtigsten Kürzel im Zusammenhang auf.



Einrichten des Drucksystems



Mögliche Kürzel für Feldeinträge der Datei /etc/printcap

Kürzel	Bedeutung
<i>lp=</i>	Name der Gerätedatei, über die gedruckt werden soll. Fehlt dieser Eintrag, wird <i>/dev/lp</i> als Gerät benutzt. Will man via Netzwerk an einem Remote-Drucker drucken lassen, so muss <i>lp=</i> ohne weitere Angabe benutzt werden, um die Voreinstellung zu überschreiben.
<i>rm=</i>	(<i>remote machine name</i>) Name eines Host-Rechners, über den remote gedruckt werden soll (beachten Sie die Angaben zu <i>lp=</i>)
<i>rg=</i>	(<i>restricted group</i>) Benutzergruppe, welche die Queue benutzen darf
<i>rp=</i>	(<i>remote printer</i>) Name des Drucker am Host
<i>lo=</i>	(<i>lock file</i>) Name der Lock-Datei (voreingestellt <i>lock</i> im Spool-Verzeichnis)
<i>st=</i>	(<i>status file</i>) Name der Statusdatei (voreingestellt <i>status</i> im Spool-Verzeichnis)
<i>sd=</i>	(<i>spool directory</i>) Name des Spool-Verzeichnisses (voreingestellt ist <i>/var/spool/lpd</i>)
<i>af=</i>	(<i>accounting file</i>) Name der Abrechnungsdatei. Über diese Datei kann die Systemverwaltung die durch den Druck verursachten Kosten nachhalten.
<i>ff=</i>	(<i>form feed</i>) Zeichenkette für Seitenvorschub, z.B. " <i>\f</i> "
<i>lf=</i>	(<i>error logging file</i>) Name der Log-Datei, in die der Druckdämon Fehlermeldungen ablegt
<i>of=</i>	(<i>output filter</i>) Name des Ausgabefilters
<i>if=</i>	(<i>input filter</i>) Name des Standardeingabefilters
<i>rs</i>	(<i>restrict remote users</i>) erlaubt nur solchen Benutzern den Zugriff auf einen Remote-Drucker, die einen Account am entsprechenden Host haben
<i>sc</i>	(<i>suppress multiple copies</i>) verhindert den Mehrfachdruck eines Dokuments, entspricht <i>mc#1</i>
<i>sf</i>	(<i>suppress form feed</i>) unterdrückt die Ausgabe eines Seitenvorschubs nach jedem Druckjob
<i>sh</i>	(<i>suppress header</i>) unterdrückt die Ausgabe einer Header-Seite vor jedem Druckjob
<i>mx#</i>	(<i>maximum file size</i>) maximale Anzahl von Blöcken, die ein Druckjob haben darf. Mit <i>mx#0</i> hebt man die Beschränkung auf.



Linux-Dateisystem



Mögliche Kürzel für Feldeinträge der Datei /etc/printcap (Forts.)



Kürzel	Bedeutung
mc#	(maximum copies) maximal zulässige Zahl an Kopien eines Druck-jobs (vergleichen Sie die Option -# des Befehls lpr)
pw#	(page width) maximale Spaltenzahl einer Druckseite
pl#	(page length) maximale Zeilenzahl einer Druckseite
px#	(page width in pixels) Seitenbreite in Pixel
py#	(page length in pixels) Seitenhöhe in Pixel
br#	(baud rate) gibt die Baudrate (für serielle Drucker) an

Tabelle 3.22



Ein Beispieleintrag für /etc/printcap könnte wie folgt aussehen:

```
laser|norm:lp=/dev/lp1:sd=/var/spool/lpd/laser:mx#0:sh
```

Die Queue kann unter den Namen *laser* oder *norm* angesprochen werden (verglei-chen Sie dazu den Schalter -P des Befehls *lpr*). Gedruckt wird über */dev/lp1*, der Spool-Bereich ist das Verzeichnis */var/spool/lpd/laser*, eine Größenbeschränkung für Druckjobs besteht nicht, und der Ausdruck einer Header-Seite wird unterdrückt.

Einen Eintrag enthält die Datei /etc/printcap normalerweise immer, dies ist die Standard-Queue *lp*. Über diese Queue wird gedruckt, wenn beim Befehl *lpr* keine Angabe für eine Queue mit dem Schalter -P erfolgt.

Über die Umgebungsvariable *PRINTER* kann ein Benutzer im Übrigen festlegen, dass eine andere Queue standardmäßig von *lpr* benutzt wird. Durch Eingabe von

```
PRINTER=laser
```

wird also festgelegt, dass *lpr* standardgemäß die Queue *laser* benutzen soll.

Druckerfilter

Sie wurden schon bei der Beschreibung des Spool-Bereichs erwähnt, die Druckerfilter. Mit ihrer Hilfe werden die zu druckenden Dateien in eine für den Drucker geeig-nete Form gebracht. Sie gleichen in ihrer Aufgabenstellung also den Druckertreibern von DOS/Windows, werden unter Linux aber „einfach“ als Shell-Skripte realisiert.

Ein Druckerfilter arbeitet in typischer UNIX-Manier. Er erhält über die Standardein-gabe (die Tastatur oder die Ausgabe eines anderen Programms) Daten, die er ver-arbeitet und an die Standardausgabe (Bildschirm oder die Eingabe eines anderen Programms) weiterleitet. Die Verarbeitung der Daten besteht in diesem Fall darin,



Einrichten des Drucksystems



sie in einer für den Drucker geeigneten Form aufzubereiten. Der Druckdämon *lpd* sorgt schließlich dafür, dass die Daten auf dem Drucker statt auf dem Bildschirm ausgegeben werden.

Die Situation wird allerdings dadurch erschwert, dass ein Druckerfilter Daten unterschiedlichster Herkunft bearbeiten muss. Gedruckt werden sollen einfache ASCII-Texte, formatierte Texte, Grafiken oder Bilddateien.

Vorausgesetzt, man hat für jedes vorliegende Dateiformat eine Konvertierungsroutine, dann lässt sich das Problem lösen, indem man

- für jeden zu druckenden Dateityp einen eigenen Filter bereitstellt, der die jeweilige Konvertierungsroutine enthält und der über eine entsprechende Option des Befehls *lpr* angesprochen wird, oder
- einen „intelligenten“ Filter verwendet, der in einem ersten Schritt den vorliegenden Dateityp erkennt (etwa mithilfe des Befehls *file*) und dann die Datei mit der entsprechenden Subroutine umwandelt.

Für den ersten Fall werden eine Reihe weiterer Kürzel für Druckerfilter in der Datei */etc/printcap* bereitgestellt. Diese entsprechen jeweils einer Option des Befehls *lpr*. Tabelle 3.23 nennt Ihnen die Kürzel und deren Bedeutung.

Mögliche Kürzel für Druckerfilter in der Datei */etc/printcap*



Kürzel Bedeutung

<i>if=</i>	Standardfilter für ASCII-Dateien (keine Option des Befehls <i>lpr</i> nötig, da dieser Filter angesprochen wird, wenn kein anderer festgelegt wird)
<i>cf=</i>	zur Ausgabe von Dateien im <i>cifplot</i> -Format (Option des Befehls <i>lpr : -c</i>)
<i>df=</i>	zur Ausgabe von Dateien im DVI-Format, z.B. T _E X-Dateien (Option des Befehls <i>lpr : -d</i>)
<i>gf=</i>	zur Ausgabe von Dateien im <i>plot</i> -Format (Option des Befehls <i>lpr : -g</i>)
<i>nf=</i>	zur Ausgabe von Dateien im <i>ditroff</i> -Format (Option des Befehls <i>lpr : -n</i>)
<i>rf=</i>	zur Ausgabe von Fortran-Dateien (Option des Befehls <i>lpr : -f</i>)
<i>tf=</i>	zur Ausgabe von Dateien im <i>troff</i> -Format (Option des Befehls <i>lpr : -t</i>)
<i>vf=</i>	zur Ausgabe von gerasterten Bildern (Option des Befehls <i>lpr : -v</i>)

Tabelle 3.23



Linux-Dateisystem



Wollen Sie etwa über eine Queue namens *latex* Dateien im DVI-Format ausdrucken und es steht Ihnen ein Filter namens *dvi* im Verzeichnis */var/lib/filters/dvi* zur Verfügung, so könnte ein entsprechender Eintrag in */etc/printcap* wie folgt lauten:

```
latex:lp=/dev/lp1:\n      :sd=/var/spool/lpd/latex:\n      :mx#0:\n      :vf=/var/lib/filters/dvi:\n      :sh
```

Gedruckt würde eine Datei namens *mathediplom* mithilfe der folgenden Befehlszeile:

```
lpr --Platex --v mathediplom
```

Der Filter *apsfilter*

Mit *apsfilter* stellen die meisten Linux-Distributionen heutzutage einen intelligenten Filter zur Verfügung, der

Überlegen Sie doch einmal, was *aps* bedeuten könnte!

- eigenständig das Dateiformat erkennt (unterstützt werden zurzeit unter anderem ASCII, DVI, PS – PostScript als Standardformat der UNIX-Welt, GIF, TIFF, PBM, Sun Raster, X11-Bitmap, diverse Datenformate, wie etwa PCL)
- gegebenenfalls die Datei dekomprimiert (unterstützt werden zurzeit *compress*, *gzip* und *freeze*) und
- die Datei entsprechend ihrem Format und der gewählten Queue filtert

Der *apsfilter* besteht im Wesentlichen aus einem Shell-Skript namens */var/lib/apsfilter/apsfilter*. Die Konfiguration erfolgt über die Datei */etc/apsfilterrc.druckernname*, die beim Einrichten des Filters aus */etc/apsfilterrc* abgeleitet wird. Darüber hinaus müssen natürlich entsprechende Einträge in der Datei */etc/printcap* den Filter für die entsprechenden Queues anmelden.

Sollten Sie einen Farbdrucker benutzen, dann können Sie fünf Queues verwenden, um via *apsfilter* auf ihm zu drucken – auch diese Queues werden bei der Einrichtung erzeugt:

- *auto-color*: Warteschlange für alle Dateiformate, Ausdruck in Farbe
- *auto-mono*: Warteschlange für alle Dateiformate, Ausdruck schwarzweiß
- *ascii-color*: Warteschlange für ASCII-Texte, Ausdruck in Farbe
- *ascii-mono*: Warteschlange für ASCII-Texte, Ausdruck schwarzweiß
- *raw*: Warteschlange für Dateien, die bereits im druckerspezifischen Format vorliegen. Eine Konvertierung findet nicht mehr statt.



Einrichten des Drucksystems



Da Sie mehrere Drucker verwenden können, die außerdem auf unterschiedlichem Papierformat und in unterschiedlicher Auflösung drucken können, werden diese Attribute dem Namen der entsprechenden Warteschlange hinzugefügt, so dass er beispielsweise wie folgt lauten könnte:

```
ljet4-a4-ascii-mono-360
```

Dies ist eine Warteschlange, für einen HP Laserjet 4, über die ASCII-Texte schwarz-weiß auf DIN-A4-Papier mit einer Auflösung von 360 dpi gedruckt werden sollen.

Doch keine Angst: Diese üppige Namensvergabe ist weitgehend automatisiert und erfolgt bei der Einrichtung des *apsfilters* über ein menügesteuertes *SETUP*-Tool.

Dieses Tool erledigt die folgenden Aufgaben:

- Eintragen der Queue-Informationen (in Abhängigkeit von der Wahl des Druckers) in die Datei */etc/printcap*
- Anlegen der entsprechenden Verzeichnisse im Spool-Bereich */var/spool/lpd*
- Erzeugen einer Konfigurationsdatei mit dem Namen */etc/apsfilterrc.druckername* (im obigen Beispiel wäre dies etwa */etc/apsfilterrc.ljet4*)
- Erstellen der Datei */etc/apsfilterrc* (falls diese nicht schon existiert)
- Bereitstellen der entsprechenden Druckerfilter wie beispielsweise */var/lib/apsfilter/bin/ljet4-a4-ascii-mono-360* (diese sind nichts anderes als symbolische Links auf */var/lib/apsfilter/apsfilter*).

Das *SETUP*-Tool für den Druckerfilter *apsfilter* lässt sich wie folgt starten (es wird vorausgesetzt, dass Sie als Benutzer *root* eingeloggt sind):

```
$ cd /var/lib/apsfilter←  
$ ./SETUP←
```

Beachten Sie, dass Sie als Benutzer *root* das aktuelle Verzeichnis gewöhnlich nicht im Suchpfad haben, daher also der Aufruf von *SETUP* mit vorangestellter Angabe des aktuellen Verzeichnisses (*./*).

Nach einem Informationsbildschirm (kommt Ihnen das nicht irgendwie bekannt vor) erscheint das Hauptmenü mit den folgenden vier Menüpunkten:

- *EXIT* Verlassen des *SETUP*-Tools,
- *LISTING* Anzeige der vorhandenen Filter,
- *ENTRY* Anlegen eines neuen bzw. Ändern eines vorhandenen Filters,
- *DELETE* Löschen eines Filters.

—○ Als Filter werden nur *apsfilter* bearbeitet



Linux-Dateisystem



Zum Einrichten wählt man natürlich den Menüpunkt *ENTRY*. Dies führt zu einem Untermenü mit folgenden Einträgen (vergleichen Sie auch Abbildung 3.4):

- *RETURN* Rückkehr zum Hauptmenü,
- *DEVICE* Festlegen des Anschlusses (parallel – z.B. */dev/lp1* – oder seriell – z.B. */dev/ttyS1*),
- *PRINTER* Auswahl des Druckers, auf dem die Ausgabe erfolgen soll, über ein Untermenü (die Auswahl ist insbesondere dann wichtig, wenn der Drucker nicht postscriptfähig ist),
- *PAPER* das Papierformat (normalerweise DIN A4),
- *COLOR* für den Fall, dass Sie einen Farbdrucker benutzen wollen,
- *SPECIAL* erlaubt über ein Untermenü einige Spezialeinstellungen wie die Vergabe eines alternativen Namens für die Queue oder das Unterdrücken von Seitenvorschüben,
- *RESET* stellt die ursprünglichen Einstellungen wieder her,
- *ADD* Anlegen eines neuen Filters mit den getroffenen Einstellungen,
- *OVERWRITE* ändert einen vorhandenen Filter,
- *DELETE* löscht die getroffenen Einstellungen.

Bis auf das Menü *PRINTER* sind alle Menüs recht einfach zu nutzen. Mithilfe der Cursortasten lassen sich Menüpunkte (bzw. *OK* zur Bestätigung der Einstellung und *Cancel* zum Abbrechen) wählen. Für Texteingaben (z.B. zur Angabe der Schnittstellen) werden in der Regel Voreinstellungen bereitgestellt, zumindest finden Sie aber einen erklärenden Text in dem Fenster (wenn auch in Neudeutsch).

Die Schwierigkeit beim Menü *PRINTER* besteht darin, dass Sie hier Ihren Drucker auswählen müssen. Ist er ein PostScript-Drucker, dann fällt die Wahl leicht: Sie wählen den entsprechenden Untermenüpunkt, legen die Druckauflösung fest – und Sie sind fertig.

Ist Ihr Drucker nicht postscriptfähig, können Sie über entsprechende Listen einen Druckertyp auswählen. Dabei ist zu beachten, dass nicht alle auf dem Markt verfügbaren Drucker unterstützt werden. Als generelle Richtlinie für einen solchen Fall gilt, dass Sie einen möglichst ähnlichen Drucker wählen sollten. Haben Sie etwa einen HP Laserjet 5 (ist zurzeit nicht in der Liste der unterstützten Drucker enthalten), so wählen Sie einfach einen HP Laserjet 4!

Schließlich erlaubt das Menü *PRINTER* die Einrichtung eines Remote-Druckers, also eines Druckers, der an einem anderen Linux-Rechner angeschlossen ist. In diesem Fall sind der Name des Host-Rechners und der Name des Remote-Druckers einzutragen. Gesetzt den Fall, der Remote-Drucker ist am Host-Rechner geeignet ein-



Einrichten des Drucksystems



gerichtet und Sie haben Zugriff auf diesen Rechner, lässt sich nun Ihr Druckauftrag über diesen entfernten (*remote*) Drucker ausführen. Auf diese Weise lassen sich in einem Netzwerk Druckaufträge an einem speziellen Druckserver bündeln.

Nachdem Sie im Menü *ENTRY* die nötigen Menüpunkte ausgewählt und Ihre Entscheidungen getroffen haben (im erklärenden Text sollten die Einstellungen in der Form "*ljet4*" "mono" "a4" "360" erscheinen), können Sie durch Wahl von *ADD* einen neuen Filter anlegen.

Wollen Sie das Ergebnis Ihrer Bemühungen überprüfen, so sollten Sie sich zunächst den Inhalt des Verzeichnisses */var/spool/lpd* anzeigen lassen. Dies sollte etwa zu folgendem Ergebnis führen:

```
ljet4-a4-ascii-mono-360    ljet4-a4-auto-mono-360
ljet4-a4-raw           lpd.lock
```

Die Verzeichnisse *ljet4** verweisen auf folgende Situation: Sie haben einen HP Laserjet 4 als Drucker festgelegt, der im DIN-A4-Format bei einer Auflösung von 360 dpi drucken soll. Es sind drei Warteschlangen eingerichtet worden, für ASCII-Dateien, vorformatierte Dateien (*raw*) und alle übrigen Dateien. Der Drucker ist nur für Schwarzweiß-Druck vorgesehen (*mono*). Die Datei *lpd.lock* ist hier eine allgemeine Lock-Datei.

Ein Blick in die Datei */etc/printcap* zeigt folgende neue Zeilen:

```
### BEGIN apsfilter: ### ljet4 a4 mono 360 ###
#   Warning: Configured for apsfilter, do not edit the
labels!
#           apsfilter setup Sun Nov 22 12:00:56 MET 1998
#
ascii|lp1|ljet4-a4-ascii-mono-360|ljet4 a4 ascii mono
360:\n
:lp=/dev/lp1:\n
:sd=/var/spool/lpd/ljet4-a4-ascii-mono-360:\n
:lf=/var/spool/lpd/ljet4-a4-ascii-mono-360/log:\n
:af=/var/spool/lpd/ljet4-a4-ascii-mono-360/acct:\n

:if=/var/lib/apsfilter/bin/ljet4-a4-ascii-mono-360:\n
:la@:mx#0:\n
:sh:sf:
#
lp|lp2|ljet4-a4-auto-mono-360|ljet4 a4 auto mono 360:\n
:lp=/dev/lp1:\n
:sd=/var/spool/lpd/ljet4-a4-auto-mono-360:\n
:lf=/var/spool/lpd/ljet4-a4-auto-mono-360/log:\n
:af=/var/spool/lpd/ljet4-a4-auto-mono-360/acct:\n

:if=/var/lib/apsfilter/bin/ljet4-a4-auto-mono-360:\n
:la@:mx#0:\n
:sh:sf:
```



Linux-Dateisystem



```
#  
raw|lp3|ljet4-a4-raw|ljet4 a4 raw:\  
:lp=/dev/lp1:\\  
: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:\\  
:sh:sf:  
#  
### END    apsfilter: ### ljet4 a4 mono 360 ###
```

Es ist leicht zu erkennen, dass in diesem Fall Einträge für drei neue Queues angelegt wurden, dies entspricht den drei neuen Verzeichnissen in `/var/spool/lpd`. Die Namen der Queues, z.B. `ljet4-a4-raw`, werden gebildet, wie oben bereits besprochen. Spool-Verzeichnis, Druckeranschluss (`/dev/lp1`), die benötigten Dateien (z.B. der Filter `/var/lib/apsfilter/bin/ljet4-a4-raw`) und besondere Einstellungen werden genannt. Neben den langen Namen für die einzelnen Queues werden auch abkürzende (z.B. `raw`) angegeben.

Beachten Sie bitte, dass die kommentierenden Zeilen zu Beginn und am Ende nicht geändert werden sollten, da ansonsten ein reibungsloses Funktionieren der Warteschlangen nicht gewährleistet ist.

Werfen Sie zum Schluss noch einen Blick auf die Konfigurationsdatei `/etc/apsfilterc.druckernname`, (`druckernname` entspricht dem Namen des im *SETUP* gewählten Druckers). Ihr Inhalt besteht aus bis zu 12 Definitionen:

```
# (0)  
#REMOTE_PRINTER="remote"  
# (1)  
#TEXINPUTS=  
# (2)  
#PRINT_DVI=  
# (3)  
#PRINT_RAW_SETUP_PRINTER=  
# (4)  
#PRINT_RAW_SUPPRESS_FORMFEED="yes"  
# (5)  
#if [ "$COLOR" = "mono" ] ; then  
#   GS_FEATURES=  
#   PRELOADS=  
#else  
#   GS_FEATURES=  
#   PRELOADS=  
#fi  
# (6)  
#USE_RECODE_NOT_A2PS="yes"  
# (7)  
#FEATURE=
```



Einrichten des Drucksystems

```
# (8)
#A2PS_OPTS=
# (9)
#DVIPS_MODE=
# (10)
#GS_RESOL=
# (11) Works only with ascii or lp on non PS printers with
gs
#      NOTE: |`la" should be disabled in printcap (done
with :la@:)
#DO_ACCOUNTING=yes
```

Die durch # eingeleiteten Zeilen sind, wie so oft in Linux, Kommentarzeilen. In vorliegendem Beispiel sind alle Zeilen Kommentare, was bedeutet, dass bei dieser Installation keine besonderen Einstellungen getroffen worden sind. Mit dem nötigen Wissen um diese 12 Definitionen lassen sich aber Einstellungen des *apsfilters* anpassen, indem die entsprechenden Kommentarzeichen aus der Datei entfernt und die Variablen auf die gewünschten Werte gesetzt werden.

Für eine genauere Beschreibung der Definitionen sollten Sie das Online-Manual für *apsfilter* zu Rate ziehen oder einfach die Datei */etc/apsfilterrc* lesen. Hier wird nur ein Überblick über die Einstellungsmöglichkeiten gegeben.

Variable der Datei */etc/apsfilterrc*

Variable	Bedeutung
<i>REMOTE_PRINTER</i>	Weiterleitung der Daten an einen Remote-Drucker
<i>TEXINPUTS</i>	Pfade für Bilder, die zur Druckzeit in eine DVI-Datei eingefügt werden sollen
<i>PRINT_DVI</i>	Definition eines eigenen DVI-Filters
<i>PRINT_RAW_SETUP_PRINTER</i>	Übergabe von Escape-Sequenzen an den Drucker, wenn dieser im raw-Modus arbeitet
<i>PRINT_RAW_SUPPRESS_FORMFEED</i>	Unterdrücken von Seitenvorschüben
<i>GS_FEATURES</i>	Aktivieren spezieller Eigenschaften von (nicht postscriptfähigen) Druckern
<i>PRELOAD</i>	Laden einer entsprechenden PostScript-Datei
<i>USE_RECODE_NOT_A2PS</i>	Umwandeln von ASCII-Daten in PostScript-Daten mittels <i>a2ps</i> oder nicht



Linux-Dateisystem

Variable der Datei */etc/apsfilterrc* (Forts.)

Variable	Bedeutung
<i>FEATURE</i>	Art der Umwandlung durch <i>a2ps</i> . Wird ignoriert, wenn <i>USE_RECODE_NOT_A2PS</i> auf <i>yes</i> gesetzt ist. Standardeinstellung ist im Übrigen <i>2n</i> , was eine Ausgabe von zwei Seiten auf einem Blatt (also quer) bedeutet.
<i>A2PS_OPTS</i>	Optionen für <i>a2ps</i> . Wird ignoriert, wenn <i>USE_RECODE_NOT_A2PS</i> auf <i>yes</i> gesetzt ist.
<i>DVIPS_MODE</i>	Einrichten eines DVIPS-Treibers
<i>GS_RESOL</i>	variable Auflösung
<i>DO_ACCOUNTING</i>	Accounting ermöglichen

Tabelle 3.24

Vielleicht haben Sie sich schon die ganze Zeit gefragt, wie ein nicht postscriptfähiger Drucker von einem *apsfilter* bedient werden kann. Schließlich ist die Ausgabe eines solchen Filters standardgemäß für einen PostScript-Drucker gedacht. Dies ist der Ansatzpunkt des Programmpakets *gs* (Ghostscript).

Dies ist ein Filter, der PostScript-Dateien als Eingabe erwartet, die er in einem Format für den gewählten Drucker ausgibt. Vereinfacht ausgedrückt, ist Ghostscript eine Sammlung von Druckertreibern. Wollen Sie sich nähere Informationen über die Möglichkeiten dieses Pakets verschaffen, so können Sie am Prompt einfach eingeben:

```
gs --h
```

Der *apsfilter* stützt sich auf *gs*, wenn es um nicht postscriptfähige Drucker geht. Die Liste der im SETUP-Tool genannten Drucker wird durch Ghostscript festgelegt.

In der Tat, es kann zu der merkwürdigen Konstellation kommen, dass der Inhalt einer gewöhnlichen ASCII-Datei mit dem *apsfilter* in PostScript-Daten umgewandelt wird, um dann, da der zugeordnete Drucker PostScript nicht versteht, wieder in ASCII-Daten zurückverwandelt zu werden. Dieser ungewöhnliche Aufwand wird aber dadurch gerechtfertigt, dass Sie mit dem *apsfilter* eine Möglichkeit zur Verfügung haben, Dateien (fast) beliebigen Formats auf (fast) beliebigen Druckern zu drucken. Wenn Sie die Einstellungen wieder loswerden wollen, löschen Sie einfach die Einträge in der */etc/printcap*. Das Werkzeug *apsfilter* ist zwar ein komfortables Tool zur Verwaltung ausgefallener Druckformate, leider ist es aber noch etwas anfällig in der Anwendung.





Aufbau eines Dateisystems

Werfen Sie nun einen Blick hinter die Kulissen und untersuchen Sie, wie Linux Dateien verwaltet. Als einfaches Modell dient Ihnen dabei zunächst das Standard-UNIX-Dateisystem.

Zu einer Datei gehören neben den Daten auch die Verwaltungsinformationen. Diese werden in zwei sauber voneinander getrennten Bereichen gespeichert:

- dem Verzeichnis
- der I-Node

Im Verzeichnis werden der Dateiname und die I-Node-Nummer, der so genannte „Link“, abgelegt. In der I-Node selbst befinden sich alle anderen Angaben zur Datei:

- der Dateityp
- die Rechtetabelle
- die Anzahl der Links
- der Dateibesitzer (genauer seine UID)
- die Gruppe (genauer ihre GID)
- die Größe in Byte (bzw. die Gerätenummern)
- Datum und Uhrzeit der letzten Änderung
- Datum und Uhrzeit der letzten Änderung der I-Node
- Datum und Uhrzeit der letzten Benutzung
- Angaben zum Ort, wo auf der Platte die Daten gespeichert sind

Dies sind bis auf den letzten Eintrag all die Informationen, die `ls -l` liefert. Beispiel:

```
-rw-r--r- 2 root root 889 Dec 29 05:34 passwd
```

Die Verzeichnisinhalte sind ihrerseits die Daten des jeweiligen Verzeichnisses. Insgesamt muss man unterscheiden zwischen den Daten und den I-Nodes. Daten und I-Nodes werden auf dem Datenträger getrennt gespeichert.

Durch Formatierung erhält eine Festplatte oder Diskette eine Einteilung in Spuren/Zylinder und Sektoren. Dadurch lassen sich vom System einzelne Blöcke ansprechen, etwa der Block auf Spur 0 in Sektor 1. Die Größe dieser Blöcke beträgt normalerweise 512 Byte.



Linux-Dateisystem



Zu speichernde Daten werden immer blockweise abgelegt. Dies spart erheblich Zeit, ist aber auch eine Verschwendungen von Speicherkapazität.

Vergleichen Sie einmal eine Festplatte und ihre Einteilung in Blöcke mit einem großen Schrank mit vielen Schubfächern. Nehmen Sie weiter an, Sie haben eine Reihe von Informationen auf einzelne Blätter geschrieben und wollen diese im Schrank ablegen. Man kann nun alle Blätter in ein Schubfach legen und, wenn dies voll ist, das nächste benutzen und so fort. So spart man Platz.

Was aber, wenn bestimmte Informationen gesucht werden? Dann müssen theoretisch alle gefüllten Schubfächer durchsucht werden. Natürlich muss auch jedes einzelne Blatt daraufhin untersucht werden, ob auf ihm Angaben zum gewünschten Thema enthalten sind.

Deshalb kann man bei der Einordnung der Blätter auch nach folgendem Ordnungsprinzip verfahren:

- Blätter, die Informationen zum selben Thema enthalten, werden in einem gemeinsamen Schubfach abgelegt. Reicht dies nicht, nimmt man ein nächstes und so weiter. Dabei ist darauf zu achten, dass „fremde“ Blätter *nicht* in diese Fächer gelangen.
- Einige der Schubfächer werden reserviert für besondere Informationen, nämlich Blätter, auf denen steht, in welchen Fächern welche Informationen enthalten sind.

Genau dies ist das Verfahren, nach dem Ihr Dateisystem verwaltet wird: Statt der Fächer nimmt Linux natürlich Blöcke, die thematisch geordneten Informationen sind die Dateiinhalte. In den meisten Fällen werden mehrere Blöcke zu einer Verwaltungseinheit zusammengefasst. So muss man gegebenenfalls unterscheiden zwischen logischen Blöcken, den Verwaltungseinheiten, und physikalischen Blöcken, den 512 Byte großen Einheiten, die sich durch die Einteilung in Spuren und Sektoren ergeben.

Der für die Dateiverwaltung reservierte Bereich ist die so genannte „I-Node-Liste“. Die eigentlichen Daten befinden sich im Datenbereich.

Die vorhandenen (physikalischen) Blöcke werden nun einfach durchnummeriert, beginnend mit 0.

Die Nutzung der Blöcke durch Linux entnehmen Sie *Tabelle 3.25*.



Aufbau eines Dateisystems



Nutzung der Blöcke eines Datenträgers durch Linux

Block	Verwendung
0	Bootblock. Enthält eine einfache Routine zum Laden (Booten) des Betriebssystems. Auf diesen Block greift der Rechner beim Hochfahren automatisch zu.
1	Superblock. Er enthält allgemeine Informationen über das Dateisystem.
2 bis x	reserviert für die I-Node-Liste – die Größe von x hängt natürlich wesentlich von der Größe der Platte ab
x+1 bis z	reserviert für Daten

Tabelle 3.25



Der Superblock enthält unter anderem folgende Angaben:

- Größe des Dateisystems in (physischen) Blöcken
- Größe der logischen Blöcke
- Name des Datenträgers, auf dem sich das Dateisystem befindet (z.B. `/dev/hda3`)
- Name des Dateisystems (z.B. `/` – das ist das *root*-Dateisystem, das immer beim Start gemountet wird)
- Größe der I-Node-Liste
- Zeiger auf den ersten freien I-Node
- Zeiger auf den ersten freien Datenblock
- Diverse Schalter, die den Zustand des Dateisystems kennzeichnen.

Die Bedeutung des Superblocks wird damit erklärlich, dass beim Anlegen einer neuen Datei das System zunächst auf den Superblock zugreift, um die Adresse des ersten freien Datenblocks und der ersten freien I-Node zu erhalten. Beim Löschen (des letzten Links) einer Datei müssen diese Adressen neu festgelegt werden.

Zugriff auf eine Datei

Für jede Datei existiert nun in der I-Node-Liste ein Eintrag, die I-Node. Über sie erfolgt der Zugriff auf die Datei. 13 Adresseinträge in der I-Node beschreiben den Ort auf der Platte, an dem die Daten zu finden sind. Die ersten zehn dieser Einträge beinhalten direkt bis zu zehn Adressen von Blocks, die Daten der Datei enthalten.



Linux-Dateisystem

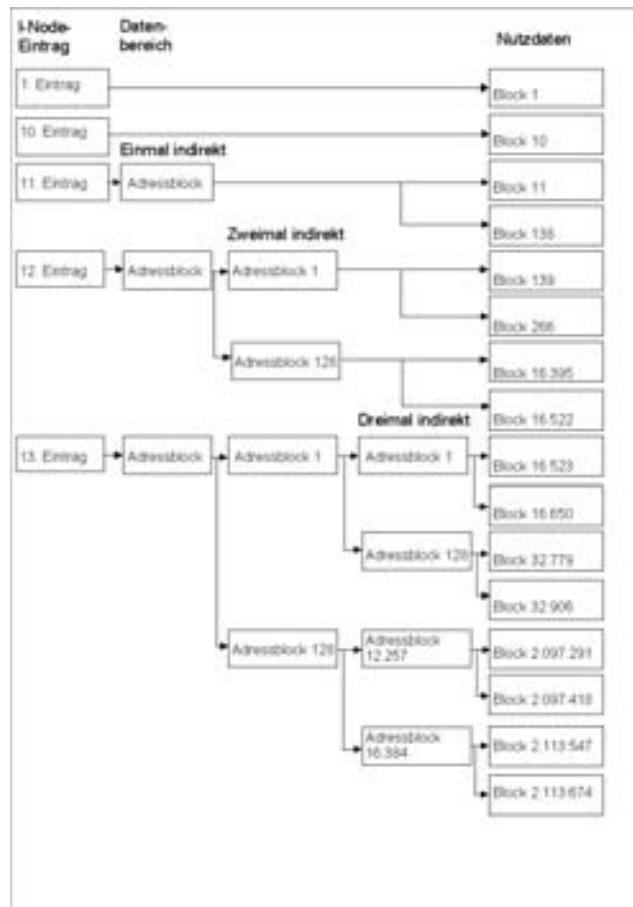


Dies reicht bei größeren Dateien natürlich nicht aus. In diesem Fall reserviert Linux einen Block aus dem Datenbereich. Dieser wird über den elften Eintrag der I-Node adressiert und enthält keine Daten, sondern bis zu 128 Adressen weiterer Datenblöcke.

Reicht auch dies nicht, werden (bis zu) 129 Blöcke des Datenbereichs reserviert. Der erste davon wird über den zwölften Adresseintrag der I-Node erreicht und enthält die bis zu 128 Adressen der anderen Blöcke. Diese ihrerseits enthalten jeder bis zu 128 Adressen von Datenblöcken.

Als letzte Möglichkeit wird über den 13. Adresseintrag eine dreifach indirekte Adressierung auf bis zu 128³ Datenblöcke vorgenommen. Dieses für große Dateien sicherlich umständliche Verfahren soll in *Abbildung 3.4* verdeutlicht werden.

Abb. 3.4 ○
Zugriff auf die Daten
einer Datei



Aufbau eines Dateisystems



Zu den Grundlagen des Standard-UNIX-Dateisystems gehört es noch, dass die Einträge in einem Verzeichnis exakt 16 Byte lang sind. Die beiden ersten Byte sind für die I-Node, die restlichen 14 für den Dateinamen. Hieraus ergaben sich zwei wesentliche Einschränkungen:

- Ein Dateiname konnte maximal 14 Zeichen lang sein.
- In einem Dateisystem konnten maximal $65.536 = 2^{16}$ Dateien enthalten sein, da nur 16 Bit (= 2 Byte) lange Einträge für die I-Node-Nummern zugelassen waren.

Linux hat diese Grenzen durch eine Umstrukturierung der Dateiverwaltung überwunden. Das von Linux gewöhnlich benutzte Dateiverwaltungssystem *ext2* enthält eine Reihe von Verbesserungen, die aber alle auf dem vorgestellten Standard-UNIX-Dateisystem aufbauen.

Ein Verzeichnis enthält nun nicht mehr nur 16 Byte je Eintrag, für den Dateinamen sind bis zu 255 Zeichen zugelassen. Sinnvollerweise wird ein Verzeichnis nun nicht mehr 255 Byte für jeden Namen reservieren, stattdessen werden der Dateiname und seine Länge gespeichert.

Einschneidende Veränderungen hat auch die I-Node selbst erfahren. Unter anderem beinhaltet sie nun Informationen, die es gestatten, die gesamte I-Node-Liste in einer so genannten „Hash-Tabelle“ zu verwalten. Dadurch wird die Suche nach freien I-Nodes erheblich beschleunigt. Außerdem wurden Algorithmen zur beschleunigten Suche nach den Daten umfangreicher Dateien eingeführt.

Wir ersparen es Ihnen an dieser Stelle, auf Einzelheiten einzugehen, da ihr Verständnis tiefer gehende Programmierkenntnisse erfordert.

Eine wesentliche Neuheit von Linux ist das *Virtuelle Dateisystem* (**VFS**, **Virtual Filesystem Switch**, kurz VFS). Der englische Begriff Switch (dt. Schalter) gibt dabei die Bedeutung des VFS besser wieder.

Es ist eigentlich eine Benutzerschnittstelle. Das VFS gibt dem Benutzer die Möglichkeit, in einem Dateisystem mit den bekannten Befehlen dessen Daten zu bearbeiten. Mit *cp* werden eben Dateien kopiert und mit *rm* gelöscht. Wie die eigentliche Verwaltung des Systems dabei aussieht, kann dem Benutzer egal sein.

Intern jedoch kann Linux sehr wohl unterscheiden, ob das zugrunde liegende reale Dateisystem ein Standard-UNIX-Dateisystem ist oder *ext2* oder *HPFS* (das Dateisystem von OS/2) oder *FAT* (das MS-DOS-Dateisystem). Insbesondere die Eigenschaft, Fremddateisysteme zu verstehen, ermöglicht es Linux, Daten zwischen diesen verschiedenen Systemen auszutauschen.

Bei der Installation von Linux hatten Sie möglicherweise auf Ihrer Festplatte eine DOS-Partition. Diese wird von Linux nicht nur erkannt, Linux ermöglicht es Ihnen sogar, die DOS-Partition in das System einzubeziehen, ja, man kann sogar Linux in der DOS-Partition installieren!



Linux-Dateisystem



Angenommen, Sie haben eine eigene Linux-Partition mit dem *ext2*-Dateisystem angelegt und die DOS-Partition ins System integriert. Dann befinden sich zwei Dateisysteme auf Ihrer Platte, auf die Linux zugreifen kann: das *FAT*-Dateisystem in der DOS-Partition und das *ext2*-Dateisystem in der Linux-Partition. Verwaltungstechnisch unterscheiden sich beide Systeme erheblich. Die Verwaltungsdaten werden unter MS-DOS im Directory und in der FAT abgelegt. Diese Verwaltungsdaten unterscheiden sich darüber hinaus in vielen Einzelheiten. So kennt MS-DOS keine Rechte, dafür aber Attribute, keine Links und nur ein Datum.

VFS gestattet es dem Benutzer nun, in beiden Dateisystemen dieselben Befehle anzuwenden, so dass z.B. das Kopieren einer Datei aus der Linux-in die DOS-Partition möglich ist. Intern werden dabei zur Bearbeitung der Dateien allerdings unterschiedliche Routinen benutzt. Das Ablegen der Daten im Datenbereich erfolgt in UNIX-Dateisystemen wie *ext2* über den Zugriff auf das Verzeichnis und die I-Node, in MS-DOS über den Zugriff auf das Directory und die FAT.

Dieser Wechsel (switch) zwischen den Dateisystemen ist der Grund dafür, dass Linux so „problemlos“ auf DOS-Daten zugreifen kann.

Arbeiten mit Dateisystemen

Mit dem Begriff „Dateisystem“ werden sowohl die darin enthaltenen Daten als auch die Art der Datenverwaltung bezeichnet

Ein Dateisystem ist immer an einen bestimmten Datenträger gebunden. Es kann sich auf einer Diskette befinden, in einer Partition auf der Festplatte oder auf einer CD-ROM. In diesem Abschnitt werden die Tools zur Verwaltung der realen Dateisysteme vorgestellt. Beachten Sie dabei, dass diese Tätigkeiten einschneidende Eingriffe in Ihr System bedeuten können. Die Autoren übernehmen keine Haftung für Schäden irgendwelcher Art, die durch das Testen der diversen Beispiele auftreten können.

Natürlich sei noch vermerkt, dass Sie diese Arbeiten als Superuser durchführen müssen.

Erstellen eines Dateisystems

Wie legt man überhaupt ein Dateisystem an? Dies erledigt der Befehl *mkfs* (make filesystem).

mkfs

Die Syntax des Befehls *mkfs* lautet:

```
mkfs [ -option] dateisysstem blöcke
```

Erstellen eines Dateisystems



Der Parameter *dateisystem* kann der Name einer Gerätedatei oder eines Verzeichnisses sein. Letzter Fall setzt sehr genaue Kenntnis des *mkfs*-Befehls voraus und sollte vermieden werden. Eine Gerätedatei könnte zum Beispiel */dev/hda3* sein. Dies bezeichnet eine Partition der Festplatte.

Der Parameter *blöcke* bezeichnet die Anzahl der physikalischen Blöcke auf dem Datenträger, die für das Dateisystem benutzt werden sollen.

Um eine Diskette mit einem Dateisystem zu versehen, lautet der entsprechende Befehl in seiner einfachsten Form:

```
mkfs /dev/fd0h1440 2880
```

Gemeint ist eine 3-Zoll-HD-Diskette im Laufwerk A:. Benutzt werden sollen alle 2.880 physikalischen Blöcke der Diskette.

Die wichtigsten Optionen für *mkfs* sind in *Tabelle 3.26* enthalten.

Optionen des Befehls *mkfs*

Option	Bedeutung
<i>-t dateisystemtyp</i>	legt fest, welcher Dateisystemtyp verwendet werden soll. Mögliche Werte entnehmen Sie der <i>Tabelle 3.27</i> .
<i>-c</i>	prüft den Datenträger auf defekte Blöcke (bad blocks)
<i>-l dateiname</i>	liest <i>dateiname</i> als Liste von bad blocks. Diese bad blocks werden von der Benutzung ausgeschlossen
<i>-V</i>	veranlasst <i>mkfs</i> , ausführliche Informationen über die einzelnen Arbeitsgänge auszugeben

Tabelle 3.26



Zu beachten ist, dass die Erstellung des Dateisystems nicht durch *mkfs* selbst erfolgt. Vielmehr ruft es ein dateisystemspezifisches Programm auf, etwa *mkfs.ext2*, das die eigentliche Arbeit übernimmt (achten Sie gegebenenfalls auf die entsprechenden Meldungen). Wenn man das Front-End *mkfs* nicht besitzt, kann man auch jederzeit mit *mke2fs* arbeiten. Der Befehl lautet dann:

```
mke2fs /dev/hdb5
```

Verfügbare Werte für Dateisystemtypen sind die aus *Tabelle 3.27*.



Linux-Dateisystem

Dateisystemtypen für *mkfs* und andere Dateisystembefehle

Wert	Dateisystem
<i>ext</i>	Extended Filesystem, ein Vorläufer von <i>ext2</i>
<i>ext2</i>	Second Extended Filesystem, das aktuelle Standarddateisystem für Linux
<i>minix</i>	Minix-Dateisystem, Vorläufer von <i>ext</i> . War das ursprüngliche Dateisystem von Linux
<i>xia</i>	ein nach seinem Entwickler benanntes Dateisystem, das <i>ext2</i> sehr ähnlich ist. Wird aber selten benutzt.
<i>nfs</i>	Network File System (NFS). Gestattet den Zugriff auf Dateien via Netzwerk.
<i>sysv</i>	Dateisystem gemäß System V
<i>coherent</i>	Dateisystem der UNIX-Version Coherent (war lange Jahre eine kostengünstige Alternative zu kommerziellen UNIX-Produkten)
<i>xenix</i>	Dateisystem von Xenix. (Eine Entwicklung von Microsoft für den PC. Wurde in späteren Jahren von der Firma SCO übernommen. Xenix war jahrelang das Standard-UNIX-System für den PC.)
<i>msdos</i>	das FAT-Dateisystem von MS-DOS
<i>umsdos</i>	Dateisystem, mit dessen Hilfe Linux in einer DOS-Partition installiert werden kann
<i>hpfs</i>	High Performance Filesystem, das Dateisystem von OS/2
<i>iso9660</i>	ISO-9660-Dateisystem, das für die meisten CD-ROMs benutzt wird

Tabelle 3.27

System V ist eine UNIX-Version, die immer wieder als Ausgang für eine Normung benutzt wurde, quasi eine Standard-UNIX-Version.

Der Liste ist zu entnehmen, dass Linux einige Fremddateisysteme beherrscht, das heißt im Falle von *mkfs*, dass Linux auch Disketten mit einem DOS-Dateisystem erstellen kann. Durch die Entwicklungen im Windows-Bereich steht zu erwarten, dass bald auch *NTFS* als Dateisystem unter Linux verfügbar sein wird.

Das Anlegen eines *ext2*-Dateisystems auf der Diskette erfolgt mit:

```
mkfs -t ext2 -c /dev/fd0h1440 2880
```

Mounten von Dateisystemen



Zusätzlich wird die Diskette auf bad blocks geprüft. Abschließend ist zu bemerken, dass *mkfs*, wenn es keine Angabe bezüglich des Dateisystems findet, ein *Minix*-Dateisystem anlegt!

Der möglicherweise kommende Standard ist das Reiser-Filesystem (benannt nach dem wichtigsten Entwickler des Dateisystems). Es ist ein so genanntes „journaled filesystem“. Zugriffe auf das Dateisystem werden protokolliert, um es nach einem möglichen Systemcrash anhand der Protokolldatei rekonstruieren zu können. Dies erspart die bisweilen sehr zeitaufwändigen Dateisystemüberprüfungen (siehe weiter unten) beim Systemstart.

Falls *reiser* bzw. *reiserfs* als Dateisystem für die Option *-t* nicht verfügbar ist, benutzt man zum Erstellen eines solchen Dateisystems anstelle von *mkfs* die direkte Variante *mkreiserfs*.



Hinweis

Mounten von Dateisystemen

Beim Booten des Systems greift die Bootroutine lesend auf die Linux-Partition der Festplatte zu (sofern Sie Linux nicht in einer DOS-Partition installiert haben). Dabei fragt sie Informationen darüber ab, welche Dateisysteme nach dem Start zur Verfügung stehen sollen.

Dieses Bereitstellen von Dateisystemen nennt man „Mounten“. In diesem Abschnitt wird der Begriff „Dateisystem“ stets im Sinne von Dateisammlung benutzt. Ist die Verwaltung der Dateien gemeint, wird dies durch den Begriff „Dateisystemtyp“ beschrieben.

Zu Beginn steht Linux nur ein Dateisystem zur Verfügung, das so genannte „root-Dateisystem“. In dieses hinein müssen zusätzlich Dateisysteme integriert – gemountet – werden. Das *root*-Dateisystem ist in der Regel die Linux-Partition der Festplatte.

Angenommen, es befinden sich Daten auf einer Diskette, so kann man auf diese erst zugreifen, wenn die Diskette gemountet ist. Man beachte, dass unter Linux keine Laufwerksbezeichnungen zur Verfügung stehen, mit denen man Zugriff auf die Diskette hätte.

Das Mounten bezweckt zweierlei:

- Der Datenträger, auf dem sich das Dateisystem befindet, wird bei Linux als solcher angemeldet, und
- es wird eine Zugriffsmöglichkeit auf dieses zusätzliche Dateisystem eingerichtet.

Diese Zugriffsmöglichkeit wird dadurch realisiert, dass Linux beim Mounten ein Verzeichnis mitgeteilt wird, über das der Zugriff auf die Dateien des neuen Dateisystems erfolgen soll. Dieses Verzeichnis nennt man auch den „Mount-Point“.

Linux-Dateisystem

mount

Hieraus ergibt sich die grundsätzliche Syntax des Befehls *mount*, mit dem das Einbinden eines zusätzlichen Dateisystems erfolgt:

```
mount [ -option] gerätedatei mount-point
```

Soll beispielsweise eine Diskette in Laufwerk A: (in Ermangelung einer Linux-eigenen Laufwerk kennung wird hier die DOS-Bezeichnung verwendet) gemountet werden, so erfolgt dies mittels:

```
mount /dev/fd0h1440 /mnt
```

Vergleichen Sie
diesbezüglich die
Einträge in */etc*, die
mit *fd* beginnen

Die Gerätedatei hängt dabei vom gewählten Laufwerk (*fd0...* entspricht A; *fd1...* B:) und dem Diskettentyp ab.

Der Mount-Point ist frei wählbar, vorgesehen ist dafür jedoch das Verzeichnis */mnt*. Theoretisch kann also der Mount-Point ein Verzeichnis sein, in dem sich bereits Daten befinden. Diese werden, so lange das andere Dateisystem gemountet ist, nicht verfügbar sein – sie werden überlagert.

Einige der verfügbaren Optionen von *mount* sind in *Tabelle 3.28* aufgeführt.

Optionen des *mount*-Befehls

Option	Bedeutung
<i>-w</i>	Mounten mit Schreib-/Lesezugriff auf das Dateisystem (voreingestellt).
<i>-r</i>	Mounten mit Lesezugriff (Daten des Dateisystems können nicht verändert werden).
<i>-n</i>	Mounten ohne Eintrag in die Tabelle <i>/etc/mtab</i> .
<i>-t Dateisystemtyp</i>	Angabe des Dateisystemtyps. Vergleichen Sie dazu die <i>Tabelle 3.27</i> .
<i>-a</i>	Automatisches Mounten aller in der Datei <i>/etc/fstab</i> genannten Dateisysteme.

Tabelle 3.28

Die Daten eines Linux-Systems verteilen sich häufig auf mehrere Datenträger. Man denke an Systeme mit einer großen Benutzerzahl und die damit verbundenen Datenmengen. Oder aber man möchte die DOS-Partition in das System einbeziehen. Dann ist es zweckmäßig, dies beim Systemstart automatisch ausführen zu lassen.



Mounten von Dateisystemen



Die Datei `/etc/fstab`

Diesem Zweck dient die Datei `/etc/fstab`. In ihr befinden sich Informationen über die Dateisysteme, die beim Booten automatisch gemountet werden sollen.

Ein einfaches Beispiel für den Inhalt einer solchen Datei ist:

```
/dev/hda3      /          ext2      defaults  1  1
/dev/hda1      /dos/c:    msdos    defaults  1  1
/dev/hda5      /dos/d:    msdos    defaults  1  1
none          /proc      proc      defaults  1  1
```

Jede Zeile der Datei entspricht einem zu mountenden Dateisystem:

- Der erste Eintrag einer solchen Zeile entspricht dem Namen der Gerätedatei, die beim `mount`-Befehl anzugeben ist.
- Der Zweite bezeichnet den *Mount-Point*.
- Der dritte Eintrag schließlich bezeichnet den Dateisystemtyp.

Weitere Einträge beziehen sich auf spezielle Optionen des `mount`-Befehls sowie Angaben darüber, wie oft das Dateisystem durch den Befehl `dump` gesichert bzw. ob das Dateisystem beim Booten mit `fsck` überprüft werden soll. Der an Einzelheiten interessierte Leser sei auf die *man pages* von `mount` verwiesen.

In vorliegendem Beispiel werden vier Dateisysteme gemountet, darunter eines vom Typ `ext2`. Es ist das *root*-Dateisystem und befindet sich in der Linux-Partition der Festplatte, zwei vom Typ `msdos`, die Laufwerke C: und D: in den entsprechenden DOS-Partitionen sowie eines vom Typ `proc`.

Dieser Dateisystemtyp wurde in *Tabelle 3.27* nicht erwähnt, da er offensichtlich aus dem Rahmen fällt. Ihm ist keine Gerätedatei zugeordnet (`none`). Das ist auch nicht notwendig, da sich dieses Dateisystem nicht auf einem Datenträger befindet.

Die in ihm vorhandenen Daten sind Informationen, welche die laufenden Prozesse im Arbeitsspeicher ablegen, die man sich mit dem Befehl `ps` anzeigen lassen kann.

Ein gemountetes Dateisystem lässt sich mit dem Befehl `umount` wieder „abmelden“.

umount

Man kann den Befehl `umount` in folgenden Varianten benutzen:

```
umount gerätedatei
```

oder

```
umount mount-point
```



Linux-Dateisystem



oder

```
umount -t dateityp
```

oder

```
umount -a
```

Die ersten beiden Möglichkeiten führen zum Abmelden des spezifizierten Dateisystems. Syntaxvariante drei meldet alle Dateisysteme eines bestimmten Typs ab, und in der letzten Variante werden alle Dateisysteme abgemeldet, die in der Tabelle */etc/mtab* eingetragen sind.

Die Datei */etc/mtab*

Dies ist eine Liste analog zu */etc/fstab*. Jedes Mounten erzeugt einen Eintrag in dieser Datei, sofern beim *mount*-Befehl nicht die Option *-n* benutzt wurde.

Beim Herunterfahren des Systems (*shutdown*) wird diese *umount*-Variante benutzt.

umount kann nicht erfolgreich durchgeführt werden, wenn auf das entsprechende Dateisystem zugegriffen wird. Dies ist bereits dadurch gegeben, dass das aktuelle Verzeichnis Mount-Point für einen Benutzer oder eines seiner Unterverzeichnisse ist.

```
cd /mnt
```

durch einen beliebigen Benutzer verhindert zunächst ein Abmelden eines Dateisystems, das in dieses Verzeichnis hineingemountet wurde. Der Superuser muss erst alle Prozesse beenden (*kill*), die auf */mnt* und seine Unterverzeichnisse zugreifen. Dies kann er aber ohne Einschränkung.

Der ungeübte Linux-Anwender, vor allem, wenn er bisher nur Einplatzsysteme kannte, sollte folgendes Problem beachten: Eine gemountete Diskette oder CD-ROM sollte nie aus dem Laufwerk entfernt werden.

Um unnötige Plattenzugriffe zu vermeiden, werden Änderungen im Dateisystem (egal, ob sie die Daten oder die Verwaltungsinformationen betreffen) zunächst immer nur im Arbeitsspeicher auf den neuesten Stand gebracht. In regelmäßigen Abständen werden diese Änderungen dann auch auf die Platte zurückgeschrieben. Die meiste Zeit befindet sich ein Dateisystem auf der Platte/Diskette in einem inkonsistenten Zustand, d.h., das, was sich auf der Platte/Diskette befindet, stimmt nicht mit dem von Linux angenommenen Zustand der Platte/Diskette überein. Dadurch kann das Entfernen einer Diskette im gemounteten Zustand zu einem unwiderruflichen Datenverlust führen.





Prüfen von Dateisystemen

Das letztgenannte Problem gilt natürlich auch für den Fall, dass der Rechner ohne ordnungsgemäßes Herunterfahren von Linux (*shutdown*) ausgeschaltet wird. Dies führt in der Regel nicht zum größten anzunehmenden Datenverlust, ist aber in jedem Fall unangenehm.

Linux stellt nun ein leistungsfähiges Werkzeug zur Überprüfung von Dateisystemen zur Verfügung, das automatisch beim Booten des Systems gestartet wird. Es ist der schon erwähnte Befehl *fsck* (*filesystem check*).

fsck

Die Syntax des Befehls *fsck* lautet:

```
fsck[ -option] dateisystem
```

Tabelle 3.29 zeigt die wichtigen zur Verfügung stehenden Optionen.

Optionen des *fsck*-Befehls

Option	Bedeutung
<i>-a</i>	automatisches Reparieren des Dateisystems ohne Rückfrage (ist mit Vorsicht zu genießen, besser ist die folgende Option <i>-r</i>)
<i>-r</i>	Sicherheitsabfrage vor möglichen Reparaturen
<i>-t dateisystemtyp</i>	Angabe des Dateisystemtyps, eine Angabe des Arguments <i>dateisystem</i> erfolgt in diesem Falle nicht. Vergleichen Sie zu den Dateisystemtypen die <i>Tabelle 3.27</i> .
<i>-A</i>	prüft alle in der Datei <i>/etc/fstab</i> genannten Dateisysteme, eine Angabe des Arguments <i>dateisystem</i> erfolgt in diesem Falle nicht

Tabelle 3.29

Der Befehl

```
fsck -r /dev/fd0h1440
```

überprüft das Dateisystem auf der Diskette und wird, falls notwendig, Reparaturen vornehmen, wenn die entsprechende Sicherheitsabfrage bejaht wird.



Linux-Dateisystem



Hinweis



Für Reiser-Dateisysteme steht der Befehl *reiserfsck* zur Verfügung.

Neben diesen, mit der nötigen Vorsicht einzusetzenden Verwaltungs-Tools kennt Linux auch einfache Informationsbefehle, die Auskunft über die Benutzung eines Dateisystems geben können. So liefert *df* (*disk free*) Informationen über den noch verfügbaren Platz im Dateisystem.

df

Die Syntax des Befehls *df* lautet:

```
df [ -option] [ pfad]
```

Dabei kann *pfad* der Mount-Point oder die entsprechende Gerätedatei sein. Ohne Angabe eines Pfades werden Informationen über alle gemounteten Dateisysteme angezeigt. Mit der Option *-i* werden statistische Angaben über die I-Node-Listen angezeigt. Die Eingabe von

```
df
```

könnte folgende Ausgabe am Bildschirm bewirken:

Filesystem	1024-blocks	Used	Available	Capacity	
Mounted on					
/dev/hda3	347529	49886	279096	15%	/
/dev/hda1	307240	175672	131568	57%	
/dos/c:					
/dev/hda5	102566	51630	50936	50%	
/dos/d:					

df gibt die Namen der Gerätedateien, die Anzahl der vorhandenen, benutzten und verfügbaren logischen Blöcke (*1024-blocks*), einen daraus resultierenden Prozentwert und den Namen des Mount-Points wieder.

Das Argument *-T* erzwingt außerdem die Ausgabe des Namens des Dateisystems.

du

Ein weiterer Informationsbefehl ist *du* (*disk usage*).

Er zeigt an, wie viele Blöcke des Dateisystems durch die einzelnen Dateien belegt werden. Das folgende Beispiel zeigt den Platzbedarf der Dateien des aktuellen Verzeichnisses.

```
4 ./ .term
2 ./bin
5 ./stuff
4 ./help/testverz/egal
```



Prüfen von Dateisystemen



```
7 ./help/testverz  
10 ./help  
36 .
```

Beachten Sie, dass *du* bei der Bearbeitung von Links Probleme hat. Die Syntax des Befehls lautet:

du [-option] [pfad]

Bezüglich der Optionen konsultieren Sie auch hier die *man pages*.





Benutzerumgebung



Grundlegende Befehle	214
Arbeiten mit der bash	243
Editoren	289
Linux-Werkzeuge	317



4

Benutzerumgebung



Linux kennt als Mehrbenutzersystem eine Reihe unterschiedlicher Methoden zur Sicherung der von ihm verwalteten Daten. Neben der Rechtesicherheit, die im *Kapitel 3 Linux-Dateisystem* ausführlich diskutiert wurde, zählt dazu auch die Login-Sicherheit. Ohne Login (Eingabe von Benutzernamen und – in der Regel – Passwort) kein Zugang zum System.

Das Einloggen ist unter Linux aber mehr als nur das Anmelden beim System. Über die Login-Prozedur wird dem Benutzer eine persönliche Arbeitsumgebung zur Verfügung gestellt, die ihn in die Lage versetzt, die Dienste des Systems zu nutzen und nach seinen Vorstellungen – sofern es das System zulässt – zu gestalten.

Die offensichtlichsten Merkmale dieser Umgebung sind die Shell, mit deren Hilfe der Benutzer mit dem System kommuniziert, und das Heimatverzeichnis, in dem der Benutzer hinreichend Rechte besitzt, um vernünftig mit Dateien arbeiten zu können, die z.B. während einer Sitzung entstehen.

Der Schwerpunkt dieses Kapitels wird die Beschreibung der bash sein. Es ist diejenige Shell, die dem Benutzer standardmäßig zur Verfügung gestellt wird. Die Festlegung der Shell geschieht beim Einrichten von Benutzern durch den Superuser, dem dazu der Befehl *adduser* zur Verfügung steht.

Darüber hinaus werden in einer kurzen Übersicht weitere bekannte Shells vorgestellt.

Abgerundet wird das Kapitel durch die Beschreibung diverser Linux-Werkzeuge, darunter die Editoren *ed*, *vi*, *joe* und der „Stream“-Editor *sed*.

Grundlegende Befehle

Nach dem Einloggen meldet sich beim Benutzer normalerweise die bash mit dem Bereitschaftszeichen, dem Prompt in der Form

\$

(ein \$-Zeichen gefolgt von einem Leerzeichen). Eventuell finden Sie davor noch einen mehr oder weniger phantasievollen Namen, der Ihren Rechner bezeichnet (dieser Rechnername wird bei der Installation vergeben und dient in der Regel dazu, den Rechner in einem Netzwerk zu identifizieren) und eventuell Ihr aktuelles Arbeitsverzeichnis.

Sollten Sie als Superuser eingeloggt sein, wird in der bash das Dollar- durch ein Nummernzeichen ersetzt:

compostor:~#

Die Tilde (~) deutet in diesem Zusammenhang an, dass sich *root* im Heimatverzeichnis befindet.



Grundlegende Befehle



Wie auch immer der Prompt aussehen mag, er signalisiert dem User, dass er nun seine Kommandos eingeben kann.

Die prinzipielle Arbeitsweise einer textorientierten Shell wie der bash besteht darin, dass sie auf Benutzereingaben wartet; dies wird durch einen Prompt angezeigt. Die tatsächlich erfolgte Eingabe wird dann von der Shell analysiert, unter der Annahme, dass der Benutzer ein Programm starten will.

Auch ein Linux-Befehl ist ein Programm.

Im Folgenden wird, wenn nicht ausdrücklich anders vermerkt, angenommen, dass die bash als Shell benutzt und ihr Prompt zu sehen ist.



Hinweis

Bevor die Arbeit mit der bash genauer untersucht wird, soll nicht unerwähnt bleiben, dass man sich nach vollbrachter Arbeit aus Sicherheitsgründen beim System abmelden sollte. So vermeiden Sie, dass jemand ungefragt Zugriff auf Ihre Daten bzw. unberechtigten Zugang zu Linux erhält.

Das Abmelden oder Ausloggen geschieht mithilfe der Befehle

`logout`

bzw.

`exit`

Besonderheiten bei der Eingabe

Bei der Eingabe (von Befehlen) sind einige Besonderheiten zu beachten, die aus anderen gängigen Betriebssystemen (speziell denen des Hauses Microsoft) so nicht bekannt sind. Zum Beispiel unterscheidet Linux Groß- und Kleinschreibung. Dies gilt nicht nur für Dateinamen sondern auch für Eingaben am Prompt.

Man beachte, dass die Mehrzahl der Befehle (Programme) in einer gleichnamigen Datei enthalten sind.



Hinweis

Will man etwa eine Liste der Dateien im aktuellen Verzeichnis angezeigt bekommen und gibt statt `ls`

`Ls`

ein, so wird Linux das entsprechende Programm `ls` nicht finden und deshalb folgende Fehlermeldung ausgeben

```
bash: Ls: command not found
```

Wie Sie sehen, wird diese Meldung tatsächlich von der bash ausgegeben.



Benutzerumgebung



Die Benutzung von Sondertasten kann zu unerwarteten Ergebnissen führen. Zwar kennt die bash einen History-Mechanismus, der über die Cursor-tasten gesteuert wird, die Funktionstasten sind aber nicht darin eingeschlossen. Vermeiden Sie deren Benutzung, auch wenn Sie keine ernsthaften Probleme dadurch bekommen.

Bezüglich der Benutzung von Leerzeichen als Worttrennern bestehen strengere Regeln als etwa unter MS-DOS. Aus der Befehlssyntax ergibt sich eindeutig, dass nach dem Befehlsnamen ein Trennzeichen stehen muss, wenn ihm ein Argument folgt.

Hinweis



Das übliche Trennzeichen ist natürlich ein Leerzeichen, kann aber auch ein Tab-stopp sein.

Die unter MS-DOS übliche Schreibweise zum Wechsel ins übergeordnete Verzeichnis `cd..` führt unter Linux zu der schon genannten Fehlermeldung

```
$ cd.. ↵  
bash: cd..: command not found
```

Die bash findet in der Befehlszeile kein Trennzeichen und nimmt deshalb an, dass der auszuführende Befehl `cd..` heißt. Dies mag den Anfänger zunächst verwirren, andererseits zeigt es die unbestechliche Logik eines Linux-Systems. Es gibt keine „Kann“- und „Muss“-Leerzeichen. Die Benutzung von Leerzeichen ist durch einfache Regeln klar gekennzeichnet.

Eine Spezialität von UNIX-Systemen ist das so genannte „type ahead“. Dies bedeutet, dass die Tastatur unmittelbar nach der Eingabe eines Befehls wieder (für die nächste Eingabe) freigegeben ist.

Der Start eines Programms durch die Eingabe seines Namens (MS-DOS) oder Anklicken seines Symbols (Windows bzw. OS/2) bedeutet in jedem Falle eine Wartezeit, in der keine weiteren Aktivitäten des Benutzers vom System akzeptiert werden (außer etwa dem Bewegen der Maus).

Hat man am Prompt einen Befehl eingegeben, kann ohne Wartezeit die Eingabe des nächsten erfolgen. Diese an sich nützliche Eigenschaft der Shell wird in bestimmten Situationen sicher verwirren, da das Bildschirmecho der Eingabe unmittelbar erfolgt. Produziert z.B. der erste eingegebene Befehl eine längere Bildschirmausgabe, wird die Bildschirmanzeige der Neueingabe (das Bildschirmecho) nicht erst dann angezeigt, wenn die Ausgabe des Vorbefehls abgeschlossen ist, sondern beide Bildschirmanzeigen werden miteinander vermischt.

Sie können dies testen, indem Sie zunächst

```
ls -l /dev
```

und unmittelbar danach

```
ls
```

eingeben.



Grundlegende Befehle



Man kann seine Eingabe bedenkenlos mit **[]** bestätigen, der entsprechende Befehl wird ausgeführt. Ist man sich seiner Sache nicht sicher, d.h., hat man bei der Eingabe eventuell einen Fehler gemacht, so kann man die Eingabe nicht mehr überprüfen, da sie ja mit der Ausgabe des vorhergehenden Befehls am Bildschirm vermischt ist. In diesem Fall lässt sich die Eingabe durch

[Strg]+[U]

abbrechen.

Diese und einige weitere Tastenkombinationen können allgemein bei der Bearbeitung einer Eingabezeile benutzt werden. Mithilfe der Cursortasten lassen sich bereits eingegebene Befehle wiederholen und editieren.

Dies gilt natürlich nicht für Terminals, die nicht über die genannten Sondertasten verfügen. Beachten Sie bitte, dass UNIX nicht nur für die heute gängige Hardware entwickelt wurde. Es unterstützt auch solche Hardware, die verglichen mit einem PC erheblich eingeschränktere Möglichkeiten besitzt. Letzteres gilt auch für Linux, und so ist es kein Wunder, wenn bestimmte Terminalfunktionen nicht zwangsläufig an bestimmte Tasten oder Tastenkombinationen gebunden sind.

Zur Information über die Terminalerstellungen und auch zu deren Veränderung kennt Linux den Befehl **stty** (set terminal type). Seine allgemeine Syntax lautet:

stty[-option]

bzw.

stty aktion tastenkombination

Eine wichtige Option ist **-a**. Damit lassen sich alle aktuellen Terminalerstellungen anzeigen. Das Ergebnis einer solchen Anzeige könnte etwa wie folgt aussehen:

```
$ stty -a
speed 38400 baud; rows 25; columns 80; line = 0;
intr = ^C; quit = ^\; erase = ^?; kill = ^U; eof = ^D; eol =
= <undef>;
eol2 = <undef>; start = ^Q; stop = ^S; susp = ^Z; rprnt =
^R; werase = ^W;
lnext = ^V; flush = ^O; min = 1; time = 0;
-parenb -parodd cs8 hupcl -cstopb cread -clocal -crtscs
-ignbrk -brkint -ignpar -parmrk -inpck -istrip -inlcr
-igncr icrnl ixon ixoff
-iuclc -ixany -imaxbel
opost -olcuc -ocrnl onlcr -onocr -onlret -ofill -ofdel
n10 cr0 tab0 bs0 vt0 ff0
isig icanon -iexten echo echoe echok -echonl -noflsh
-xcase -tostop -echoprt
-echoctl echoke
```



Benutzerumgebung



Es werden allgemeine Einstellungen wie die Baudrate oder die Zeilen- und Spaltenzahl des Terminals angezeigt. Darüber hinaus erfährt man verschiedene aktuelle Tastenkombinationen: Das Caret-Zeichen (^) symbolisiert dabei die **[Strg]-Taste**. So bedeutet etwa **[Strg]+[C]**, dargestellt als ^C, den Abbruch des aktuellen Prozesses, dies entspricht der Aktion *intr*. Schließlich werden eine Reihe grundsätzlicher Einstellungen etwa bezüglich der Bildschirmanzeige bestimmter Tastendrücke dargestellt. So bedeutet *echoctl*, dass das Drücken der **[Strg]-Taste (Control!)** ein Bildschirmecho erzeugt.

Diese Einstellungen lassen sich mithilfe der zweiten Syntax des Befehls verändern. Mit

```
stty intr ^x
```

legt man z.B. fest, dass die Unterbrechung eines laufenden Prozesses ab sofort mit **[Strg]+[X]** erfolgt.

Die wohl bekannteste Tastenkombination ist **[Strg]+[D]**, mit der das Ende einer (Text-)Datei gekennzeichnet wird. In obigem Listing finden Sie diese Kombination unter *eof* aufgeführt. Bereits im *Kapitel 3 Linux-Dateisystem* wurde hiervon reger Gebrauch gemacht, als mittels *cat* einfache Dateien erstellt wurden.

In vielen UNIX-Varianten kann man mit **[Strg]+[D]** ausloggen. Die jeweilige Shell interpretiert **[Strg]+[D]** konsequenterweise als Ende des einzugebenden Textes und beendet ihre Arbeit. Wegen der erweiterten Funktionalität der bash ist das Ausloggen in dieser Form nicht mehr möglich. Immerhin erhalten Sie aber eine Meldung, wie das Ausloggen erfolgen muss.

Es spielen auch Sicherheitsaspekte eine Rolle. Da viele Aktivitäten durch **[Strg]+[D]** beendet werden, könnte ein zu konsequentes Drücken dieser Tasten ein unfreiwilliges Ausloggen bedeuten.

Informationen über Benutzer

Unter Beachtung der oben genannten Besonderheiten können Sie sich mit einigen einfachen Befehlen über Ihre Benutzerumgebung informieren.

Im Allgemeinen sind Sie im Mehrbenutzersystem Linux nicht der einzige eingeloggte User. Der Befehl *who* informiert Sie über Ihren persönlichen Account sowie über mögliche weitere eingeloggte Benutzer. Seine Syntax lautet:

```
who [ -option ]
```

bzw.

```
who am { i | I }
```



Grundlegende Befehle



Account lässt sich in etwa mit „Zugriff auf das System“ übersetzen.

Die zweite Variante zeigt Informationen zum eigenen Account, etwa in folgender Form:

```
composter!rattle    tty1      Dec 10 08:27
```

Man ist in diesem Beispiel als Benutzer *rattle* am Rechner *composter*, Terminal *tty1* seit dem 10.12. 8:27 eingeloggt.

Man beachte, dass bei der Bezeichnung der eigenen Person Groß- und Kleinschreibung nicht unterschieden wird.

Im anderen Fall wird eine Liste der eingeloggten Benutzer angezeigt, die etwa wie folgt aussehen könnte:

```
$ who ↵
root      tty1      Dec 10 08:55
rattle    tty4      Dec 10 09:04
$
```

Der erste Eintrag nennt den Namen des Benutzers, der zweite das Terminal, an dem er eingeloggt ist, und der dritte die Login-Zeit.

Weitergehende Informationen liefern die Schalter *-i* (*idle time*, die Zeit, seit der am entsprechenden Terminal keine Aktivität mehr stattgefunden hat) oder *-T* (zeigt über +/- an, ob ein Terminal Nachrichtenempfang zulässt oder nicht). Der Schalter *-H* zeigt zusätzlich eine Überschriftenzeile an (*Header*). Eine Beispielausgabe zeigt das folgende Listing:

```
$ who -HTi ↵
USER      MESG   LINE      LOGIN-TIME     IDLE   FROM
rattle    +      tty1      Dec 11 13:27   .
root      +      tty3      Dec 11 13:30 00:02
cshuser   --     tty4      Dec 11 13:30 00:01
$
```

Die Spalte *USER* nennt den Benutzernamen. *MESG* vermerkt, ob der User eine Nachricht empfangen kann (+) oder nicht. *LINE* nennt das Terminal, an dem der entsprechende Benutzer eingeloggt ist. *LOGIN-TIME* ist der Zeitpunkt des Einloggens, *IDLE* stellt fest, wie lange (in Minuten) am Terminal des entsprechenden Benutzers keine Aktivität mehr stattgefunden hat: Ein Punkt bedeutet, dass am entsprechenden Terminal in der letzten Minute eine Eingabe erfolgte. *FROM* nennt gegebenenfalls den Namen des Rechners, über den der Benutzer Zugang zum System hat: Hat er direkten Zugang und nicht über ein Netzwerk, erscheint hier kein Eintrag.

Will man nur den Benutzernamen feststellen, genügt auch die Eingabe des Befehls

```
logname
```



Hinweis



Hinweis



219

Benutzerumgebung



Dieser einfache Befehl ohne Optionen oder Argumente wird vornehmlich in Shell-Skripten Anwendung finden. Weiter gehende Informationen als *who* liefert der Befehl *w*:

```
9:46am  up 51 min,  2 users,  load average: 0.00, 0.00, 0.00
User     tty     from      login@    idle    JCPU    PCPU   what
root     tty1          8:55am           1       1   -bash
rattle   tty4          9:04am           w
```

Die erste Zeile enthält allgemeine Systeminformationen, wie die aktuelle Zeit, die Zeit, seit der Linux aktiv ist, die Anzahl der eingeloggten Benutzer und die Ladeaktivitäten in den letzten 1, 5 und 15 Minuten.

Neben den bereits von *who* bekannten Informationen erfährt man noch, was der Benutzer gerade macht (die Spalte *what* versucht, anhand des vom Terminal kontrollierten aktiven Prozesses die Befehlszeile zu rekonstruieren, mit der er gestartet wurde), und wie viel CPU-Zeit er insgesamt (*JCPU*) sowie wie viel der aktive Prozess (*PCPU*) verbraucht hat.

Der Befehl *finger* kann ebenfalls zur Anzeige der eingeloggten Benutzer eingesetzt werden. Er liefert dann ähnliche Informationen wie *who* oder *w*. Seine Syntax lautet

```
finger[ -option] [ benutzerliste]
```

Bei Angabe eines Benutzernamens werden auch dann Informationen angezeigt, wenn der Benutzer nicht eingeloggt sein sollte. Die Namen in *benutzerliste* können Benutzernamen des aktuellen Linux-Rechners oder auch solche an einem entfernten Host-Rechner in der Form *username@hostname* oder *@hostname* sein. Im letzten Fall werden wieder alle eingeloggten Benutzer am entsprechenden Host angezeigt.

Der Befehl *finger* gibt an zusätzlichen Informationen noch den „tatsächlichen“ Namen des Benutzers, sein Heimatverzeichnis und seine Standard-Shell aus. Schließlich gibt er Auskunft über dessen Pläne und Projekte.

Sowohl *finger* als auch *who* und *w* sind im Prinzip nichts anderes als besondere Dateiausgabebefehle. Sie beziehen die Information, die sie darstellen, aus bestimmten Dateien, deren Inhalte sie formatiert ausgeben. Diese Dateien sind:

- /var/adm/utmp,
- /var/adm/wtmp und
- /etc/passwd.

Sowohl *utmp* als auch *wtmp* befinden sich bei vielen UNIX-Varianten im Verzeichnis */etc*. In diesem Verzeichnis können Sie unter Linux einen Link auf die beiden entsprechenden Dateien in */var/adm* finden.



Grundlegende Befehle



Die Passworddatei `/etc/passwd` kennen Sie bereits aus dem Kapitel 3 *Linux-Dateisystem*. Sie enthält benutzerrelevante Informationen, die beim Einloggen überprüft werden. Sie wird vor allem von `finger` benutzt, um über nicht eingeloggte Benutzer zu informieren und um die zusätzlichen Angaben wie Heimatverzeichnis, Shell und „tatsächlicher“ Name machen zu können.

Seine Angaben zu Plänen und Projekten bezieht er aber aus den Dateien `.plan` und `.projekt` im Heimatverzeichnis des jeweiligen Benutzers. Sie sind ein einfaches Beispiel für die vielfältigen Kommunikationsmöglichkeiten, die unter Linux vorhanden sind. Werden sie von den Benutzern gepflegt, d.h. mit aktuellen Inhalten versehen, kann sich ein anderer Benutzer mit `finger` ohne störende Nachfragen wichtige Nachrichten beschaffen.

Die Datei `/var/adm/utmp` ist eine Liste der verfügbaren Terminals. Zu jedem Terminal wird außerdem vermerkt, wer sich dort wann eingeloggt hat. Einträge über Terminals, an denen niemand eingeloggt ist, erhalten eine entsprechende Information (nämlich die, dass der Login -Prozess an diesem Terminal aktiv ist). Diese Liste benutzen alle genannten Befehle, um die Namen der eingeloggten Benutzer festzustellen sowie deren Terminal und Login-Zeit. Die Angaben sind nicht in einem einfachen Textformat enthalten, was Sie mit

```
more /var/adm/utmp
```

überprüfen können; `who`, `w` und `finger` setzen sie in für uns lesbare Tabellen um.

Die Datei `/var/adm/wtmp` schließlich ist eine Login-Statistik und enthält weitere Angaben über Benutzer- und Systemaktivitäten. Diese geben z.B. Auskunft über die *idle-time* oder die verbrauchte CPU-Zeit.

Auch der Befehl `last` bezieht seine Informationen im Wesentlichen aus `/var/adm/wtmp`. Seine Syntax lautet:

```
last [ -option] [ benutzer]
```

Ohne Argument gibt er den gesamten Inhalt der Datei als Tabelle aus. Da diese recht lang sein kann, lassen sich die Informationen durch verschiedene Optionen oder die Angabe eines Benutzernamens filtern.

idle-time ist die Zeit, seit der an einem Terminal keine Aktivität mehr stattgefunden hat

Optionen des Befehls `last`



Option	Bedeutung
<code>-anzahl</code>	beschränkt die Ausgabe auf die ersten <code>anzahl</code> Zeilen. Da die jüngsten Informationen zuerst ausgegeben werden, erhält man so Auskunft über die aktuellsten Login-Ereignisse.
<code>-f datei</code>	ersetzt <code>/var/adm/wtmp</code> durch die angegebene Datei



Benutzerumgebung

Optionen des Befehls *last* (Forts.)

Option	Bedeutung
<i>-t terminal</i>	zeigt nur Informationen über Logins am genannten Terminal. Es genügt im Übrigen die Angabe des Terminals ohne -t
<i>-h host</i>	zeigt nur Informationen über Logins, die über den genannten Hostcomputer erfolgt sind
<i>-y</i>	zusätzliche Ausgabe des Jahres bei Datumsangaben

Tabelle 4.1

Will man sich die letzten zehn Login-Aktivitäten am Terminal *tty4* anzeigen lassen, geschieht dies wie folgt:

```
$ last -10 tty4
rattle    tty4          Fri Jan  2 09:04  still logged in
root      tty4          Mon Jan  1 08:58 -- down   (00:03)
cshuser   tty4          Sun Dec 29 05:34 -- 05:58 (00:24)
root      tty4          Sun Dec 29 05:34 -- 05:34 (00:00)
root      tty4          Sun Dec 29 05:33 -- 05:34 (00:01)
root      tty4          Sun Dec  8 07:25 -- down   (00:01)
rattle    tty4          Tue Nov 19 17:52 -- 18:07 (00:15)
rattle    tty4          Tue Nov 19 17:49 -- 17:51 (00:01)
rattle    tty4          Tue Nov 19 17:37 -- 17:49 (00:11)
root      tty4          Mon Nov 18 17:33 -- down   (00:01)
$
```

Die erste Spalte zeigt den jeweiligen Benutzer, die zweite das Terminal. Es folgen Login-Datum und -Zeit. Die folgenden Angaben hängen davon ab, ob der Benutzer noch eingeloggt ist (*still logged in*) oder nicht. Hat er sich bereits ausgeloggt, werden seine Logout-Zeit und die Dauer seines Logins vermerkt. Gegebenfalls wird anstelle der Logout-Zeit angezeigt, dass er das System heruntergefahren bzw. neu gestartet hat.

Die Datei */etc/passwd* und das Login

Während die Dateien */var/adm/utmp* und */var/adm/wtmp* Binärdaten enthalten und nicht direkt lesbar sind, kann man */etc/passwd* mit einem der gewöhnlichen Dateiausgabebefehle ansehen. Ein Auszug aus der Datei sieht etwa wie folgt aus:

```
root::0:0:root:/root:/bin/bash
rattle:1/vfl9fumtNV2:501:100:Guy
Rattlestern:/home/rattle:/bin/bash
rumpel:AcsJsMFPFn16s:502:100:Gerd
Schwatzke:/home/rumpel:/bin/bash
```



Grundlegende Befehle



Die durch Doppelpunkt getrennten Felder eines Eintrags sind im Einzelnen:

- Der Benutzername, z.B. *rumpel*.
- Das (verschlüsselte) Passwort – beachten Sie die Sicherheitslücke, die sich in diesem System dadurch ergibt, dass *root* kein Passwort hat beziehungsweise der entsprechende Eintrag fehlt (seltsamerweise ist in Linux der Superuser nicht gezwungen, ein Passwort zu benutzen).
- Die Benutzerkennnummer UID, z.B. 502, über die jeder Benutzer im System identifiziert wird.
- Die Gruppenkennnummer GID, z.B. 100, die festlegt, welcher Gruppe der Benutzer nach dem Einloggen angehört.
- Das Kommentarfeld, in dem üblicherweise Informationen über den realen Benutzer zu finden sind wie sein wirklicher Name, seine (dienstliche) Telefonnummer oder Angaben über sein Büro.
- Das Heimatverzeichnis, z.B. */home/rumpel* und
- die Standard-Shell, z.B. */bin/bash*, die dem Benutzer nach dem Einloggen zur Verfügung steht.

Das Heimatverzeichnis von *root* ist abweichend von den Heimatverzeichnissen anderer Benutzer nicht in */home*, sondern in der Wurzel zu finden.



Hinweis

Benutzername und Passwort dienen der Identifikation beim Einloggen. Der Login-Prozess prüft die vom Benutzer gemachten Eingaben gegen die beiden ersten Einträge (jeder Zeile) der Datei */etc/passwd*. Wird keine Übereinstimmung festgestellt, wiederholt der Login-Prozess seine Eingabeaufforderung – man hat keinen Zugang zum System.

Hat man jedoch Benutzername und Passwort korrekt eingegeben, erfolgt eine genauere Prüfung der passenden Zeile von */etc/passwd* durch den Login-Prozess. Der nun von Linux akzeptierte Benutzer erhält UID und GID, die ihn bei allen folgenden Aufgaben auswiesen.

Legt er etwa eine neue Datei an, werden beide Nummern in deren I-Node gespeichert. Rechte werden, wie im entsprechenden Abschnitt in *Kapitel 3 Linux-Dateisystem* näher beschrieben, auf Grundlage der Rechtetabelle durch Vergleich der gespeicherten UID und GID einerseits mit den beiden beim Login erhaltenen Identifikationsnummern andererseits ermittelt.

Weiter erhält der Benutzer nun sein Heimatverzeichnis zugewiesen. Das bedeutet zum einen, dass das in */etc/passwd* eingetragene Verzeichnis zum aktuellen Verzeichnis wird. Zum anderen wird die Umgebungsvariable *HOME* initialisiert. Sie erhält als Wert den Namen dieses Verzeichnisses.



Benutzerumgebung

Hinweis



Über die Verwendung von Variablen vergleichen Sie den entsprechenden Abschnitt in diesem Kapitel.

Zuletzt wird das Programm gestartet, dessen Name der Eintrag im letzten Feld einer Zeile von */etc/passwd* ist. Dies wird in der Regel die bash sein. Prinzipiell lässt sich natürlich jede ausführbare Datei dort eintragen, aber der Benutzer könnte dann nicht mehr interaktiv mit Linux in Verbindung treten – das Programm würde ausgeführt, dann würde er ausgeloggt.

Die bash selbst wird beim Start noch zwei Shell-Skripte ausführen, mit deren Hilfe sie konfiguriert werden kann. Das ist zum einen die Datei */etc/profile*. Diese Konfigurationsdatei gilt für alle Benutzer. In ihr werden standardmäßig folgende Aufgaben abgewickelt:

- Festlegen der allgemein üblichen Bereitschaftszeichen in Abhängigkeit von der jeweiligen Shell,
- Festlegen der Rechtemaske für neue Dateien mit *umask*,
- Festlegen des Terminaltyps,
- Aufruf des Programms *fortune*, das für die Bildschirmausgaben der „Lebensweisheiten“ nach dem Login verantwortlich ist

Über */etc/profile* kann der Systemverwalter Anpassungen der Benutzerumgebung für alle User vornehmen. Die Datei *.profile* im Heimatverzeichnis eines jeden Benutzers dient dagegen der individuellen Anpassung der Umgebung. Da sie sich im Heimatverzeichnis des Benutzers befindet, kann er selbst darüber eine Konfiguration der Shell vornehmen.

Der Inhalt dieses Shell-Skripts könnte z.B. wie folgt aussehen:

```
PS1=\$\
PATH=$PATH:/home/rattle/bin
export PS1 PATH
alias l="ls -l"
```

In vorliegendem Beispiel werden zwei wichtige Shell-Variablen gesetzt und exportiert sowie ein Alias-Name für einen Befehl festgelegt.

Hinweis



Einzelheiten zu Variablen und Alias-Namen entnehmen Sie bitte den entsprechenden Abschnitten dieses Kapitels.

Die folgende Abbildung zeigt die einzelnen Schritte, die der Login-Prozess durchführt.

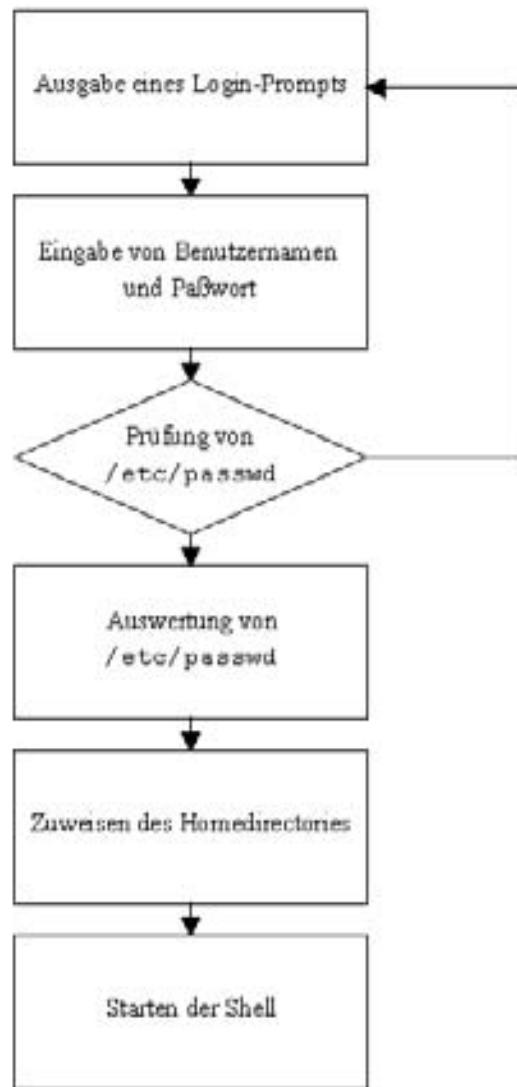


Grundlegende Befehle



Abb. 4.1

Der Login-Prozess



Benutzergruppen

Das vierte Feld einer Zeile der Datei `/etc/passwd` gibt die Gruppennummer GID an. Dieser Gruppe gehört der Benutzer nach dem Einloggen an. Durch die Gruppenzugehörigkeit werden Rechte kalkuliert und Besitzverhältnisse an Dateien festgelegt.



Benutzerumgebung



Angenommen, man gehört der Gruppe 100 an (der Name dieser Gruppe ist standardmäßig *users*). Erstellt man in dieser Situation eine Datei, so wird die Gruppe 100 als Besitzergruppe gespeichert. Ein Beispiel:

```
$ who am i ↵
compostor!rattle    tty1      Dec 10 08:27
$ grep rattle /etc/passwd ↵
rattle:1/vfl9fumtNV2:501:100:Guy
Rattlestern:/home/rattle:/bin/bash
$ cat > testdatei ↵
Dies ist eine Testdatei ↵
[Srg]+[D]
$ ls -ln testdatei ↵
-rw-r--r-- 1 501          100          24 Jan 13 02:22
testdatei
$
```

Hinweis



In die Gruppe *users* werden alle neuen Benutzer aufgenommen, sofern der Systemverwalter nicht ausdrücklich eine andere Gruppe wählt

Zunächst prüfen Sie mit *who am i*, unter welchem Benutzernamen Sie eingeloggt sind. Dann suchen Sie mit *grep* nach diesem Benutzernamen in der Datei */etc/passwd*. Dadurch lassen sich Benutzer- und Gruppennummer ermitteln. Dies sind im Beispiel 501 (dritter Eintrag der Passworddatei, also UID) und 100 (vierter Eintrag, die GID).

Sie erstellen nach der schon mehrfach angewandten Methode mittels *cat* eine neue Datei und lassen sich abschließend mit *ls -ln* ausführliche Informationen über sie geben. Die Verwendung der Option *-n* erklärt sich aus der Tatsache, dass Sie statt der Namen die Identifikationsnummern wissen wollen. In der Tat, die neue Datei gehört Benutzer 501 und der Gruppe 100.

Man kann nun während der laufenden Sitzung seine Gruppenzugehörigkeit wechseln. Grundvoraussetzung dafür ist, dass in der Datei */etc/group* ein entsprechender Eintrag zu finden ist. Diese Datei ist eine */etc/passwd*-ähnliche Liste und sieht in einem Linux-System mit nur wenigen Benutzern typischerweise wie folgt aus:

```
root::0:root
bin::1:root,bin,daemon
daemon::2:root,bin,daemon
sys::3:root,bin,adm
adm::4:root,adm,daemon
tty::5:
disk::6:root,adm
lp::7:lp
mem::8:
kmem::9:
wheel::10:root
floppy::11:root
mail::12:mail
```



Grundlegende Befehle



```
news::13:news
uucp::14:uucp
man::15:man
users::100:games
buch::101:rumpel,rattle
nogroup::-1:
```

Jede Zeile entspricht einer Gruppe. Die einzelnen Felder sind

- der Gruppenname, z.B. *users*,
- das Gruppenpasswort,
- die GID und schließlich
- eine durch Kommata getrennte Liste der Benutzer, die für diese Gruppe zugelassen sind.

Überraschenderweise ist das Passwortfeld für alle Gruppen leer. Noch überraschender mag es Ihnen erscheinen, dass es keinen Befehl gibt, dieses Passwort zu erstellen oder zu ändern – man müsste dazu die Datei editieren. Sinn macht das nicht, da ein Verschlüsselungsalgorithmus bekannt sein müsste. Die Datei kann ja von jedermann im System gelesen werden.

Eine weniger merkwürdige Besonderheit sind die Listen der Benutzer der Gruppe. Die Gruppe *users* mit der GID 100 enthält nur einen Benutzereintrag, obwohl wir weiter oben darauf hingewiesen haben, dass sie eine Standardgruppe ist, der im Allgemeinen die meisten Benutzer der System angehören.

Das liegt daran, dass ein Benutzer nicht bei seiner Login-Gruppe (das ist die durch die GID in */etc/passwd* festgelegte Gruppe) eingetragen sein muss. Diese lässt sich ja über den entsprechenden Eintrag in */etc/passwd* ermitteln.

Nehmen Sie wieder den Referenzbenutzer *rattle*, der ja gemäß Beispiel der Gruppe *users* angehört. Da er ferner in */etc/group* als Mitglied der Gruppe *buch* geführt wird, kann er nun wahlweise in einer der beiden Gruppen sein. Nach dem Login ist er in der Gruppe *users*, mit dem Befehl *newgrp* kann er jederzeit zwischen den Gruppen wechseln. Die Syntax ist:

```
newgrp [ gruppe]
```

Ein Wechsel zur Gruppe *buch* erfolgt also mit:

```
newgrp buch
```

Will man zu seiner Login-Gruppe zurückwechseln, geschieht dies ohne Angabe eines Gruppennamens. Dies veranlasst *newgrp*, die Datei */etc/passwd* statt */etc/group* zu untersuchen. Die Eingabe von *newgrp users* würde im vorliegenden Fall sogar eine Fehlermeldung hervorrufen; denn ein Eintrag *rattle* liegt unter *users* nicht vor! Der korrekte Wechsel zur Login-Gruppe erfolgt einfach mit:

```
newgrp
```



Benutzerumgebung

Hinweis



Beachten Sie, dass bei der Datei `/usr/bin/newgrp` das setuid-Bit gesetzt sein muss, da für das Ändern der GID Superuser-Rechte nötig sind.

Tipp



Als User root können Sie (ohne Passworteingabe!) eine beliebige Benutzeridentität annehmen

Man kann nun nicht nur die Gruppe während einer Sitzung wechseln, man kann sogar seine Identität ändern. Dies geschieht mithilfe des Befehls `su` (*substitute user*). Seine Syntax lautet:

```
su [ - ] [ benutzername]
```

Natürlich können Sie diesen Identitätswechsel nur durchführen, wenn Sie Benutzernamen *und* Passwort des entsprechenden Benutzers kennen. `su` prüft nämlich die Datei `/etc/passwd` und fragt gegebenenfalls das Passwort ab.

```
su rattle
```

lässt Sie zu Benutzer *rattle* werden, d.h., Sie erhalten seine UID und GID, mit denen die Rechte entsprechend kalkuliert werden. Ohne Angabe eines Benutzernamens, also bei Eingabe von

```
su
```

ergänzt der Befehl automatisch den Benutzernamen *root*, d.h., Sie nehmen, falls Sie das Passwort kennen, die Identität des Superusers an.

Dies kann zu überraschenden Situationen führen, wenn etwa der „neue“ Benutzer am aktuellen Verzeichnis keine Rechte besitzt

Im Normalfall wird nach erfolgreicher Abwicklung von `su` die Login-Shell (aus `/etc/passwd`) des entsprechenden Benutzers gestartet, jedoch nicht die komplette Login-Prozedur durchlaufen. Zu beachten ist zunächst also, dass ein direktes Ausloggen wie auch bei Verwendung von `newgrp` nicht möglich ist. Der Befehl `exit` beendet lediglich die entsprechende Shell, wodurch die ursprünglichen UID und GID wiederhergestellt werden.

Im Unterschied zu einem gewöhnlichen Login werden folgende Aktivitäten *nicht* ausgeführt:

- Festlegen des Heimatverzeichnisses als aktuelles Verzeichnis und Neudefinition der Variablen `HOME` – das aktuelle Verzeichnis wird nicht gewechselt,
- Ausführen der Konfigurations-Shell-Skripte `/etc/profile` und `.profile` im Heimatverzeichnis.



Grundlegende Befehle



Durch Angabe der Option – wird jedoch eine normale Login-Prozedur erzwungen.

Für Systemverwalter des Linux-Systems empfiehlt es sich im Übrigen, mit *su* als Superuser tätig zu werden. Ein normales Login als *root* birgt sehr häufig die Gefahr in sich, dass man, sich in einer normalen Sitzung wähnend, Befehle benutzt, die das System empfindlich treffen könnten.

Alle oben genannten Befehle (*who*, *w*, *logname* oder die Variante *who am i*) registrieren den Identitätswechsel nicht, sie geben den ursprünglichen Benutzernamen an. Will man sich über die aktuelle Benutzeridentität informieren, hilft der Befehl *id*.

Seine Syntax lautet

```
id[ -option] [ benutzer]
```

Wird ein Benutzername angegeben, liest *id* die Datei */etc/passwd* und gibt deren Benutzerbeschreibung aus. Ansonsten werden die aktuellen UID und GID angezeigt. Das folgende Listing zeigt eine Beispielausgabe von *id*.

```
$ id←
uid=501(rattle) gid=101(buch)
groups=100(users),101(buch)
$
```

Die aktuelle UID ist also *501*, die GID *101*. Ferner werden die Gruppen aufgelistet, in die man mit *newgrp* wechseln kann.

Gruppen- und insbesondere der Identitätswechsel stellen eine potenzielle Gefährdung der Systemsicherheit dar, und zumindest Letzteres ist deshalb durch Passwortabfrage geschützt. Überhaupt sollten Benutzer, die von Einplatzsystemen wie DOS, OS/2 oder Windows 95 zu Linux gekommen sind, die Passwortsicherheit nicht unterschätzen. Sie ist der wichtigste Schutz vor unberechtigtem Zugriff auf Daten.

Die Vergabe und Änderung von Passwörtern besorgt das Programm *passwd*. Es hat folgende Syntax:

```
passwd[ benutzername]
```

Ohne Argument wird das Passwort desjenigen Benutzers geändert, der *passwd* aufgerufen hat. Dabei prüft das Programm aber, ob die aktuelle UID und der Benutzername zueinander passen, d.h., ein Identitätswechsel mit *su* wird erkannt. In diesem Fall erlaubt *passwd* die Passwortänderung nur, wenn die aktuelle UID 0 ist.

Mit Argument aufgerufen wird durch *passwd* das Passwort des angegebenen Benutzers geändert. Diese Variante ist natürlich nur dem Superuser vorbehalten.

passwd liest die entsprechende Zeile der Passworddatei */etc/passwd* und fragt das alte Passwort ab (Enter old password:). Die Eingabe erfolgt aus Sicherheitsgründen ohne Bildschirmecho. Die Eingabe des alten Passworts entfällt für den Superuser.

Beachten Sie, dass die Login-Gruppe von *rattle* in diesem Fall *users* (GID 100) ist.

Ist die aktuelle UID 0, dann ist man faktisch Superuser

Benutzerumgebung



Danach muss man das neue Passwort eingeben. Da die eingegebenen Zeichen nicht am Bildschirm dargestellt werden, muss man diese Eingabe wiederholen. Erst wenn das neue Passwort zweimal gleich eingegeben wurde, erlangt es Gültigkeit.

Ein Passwort sollte man nicht erst dann ändern wollen, wenn es bereits „geknackt“ worden ist – dann dürfte es ohnehin für eine Änderung zu spät sein. Gewöhnen Sie sich an, in mehr oder weniger regelmäßigen Abständen Ihr Passwort zu ändern. Benutzen Sie Passwörter, die nicht allzu leicht zu entschlüsseln sind (Namen sind als Passwort denkbar ungeeignet), und vor allem: Schreiben Sie es nicht auf – irgendjemand wird Ihre Aufzeichnungen finden und nutzen können.

Onlinehilfen

Auch der professionelle Linux-Anwender wird bei der Fülle der Befehle und ihrer Optionen bisweilen Hilfe brauchen. Mehr noch der Anfänger. Linux hält eine Reihe von Hilfen bereit, die online – also am Terminal direkt – benutzt werden können.

Externe Befehle liegen als ausführbare Datei vor, interne Befehle sind in der jeweiligen Shell implementiert

Die meisten externen Befehle kennen die Option `-help`, über die eine Kurzhilfe zum Befehl verfügbar ist. Beachten Sie, dass der Option zwei Minuszeichen vorangestellt sein müssen, da der Befehl sie ansonsten zeichenweise interpretieren würde; `-help` wird wegen der Möglichkeit, Schalter zu gruppieren, als `-h -e -l -p` verstanden.

Die Kurzhilfe stellt in knapper Form die verfügbaren Schalter und ihre Bedeutungen vor. Wenn Sie diese Form der Hilfe nutzen, werden Sie sicher feststellen, dass eine Reihe von Optionen in zwei Varianten vorliegen: als Buchstabenkürzel und als Wort.

Der Befehl `ls` etwa kennt den Schalter `-R`, mit dessen Hilfe man ganze Verzeichnisbäume einsehen kann. Der englische Begriff für das sukzessive Durchsuchen von Unterverzeichnissen lautet recursive. Eine zu `-R` alternative Option heißt deshalb `-recursive`.

Diese zusätzlichen Optionen haben den Vorteil, dass sie, zumindest für diejenigen, welche die englische Sprache beherrschen, besser verständlich sind als irgendwelche willkürlich gewählten Buchstabenkürzel. Sie erfordern natürlich mehr Schreibarbeit. Außerdem ist zu beachten, dass zwei Minuszeichen als Einleitung benutzt werden müssen.

Neben `-help` kennen viele Befehle standardmäßig noch `-version` als Option. Mit ihr werden Versionsinformationen angezeigt. Dies ist insofern von Bedeutung, als Linux ein „lebendes“ Betriebssystem ist, dessen Bestandteile ständig verbessert werden. Eine neue Version eines Befehls kann neue „Goodies“ enthalten.

Für interne Befehle ist ebenfalls eine Kurzhilfe verfügbar, die sich in der bash mittels

```
help befehl
```

aktivieren lässt. Für den (internen) Befehl `pwd` etwa liefert `help` folgende Information



Grundlegende Befehle



```
pwd: pwd  
      Print the current working directory.
```

Nun besitzt *pwd* keine Optionen – demzufolge fällt die Ausgabe von *help* etwas dürftig aus.

Falls Sie nicht sicher sind, ob der Befehl, zu dem Sie Informationen haben wollen, extern oder intern ist, probieren Sie einfach beide Varianten, den Schalter und den Befehl *help*.

Umfassende Hilfe und häufig auch weiterführende Informationen liefern die *man pages*. Der Befehl *man* stützt sich auf eine umfangreiche Datenbank, die im Verzeichnis */usr* zu finden ist. Diese gibt er mithilfe von *more* Seitenweise aus.

Die *man pages* folgen dabei einem allgemeinen Schema, das am Beispiel der *man page* für *passwd* verdeutlicht werden soll.

Nach der Überschrift wird in einer Rubrik *NAME* der Befehl mit Namen und Bedeutung kurz skizziert.

- Die Rubrik *SYNOPSIS* enthält eine allgemeine Syntaxbeschreibung, deren Regeln auch in diesem Buch weitgehend verwendet werden.
- Der Abschnitt *DESCRIPTION* enthält eine je nach Befehl mehr oder weniger umfangreiche Beschreibung des Befehls. Hier erhält man auch ausführliche Informationen über die diversen Schalter.
- Benötigt der Befehl zu seiner Ausführung Dateien – sei es, um darin enthaltenen Informationen zu lesen, sei es, um Daten in der Datei zu verändern – werden diese im Abschnitt *FILES* genannt.
- Die Rubrik *SEE ALSO* verweist auf verwandte Befehle bzw. Teile des Online-Manuals, die verwandte Themen behandeln.
- In einigen Fällen gibt es einen Abschnitt *BUGS*, in dem bekannte Fehler oder Unzulänglichkeiten des Programms beschrieben werden. Dieses offene Verhältnis zu Programmfehlern ist typisch für UNIX und in besonderem Maße für Linux.

Auch wenn der Besitzer einer nicht unbekannten Softwareschmiede behauptet, dass sich keine „nennenswerte Anzahl“ von Anwendern seiner Produkte über „wesentliche Probleme“ mit diesen beschweren, bleibt festzuhalten, dass es derzeit keine fehlerfreie Software gibt!

Wenn ein Textverarbeitungsprogramm abstürzt und dieser Absturz den Verlust von zehn Seiten Text bedeutet, mag das kein wesentliches Problem sein. Wenn solche Abstürze pro Anwender nur einmal im Jahr passieren, mag dies nicht nennenswert sein – in jedem Fall ist es aber ärgerlich und bedeutet zusätzliche Arbeit.



Benutzerumgebung



Dies wäre einfacher zu verschmerzen, wenn man auf solche Fehlfunktionen vorbereitet wäre. Ein bekannter Fehler lässt sich eventuell vermeiden. Außerdem: Den Benutzer auf bekannte Fehler hinzuweisen, zeugt vom ehrlichen Bemühen, es besser zu machen!

PASSWD(1) Linux Programmer's Manual PASSWD(1)

NAME

passwd -- change password

SYNOPSIS

passwd [name]

DESCRIPTION

passwd will change the specified user's password. Only the superuser is allowed to change other user's passwords. If the user is not root, then the old password is prompted for and verified.

A new password is prompted for twice, to avoid typing mistakes. Unless the user is the superuser, the new password must have more than six characters, and must have either both upper and lower case letters, or non-letters. Some passwords which are similar to the user's name are not allowed.

FILES

/etc/passwd
/etc/shells

SEE ALSO

chsh(1), chfn(1)

BUGS

A password consisting of all digits is allowed. No warnings are printed if the superuser chooses a poor password. The -f and -s options are not supported.

AUTHOR

Peter Orbaek (poe@daimi.aau.dk)

Linux 1.0

22 June 1994

1

Ach ja, die Angaben zum Autor sind ohne Gewähr.

Weitere Einzelheiten über die Benutzung des Online-Manuals erhalten Sie durch Eingabe von:

man man





Weitere nützliche Befehle

An dieser Stelle sollen in loser Folge noch eine Reihe von Befehlen vorgestellt werden, die Ihnen die tägliche Arbeit erleichtern können, die aber keinem der bisher behandelten Themen zuzuordnen sind.

Beim Arbeiten mit der bash ist es häufig nützlich, den Bildschirm von vorhandenen Informationen zu befreien, ihn zu löschen. Dies geschieht mithilfe von:

```
clear
```

Da unter Linux eine ganze Reihe unterschiedlicher Terminalsysteme eingesetzt werden können, ist die Realisation dieses auf den ersten Blick einfachen Befehls nicht ganz trivial. Er nutzt die Informationen der Terminaldatenbanken *termcap* (*terminal capabilities*) bzw. *terminfo*.

Ein speziell für Shell-Skripte gedachter Befehl ist:

```
echo[ -option] [ text]
```

Er gibt text unmittelbar auf dem Bildschirm aus. Am Prompt macht *echo* daher wenig Sinn, man kann mit seiner Hilfe allerdings einige Besonderheiten der bash verdeutlichen. Die Option *-n* unterdrückt die Ausgabe eines Zeilenumbruchs, die ansonsten auch erfolgt, wenn kein text angegeben wird.

Mit

```
date[ -option] [ +format]
```

kann man sich Datum und Uhrzeit in nahezu beliebiger Form anzeigen lassen. Dabei ist format ein beliebiger Text, der von *date* ausgegeben wird.

```
date Hallo
```

würde lediglich das Wort Hallo am Bildschirm anzeigen. format kann aber auch eine Reihe von Platzhaltern beinhalten, die *date* zur Anzeige bestimmter Informationen veranlassen.

Eine Liste dieser Platzhalter oder Formatelemente zeigt *Tabelle 4.2*. Sofern sie sich auf Zeit- oder Datumsangaben beziehen, ist die aktuelle Systemzeit bzw. das Systemdatum gemeint.



Man beachte das +Zeichen als Einleitung des Ausgabe-formats

Formatelemente des Befehls *date*



Formatelement	Bedeutung
%H	Stunde im Format 00 bis 23
%I	Stunde im Format 01 bis 12



Benutzerumgebung

Formatelemente des Befehls `date` (Forts.)

Formatelement	Bedeutung
<code>%k</code>	Stunde im Format 0 bis 23
<code>%l</code>	Stunde im Format 1 bis 12
<code>%M</code>	Minuten im Format 00 bis 59
<code>%S</code>	Sekunden im Format 00 bis 59
<code>%p</code>	vormittags/nachmittags (AM oder PM)
<code>%r</code>	Uhrzeit im 12-Stunden-Format hh:mm:ss AM oder hh:mm:ss PM (z.B. 10:00:00 AM für 10 Uhr morgens)
<code>%T</code>	Uhrzeit im 24-Stunden-Format hh:mm:ss
<code>%X</code>	entspricht <code>%H:%M:%S</code>
<code>%Z</code>	Zeitzone (z.B. MEZ)
<code>%a</code>	Wochentag dreistellig abgekürzt (z.B. Sun für Sonntag)
<code>%A</code>	Wochentag (z.B. Sunday)
<code>%w</code>	Wochentag im Format 0 bis 6, 0 entspricht Sonntag
<code>%b</code>	Monatsname dreistellig abgekürzt (z.B. Jan)
<code>%h</code>	entspricht <code>%b</code>
<code>%B</code>	Monatsname (z.B. January)
<code>%m</code>	Monat im Format 01 bis 12
<code>%d</code>	Tag des Monats im Format 01 bis 31
<code>%D</code>	Datum im Format <code>mm/dd/yy</code> (z.B. 01/10/97 für den 10.1.1997)
<code>%x</code>	wie <code>%D</code>
<code>%y</code>	die letzten beiden Ziffern der Jahreszahl (z.B. 97)
<code>%Y</code>	Jahr vierstellig
<code>%j</code>	Tag des Jahres im Format 001 bis 366
<code>%U</code>	Woche des Jahres im Format 00 bis 53 (erster Wochentag: Sonntag)
<code>%W</code>	wie <code>%U</code> , aber mit Montag als erstem Wochentag
<code>%s</code>	Anzahl der Sekunden seit dem 1.1.1970, 0 Uhr (nun grübeln Sie mal schön)



Grundlegende Befehle



Formatelemente des Befehls `date` (Forts.)

Formatelement	Bedeutung
<code>%c</code>	Datum und Zeit, fasst <code>%a %b %X %Z %Y</code> zusammen
<code>%n</code>	Zeilenvorschub
<code>%t</code>	Tab-Sprung
<code>%%</code>	<code>%</code> -Zeichen

Tabelle 4.2



Mit

```
date "+Es ist %H:%M Uhr"
```

kann man sich also die Uhrzeit in der Form

```
Es ist 01:55 Uhr
```

anzeigen lassen. Bei Angabe eines Formats empfiehlt es sich generell, dieses in Anführungszeichen zu setzen. Der Grund dafür wird im nächsten Abschnitt (*Arbeiten mit der bash*) erläutert.

In der Form

```
date MMDDhhmm[CC] [YY] [.ss]
```

kann der Superuser den Befehl zum Setzen der Systemzeit und des Systemdatums benutzen.

```
date 12010800
```

legt das Datum auf den 1.12. und die Zeit auf 8 Uhr fest.

Eine für Linux (aber auch UNIX) typische Spielerei stellt der Befehl `cal` dar. Er zeigt einen Kalender an. Ohne Angabe eines Arguments ist dies der Kalender des aktuellen Monats in der Form:

```
March 2001
S M Tu W Th F S
          1 2 3
4 5 6 7 8 9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30 31
```



Benutzerumgebung



Die genaue Syntax des Befehls lautet:

```
cal[ -option] [[ monat] jahr]]
```

An Optionen verfügbar sind *-y*, was eine Ausgabe des Kalenders für das gesamte Jahr bewirkt, und *-j*, was eine Durchnummerierung der Tage über das gesamte Jahr hinweg zur Folge hat. Unter Verwendung des Schalters *-j* wird aus dem 31.12. der 365. bzw. 366. Tag des Jahres.

Gibt man als Argument ein Jahr an, verfährt *cal* wie bei Benutzung des Schalters *-y* und zeigt einen Kalender für 12 Monate. Doch Achtung:

```
cal 96
```

zeigt den Kalender des Jahres 96, nicht 1996. Gültige Werte für das Jahr sind die Zahlen 1 bis 9999.

Zusätzlich zum Jahr kann der Monat mit angegeben werden. So erzeugt

```
cal 9 1752
```

die folgende Ausgabe

September 1752
S M Tu W Th F S
1 2 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30

Wie gesagt, *cal* ist eine kleine Spielerei. Die Ausgabe für September 1752 berücksichtigt die seinerzeit in vielen Ländern vollzogene gregorianische Kalenderreform. Zur Angleichung des julianischen Kalenders an die realen Gegebenheiten wurden die Tage 2.9. bis 13.9.1752 übersprungen.

Die Dateien sind in
der Regel unter
/usr/games/fortunes

Dem neugierigen Benutzer sei noch der bereits erwähnte Befehl *fortune* empfohlen. Auch er ist im eigentlichen Sinne ein Dateiausgabebefehl. Der Inhalt der Dateien, die *fortune* ausgibt, sind Aphorismen, Anekdoten, Sprüche, Buchzitate – kurzum Weisheiten der Form:

Cahn's Axiom:

When all else fails, read the instructions.

Der Clou des Programms ist, dass es keinen bestimmten Text, sondern eine zufällig ausgewählte Passage der Datei ausgibt. Dazu ermittelt es zunächst eine Zufallszahl, um mit deren Hilfe den entsprechenden Eintrag zu finden und anzuzeigen. Da die Dateien recht umfangreich sind, kann man so eine ganze Weile mehr oder weniger witzige Weisheiten lesen, ohne dass sich *fortune* zu wiederholen scheint.





Kommunikation

Dass Sie im System nicht alleine arbeiten, zeigt Ihnen der Befehl `who`. Sie können mit denen da draußen kommunizieren.

Eine einfache Möglichkeit dazu ist der Befehl `write`. Seine Syntax lautet:

```
write benutzer[ terminal]
```

Falls ein Benutzer an mehreren Terminals eingeloggt ist, muss zur Klarstellung das Terminal angegeben werden, zu dem die Nachrichten geleitet werden sollen.

Die Abwicklung des `write`-Befehls soll an einem Beispiel veranschaulicht werden.

```
$ write rattleHallo rattle, kennst Du einen Befehl zum Vergleich vonzwei DateienWenn ja, bitte ich um baldige BenachrichtigungStrg+D$
```

Nach der Bestätigung des Befehls mit wartet er auf Eingaben von der Tastatur. Die Eingabe wird wie gehabt durch abgeschlossen. Im Prinzip arbeitet `write` wie `cat`, wenn man diesem Befehl keinen Dateinamen als Argument mitgegeben hat. Der Unterschied besteht darin, dass die Ausgabe nicht auf dem eigenen Terminal erfolgt, sondern auf dem des genannten Benutzers.

Selbstverständlich kann man nur Benutzer mit `write` ansprechen, die auch eingeloggt sind. Man sollte dabei berücksichtigen, dass die Meldung, die man einem anderen Benutzer sendet, auf dessen Bildschirm genau dort erscheint, wo sich sein Cursor befindet.

Die Meldung, die er durch Ihre Aktivität auf dem Bildschirm erhält, sieht etwa wie folgt aus:

```
Message from stern@composter on tty5 at 08:04 ...  
Hallo rattle, kennst Du einen Befehl zum Vergleich von  
zwei Dateien  
Wenn ja, bitte ich um baldige Benachrichtigung  
EOF
```

Zunächst ein Hinweis darüber, wer wann und von wo aus die Nachricht abgesandt hat, dann die Nachricht selbst und schließlich der Hinweis auf deren Ende (*EOF*). Im Übrigen ist *stern* der Benutzer, @ leitet den Namen des Hosts ein, von dem aus der Benutzer seine Nachricht abgesetzt hat.

Stellen Sie sich nun vor, Ihr Gegenüber will gerade eine komplexen Befehl eingeben und hat dies schon zu einem guten Teil erledigt. In diesem Moment trifft Ihre Meldung ein.



Benutzerumgebung



Das sieht etwa wie folgt aus (die Eingabe von *rattle* unterstrichen, der Beginn Ihrer Meldung normal dargestellt):

```
$ find / -name testdat -user rattle -pMessage from stern@composter on tty5 at 08:04 ...
```

Der zusätzliche Nachteil des Benutzers *rattle* ist natürlich der, dass er keine Unterscheidung aufgrund der Unterstreichung treffen kann. Auf diesen Effekt des „type ahead“ wurden bereits weiter oben aufmerksam gemacht.

Sie sehen also, Nachrichten zu empfangen, kann ganz schön lästig sein.

Andererseits weiß nun natürlich *rattle*, von wem die Nachricht stammt, und kann reagieren, indem er seinerseits eine Nachricht an *stern* sendet. Daraus kann sich eine regelrechte Sitzung in der Art eines Telefonats ergeben.

Wenn nämlich beide Benutzer ihre Meldungen nicht sofort beenden, sondern auf die Reaktion des Gegenübers warten, haben sie unmittelbar die Gelegenheit, auf die Meldung des Partners zu antworten. Der Kontakt zum anderen Benutzer bleibt schließlich solange aufrecht erhalten, bis er durch Eingabe des Dateiendezeichens (**Strg+D**) abgebrochen wird.

Nun bedeutet Telefonieren nicht immer Kommunizieren im Sinne von „Informationen austauschen“. Nur wenn bestimmte Regeln eingehalten werden, kann es zu einem vernünftigen Gespräch kommen. Reden etwa beide gleichzeitig, kann nicht davon ausgegangen werden, dass einer der beiden dem anderen überhaupt zuhört.

So gilt etwa beim Telefonieren die einfache Konvention, dass sich derjenige, der angerufen wird, mit seinem Namen meldet, woraufhin der Anrufer seinen Namen nennt, um dann sein Anliegen vorzutragen. Durch verschiedene sprachliche Mittel zeigt man dem Gegenüber jeweils an, dass man vorerst genug gesagt hat und auf eine Antwort wartet.

Ähnliche Regeln sollten auch bei einem „Gespräch“ mithilfe von *write* eingehalten werden. Normalerweise braucht man nicht davon auszugehen, dass der „Angerufene“ sich zu melden hat. Vielmehr wird man als „Anrufer“ zunächst seinen Wunsch vortragen, um dann auf eine Antwort zu warten.

Es ist zwar nur eine Konvention, aber es ist sinnvoll, sich daran zu halten, dass man dem Gegenüber mitteilt: „Ich bin so weit fertig, du bist dran!“ Üblicherweise bedient man sich dabei den Gepflogenheiten, die auch von den Funkern her bekannt sind. Dort wird das Ende einer Übermittlung mit dem Wort „out“ gekennzeichnet, das Ende der gesamten Übertragung mit „over and out“.

Ähnliches kann man auch bei einer Kommunikation mit *write* machen. Üblicherweise kennzeichnet man hier das Ende einer Meldung mit *(o)*, das der Übertragung mit *(oo)*.



Grundlegende Befehle



Das vorgenannte Beispiel wäre also korrekterweise wie folgt abzuwickeln:

```
$ write rattle←
Hallo rattle, kennst Du einen Befehl zum Vergleich von
zwei Dateien←
Wenn ja, bitte ich um baldige Benachrichtigung (oo)←
Strg+D
$
```

Dadurch wird signalisiert, dass keine unmittelbare Reaktion des anderen Benutzers erwartet wird.

In besonderen Fällen wird ein Benutzer nicht wollen, dass man ihn durch Meldungen in seiner Arbeit stört. Dann kann er mithilfe des Befehls *mesg* den Nachrichtenempfang regeln.

mesg n

verhindert den Empfang,

mesg y

lässt ihn wieder zu und

mesg

zeigt an, ob Nachrichtenempfang zugelassen ist oder nicht. Wollen Sie wissen, ob andere Benutzer Nachrichtenempfang zulassen oder nicht, können Sie das etwa so tun:

```
$ who -T
rattle    --    tty01      Jan 19 09:50
stern     +    tty02      Jan 19 08:50
root      +    tty1       Jan 19 09:31
```

Die Benutzer *root* und *stern* haben Nachrichtenempfang zugelassen (Zeichen + in der zweiten Spalte), Benutzer *rattle* nicht.

Eine Variante von *write* ist der Befehl *wall*, mit dem Nachrichten an alle Benutzer gleichzeitig gesandt werden. Der Ablauf entspricht dem von *write*.

```
$ wall←
Have a lot fun←
Strg+D
```

```
Message from stern@composter on tty5 at 08:04 ...
Have a lot fun
EOF
$
```



Benutzerumgebung



Nachdem Sie die Nachricht abgeschickt haben, erscheint sie auf *jedem* Bildschirm, also auch auf Ihrem. Seien Sie sparsam im Einsatz von *wall*: Bei der Arbeit könnten unnötige Mitteilungen als störend empfunden werden.

Eine stille Art der Nachrichtenübermittlung ist die „elektronische Post“ (*electronic mail* oder kurz *E-Mail*). Ist der Befehl *write* eine Nachrichtenübermittlung, die analog zum Telefon funktioniert, so kann man die elektronische Post sehr gut mit der „Gelben Post“ der Bundesrepublik vergleichen. Man schreibt seine „Post“, versieht sie mit einem Absender und wirft sie in den Briefkasten. Anders als bei der richtigen Post wird der Briefkasten nicht in regelmäßigen Abständen geleert, sondern unmittelbar nach dem Abwerfen in den Briefkasten.

Auf der anderen Seite jedoch wird der Briefkasten in regelmäßigen Abständen kontrolliert. Ist Post angekommen, so wird der Benutzer darüber informiert. Ein Dämonprozess kontrolliert in regelmäßigen Abständen, ob im Briefkasten neue Post angekommen ist. In diesem Fall erhält der Benutzer die folgende Nachricht:

```
You have new mail
```

Dieser Briefkasten befindet sich, wie bereits im *Kapitel 3 Linux-Dateisystem* erwähnt, als Unterverzeichnis in */var/spool/mail*. Das Versenden von Post ist ein schreibender (anhängend) Zugriff auf das Unterverzeichnis des entsprechenden Benutzers, das Öffnen der Post ein lesender und bisweilen schreibender Zugriff auf das eigene „Post“-Verzeichnis.

Das Versenden einer Nachricht erfolgt mittels:

```
mail benutzername
```

Will man Nachrichten über ein Netzwerk hinweg versenden, muss der Name des Hosts, an dem der Empfänger Zugang hat, zusätzlich in der folgenden Form angegeben werden:

```
mail benutzername@hostname
```

Man kann mehreren Benutzern gleichzeitig eine Mitteilung zukommen lassen, indem man all ihre Benutzernamen in die Befehlszeile aufnimmt:

```
mail root stern
```

sendet die Post an *root* und *stern*.

Der Ablauf gestaltet sich etwa wie folgt:

```
$ mail stern
Subject: Information zu Großwetterlage
Neueste Nachrichten zum Wetter sind augenblicklich
in der Datei /usr/all/wetter zu finden.
Datei wird alle drei Stunden upgedated
Nachrichtenquelle: (dpa)
$
```



Grundlegende Befehle



Zunächst fragt *mail* nach einem *Subject*. Dies lässt sich am ehesten mit einem Betreff in einem geschäftlichen Brief vergleichen, dieses *Subject* ist nämlich für den Empfänger ein erster Hinweis auf den Inhalt des „Briefes“. Liest er seine Post, so werden ihm in einer Übersicht zunächst sämtliche dieser *Subjects* angezeigt.

Der eigentliche Inhalt der Post wird dann eingegeben, wie man es schon gewohnt ist. Man kann zu diesem Zeitpunkt das Versenden der Post noch dadurch verhindern, dass man zweimal **[Strg]+C** drückt. Das (normale) Ende der Eingabe des Brieftextes erfolgt dadurch, dass man einen Punkt als einziges Zeichen einer Zeile eingibt oder wie gewöhnlich die Tastenkombination **[Strg]+D** drückt.

Um mögliche Schwierigkeiten beim Editieren des Textes zu vermeiden, kann man diesen mit einem Editor oder einem Textverarbeitungsprogramm vorbereiten.

Eine elegantere Methode ist die Benutzung eines ~-Befehls:

```
$ mail rattle←
Subject: Kurzreferenz zur bash←
~r bash.ref←
~.←
$
```



Auch unter *mail* gilt,
dass die Bestäti-
gung einer Zeile mit
← ein nachträ-
gliches Editieren
ausschließt.

Die Tilde (~) signalisiert *mail*, dass kein gewöhnlicher Text, sondern ein Befehl folgt. In diesem Fall wird mit *r dateiname* eine Datei in die zu übermittelnde Post übernommen.

Gelesen wird die Post mit:

```
mail
```

Man erhält eine Liste der eingegangenen Nachrichten mit Absender, Zeit und Betreff (dem *Subject*). Diese Liste sieht etwa wie folgt aus:

```
Mail Type ? for help.
"/var/spool/mail/rattle" 3 messages 1 new 1 unread
  1 stern Mon Jan 6 16:45 10/86    Nächste Termine
U  2 root Tue Jan 6 10:14 3/70
Passwortänderung vornehemen
>N 3 root Tue Jan 6 17:35 5/123 Wartung des
Systems
&
```

Gleich in der ersten Zeile wird man darauf hingewiesen, wie man Hilfe zu *mail* anfordern kann. Es folgt eine Statistik über das Postverzeichnis, die in diesem Falle besagt, dass drei Nachrichten vorliegen: eine neue, zwei schon bekannte, davon eine noch nicht gelesen.



Benutzerumgebung

Es folgen die Informationen zu den einzelnen Nachrichten. In der ersten Spalte steht eine Kennung über den Status der Nachricht (*N* = neu, *U* = ungelesen). Mit > ist zusätzlich die aktuelle Nachricht gekennzeichnet. Befehle innerhalb von *mail*, die eine oder mehrere Nachrichten betreffen, aber keine Angabe zur Nummer dieser Nachrichten machen, beziehen sich auf die aktuelle.

Die nächsten Spalten enthalten in dieser Reihenfolge:

- eine fortlaufende Nummer zur Identifizierung der Nachricht innerhalb von Befehlen,
- den Absender,
- Datum und Uhrzeit des Postversandes,
- Anzahl der Zeilen und Wörter der Nachricht (die erste enthält also 10 Zeilen mit insgesamt 86 Wörtern),
- den Betreff.

Zuletzt gibt *mail* einen Prompt (das &-Zeichen) aus, um anzuzeigen, dass es Befehle des Benutzers zur Bearbeitung der Post erwartet.

Die wichtigsten Befehle entnehmen Sie der *Tabelle 4.3*.

Befehle zur Bearbeitung der Post mit *mail*

Befehl	Bedeutung
<i>t[nachrichten]</i>	Nachrichten anzeigen
<i>d[nachrichten]</i>	Nachrichten löschen (das tatsächliche Löschen erfolgt erst beim Verlassen mit <i>q</i>)
<i>p[nachrichten]</i>	Nachrichten drucken
<i>e[nachrichten]</i>	Nachrichten bearbeiten
<i>u[nachrichten]</i>	gelöschte Nachrichten wiederherstellen
<i>s[nachrichten] datei</i>	Nachrichten an datei anhängen
<i>R[nachrichten]</i>	Nachricht beantworten
<i>m[benutzerliste]</i>	eigene Nachricht an Benutzer in benutzerliste
<i>n</i>	Anzeige der nächsten Nachricht
◀	wie <i>n</i>



Arbeiten mit der bash



Befehle zur Bearbeitung der Post mit *mail* (Forts.)

Befehl	Bedeutung
<i>q</i>	Verlassen von <i>mail</i> , endgültiges Löschen der Nachrichten in <i>/var/spool/mail/benutzername</i> , Kopieren der gelöschten Nachrichten in die Datei <i>mbox</i> im Heimatverzeichnis
<i>x</i>	Verlassen von <i>mail</i> , ohne Nachrichten zu löschen
<i>h</i>	Ausgabe der Liste der Nachrichten (wie zu Beginn der Sitzung mit <i>mail</i>)
<i>!befehl</i>	Ausführung eines beliebigen Linux-Befehls
<i>?</i>	Hilfe zu <i>mail</i>

Tabelle 4.3



Die nach den Befehlskürzeln optional verwendbaren *nachrichten* sind eine Liste von Nummern. Beispielsweise löscht *d 2* die Nachricht Nummer 2, *d 5-7* löscht die Nachrichten 5 bis 7.

Weitere Einzelheiten können interessierte Anwender den *man pages* von *mail* entnehmen.

Arbeiten mit der bash

Die Hauptaufgabe einer Shell besteht darin, Schnittstelle zwischen Mensch und Maschine, Anwender und System zu sein, d.h., die Shell wartet auf Eingaben des Benutzers und versucht, sie in Aktionen umzusetzen. In diesem Sinne ist sie wichtigster Bestandteil der Benutzerumgebung unter Linux.

In der UNIX-Philosophie allerdings ist die Shell eines von vielen Anwendungsprogrammen, aus denen man Nutzen ziehen kann. Dass dieser Nutzen darin besteht, interaktiv weitere Programme verwenden zu können, steht auf einem anderen Blatt.

Insofern ist es nicht verwunderlich, wenn UNIX und mit ihm Linux eine ganze Reihe unterschiedlicher Shells kennt. Die unter den Standard-UNIX-Varianten verbreitetste ist sicher die nach ihrem Entwickler benannte *Bourne-Shell*. Sie ist das Vorbild vieler anderer textorientierter Benutzeroberflächen nicht nur für UNIX/Linux geworden.

Wie jedes Programm hat auch die *Bourne-Shell* Schwächen, eine wesentliche ist die mangelhafte Unterstützung bei der Befehlseingabe. Stellen Sie sich vor, Sie haben am Beginn der Befehlszeile einen Tippfehler gemacht und können ihn nur dadurch korrigieren, dass Sie die gesamte Zeile bis zum Fehler löschen!

Benutzerumgebung

Die „wiedergeborene Muschel“

Solche Unzulänglichkeiten haben eine Weiterentwicklung dieser Shell geradezu provoziert. Eine dieser Weiterentwicklungen ist die *bash*. Ihr Name ist ein Wortspiel, er ist die Abkürzung für **b**orn **a**gain **s**hell, eine unüberhörbare Erinnerung an den Namen Bourne. Die *bash* ist die Standard-Shell für Linux.

Neben *Bourne* und *born again shell* gibt es viele andere textorientierte Benutzeroberflächen von UNIX/Linux, von denen an dieser Stelle vor allem die *C-Shell* Erwähnung finden soll, da sie einen nicht unerheblichen Einfluss auf die Entstehung der *bash* hatte. Unter anderem beinhaltet diese an der University of California (Berkeley) entstandene Shell einen *history*-Mechanismus, der eine raffinierte Methode der Befehlwiederholung darstellt.

Die Programmierer der *C-Shell* sind allerdings eigene Wege bei der Realisierung ihrer Benutzerschnittstelle gegangen, was dazu geführt hat, dass die beiden wichtigsten Benutzeroberflächen für UNIX nicht kompatibel zueinander sind. Dies wirkt sich besonders unangenehm bei der Nutzung von Shell-Skripten aus. Die unterschiedliche Handhabung von Variablen sowie ein unterschiedlicher Satz an internen Befehlen verhindert, dass Skripte, die unter der einen Shell entwickelt wurden, von der anderen korrekt ausgeführt werden können.

Da sowohl *Bourne*- als auch *C-Shell* eine weite Verbreitung in der UNIX-Welt haben, ist dieser Umstand insofern misslich, als dass zwei UNIX-Anwender unter Umständen schon aufgrund der Tatsache, dass sie verschiedene Shells benutzen, eine unterschiedliche Sprache sprechen. Auf die Problematik unterschiedlicher UNIX-Varianten soll an dieser Stelle gar nicht erst eingegangen werden.

Versuche, beide Benutzeroberflächen zu harmonisieren, gibt es schon sehr lange. Einer dieser Ansätze hat der UNIX-Gemeinde die ebenfalls nach ihrem Programmierer benannte *Korn-Shell* beschert. Sie entspricht im Wesentlichen der *Bourne-Shell*, beinhaltet aber auch Elemente der *C-Shell*.

Ähnlich ist in diesem Zusammenhang die *bash* zu bewerten. Sie ist prinzipiell eine *Bourne-Shell*, d.h., Anwender mit UNIX-Kenntnissen werden sehr viele ihrer Features kennen. Einige ihrer Möglichkeiten sind allerdings gegenüber dem Original neu, wie etwa der Befehlszeilen-Editor, eine nützliche Hilfe für die Eingabe.

Da nun die *bash* standardmäßig jedem Linux-Benutzer nach dem Einloggen zugewiesen wird, dient sie als Beispiel zur Beschreibung der Funktionsweise einer Shell. Andere textorientierte Benutzeroberflächen wie die *C-Shell* werden nicht weiter im Detail beschreiben. Auf wichtige Besonderheiten soll aber in einem eigenen Abschnitt eingegangen werden.





Dateinamenserweiterung

Zunächst ein einfacher Mechanismus, der in seinen Grundzügen in allen gängigen Betriebssystemen für PCs vorhanden ist, die Expansion von Dateinamen. Sollen mehrere Dateien gleichzeitig kopiert werden, kann man in der Befehlszeile natürlich die Namen aller Dateien angeben, z.B. kopiert

```
cp test1 test2 test3 ..
```

die drei Dateien *test1*, *test2* und *test3* ins übergeordnete Verzeichnis (nämlich ...). Linux bietet aber ebenso wie MS-DOS, Windows oder OS/2 die Möglichkeit, dass der Benutzer alle drei Dateinamen zu einem einzigen Argument der Befehlszeile zusammenfasst.

```
cp test* ..
```

Der * (Asterisk) ersetzt eine beliebige Zeichenfolge, d.h., *test** repräsentiert in dieser Befehlszeile alle Dateien, die mit der Zeichenfolge „*test*“ beginnen und beliebig enden. Dazu gehören *test1*, *test2* und *test3*.

Natürlich würden dazu auch, falls sie existierten, *test4*, *test31* oder *testastestcan* gehören. Der Einsatz des * in der Befehlszeile will also überlegt sein.

```
rm test*
```

löscht leider nicht nur *test1*, *test2*, *test3*, sondern eben auch alle anderen Dateien, deren Namen mit „*test*“ beginnen.

Einige lästige Einschränkungen, die in den eben genannten Betriebssystemen bezüglich des * gelten, entfallen in Linux. Er steht tatsächlich für eine *beliebige* Zeichenfolge, seine Wirkung endet nicht am Punkt vor der Dateinamenserweiterung!

*test** steht auch für Dateinamen wie *test.doc*, wie das folgende Beispiel zeigt:

```
$ ls
info          test.doc      test2      testfall
stuff         test1       test3      zdat
$ rm test*
$ ls
info          stuff        zdat
$
```

Der Punkt zu Beginn eines Dateinamens wird aber, wie im Kapitel 3 *Linux-Dateisystem* erläutert, gesondert behandelt. Der * ersetzt diesen Punkt nicht! Deshalb muss ein Löschen aller Dateien mit

```
rm * .*
```

erfolgen. Diese Eingabe bedeutet konkret: Lösche alle Dateien (sofern ihr Name nicht mit . beginnt) und außerdem alle Dateien, deren Name mit . beginnt.

○—————
Zu den Einzelheiten
der Befehlssyntax
von *cp* vergleichen
Sie Kapitel 3 *Linux-
Dateisystem*

Benutzerumgebung

Lassen Sie in der Befehlszeile dem * als „Platzhalter“ im Dateinamen noch weitere Zeichen folgen, dann wird dies bei der Dateinamensexpression berücksichtigt.

```
cp a*z /mnt
```

kopiert alle Dateien, deren Namen mit *a* beginnen und mit *z* enden (!) ins Verzeichnis */mnt*.

Linux kennt insgesamt folgende Zeichen, die für die Dateinamensexpression benutzt werden.

Zeichen für die Dateinamensexpression

Zeichen	Bedeutung
*	ersetzt eine beliebige, eventuell auch leere Zeichenfolge an der entsprechenden Stelle
?	ersetzt genau ein Zeichen an der entsprechenden Stelle
[zeichenliste]	ersetzt ein einzelnes Zeichen an der entsprechenden Stelle, sofern es in der zeichenliste ist
[!zeichenliste]	ersetzt ein einzelnes Zeichen an der entsprechenden Stelle, sofern es <i>nicht</i> in der zeichenliste ist

Tabelle 4.4

Das ? ist Ihnen wahrscheinlich in seiner Bedeutung vertraut. Es steht für ein einzelnes Zeichen in dem Sinne, dass

```
cp test? ..
```

alle Dateien kopiert, deren Namen mit „test“ beginnen, wobei der Name exakt fünf Zeichen lang sein muss. Eine Datei *test* würde *nicht* kopiert.

Die zeichenliste ist im Normalfall eine Aufzählung der Zeichen, die genau ein Zeichen ersetzen sollen. [abc] ersetzt etwa eines der drei Zeichen *a*, *b* oder *c*. Die Reihenfolge der Zeichen ist dabei unerheblich. Die einzelnen Zeichen werden ohne eine Trennung in den Klammern angegeben. Die Aneinanderreihung der Zeichen in den Klammern wird wie eine ODER-Verknüpfung interpretiert.

Eine elegante Lösung, die drei Dateien *test1*, *test2* und *test3* gleichzeitig zu kopieren (aber auch nur sie), ist mit dieser Form der Dateinamensexpression gegeben:

```
cp test[123] ..
```

Die Befehlszeile bedeutet: Kopiere die Dateien *test1* oder *test2* oder *test3* ins übergeordnete Verzeichnis.



Arbeiten mit der bash



Die Benutzung eines – (Minuszeichen) in zeichenliste wird von der bash als von ... bis interpretiert. `[o-9]` meint alle Ziffern, `[a-h]` die Kleinbuchstaben *a* bis *h*. Es lassen sich auch Kombinationen wie folgt benutzen:

```
chmod a-w /dev/tty0[1-46-9]
```

entzieht allen Benutzern das Schreibrecht *w* an den Dateien *tty01* bis *tty04* und *tty06* bis *tty09* im Verzeichnis */dev*. Die Datei *tty05* ist von diesem Rechteentzug nicht betroffen, da die 5 nicht in der zeichenliste auftaucht.

Schließlich bewirkt das Ausrufezeichen `!` eine Negation.

```
rm [ !a-zA-Z ] *
```

löscht alle Dateien, deren Namen nicht mit einem (Groß- oder Klein-)Buchstaben beginnen. Beachten Sie, dass in diesem Fall sowohl der `*` als auch [...] benutzt werden.

Ein Hinweis noch an DOS-User: Ein gleichzeitiges Kopieren und Umbenennen mehrerer Dateien nach dem Muster

```
copy test.* /tmp/alt.*
```

funktioniert unter UNIX/Linux so nicht!

Datenumleitung

Die Dateinamensexpression hat auf dem Weg über MS-DOS Eingang in die heute gängigen PC-Betriebssysteme gefunden. Wer in einem Anwendungsprogramm, das unter einer grafischen Benutzeroberfläche wie „Fenster 95“ oder auch „Fenster Neue Technologie“ läuft, eine Datei „öffnen“ will, wird feststellen, dass die Liste der verfügbaren Dateien dadurch eingeschränkt ist, dass nur solche Dateien angezeigt werden, die einem bestimmten Namensmuster entsprechen. Dieses wird meist mithilfe des `*` dargestellt, in der Regel unter Angabe einer Dateiendung, z.B. `*.txt`. Will man andere Dateien sehen, kann man das u.a. durch Änderung dieses Musters erzwingen.

Der Begriff „Öffnen“ meint eigentlich „Eine Datei zum Gebrauch bereitstellen“, nicht aber, sie in den RAM zu „laden“ und sie dann am Bildschirm anzuseigen.



Hinweis

Während Ihnen die Dateinamensexpression auch in anderen modernen Betriebssystemen ständig begegnen wird, ist die Datenumleitung in der Regel nur denjenigen geläufig, die mit DOS gearbeitet haben oder aber denen zumindest die Arbeit am Prompt bekannt ist.

Eine ganze Reihe von Linux-Befehlen, z.B. `cat` oder `ls`, nutzen den Bildschirm, um die Ergebnisse ihrer Arbeit anzuzeigen. Nun ist der Bildschirm aus Sicht von Linux eine Datei. Es macht für ein Programm, das unter Linux läuft, keinen wesentlichen Unterschied, ob es Daten in die Gerätedatei */dev/tty01*, d.h. auf den Bildschirm, oder in eine gewöhnliche Datei, wie zum Beispiel */home/rattle/testdatei*, ausgibt.

247



Benutzerumgebung

Bisweilen spricht man auch von „Dateideskriptoren“.

Bei gewöhnlichen Dateien spricht man natürlich vom „Speichern“ statt vom „Ausgeben der Daten“

Im zweiten Fall muss es Linux auffordern, die Datei zu öffnen. Dabei stellt das Betriebssystem eine Nummer, „handle“ genannt, zur Verfügung, über welche die weitere Behandlung der Datei (das Ausgeben der Daten) erfolgt. Im ersten Fall gibt es diese Nummer bereits.

Wie auch immer: Das Programm und Linux kommunizieren mithilfe des *handles*, um Daten in eine Datei auszugeben. Dabei ist es recht einfach für das System, diese Nummern zu manipulieren (dies ist hier nicht negativ gemeint!) und somit die Richtung des Datenstroms zu beeinflussen. Dies geschieht z.B. dann, wenn man es ausdrücklich in der Befehlszeile durch >, das „Umleitungszeichen“, anfordert.

Es wurde im *Kapitel 3 Linux-Dateisystem* bereits mehrfach benutzt, um auf einfache Weise Dateien zu erstellen. Z.B. erzeugt

```
$ cat > testdatei←  
Dies ist eine Testdatei←  
[Strg]+D  
$
```

die Datei *testdatei*, indem das System die Daten des Befehls *cat* statt auf dem Bildschirm in diese Datei ausgibt. Dabei wird sie zunächst geöffnet, wodurch wird ein neues *handle* verfügbar wird. Dieses wird von Linux statt des „Bildschirm“-handles benutzt, sobald *cat* Daten auf dem Bildschirm ausgeben will.

Jeder Prozess, d.h. jedes laufende Programm erbt von seinem Elternprozess drei *handles*: für die Standardeingabe (die Tastatur), die Standardausgabe (den Bildschirm) und die Standardfehlerausgabe (ebenfalls der Bildschirm). Einzelheiten entnehmen Sie *Tabelle 4.5*.

Hinweis



Einzelheiten zur Prozesssteuerung entnehmen Sie bitte dem *Kapitel 3 Linux-Dateisystem*.

Die *handles*, die jeder Prozess erbt (in Klammern der assoziierte Name)

Handle	Bedeutung
0	(<i>stdin</i>) Standardeingabe, d.h. Tastatur
1	(<i>stdout</i>) Standardausgabe, d.h. Bildschirm
2	(<i>stderr</i>) Standardfehlerdatei, das ist ebenfalls der Bildschirm

Tabelle 4.5



Arbeiten mit der bash



Öffnet die Shell nun eine zusätzliche Datei (nämlich *testdatei*), so erhält diese ebenfalls ein *handle*, zum Beispiel 3. Immer wenn *cat* auf den Bildschirm ausgeben will (*cat* will den Linux-Kernel veranlassen, Daten in die Datei mit dem *handle* 1 zu schreiben), ersetzt die Shell diese Nummer durch das *handle* 3. Linux wird die Daten also in *testdatei* hineinleiten! (Hier mogelt Linux also!)

Das Zeichen > in der Befehlszeile wird von der bash also dahingehend interpretiert, eine neue Datei zu erzeugen und bei allen Ausgabeoperationen, genauer: schreibenden Zugriffen auf eine Datei, das *handle* 1 durch das *handle* der neuen Datei zu ersetzen.

Das folgende Beispiel zeigt nun, dass nicht alle Ausgaben mithilfe des Zeichens > umgeleitet werden können.

```
$ cp altedatei neuedatei > unwichtig←
cp: altedatei: No such file or directory
$ ls alt*←
altdatei
$
```

Trotz der Ausgabumleitung erfolgt eine Bildschirmmeldung von *cp*, nämlich „No such file or directory“. Die Art der Ausgabe verdeutlicht aber sofort, dass ein Problem vorliegt: Die zu kopierende Datei wurde nicht gefunden. Eine Überprüfung mit *ls* zeigt den Grund: Der Dateiname war nicht korrekt eingegeben. *cp* hat also eine Fehlermeldung angezeigt.

Unter Linux können „normale“ Ausgaben von Fehlermeldungen unterschieden werden. Wenn ein Programm Daten am Bildschirm ausgeben will, hat es, wie Tabelle 4.5 zeigt, mindestens zwei Möglichkeiten, nämlich die Ausgabe über das *handle* 1, die Standardausgabe, und über das *handle* 2, die Standardfehlerdatei!

Ohne besondere Vorkehrung erfolgen Ausgaben dabei stets über *handle* 1. Nur wenn der Programmierer dies ausdrücklich anfordert, erfolgen sie mittels *handle* 2. Gehen Sie davon aus, dass für alle Standardbefehle von Linux gilt, dass im Fehlerfall die Standardfehlerdatei zur Ausgabe benutzt wird. Das Beispiel oben belegt dies.

Natürlich lässt sich auch die Ausgabe solcher Fehlermeldungen umleiten. Das geschieht einfach dadurch, dass dem Umleitungszeichen die Nummer des *handles* der Standardfehlerdatei vorangestellt wird.

```
cp altedatei neuedatei 2> unwichtig
hätte den Text
cp: altedatei: No such file or directory
in die Datei unwichtig umgeleitet und nicht am Bildschirm angezeigt.
```



Benutzerumgebung

Dieses Verfahren lässt sich verallgemeinern. Stellt man dem > eine Nummer voran, so veranlasst dies die bash, das entsprechende *handle* (in vorliegendem Beispiel 2 für die Standardfehlerdatei) durch das *handle* der neu zu erstellenden Datei zu ersetzen. Letztendlich ist der Gaudizacken > nur eine abkürzende Schreibweise für >, was Sie mit folgendem Beispiel leicht nachprüfen können.

```
$ echo Hallo 1> umleitung1
$ cat umleitung1
Hallo
$
```

Es handelt sich hierbei um den „Anfügen-Modus“
(append mode)

Durch die Ausgabeumleitung mit > wird die Datei, deren Name dem Umleitungszeichen folgt, neu erstellt. Das ist problematisch, wenn eine Datei dieses Namens bereits existiert; denn dann wird der Inhalt der vorhandenen Datei durch die neuen Daten überschrieben.

Will man dies vermeiden, benutzt man zwei Umleitungszeichen, also >>.

Ein einfaches Beispiel soll die Nutzung von >> demonstrieren.

```
$ echo Pfade für Dateien namens passwd > passwd.pfad
$ find / -name passwd -print >> passwd.pfad 2>
/dev/null
$ cat passwd.pfad
Pfade für Dateien namens passwd
/usr/bin/passwd
/etc/passwd
$
```

Zunächst wird ein Text in eine Datei *passwd.pfad* umgeleitet. Anschließend lässt man den Befehl *find* nach Dateien suchen. Seine Meldungen werden ebenfalls nach *passwd.pfad* umgeleitet. Durch Benutzung zweier Größerzeichen wird verhindert, dass die Datei dabei neu angelegt wird, stattdessen werden die neuen Daten an die alten angehängt.

Sie können >> natürlich auch einsetzen, wenn die Zielfile noch nicht existiert, dann wird sie angelegt. In diesem Fall haben > und >> die gleiche Wirkung.

Auch das Voranstellen von *handles* ist im Zusammenhang mit >> möglich.

```
find / -name passwd -print 2>> error.dat
```

leitet mögliche Fehlermeldungen von *find* in die Datei *error.dat* um, ohne ihren bisherigen Inhalt zu löschen.

Das letzte Beispiel zeigt noch zwei Besonderheiten. Offensichtlich ist es möglich, die Standard- und die Standardfehlerausgabe gleichzeitig umzuleiten. Die „normale“ Ausgabe des Befehls *find* erfolgt in die Datei *passwd.pfad*, Fehlermeldungen werden in die Datei */dev/null*. Diese kann man als Zielfile angeben, wenn



Arbeiten mit der bash



- keine neue Datei erzeugt und gleichzeitig
- keine bestehenden Daten überschrieben werden sollen.

Die Umleitung nach `/dev/null` ist eine elegante Möglichkeit, die Ausgabe von Informationen am Bildschirm zu unterdrücken, ohne dabei eine neue Datei anzulegen.

Wollen Sie gleichzeitig die Standardausgabe *und* die Fehlermeldungen in ein und dieselbe Datei umleiten, so müssen Sie beachten, dass die Datei durch `>` bzw. `2>` zweimal angelegt wird. Ferner würde die Shell annehmen, dass sowohl Fehler als auch „normale“ Ausgaben zunächst an den Anfang der Datei zu schreiben wären. Dies kann zu recht merkwürdigen Ergebnissen führen. In diesem Fall ist das doppelte Umleitungszeichen nützlich.

```
find / -name passwd -print >> fp 2>> fp
```

ist eine saubere Lösung für das geschilderte Problem. Alle Ausgaben von `find` werden nach und nach in die Datei `fp` geschrieben. Das gleiche Ergebnis erzielt man, wenn folgende Umleitungsvariante benutzt:

```
find / -name passwd -print > fp 2>&1
```

Die Angabe `2>&1` veranlasst die bash, die Fehlermeldungen in den normalen Datenstrom einzubinden.

Es gibt eine ganze Reihe weiterer Umleitungsvarianten. Der an Einzelheiten interessierte Leser sei in diesem Zusammenhang auf die *man pages* für die bash verwiesen.

Neben der Ausgabe lässt sich auch die Eingabe umleiten. Sie sollten dabei aber beachten, dass die umgeleiteten Eingabedaten für das entsprechende Programm geeignet sind.

So ruft

```
rm -i *
```

den Löschenbefehl `rm` als interaktives Tool auf. Für jede einzelne Datei wird auf eine Beantwortung der Sicherheitsabfrage gewartet, wobei `Y` ein Löschen bewirkt und die Eingabe jedes anderen Zeichens ein Löschen verhindert.

Eine Umleitung derart, dass `rm` seine Eingaben aus einer Datei, z.B. `.profile`, statt von der Tastatur erhält, ist unvernünftig. Die Datei `.profile` enthält zwar recht nützliche Informationen, leider lassen sich diese im Zusammenhang mit `rm` aber nicht sinnvoll einsetzen.

Bedenken Sie immer, dass bei Eingabeumleitungen der interaktive Charakter eines Befehls – sofern er interaktiv arbeitet – verloren geht. Eingriffe über die Tastatur sind nach der Bestätigung der Befehlszeile mit `Esc` nicht mehr möglich!



Die Datei `/dev/null` ist eine „Pseudo-Datei“, d.h. sie hat keinen Inhalt. Daten, die durch Umleitung oder Kopie oder ... in die Datei ausgegeben werden, landen im Nirgendwo, dem „Computer-Nirwana“, wie man es gelegentlich nennt. Sie werden weder (in einer gewöhnlichen Datei) gespeichert noch an einem Gerät (über die entsprechende Gerätedatei) ausgegeben.



Benutzerumgebung



Die Umleitung der Eingabe wird durch das Zeichen < ermöglicht. So kann man mit

```
write root < text
```

dem Superuser eine vorbereitete Mitteilung zukommen lassen. Die übliche Eingabe über Tastatur wird ersetzt durch den Inhalt der Datei *text*. Das Dateiendezeichen (üblicherweise über die Tastatur mit **Strg**+**D** erzeugt) braucht natürlich nicht explizit eingegeben zu werden, es wird mit dem Ende der Datei automatisch übermittelt.

Analog lässt sich auch vorbereitete Post versenden.

```
mail -s "Das Wetter morgen" root < info
```

sendet die Datei *info* in die Mailbox des Superusers. Ein *Subject* wird mithilfe der Option -s vorangestellt.

Natürlich lässt sich auch die Eingabe für den Befehl *cat* umleiten. Betrachten Sie etwa die Befehlszeile

```
cat < .profile
```

Dem Befehl *cat* ist keine Datei als Argument übergeben worden. Er würde deshalb die Standardeingabe, also die Tastatur, als Argument ergänzen, d.h., *cat* würde auf eine Eingabe von der Tastatur warten.

Andererseits ist die Eingabe von der Tastatur umgelenkt worden auf die Datei *.profile*. Also wird *cat* deren Inhalt ausgeben. Dies entspricht natürlich der Eingabe von

```
cat .profile
```

Auch hier gibt *cat* den Inhalt der Datei *.profile* aus. Allerdings wird in diesem Fall die bash nur insofern aktiv werden, als sie den Befehl *cat* zur Ausführung bringt. Im ersten Fall jedoch muss sie zusätzlich die Datenströme umleiten. Die interne Abwicklung beider Befehle ist also unterschiedlich, für den Anwender ist das Ergebnis aber in beiden Fällen gleich.

Die besondere Konstruktion der Befehle *cat*, *less* oder *more* erlaubt solche Umleitungen. Notwendig sind sie allerdings nicht. Man kann es einfacher haben.

Here-Dokumente

Eine besonders raffinierte Form der Eingabeumleitung stellen die *Here*-Dokumente dar. Die Umleitung wird durch << eingeleitet. Sinnvolle Beispiele für die Anwendung von *Here*-Dokumenten ergeben sich allerdings erst bei der Erstellung von Shell-Skripten. An dieser Stelle sollen nur Syntax und Funktionsweise erklärt werden.



Arbeiten mit der bash



Die Umleitung der Eingabedaten erfolgt nicht aus einer zu benennenden Datei. Die Quelle der Daten sind vielmehr die folgenden Zeilen von der Tastatur oder, im Falle der Shell-Skripte, die folgenden Zeilen des Shell-Skripts selbst. Ein einfaches Beispiel soll den Ablauf verdeutlichen.

```
$ grep Datei <<+  
> cp: Dateien kopieren  
> mv: Dateien verschieben oder umbenennen  
> rm: Dateien löschen  
> mkdir: Verzeichnisse anlegen  
> rmdir: Verzeichnisse löschen  
> +  
cp: Dateien kopieren  
mv: Dateien verschieben oder umbenennen  
rm: Dateien löschen  
$
```

Der Befehl *grep* durchsucht normalerweise eine Datei nach Text und gibt die Zeilen der Datei aus, in denen der Text vorkommt. In diesem Beispiel wird eine Umleitung vorgenommen, um zu veranlassen, dass *grep* seine Daten weder aus einer Datei noch von der Standardeingabe liest. Die bash erkennt nämlich die Umleitung und fordert uns durch einen geänderten Prompt (nämlich *>*) auf, die von *grep* zu durchsuchenden Daten einzugeben.

Das *+* kennzeichnet dabei Beginn und Ende Ihrer Eingabe, ähnlich wie in vielen Fällen die Begrenzung von Text durch „...“ erfolgt. Sie können dabei statt des *+* auch ein beliebiges anderes Zeichen verwenden, sollten aber beachten, dass es keine besondere Bedeutung für die Shell hat.

Natürlich wäre das obige Beispiel auch ohne Umleitung möglich gewesen.

```
$ grep Datei  
cp: Dateien kopieren  
mv: Dateien verschieben oder umbenennen  
rm: Dateien löschen  
mkdir: Verzeichnisse anlegen  
rmdir: Verzeichnisse löschen  
Strg+D  
cp: Dateien kopieren  
mv: Dateien verschieben oder umbenennen  
rm: Dateien löschen  
$
```

Hier wartet *grep* auf die Eingabe über die Standardeingabe. Diese am Prompt einfache Form lässt sich nicht in Shell-Skripten realisieren, speziell wenn Interaktivität nicht gewünscht ist.



Benutzerumgebung



Die Idee der Umleitung in diesem Beispiel ist die Erstellung einer kleinen Hilfesdatei zu einfachen Linux-Befehlen. Zielvorstellung ist, diese Datei als Shell-Skript aufrufen zu können unter Angabe eines Suchbegriffs, etwa dem Wort „Datei“, was dann eine Ausgabe aller Befehle zur Dateibearbeitung zur Folge hätte.

Ein erster Schritt in diese Richtung wäre etwa

```
$ cat > minihelp<input type="button" value="↶" style="border: none; border-bottom: 1px solid black; padding: 0 5px; margin-right: 5px;"/>
grep Datei <<+<input type="button" value="↶" style="border: none; border-bottom: 1px solid black; padding: 0 5px; margin-right: 5px;"/>
cp: Dateien kopieren<input type="button" value="↶" style="border: none; border-bottom: 1px solid black; padding: 0 5px; margin-right: 5px;"/>
mv: Dateien verschieben oder umbenennen<input type="button" value="↶" style="border: none; border-bottom: 1px solid black; padding: 0 5px; margin-right: 5px;"/>
rm: Dateien löschen<input type="button" value="↶" style="border: none; border-bottom: 1px solid black; padding: 0 5px; margin-right: 5px;"/>
mkdir: Verzeichnisse anlegen<input type="button" value="↶" style="border: none; border-bottom: 1px solid black; padding: 0 5px; margin-right: 5px;"/>
rmdir: Verzeichnisse löschen<input type="button" value="↶" style="border: none; border-bottom: 1px solid black; padding: 0 5px; margin-right: 5px;"/>
+
Strg+D
$ chmod u+x minihelp<input type="button" value="↶" style="border: none; border-bottom: 1px solid black; padding: 0 5px; margin-right: 5px;"/>
$ minihelp<input type="button" value="↶" style="border: none; border-bottom: 1px solid black; padding: 0 5px; margin-right: 5px;"/>
cp: Dateien kopieren
mv: Dateien verschieben oder umbenennen
rm: Dateien löschen
$
```

Die Datei *minihelp* wird erzeugt. Mit *chmod* wird sie ausführbar gemacht und schließlich aufgerufen.

Erste Zeile der Datei ist der Befehl *grep*. Die Umleitung ermöglicht es, die zu durchsuchenden Daten unmittelbar an diese Befehlszeile anzuhängen, eine zweite Datei ist nicht notwendig. *grep* durchsucht die durch das +-Zeichen begrenzten Zeilen nach dem Wort „Datei“. Verbesserungswürdig ist natürlich noch, dass *grep* nur nach diesem Wort sucht und durch den Aufruf von *minihelp* nicht veranlasst werden kann, ein anderes Wort aufzuspüren.

Pipelines

Mit der Umleitung eng verwandt ist die Möglichkeit der Pipelines. Sie ist in ihrem Ergebnis vergleichbar einer Ausgabeumleitung mit einer anschließenden Eingabeumleitung. Beteiligt sind zumindest zwei Befehle, wobei der eine Daten – für den Bildschirm – bereitstellt, die der zweite als Eingabe benutzt.

Eine Möglichkeit dieses Datenaustauschs über Umleitung zeigt das folgende Beispiel.

```
$ echo Das System wird heruntergefahren > down.txt<input type="button" value="↶" style="border: none; border-bottom: 1px solid black; padding: 0 5px; margin-right: 5px;"/>
$ wall < down.txt<input type="button" value="↶" style="border: none; border-bottom: 1px solid black; padding: 0 5px; margin-right: 5px;"/>
```



Arbeiten mit der bash



Der Text wird mittels *echo* in eine Datei umgeleitet. Aus dieser heraus benutzt sie der Befehl *wall*. Der Vorteil dieser Methode ist, dass man Standardtexte nicht immer wieder neu schreiben muss, der Inhalt von *down.text* steht auch weiterhin zur Verfügung.

```
$ ls -l /dev > dev.inhalt
$ less < dev.inhalt
```

Das vorstehende Beispiel zeigt eine andere Variante dieses Verfahrens. Der Nachteil in diesem Fall ist, dass nebenbei eine Datei entstanden ist, die im Weiteren nicht mehr benötigt wird. Sie ist nun nachträglich zu löschen.

Darüber hinaus wird bei dieser Art der Datenübergabe eines Befehls an einen anderen vorausgesetzt, dass die Rechtesituation im aktuellen Verzeichnis das Anlegen von Dateien erlaubt. Ist das aktuelle Verzeichnis die Wurzel, so wird man als gewöhnlicher Benutzer beim ersten Befehl durch eine Fehlermeldung darauf aufmerksam gemacht, dass die gewünschte Operation nicht möglich ist.

Pipelining berücksichtigt nun die Rechtesituation dadurch, dass keine neuen Dateien angelegt werden. Der Austausch der Daten zwischen den Programmen erfolgt über einen im Arbeitsspeicher bereitgestellten Puffer.

Das letzte Beispiel zur Ausgabe des Inhalts des Verzeichnisses */dev* bei Benutzung von Pipelines liest sich nun so

```
ls -l /dev | less
```

ls übergibt via Puffer seine Daten an *less* zur seitenweise Anzeige. Diese Methode ist vielen Lesern bekannt, die mit kommandozeilenorientierten Systemen Erfahrung haben. Allerdings verblüfft Linux immer wieder durch seine Vielfalt an Möglichkeiten zum Einsatz von Pipelines. Dies liegt darin begründet, dass die Ausgabebefehle in der Regel so konstruiert sind, dass sie sowohl Dateien bearbeiten können als auch Eingaben über die Tastatur (Standardeingabe). Eines der vielen Beispiele ist *less*.

Wer die Ausgabe der diversen Befehle näher analysiert, kann auf dieser Basis Befehle sehr zweckmäßig miteinander verbinden.

Sie wollen wissen, wie viele Dateien im Verzeichnis */usr/bin* vorhanden sind, ohne selbst zählen zu müssen?

```
ls /usr/bin | wc -w
```

liefert das gewünschte Ergebnis. *ls* selbst gibt ausschließlich die Dateinamen aus, *wc -w* zählt die einzelnen Wörter.

```
who | wc -l
```

etwa zählt die Anzahl der eingeloggten Benutzer; denn *who* gibt für jeden Benutzer exakt eine Zeile aus.



Benutzerumgebung



```
who | grep root
```

zeigt, ob der Superuser eingeloggt ist. Dies hat man bei kleinen Systemen natürlich auch ohne Pipelining schnell gesichtet, was aber, wenn sich 50 oder 100 Benutzer eingeloggt haben?

```
cat info | mail -s "Wetterinformationen" root
```

sendet den Inhalt der Datei *info* als Post an den Superuser. Mit Hilfe des Schalters *-s* wird der Information ein Betreff vorangestellt.

Auch komplexe Pipelines lassen sich konstruieren.

```
ls -l /dev | grep fd | more
```

listet Seitenweise ausführliche Informationen über Floppies auf (genauer Gerätedateien, deren Namen die Buchstabenkombination *fd* enthalten – es muss natürlich nicht unbedingt der Name sein, aber andere Angaben, die diese Kombination enthalten, dürften in der Regel nicht zu finden sein).

Ein nützliches Werkzeug innerhalb von Pipelines ist der Befehl *tee*. Seine allgemeine Syntax lautet

```
tee[ -option][ datei]
```

Er erwartet Daten über die Standardeingabe und gibt sie dann sowohl auf dem Bildschirm als auch in *datei* aus. Eine wichtige Option ist *-a*, die verhindert, dass möglicherweise in der Datei enthaltene Daten überschrieben werden. Die neuen Daten werden an die bisherigen angehängt (*append*).

Der Befehl zeigt seine Stärke besonders im Zusammenhang mit Pipelines. Die Eingabe über Tastatur wird dann durch die Daten ersetzt, die der in der Pipeline vorhergehende Befehl produziert.

```
who | tee angemeldet | grep root
```

wiederholt im Wesentlichen das Beispiel von oben (es soll in Erfahrung gebracht werden, ob der Superuser eingeloggt ist), mit dem Unterschied, dass man sich nun bei Bedarf zu einem späteren Zeitpunkt die Liste der eingeloggten Benutzer noch einmal anzeigen lassen kann, sie wird nämlich in der Datei *angemeldet* abgelegt.

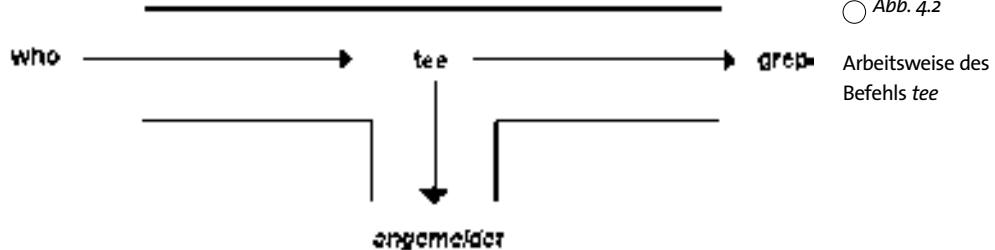
Das Funktionsprinzip von *tee* lässt sich der Abbildung 4.2 entnehmen.



Arbeiten mit der bash



Abb. 4.2



Arbeitsweise des Befehls **tee**

Die Herkunft des Namens *tee* erschließt sich ebenfalls aus diesem kleinen Beispiel.

Wie eine Shell arbeitet

Auf der Basis der bereits bekannten Möglichkeiten der Shell, kann nun Einblick in die Arbeitsweise der bash gegeben werden.

Es wurde bereits gezeigt, dass eine Reihe von Zeichen, die so genannten „Metazeichen“, für die Shell von besonderer Bedeutung sind. Wie konsequent die Behandlung der Metazeichen durch die Shell ist, mögen die folgenden Beispiele verdeutlichen.

```
$ echo Hallo Welt
Hallo Welt
$ echo "Hallo Welt"
Hallo Welt
$
```

Die Abarbeitung der ersten Befehlszeile zeigt, dass von der Eingabe bis zur Ausgabe einige Leerzeichen verloren gehen. Man möchte annehmen, dass dem Befehl *echo* der Text *Hallo Welt* zur Ausgabe am Bildschirm übergeben wurde. Wäre dies so, müssten Sie schlussfolgern, dass *echo* Ihren Text vor der Ausgabe bearbeitet, indem *echo* Leerzeichen entfernt.

Die zweite Variante zeigt eine Abarbeitung ohne Verlust von Leerzeichen. Offenbar haben die Anführungszeichen verhindert, dass Leerzeichen entfernt wurden. Nun, sowohl die Leerzeichen (in ihrer Funktion als Trennzeichen) als auch die Anführungszeichen sind Metazeichen der bash, die von ihr bei der Bearbeitung der Befehlszeile besonders behandelt werden. Überflüssige Leerzeichen werden ebenso entfernt wie die Anführungszeichen, mit der Besonderheit, dass der Text zwischen den Anführungszeichen unverändert bleibt.

Noch deutlicher zeigt das nächste Beispiel, wie das Zusammenspiel zwischen bash und auszuführendem Befehl funktioniert.



Benutzerumgebung

```
$ ls[]  
datei1      datei2  
$ echo *[]  
datei1  datei2  
$
```

Es scheint, als sei der *ls*-Befehl überflüssig, da doch *echo* das Gleiche leistet. Das merkwürdige Verhalten von *echo* liegt natürlich wieder darin begründet, dass mit dem * in der Befehlszeile ein bash-Metazeichen enthalten ist. Hier wird deutlich, dass der Befehl das Metazeichen gar nicht zur Ausgabe erhält: Bevor *echo* gestartet wird, interpretiert die bash die in der Befehlszeile enthaltenen Metazeichen.

Der Asterisk * steht dabei, wie oben beschrieben, für Dateinamensexpression. Die bash erzeugt eine Liste der Dateinamen, die dem gegebenen Muster entsprechen – hier: alle Dateien des aktuellen Verzeichnisses, sofern ihr Name nicht mit . beginnt. Es ist einsichtig, dass eine solche Expansion nicht von einem einfachen Ausgabebebefehl durchgeführt wird.

Zu den gleichen Ergebnissen gelangt man, wenn man die Ausgabeumleitung wie folgt einsetzt.

```
$ ls -l[]  
total 2  
-rw-r--r--  1 rattle  users       39 Jan 19 01:36 datei1  
-rw-r--r--  1 rattle  users      75 Jan 19 01:36 datei2  
$ > neudat[]  
$ ls -l[]  
total 2  
-rw-r--r--  1 rattle  users       39 Jan 19 01:36 datei1  
-rw-r--r--  1 rattle  users      75 Jan 19 01:36 datei2  
-rw-r--r--  1 rattle  users        0 Jan 19 01:36 neudat  
$
```

Sie können sich von der Korrektheit der Angaben leicht überzeugen, indem Sie das Beispiel nachvollziehen

Die zweite Eingabezeile – > *neudat* – enthält keinen Befehl, wohl aber ein Metazeichen mit dem in diesem Falle dazugehörigen Dateinamen. Die bash führt ungehört ihren Auftrag aus und legt die Datei *neudat* an, wie eine Prüfung vor und nach der Eingabe durch *ls* belegen. Da kein Befehl in der Zeile enthalten ist, geschieht sonst nichts, daher verwundert es auch nicht, wenn die Größe der neuen Datei mit 0 angegeben wird.

Noch skurriller erscheint auf den ersten Blick der Versuch

```
ls -l > neudat2
```

Führen Sie einen Vergleich des Inhalts des aktuellen Verzeichnisses mit dem Inhalt der Datei *neudat2* durch.



Arbeiten mit der bash



```
$ ls -l ↵
total 3
-rw-r--r-- 1 rattle  users   39 Jan 19 01:36 datei1
-rw-r--r-- 1 rattle  users   75 Jan 19 01:36 datei2
-rw-r--r-- 1 rattle  users    0 Jan 19 01:36 neudat
-rw-r--r-- 1 rattle  users  257 Jan 19 01:37 neudat2
$ cat neudat2 ↵
total 2
-rw-r--r-- 1 rattle  users   39 Jan 19 01:36 datei1
-rw-r--r-- 1 rattle  users   75 Jan 19 01:36 datei2
-rw-r--r-- 1 rattle  users    0 Jan 19 01:36 neudat
-rw-r--r-- 1 rattle  users    0 Jan 19 01:37 neudat2
$
```

Natürlich, die Datei *neudat2* existiert, und ihr Inhalt ist die Liste der Dateien des aktuellen Verzeichnisses in ausführlichem Format – schließlich wurden die Daten von *ls -l* umgeleitet.

Was aber bei der Analyse des Inhalts von *neudat2* auffällt, ist das Erscheinen von *neudat2* in der Dateiliste und natürlich die Größe von 0 Byte. Dies erklärt sich aber aus der Reihenfolge, in der die Befehlszeile

```
ls -l > neudat2
```

abgearbeitet wurde. Die bash hat zunächst aufgrund des Metazeichens die Datei *neudat2* erstellt, natürlich ohne sie mit Inhalt zu füllen (vergleichen Sie dazu auch das vorhergehende Beispiel). Nachdem sie nun existiert, wird der Befehl *ls* ausgeführt, der seinerseits die neue Datei mit ihren 0 Byte im aktuellen Verzeichnis findet und dann seine gesamten Informationen auf dem Bildschirm ausgeben will. Die bash allerdings erwirkt durch Manipulation der *handles*, dass die Daten in die neue Datei geschrieben werden. Danach ist sie 257 Byte groß, was *ls* ja schon angezeigt hat.

Die Abbildung 4.3 enthält ein vereinfachtes Ablaufschema der Arbeit der bash. Nach der Benutzereingabe erfolgt also zunächst eine Untersuchung der Eingabe auf Metazeichen, bei deren Vorhandensein eine entsprechende Aktion ausgeführt wird (*, ? und [...] führen zur Dateinamensexpansion, > und seine Variationen gegebenenfalls zur Erzeugung einer Datei und einer Manipulation von *handles*, um bereits bekannte Beispiele zu nennen). Das bearbeitete Metazeichen wird dabei natürlich aus der Zeile entfernt.

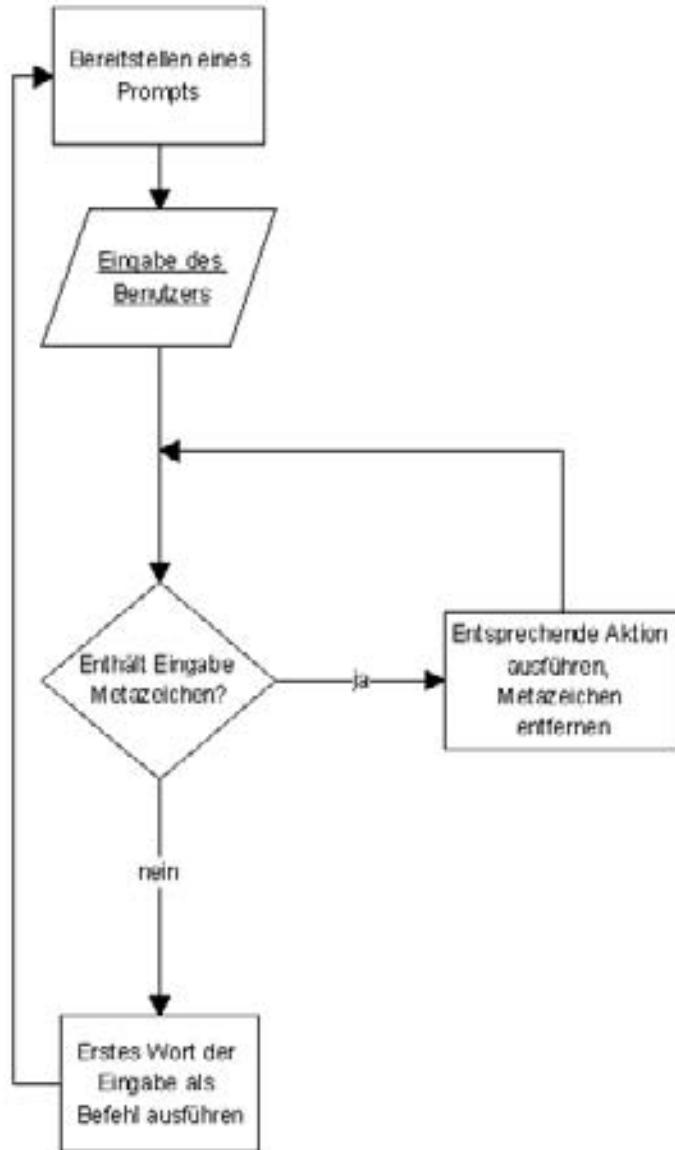
Danach wird das erste Wort der Befehlszeile genommen und als Befehl zur Ausführung gebracht. Dies werden Sie noch genauer betrachten haben, wenn Sie Suchpfade kennen gelernt haben.

Wie Sie gesehen haben, muss ein solches erstes Wort gar nicht existieren, dann ist die Arbeit der Shell bereits mit der Abarbeitung beendet.



Benutzerumgebung

Abb. 4.3 ○
Arbeitsweise der bash



Zum Abschluss dieses Abschnitts sollen die vorgenannten Ausführungen mit einem etwas absonderlichen Beispiel erhärtet werden.

```
$ mkdir neu[←]  
$ cd neu[←]  
$ ln /usr/bin ls[←]  
$ cat > test1[←]  
Dies ist eine erste Testdatei[←]
```



Arbeiten mit der bash

```
[Strg]+D
$ cat > test2
Dies ist eine zweite Testdatei
[Strg]+D
$ ls
ls      test1      test2
$ *
test1      test2
$ rm t*
$ ls
ls
$ *
ls
$
```

Für dieses Beispiel benötigen Sie ein leeres Verzeichnis, das mit dem Namen *neu* erstellt und zum aktuellen Verzeichnis gemacht wird. Dann erzeugen Sie einen Hard-Link auf die Datei */usr/bin* und nennen ihn *ls*. Schließlich legen Sie mittels *cat* zwei einfache Testdateien an und prüfen den Inhalt des Verzeichnisses mit *ls*.

Die dann folgende Zeile ist der Clou des Beispiels, lediglich ein * gefolgt von *ls*.

Erinnern Sie sich, wie die bash in jedem Fall eine Eingabezeile bearbeitet: Sie sucht nach Metazeichen. In diesem Fall findet sie den *. Die Aktion, die er auslöst, ist die Erzeugung einer Liste mit Dateinamen, die dem gewählten Muster entsprechen, also eine Liste aller Namen des aktuellen Verzeichnisses. Dabei wird der * aus der Befehlszeile entfernt.

Diese Liste haben Sie aber unmittelbar zuvor ausgeben lassen. Die von der Shell aufbereitete Befehlszeile lautet also:

```
ls test1 test2
```

Zufälligerweise ist das erste Wort dieser Zeile ein Linux-Befehl. *ls* soll die beiden Dateien *test1* und *test2* anzeigen, was auch tatsächlich geschieht.

Als Gegenprobe werden nun die beiden Testdateien gelöscht, und es wird ein erneuter Versuch mit

```
* ls
```

unternommen. Da nur noch die eine Datei im Verzeichnis verbleibt, lautet die umgewandelte Befehlszeile nun

```
ls
```

Zur Ausführung gebracht, wird der Inhalt des aktuellen Verzeichnisses angezeigt, die Datei *ls* eben.

Benutzerumgebung



Maskierung der Metazeichen

Wie gezeigt, besteht die erste Aufgabe der Shell bei der Analyse der Befehlszeile darin, die Eingabe nach Metazeichen zu durchsuchen. Sie haben schon im *Kapitel 3 Linux-Dateisystem* bei der Beschreibung des Befehls *grep* Beispiele dafür kennen gelernt, dass es bisweilen notwendig ist, der bash die Möglichkeit zu nehmen, Metazeichen zu interpretieren.

Kehren Sie noch einmal zu dem einfachen Beispiel zu Beginn des letzten Abschnitts zurück. Mittels *echo* sollte ein Text mit mehreren aufeinander folgenden Leerzeichen ausgegeben werden. Dabei wurde gezeigt, dass die Shell auch die „überflüssigen“ Leerzeichen eliminiert hat. Eine Lösung dieses kleinen Problems wurde in Form von

```
echo "Hallo    Welt"
```

mitgeliefert. Die Anführungszeichen „...“ sind eine von drei Möglichkeiten, die bash an der Interpretation von Metazeichen zu hindern. Man nennt diesen Vorgang auch „Maskierung von Metazeichen“. Tabelle 4.6 gibt einen kurzen Überblick über die Arten der Maskierung.

Maskierung von Zeichen(folgen)



Maskierung	Bedeutung
\zeichen	Maskierung von zeichen, z.B. maskiert * den asterisk, eine Dateinamensexpansion erfolgt nicht
"zeichenfolge"	maskiert die komplette zeichenfolge. Das Zeichen " darf natürlich nicht in zeichenfolge enthalten sein. Beispiel: "<>*" lässt die drei Zeichen unverändert in der Befehlszeile.
„zeichenfolge“	maskiert die komplette zeichenfolge. Ausgenommen von der Maskierung sind allerdings die Zeichen \, \$, „, und die Anführungszeichen („) selbst. Unterdrückt in jedem Fall die Dateinamensexpansion.

Tabelle 4.6



Die der Maskierung dienenden Zeichen sind natürlich ebenfalls Metazeichen. Sie haben schließlich für die bash eine wichtige Bedeutung: Sie teilen ihr mit, das oder die folgenden Zeichen nicht zu interpretieren. Die Bildschirmausgabe eines * lässt sich also mit

```
echo \*
```



Arbeiten mit der bash



erzwingen. Der vorangestellte *backslash* (\) verhindert die Interpretation durch die bash, das Metazeichen \ wird bei diesem Vorgang aus der Befehlszeile entfernt, so dass als effektive Befehlszeile, als Befehlszeile also, die von der Shell nicht weiter bearbeitet wird, übrig bleibt:

```
echo *
```

Der * wird nun dem echo-Befehl übergeben, so dass er ihn am Bildschirm anzeigen kann.

Es lässt sich mithilfe der Maskierungszeichen jede beliebige Zeichenfolge eingeben, ohne dass die Shell die Zeichen interpretiert. Dabei sollte jedoch speziell der Anfänger Vorsicht walten lassen – insbesondere, wenn es um Dateinamen geht.

Die Dateinamensregel erlaubt bekanntlich bis auf den / jedes Zeichen im Dateinamen. Zulässig sind also alle der folgenden Operationen:

```
cat > "*.*"  
cat > \"\""  
cat > "\\\\""  
cat > "Dies ist ein sonderbarer \"Dateiname\""
```

Zunächst soll klar gestellt werden, welche Dateien in den vier genannten Beispielen erzeugt werden.

Im ersten Fall erfolgt die Maskierung mit "...", der sich ergebende Name ist also **. Das Löschen dieser Datei ist natürlich mit gewissen Risiken verbunden. Unterlässt man beim Löschen die Maskierung

```
rm *.*
```

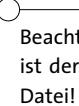
werden natürlich alle Dateien gelöscht, deren Namen einen Punkt enthalten! Ohne Maskierung erfolgt in diesem Fall Dateinamensexpression.

Im zweiten Beispiel erfolgt zunächst eine Maskierung des ersten Anführungszeichens durch \. Das folgende Zeichen dient erneut der Maskierung. Der resultierende Dateiname ist also ..

Das dritte Beispiel ist eindeutig. Die Maskierung mit "... " betrifft die gesamte Zeichenfolge, die neue Datei wird \\ heißen.

Im letzten Fall wird die Besonderheit der gewöhnlichen Anführungszeichen („...“) gegenüber den Apostrophs ("...") deutlich. Zunächst liegt ein einfacher Text mit Leerzeichen vor. Diese werden maskiert, damit die Shell den gesamten folgenden Text nicht interpretiert und ihn so dem Befehl cat als Dateinamen übergibt.

Innerhalb der Zeichenfolge aber taucht ein \ auf. Im Gegensatz zu der Maskierung mit "... " erkennt die Shell dieses Zeichen als Metazeichen und wird das darauf folgende Zeichen nicht interpretieren. So kann man Anführungszeichen in längere Dateinamen einschleusen.



Beachten Sie: Das ist der Name einer Datei!



Benutzerumgebung



Das Ergebnis ist im Übrigen der wirklich sonderbare Dateiname:

Dies ist ein sonderbarer „Dateiname“

Dass das Dollarzeichen (\$) innerhalb der gewöhnlichen Anführungszeichen von der bash als Sonderzeichen erkannt wird, erschließt sich erst, wenn man mit Shell-Variablen Erfahrung hat.

Lassen Sie uns am Ende dieses Abschnitts die Mahnung aussprechen, nicht zu sehr mit der Maskierung der Metazeichen zu experimentieren. Allzu leicht entstehen Dateien mit Namen, die ein einfaches Löschen verhindern. Erst der korrekte Gebrauch der Metazeichen wird in diesen Fällen zum Erfolg führen.

Die Befehlseingabe

Bereits im Kapitel 3 *Linux-Dateisystem* wurde die allgemeine Syntax eines Befehls erläutert. Diese Ausführungen sollen nun um einige Bemerkungen ergänzt werden.

`befehl [-option] [argliste] [umliste]`

Zu Beginn der Zeile steht also der Befehl selbst. In diesem Zusammenhang sei noch einmal darauf hingewiesen, dass, wenn dem Befehl weitere Elemente folgen sollen, diese unbedingt mit Leerzeichen, Tastopp oder Zeilenumbruch vom Befehl zu trennen sind. Die bash liest die Befehlszeile Wort für Wort!

Das zweite Element der Zeile sind die Optionen, eingeleitet durch das Minuszeichen. Mehrere Optionen können zusammengefasst werden. Dann folgt die Liste der Argumente (*argliste*). Das können Pfadnamen sein (z.B. bei Datei- und Verzeichnisbefehlen), Texte (*echo*), Benutzernamen (*chown*) oder auch einfach Zahlenangaben (*umask*, *chmod*), über die bestimmte Einstellungen getroffen werden sollen.

Optionen und Argumente gehören unmittelbar zum Befehl und werden diesem vor der Ausführung übergeben. Zu beachten ist dabei, dass eventuell in den Argumenten vorhandene Metazeichen von der bash zuvor aufgelöst werden.

Abgeschlossen werden kann solch eine einfache Befehlszeile durch eine Reihe von Umleitungen, deren Reihenfolge im Übrigen beliebig ist. Prinzipiell müssen die Umleitungen nicht am Ende der Befehlszeile erfolgen. Aus Gründen der Übersichtlichkeit ist aber davon abzuraten, Umleitungen an beliebiger Stelle der Befehlszeile zu platzieren.

Wer den vorletzten Abschnitt genau in Erinnerung hat, wird bemerkt haben, dass die dargestellte Syntax nicht ganz korrekt ist. Alle Elemente außer dem Befehl durch die eckigen Klammern wurden als optional bezeichnet. Aber auch dieser kann, wie wir gesehen haben, entfallen.



Arbeiten mit der bash



Es lassen sich nun, aufbauend auf dieser einfachen Form, Befehle auf vielfältige Weise miteinander verknüpfen, wobei dies durch

`befehl1 op befehl2`

geschieht. Dabei ist *op* ein entsprechender Verknüpfungsoperator. Die Syntax von *befehl1* und *befehl2* entspricht dem oben Beschriebenen.

Beispiel für eine Verknüpfung ist natürlich die oben beschriebene Pipeline. Solche Verknüpfungen führen häufig zu langen Eingabezeilen. Um den Arbeitsaufwand möglichst gering zu halten, werden deshalb zunächst die Möglichkeiten der Befehlwiederholung und der Bearbeitung einer Befehlszeile besprochen.

Der Befehlszeileneditor

Ein leidiges Problem der Bourne-Shell war die unzureichende Unterstützung des Anwenders bei der Befehlseingabe. Hatte man sich (speziell in einer längeren Befehlszeile) vertippt, so hatte man drei gangbare Möglichkeiten:

- Löschen der Zeile bis zum Fehler mit (Backspace, nicht Cursor!)
- Abbruch der Eingabe mit + und Neueingabe des Befehls
- Bestätigung der Befehlszeile mit , Warten auf die Fehlermeldung und Neueingabe des Befehls

Im letzten Fall musste natürlich darauf geachtet werden, dass die Situation eindeutig war und nicht aufgrund des Fehlers etwa eine falsche Datei gelöscht würde.

Die C-Shell der University of California in Berkeley verfügte als erste UNIX-Shell über einen ausgeklügelten History-Mechanismus, der auch in die bash Eingang gefunden hat.

Die bash pflegt heute eine Datei namens *.bash_history* im Heimatverzeichnis des Benutzers. In dieser werden alle eingegebenen Befehle protokolliert und für den erneuten Gebrauch bereitgehalten.

Eine traditionelle, d.h. von der C-Shell übernommene Nutzung dieses „History-Files“ erfolgt über eine Reihe von *!*-Befehlen. Ist man am Prompt und gibt einen dieser durch das Ausrufezeichen eingeleiteten *history*-Befehle ein, erfolgt eine Bearbeitung der Datei *.bash_history*.

Einige wichtige dieser Befehle sind in *Tabelle 4.7* zusammengefasst.



Benutzerumgebung

!-Befehle des *history*-Mechanismus

Befehl	Wirkung
!!	führt den letzten Befehl erneut aus
!nummer	führt den Befehl erneut aus, der im „History-File“ an entsprechender Stelle steht
!zeichenfolge	führt den letzten der Befehle erneut aus, die mit <i>zeichenfolge</i> beginnen
!?zeichenfolge	Führt den letzten der Befehle erneut aus, die <i>zeichenfolge</i> enthalten
^text1^text2	führt den letzten Befehl erneut aus, ersetzt aber den dort enthaltenen <i>text1</i> durch <i>text2</i>

Tabelle 4.7

Weiß man die Nummer des gewünschten Befehls nicht mehr, kann man sich mit

`history nummer`

die letzten nummer Befehlszeilen anzeigen lassen.

Ebenso interessant wie die kurz vorgestellten Wiederholungsbefehle sind die für die bash entwickelten Editiermöglichkeiten der Befehlszeile.

Weiter oben wurden schon die Cursortasten erwähnt. Mit \uparrow und \downarrow kann man die Liste der zuletzt benutzten Befehle einsehen, mit $\left[\begin{matrix} \text{Z} \\ \text{C} \end{matrix}\right]$ und $\left[\begin{matrix} \text{Z} \\ \text{V} \end{matrix}\right]$ kann man in der gefundenen Befehlszeile zeichenweise hin- und herwandern.

Einige grundlegende Tastenkombinationen zur Bearbeitung der Befehlszeile zeigt Tabelle 4.8.

Bewegen in der `.bash_history` und in der aktuell gewählten Befehlszeile

Einige Tastenkombinationen des Befehlszeileneditors der bash

Kombination	Wirkung
Bewegen in der <code>.bash_history</code> und in der aktuell gewählten Befehlszeile	
$\text{Strg}+\text{P}$	einen Befehl zurückblättern (wie \uparrow)
$\text{Strg}+\text{N}$	einen Befehl vorblättern (wie \downarrow)
$\text{Alt}+\text{K}$	zum ersten Befehl



Arbeiten mit der bash



Einige Tastenkombinationen des Befehlszeileneditors der bash (Forts.)

Kombination

Wirkung

[Alt]+[↑]+[<]	zum letzten Befehl
[Strg]+[R]text	rückwärtssuche nach Befehlen, die text enthalten
[Strg]+[S]text	vorwärtssuche nach Befehlen, die text enthalten
[Strg]+[A]	zum Anfang der aktuellen Befehlszeile
[Strg]+[E]	zum Ende der aktuellen Befehlszeile
[Strg]+[F]	Zeichen nach rechts (wie [→])
[Strg]+[B]	Zeichen nach links (wie [←])
[Alt]+[F]	Wort nach rechts
[Alt]+[B]	Wort nach links
Korrektur, Löschen und Einfügen	
[Strg]+[D]	Zeichen am Cursor löschen
[←]	Zeichen vor Cursor löschen
[Strg]+[T]	Zeichen am Cursor vertauschen
[Alt]+[T]	Wörter vertauschen
[Alt]+[U]	Wort in Großbuchstaben setzen
[Alt]+[L]	Wort in Kleinbuchstaben setzen
[Alt]+[C]	Anfangsbuchstabe des Wortes groß setzen
[Alt]+/[Versuch, das aktuelle Wort als Dateiname zu ergänzen
[Alt]+~	Versuch, das aktuelle Wort als Benutzername zu ergänzen
[Alt]+[.]	Einfügen des letzten Elements der letzten Befehlszeile
[Strg]+[K]	Löschen bis Zeilenende (gelöschter Text in Löschpuffer)
[Strg]+[U]	Löschen bis Zeilenanfang (gelöschter Text in Löschpuffer)
[Alt]+[D]	Löschen bis Wortende (gelöschter Text in Löschpuffer)
[Strg]+[Y]	Einfügen des Inhalts des Löschpuffers

Tabelle 4.8



Benutzerumgebung

Versuchen Sie doch einmal, die Taste **[Alt]** in Verbindung mit der Tilde ~ zu drücken

Diese Tastenkombinationen (und noch einige mehr, der interessierte Leser sei wieder einmal auf die *man pages* der bash verwiesen) lassen sich mit dem Befehl *bind* anpassen. Dies kann schon deshalb nützlich sein, weil eine wichtige vordefinierte Kombination nicht benutzbar ist – schließlich sind die Tastaturlayouts nicht immer geeignet, bestimmte Sondertasten in Verbindung miteinander zu benutzen.

Die Syntax des Befehls lautet:

`bind -option`

bzw.

`bind tastenfolge: befehl`

Die *Tabelle 4.9* zeigt einige Standardkombinationen in der Schreibweise, wie *bind* sie erwartet. Dabei steht **\C** für die **[Strg]-Taste** und **\e** für **[Alt]**. Die Namen der verfügbaren Befehle finden Sie insgesamt in den *man pages* der bash, sie lassen sich aber auch mittels

`bind -l`

anzeigen.

Zur Verdeutlichung sind in der ersten Spalte die Tastenkombinationen in der Form angegeben, wie Sie diese aus *Tabelle 4.8* kennen. Die Tabelle selbst ist unvollständig, sie soll nur einen Einblick in die Syntax des *bind*-Befehls geben.

Tastaturbelegung des Befehlszeileneditors mit *bind*

Kombination	Schreibweise für den <i>bind</i> -Befehl
[Alt] + [Left]	<code>,,\le<": beginning-of-history</code>
[Strg] + [A]	<code>,,\lC-a": beginning-of-line</code>
[Strg] + [E]	<code>,,\lC-e": end-of-line</code>
[Strg] + [F]	<code>,,\lC-f": forward-char</code>
[Strg] + [B]	<code>,,\lC-b": backward-char</code>
[Alt] + [F]	<code>,,\lef": forward-word</code>
[Alt] + [B]	<code>,,\leb": backward-word</code>

Tabelle 4.9

Soll etwa die Tastenkombination für das Löschen eines Wortes auf **[Strg]+[Z]** festgelegt werden, geschieht dies mit:



Arbeiten mit der bash



```
bind "" "\C-z": kill-word"
```

Die Maskierung von tastenkombination und befehl ist nötig, damit die bash nicht die für den Befehl *bind* nötigen „...“ aus der Befehlszeile entfernt.

Hier nun die Optionen des Befehls.

Optionen des Befehls *bind*

Option	Bedeutung
-d	Liste der aktuellen Belegungen. Wird sie in eine Datei umgeleitet, kann man sie editieren und mithilfe der Option -f die so veränderte Belegung aktivieren. Im Gegensatz zu -v wird eine Liste erzeugt, die ausschließlich Einträge hat, welche der Form aus <i>Tabelle 4.9</i> entsprechen.
-f dateiname	Die Datei <i>dateiname</i> wird als neue Tastaturbelegungstabelle genommen. Dies setzt voraus, dass die Einträge in der Datei im Format der <i>Tabelle 4.9</i> angegeben sind.
-l	Anzeige der verfügbaren Befehle
-m tastaturtabelle	Auswahl einer vordefinierten Tastaturtabelle. Mögliche Tabellen sind emacs, emacs-standard, emacs-meta, emacs-ctlx, vi, vi-move, vi-command und vi-insert. Mithilfe dieser Tabellen erfolgt eine Anpassung an die bekannten Editoren emacs oder vi.
-q befehlsname	gibt die aktuelle Belegung des genannten befehls aus
-v	Liste der verfügbaren Befehle und ihrer aktuellen Belegungen

Tabelle 4.10



Die Änderung der Tastenbelegung lässt sich durch Benutzung der Datei *.inputrc* im Heimatverzeichnis automatisieren. Sie könnte etwa durch

```
bind -d > .inputrc
```

mit dem aktuellen Status als Inhalt angelegt werden.



Benutzerumgebung



Einfache Befehlsverkettungen

Nachdem Sie nun alle Tastenkombinationen des Befehlszeileneditors auswendig gelernt haben, sollten Sie ohne Probleme längere Befehlszeile editieren können. Damit sind Sie bereit, einige weitere Formen der Befehlsverkettung kennen zu lernen.

Mit der Pipeline kennen Sie bereits eine der komplexeren Varianten. Hier werden zwei Befehle nicht einfach nacheinander ausgeführt, es gibt auch eine Interaktion zwischen ihnen. Die einfachste Form der Verknüpfung wird durch das Semikolon (;) erreicht.

Eine der vielen
Parallelen zur
Programmier-
sprache C

befehl1 ; befehl2 [; ...]

bewirkt, dass beide Befehle nacheinander abgearbeitet werden. Mit *befehl?* ist ein Befehl in der Syntax des vorletzten Abschnitts gemeint. Als Beispiel betrachten Sie die folgende Befehlszeile:

```
ls -l /bin ; ls -l /usr/bin > commands
```

Zunächst wird der Inhalt des Verzeichnisses */bin* am Bildschirm angezeigt, danach wird der Inhalt des Verzeichnisses */usr/bin* in die Datei *commands* umgeleitet.

Die Umleitung bezieht sich also nur auf *ls -l /usr/bin!* Will man beide Ausgaben gemeinsam umleiten, kann man zusätzlich die geschweiften Klammern als Zusammenfassung der Befehle verwenden.

```
{ ls -l /bin ; ls -l /usr/bin } > commands
```

In dieser Form bezieht sich die Umleitung auf beide Befehle gemeinsam. Sie werden ausgeführt, ihre Daten werden zusammen in die Datei *commands* geleitet.

Bei der Benutzung der Klammern ist auf den Gebrauch von Leerzeichen zu achten. Unmittelbar nach der öffnenden und vor der schließenden Klammer *muss* ein Leerzeichen stehen.

Statt der geschweiften können auch die runden Klammern benutzt werden.

```
( ls -l /bin ; ls -l /usr/bin ) > commands
```

Führt zum gleichen Ergebnis wie die obige Eingabe. In einige Fällen jedoch ist es nützlich, den Unterschied zwischen beiden Möglichkeiten zu kennen. Bei Gebrauch der runden Klammern wird eine Sub-Shell gestartet – ein Clone der aktuellen Shell –, unter deren Zuständigkeit die Befehle abgearbeitet werden.

Das hat zur Konsequenz, dass Änderungen an der Systemumgebung nur in der Sub-Shell Einfluss haben. Nach Abarbeitung der Befehle wird die Sub-Shell beendet, Änderungen der Umgebung sind in der aktuellen Shell nicht wirksam geworden.



Arbeiten mit der bash



Die durch geschweifte Klammern gruppierten Befehle werden von der aktuellen Shell ausgeführt. Ändert einer der Befehle die Systemumgebung, bleibt diese Änderung nach Abarbeitung der Befehlszeile wirksam.

Ein einfaches Beispiel soll diesen Unterschied erklären helfen:

```
$ pwd(←)
/home/rattle
$ { cd .. ; ls }(←)
cshuser/      rattle/      rumpel/      stern/
$ pwd(←)
/home
$ cd(←)
$ pwd(←)
/home/rattle
$ ( cd .. ; ls )(←)
cshuser/      rattle/      rumpel/      stern/
$ pwd(←)
/home/rattle
```

Zunächst wird überprüft, welches das aktuelle Verzeichnis ist. Danach werden die beiden Befehle `cd ..` und `ls` in einer gewöhnlichen Verknüpfung ausgeführt. Nach Ausführung der kleinen Befehlssequenz (die Klammern wären nicht nötig gewesen, Probleme wegen Datenleitung oder Pipelining sind in dieser einfachen Konstruktion nicht gegeben) ist `/home` das aktuelle Verzeichnis. Sie wechseln zum Heimatverzeichnis und prüfen noch einmal, welches Verzeichnis das aktuelle ist. Schließlich wird die gleiche Befehlssequenz unter Verwendung der runden Klammern gestartet.

Prüfen Sie nun das aktuelle Verzeichnis, so stellen Sie fest, dass es `/home/rattle` ist, obwohl in der Sequenz ein Verzeichniswechsel nach `/home` erfolgte. Das aktuelle Verzeichnis ist neben den Systemvariablen eines der Teile der Systemumgebung, die oben angesprochen wurden. Die Änderung des aktuellen Verzeichnisses in einer Sub-Shell hat keine Auswirkungen auf die Arbeit in der übergeordneten Shell.

Bedingte Befehlsausführung

Eine besondere Form der Verknüpfung ist die bedingte Befehlsausführung. Mit ihr kann man schon am Prompt einfache Kontrollstrukturen realisieren, wie sie sonst nur in der Programmierung üblich sind.

Die bedingte Befehlsausführung basiert auf der Tatsache, dass alle Linux-Programme nach ihrer Beendigung einen Wert an das aufrufende Programm, in diesem Fall die Shell, zurückliefern, vorausgesetzt, die Programmierer halten sich an die Konventionen. Dieser signalisiert, ob das Programm erfolgreich beendet wurde oder mit einem Fehler abschloss. Gemäß Konvention bedeutet der Rückgabewert 0, dass das Programm fehlerfrei endete. Andere Rückgabewerte signalisieren Fehler (eventuell kann über den genauen Wert die Art des Fehlers gekennzeichnet werden).



Benutzerumgebung



Die Shell ist also in der Lage, zu prüfen, ob ein Befehl erfolgreich – oder besser ohne Fehler – beendet wurde oder nicht. Dies kann in besonderen Verknüpfungen benutzt werden, um den zweiten Befehl der Sequenz in Abhängigkeit vom Ergebnis des ersten auszuführen.

```
befehl1 && befehl2
```

verknüpft nun die Befehle so, dass *befehl2* nur dann ausgeführt wird, wenn *befehl1* erfolgreich war.

```
ls -l testdatei && cat testdatei
```

zeigt ausführliche Informationen über die Datei namens *testdatei*. Ist sie tatsächlich vorhanden, wird zusätzlich ihr Inhalt angezeigt.

Das **&&** – Leser mit Erfahrung in C-Programmierung werden dies wissen – bedeutet eine logische UND-Verknüpfung, d.h., das Gesamtergebnis der Verknüpfung kann nur WAHR werden, wenn beide Teile WAHR sind. Oder umgekehrt: Ist bereits der erste Teil FALSCH, so kann das Gesamtergebnis nicht mehr WAHR werden.

Von den meisten Programmiersprachen werden solche logischen UND-Verknüpfungen so behandelt, dass die zweite Prüfung gar nicht erst durchgeführt wird, wenn die erste Prüfung mit einem negativen Ergebnis endet.

In diesem Zusammenhang ist die Wirkung von **&&** doch logisch – oder?

Die zweite Möglichkeit ist die ODER-Verknüpfung. Der zweite Befehl wird nur ausgeführt, wenn der erste *nicht* erfolgreich war. War er erfolgreich, so ist nach der Logik der Programmierer bereits alles getan.

Die Syntax der ODER-Verknüpfung ist (auch hier ein Aha-Erlebnis für C-erfahrene Leser):

```
befehl1 || befehl2
```

Beispielsweise versucht die folgende Befehlszeile zunächst, mittels *write* Kontakt mit dem Superuser herzustellen. Falls das nicht gelingt, aber auch nur dann, wird ihm mittels *mail* Post gesandt.

```
write root || mail root
```

Befehlssubstitution

Eine recht anspruchsvolle Variante der Befehlsverknüpfung ist die „Substitution“, die einiges mit dem Pipelining gemeinsam hat. So wird der eine Befehl Daten erzeugen, die der andere zur Weiterverarbeitung benutzt. Der zugrunde liegende Mechanismus ist allerdings etwas anders gelagert und erlaubt einen sehr variablen Einsatz der Substitution.



Arbeiten mit der bash



Die allgemeine Syntax lautet:

`befehl1 | `befehl2 | ``

oder alternativ

`befehl1 $ (befehl2)`

Wichtig ist, dass Sie die Bedeutung von „,“ und ` unterscheiden. Die ersten beiden Zeichen dienen der Maskierung von Metazeichen, das` (accent grave) der Substitution. Allerdings kann man der Substitution einen maskierenden Charakter nicht absprechen. Alle Zeichen die sich zwischen den ` ... ` befinden, werden von der bash als eigenständige Befehlszeile aufgefasst, die sie zuerst interpretiert und deren Befehl sie gegebenenfalls zur Ausführung bringt.

Sehr wahrscheinlich wird es auch in dieser Ausgabe dem Verlag nicht gelingen, accent grave (von links nach rechts abfallendes Hochkomma) und (normales) Hochkomma im Druck unterschiedlich darzustellen. Wir bitten den Leser dafür um Entschuldigung und bemühen uns von nun an, die alternative Schreibweise zu benutzen.



Hinweis

Mögliche Bildschirmausgaben von `befehl2` werden jedoch anstelle von `befehl2` in die ursprüngliche Befehlszeile integriert. In diesem Sinne erfolgt die Substitution eines Befehls.

`echo $(who | wc -l)` Personen sind eingeloggt

wird von der bash in folgender Reihenfolge abgearbeitet:

- Bearbeiten der „Sub“-Befehlszeile `who | wc -l` und damit Erzeugen einer Pipeline,
- Ausführen des Befehls `who`,
- Weiterleiten der Daten an `wc -l`,
- Ersetzen von `who | wc -l` in der Befehlszeile durch die Ergebnisse des Befehls `wc -l`,
- Ausführen des `echo`-Befehls.

Das Ergebnis wird die Ausgabe der Anzahl der eingeloggten Benutzer in der folgenden Form sein:

`10 Personen sind eingeloggt`

Das Beispiel zeigt, dass die Substitution nicht durch das Pipelining ersetzbar (oh! – substituierbar) ist. Es erfolgt eine Ausgabe von Daten, die einerseits aus der Befehlszeile stammen, andererseits Ergebnisse des substituierten Befehls sind.



Benutzerumgebung



Zu beachten ist ferner, dass die originale Befehlszeile von der Shell noch einmal interpretiert werden muss, da der substituierte Befehl Metazeichen als Ergebnis liefern kann.

Hintergrundverarbeitung

Hintergrundprozesse stellen eine Möglichkeit dar, mehrere Programme parallel laufen zu lassen. Gibt man am Terminal einen Befehl ein, so wird zwar sofort die Tastatur zur Eingabe des nächsten freigegeben, sofern der erste nicht interaktiv auf eine Eingabe wartet wie etwa `rm -i`, in jedem Fall muss aber der zweite Befehl auf seine Ausführung warten, bis der erste beendet ist.

Bei Suchroutinen in umfangreichen Dateisystemen oder großen Dateien kann dies natürlich zu erheblichen Wartezeiten führen. Mit den Mitteln der Hintergrundverarbeitung lässt sich jeder Prozess so starten oder auch nachträglich in einen Zustand versetzen, dass er zwar weiter seine Aufgabe erfüllt, aber gleichzeitig das Ausführen weiterer Prozesse erlaubt.

Einen Prozess, der es erlaubt, am Terminal, von dem aus er gestartet wurde, weitere Prozesse zu starten, nennt man „Hintergrundprozess“. Der Start eines Hintergrundprozesses erfolgt, indem man der eigentlichen Befehlszeile ein & anhängt. Also:

```
befehl &
```

Zum Beispiel startet

```
find / -name mkfs -print &
```

die Suche nach der Datei `mkfs` im gesamten Dateisystem als Hintergrundprozess.

Ungeschickt an dieser Eingabe ist allerdings, dass `find` Zugang zum Bildschirm hat. Startet man nun einen zweiten Prozess, welcher ebenfalls Bildschirmausgaben bewirkt, so vermischen sich diese mit denen des `find`-Befehls. Will man diese Nebeneffekte vermeiden, sollte man Ausgaben und gegebenenfalls auch Eingaben umleiten, d.h., der Aufruf sollte in dieser Form erfolgen:

```
find / -name mkfs -print > datei 2>&1 &
```

In jedem Falle hat die bash mit einer Ausgabe folgender Art reagiert:

```
[1]+ 456
```

Die Nummer in den eckigen Klammern ist eine laufende Nummer, über die mit den Befehlen `bg` und `fg` der Status des Prozesse verändert werden kann. Das + zeigt, dass es der aktuelle Prozess ist. 456 schließlich ist die eigentliche Prozessnummer, über die dem Prozess verschiedene Signale gesandt werden können, darunter das zur Beendigung eines Prozesses.

```
kill -9 456
```



Arbeiten mit der bash



würde die eingeleitete Suche abrupt beenden.

Laufen an Ihrem Terminal mehrere Prozesse, die man auch gerne „Jobs“ nennt, dann können Sie sich einen Überblick mit dem Befehl *jobs* verschaffen.

```
$ find / -name mkfs -print > mdatei 2>&1 &  
[2]+ 456  
$ jobs  
[1]- Running      find / -user rattle -print > rdatei  
2>&1 &  
[2]+ Running      find / -name mkfs -print > mdatei  
2>&1 &  
$
```

Auch hier wird ein Suchvorgang wohl vorbereitet durch Umleitung gestartet. Eine Überprüfung der *Job-Liste* ergibt, dass ein zweiter Suchlauf, hier nach Dateien eines bestimmten Benutzers, läuft.

Mit dem Befehl *fg* (für foreground = Vordergrund) lässt sich ein Hintergrundprozess wieder in den Vordergrund holen. So würde durch

```
fg 1
```

der Suchlauf nach den Dateien des Benutzers *rattle* zum Vordergrundprozess und somit der direkten Beeinflussung durch die Tastatur zugänglich. Ein laufender Vordergrundprozess lässt sich z.B. durch Drücken der Tastenkombination **Strg+C** abbrechen.

Mit der Tastenkombination **Strg+Z** wird ein im Vordergrund laufender Prozess angehalten, was zu folgender Meldung führt:

```
"[1]+ Stopped find / -user rattle -print > rdatei 2>&1"
```

Ein solcher angehaltener Prozess lässt sich schließlich mit

```
bg jobnummer
```

in den Hintergrund stellen.

Eine Besonderheit der Befehlausführung im Hintergrund stellt der Befehl *nohup* dar (no hang up). Normalerweise wird ein Prozess, den man als Kind-Prozess der Shell am Prompt startet, spätestens dadurch beendet, dass man ausloggt.

Einzelheiten zur Prozesssteuerung entnehmen Sie bitte dem *Kapitel 7 Systemverwaltung*

Bei zeitaufwändigen Aufgaben, wie etwa der Reorganisation von Datenbanken, ist es mitunter sinnvoll, diese abends zu beginnen, um am nächsten Tag die Ergebnisse vorliegen zu haben.



Hinweis



Benutzerumgebung



Weiter ist es möglich, dass der Zugriff auf das System über einen Dienst der Telekom erfolgt. Längere Arbeit im System bedeutet dann natürlich einen entsprechenden Kostenfaktor.

Der Befehl *nohup* erlaubt nun das Starten eines Hintergrundprozesses, der nicht durch das Ausloggen beendet wird. Die Syntax lautet

```
nohup befehl &
```

Dabei kann *befehl* eine beliebige Befehlszeile sein, die lediglich durch Anhängen des & als Hintergrundprozess kenntlich gemacht sein muss.

```
$ nohup find / -name mkfs -print > mdatei 2>&1 &  
$ logout  
composter login:
```

ist nun problemlos möglich. Der Suchvorgang läuft weiter, auch wenn man ausgeloggt ist. Es muss natürlich sichergestellt sein, dass das Linux-Gerät nicht vor Beendigung des Auftrags heruntergefahren wird.

Zugegeben: Die Ausführungen des letzten Abschnitts waren aus Anfängersicht eher theoretisch. Mit zunehmender Erfahrung werden Sie aber auch solche Kenntnisse benötigen, um sich die Arbeit mit dem System vereinfachen zu können.

Alias-Namen

Ein von Beginn an nützliches Feature der bash ist das von der C-Shell übernommene *Aliasing*. Dieses Verfahren erlaubt es, Befehlen ohne Aufwand neue Namen zu geben. *Alias-Namen* werden von der Shell während der Sitzung im Speicher verwaltet und gehen daher mit dem Abmelden verloren. Ein dauerhaftes *Aliasing* erreicht man durch Einbinden der entsprechenden Befehle in eine der Startdateien, z.B. in *.profile* im Heimatverzeichnis.

Das Setzen eines *Alias-Namens* erfolgt mit:

```
alias name=wert
```

Z.B. wird durch

```
alias u=umask
```

ein neuer Name für den Befehl *umask* vergeben. Die Eingabe von

```
u
```

wird ab sofort (bis zum Ausloggen) die Anzeige der Rechtemaske für neue Dateien bewirken. Natürlich lassen sich auch Optionen oder Argumente der Befehlszeile in den *Alias-Namen* einbeziehen. Wegen der Bedeutung der Leerzeichen für die bash muss bei der Setzung eine Maskierung erfolgen. Damit lässt sich auch ein so komplexer Befehl wie *mount* einfacher behandeln.



Arbeiten mit der bash



```
alias a:="mount /dev/fd0h1440 /mnt ; cd /mnt"
```

ist eine Abkürzung für das Mounten einer Diskette im ersten Disketten-Laufwerk (A:) und anschließendes Wechseln zum Verzeichnis `/mnt`. Da das Mounten Aufgabe des Superusers ist, kann dieser neue Befehl auch nur von ihm benutzt werden.

Wie Sie sehen, sind einfache Befehlssequenzen oft recht nützliche Hilfsmittel bei der täglichen Arbeit mit Linux.

Will man sich die Liste der vorhandenen *Alias-Namen* anzeigen lassen, gibt man `alias` ohne Argument ein.

```
$ alias↔
alias d="dir"
alias dir="/bin/ls $LS_OPTIONS --format=vertical"
alias l="ls -l"
alias ls="/bin/ls $LS_OPTIONS "
alias v="vdir"
alias vdir="/bin/ls $LS_OPTIONS --format=long"
```

Hier taucht der wohl bekannteste *Alias-Name* auf, `dir`. Er ist eine Reminiszenz für die DOS-Benutzer, die auf Linux umsteigen.

Hier wird ein nicht zu unterschätzender Aspekt des *Aliasing* deutlich. Mit ihm kann man auf bequeme Weise seine Benutzerumgebung so anpassen, dass einem die – möglicherweise – neue Umgebung vertraut erscheint.

Dies gilt auch für Umsteiger aus anderen Betriebssystemen. So ist etwa der Befehl `/` in der im Listing zu erkennenden Form in SCO-UNIX-Varianten implementiert. Es wurde schon einmal erwähnt, dass einer der Autoren mit dieser UNIX-Spielart „groß“ geworden ist.



Der Doppelpunkt gehört zum Alias-Namen!



Anmerkung: Eine besonders beliebte Variante ist auch `//` für den Befehl `ls -l`

Funktionen

Den Alias-Namen verwandt sind die Funktionen. Auch sie sind Ersatz für einen Befehl oder eine Befehlsfolge. Sie müssen – wie Alias-Namen auch – definiert werden und verlieren ihre Gültigkeit, wenn man sich ausloggt. Sie haben aber gegenüber den Alias-Namen den Vorteil, dass sie Positionsparameter verarbeiten können.

Zu Positionsparametern siehe *Kapitel 15 Programmierung*.



Hinweis

Die Funktionsdefinition basiert auf der Gruppierung von Befehlen mit geschweiften Klammern. Man vergibt einer solchen Befehlsgruppe einen Namen, danach lässt sie sich wie ein Befehl einsetzen.

Die Syntax für die Funktionsdefinition ist:

```
name() { befehl1; [befehl2; ... ] }
```



Benutzerumgebung



Im folgenden Beispiel wird eine Funktion definiert, die Entsprechendes leistet wie der Alias-Befehl *a*: aus dem letzten Abschnitt.

```
# wechsel(){ mount /dev/fd0h1440 /mnt ; cd /mnt; }  
# rwechsel(){ cd ; umount /dev/fd0h1440 ; }  
# wechsel  
# pwd  
/mnt  
# rwechsel  
# pwd  
/root  
#
```

Wie oben erwähnt, braucht man Superuser-Status, um das Mounten eines Dateisystems ausführen zu können. Daher sollte man für dieses Beispiel als Superuser eingeloggt sein. Um Sie nicht unnötig mit der gemounteten Diskette im Regen stehen zu lassen, haben wir gleich eine zweite Funktion definiert, die einen Rückwechsel ins Heimatverzeichnis und das Abmelden der Diskette (*umount*) erledigt.

Die Setzung der Leerzeichen sollte in jedem Fall beachtet werden, dies wurde schon bei der Erörterung von { ... } besprochen. Die runden Klammern bei der Funktionsdefinition müssen gesetzt werden – auch wenn der Aufruf ohne sie erfolgt.

Shell-Variable

Variablen sind wie Alias-Namen temporär, d.h., sie werden von der bash im Speicher verwaltet und entziehen sich mit beim Ausloggen dem Zugriff des Benutzers. Werden sie nicht in einer der Startdateien – wie *.profile* – angelegt, müssen sie theoretisch nach jedem Einloggen neu definiert werden.

Es handelt sich bei ihnen, wie bei Variablen in den meisten Programmiersprachen auch, um symbolische Speicheradressen, über die also auf bestimmte Speicherinhalte zugegriffen werden kann.

Wie es der Name schon sagt, sind Variablen veränderbar, genauer: der Speicherinhalt, der über Variablen angesprochen wird. Über Variablen kann die Shell Programme, die in ihr gestartet werden, über bestimmte Zustände informieren, über sie kann gegebenenfalls ein Datenaustausch stattfinden, sie können Notfall-Informationen für Programme enthalten – kurz: sie sind nützlich im Zusammenspiel zwischen Shell und Programmen.

Eine Variable wird gesetzt durch

name=wert

So lässt sich durch

PROG=/usr/bin



Arbeiten mit der bash



eine Variable namens *PROG* definieren, die aktuell den Wert */usr/bin* annimmt. Der wesentliche Grund für die Benutzung von runden Klammern bei der Definition von Funktionen ist der, dass die bash durch sie Funktionen von Variablen unterscheiden kann.

Es ist Konvention, in der bash Variablennamen großzuschreiben. In der C-Shell werden sie üblicherweise kleingeschrieben.



Hinweis

Eine der wichtigsten Variablen ist *HOME*. Ihr aktueller Wert ist normalerweise der Pfad des Heimatverzeichnisses. Sie wird vom Befehl *cd* benutzt, wenn er ohne Argument eingegeben wird. Die Aussage, dass *cd* ohne Argument ins Heimatverzeichnis wechselt, beruht auf der Tatsache, dass die *HOME*-Variable den entsprechenden Wert hat.

Natürlich lässt sich der Wert von *HOME* nach dem oben beschriebenen Muster ändern. Als Konsequenz ändert sich das Verhalten von *cd*, wie folgendes Beispiel zeigt.

```
$ HOME=/  
$ cd  
$ pwd  
/  
$
```

cd ohne Argument eingegeben, macht das Verzeichnis zum aktuellen, dessen Pfad in *HOME* abgelegt ist.

Angenommen, Sie wollen den Inhalt (Wert) einer Variablen abfragen, so bietet sich als erste Möglichkeit der Befehl *echo* an. Aber leider:

```
$ echo HOME  
HOME  
$
```

Woher soll *echo* auch wissen, dass es den Inhalt einer Variablen und nicht das Wort „HOME“ ausgeben soll?

Daher ist es beim Gebrauch von Variablen notwendig, sie als solche zu kennzeichnen. Dies geschieht durch Voranstellen des \$-Zeichens.

```
$ echo $HOME  
/  
$
```

Natürlich lässt sich die Variable von beliebigen anderen Programmen nutzen, da die Shell vor Programmausführung den Namen einer Variablen durch ihren Wert ersetzt. So kopiert

```
cp * $HOME/help
```



Benutzerumgebung



alle Dateien des aktuellen Verzeichnisses, sofern sie nicht mit . beginnen, in das Unterverzeichnis *help* des Heimatverzeichnisses.

Das \$-Zeichen wird bekanntlich bei der Maskierung mit „...“ von der bash als eines der wenigen Metazeichen erkannt. Dies erweist sich im Zusammenhang mit Variablen als besonders nützlich.

```
$ HOME=/home/rattle  
$ echo "$HOME/*  
/home/rattle/*  
$
```

Die Variable wird als solche erkannt, das Zeichen für die Dateinamensexpression jedoch nicht.

Variablen gelten nur in der Shell, in der sie definiert wurden. Insbesondere im Zusammenhang mit der Ausführung von Shell-Skripten, die meist durch den Start einer Sub-Shell ausgeführt werden, kann das zu Problemen führen. Will man allen Sub-Shells eine Variable verfügbar machen, muss man sie exportieren. Dies geschieht mit

```
export variable1[ variable2 ...]
```

Insbesondere die Variablen *HOME*, *PATH* und *PS1* sind Variablen von allgemeiner Bedeutung und insofern allgemein zugänglich, d.h. exportiert.

Beispiele für das Exportieren finden Sie im *Kapitel 16 Shell-Programmierung*.

Hinweis



Mit dem Befehl *set* kann man sich die aktuellen Werte von definierten Variablen anschauen. Die meisten dieser Variablen sind von Linux z.B. über die Startdateien vordefiniert. Die wichtigsten von ihnen entnehmen Sie der *Tabelle 4.11*.

Wichtige bereits definierte Shell-Variablen und ihre Bedeutung



Variable	Bedeutung
<i>IFS</i>	(Internal Field Separators) eine Liste der Zeichen, welche die bash zum Trennen von Wörtern in der Befehlszeile braucht. Da darunter normalerweise das Leerzeichen, [Space] -Taste und [Enter] sind, ist die Anzeige dieser Variablen eher verwirrend. Wegen ihrer Bedeutung sollte sie nicht geändert werden.
<i>PATH</i>	Liste von Verzeichnissen, welche die bash durchsucht, wenn der eingegebene Befehl extern ist
<i>HOME</i>	Heimatverzeichnis, benutzt u.a. vom Befehl <i>cd</i>
<i>CDPATH</i>	Verzeichnisliste für <i>cd</i> , wenn kein gültiger Pfad als Argument für <i>cd</i> angegeben wurde



Arbeiten mit der bash



Wichtige bereits definierte Shell-Variablen und ihre Bedeutung (Forts.)

Variable	Bedeutung
<i>MAIL</i>	Name der Datei, in der die Post abgelegt wird
<i>MAILCHECK</i>	Angabe der Zeit (in Sekunden), die zwischen zwei Überprüfungen der Mailbox auf eingegangene Post verstreichen soll
<i>PS1</i>	der Prompt
<i>PS2</i>	Der Hilfsprompt. Wird von der bash ausgegeben, wenn eine Eingabe als unvollständig angesehen wird (z.B. vergessenes zweites Anführungszeichen bei der Maskierung von Metazeichen).
<i>PS4</i>	Prompt für erweiterte Kommandos
<i>HISTSIZE</i>	Anzahl der Zeilen des History-Puffers
<i>HISTFILE</i>	Datei, in der die benutzten Befehle für die Befehlswiederholung gespeichert werden (normalerweise <i>\$HOME/.bash_history</i>)
<i>HISTFILESIZE</i>	Maximalzahl der Zeilen für die <i>history</i> -Datei
<i>HOSTNAME</i>	Name des Rechners (erscheint u.a. beim Login)
<i>HOSTTYPE</i>	Rechntyp, z.B. i486
<i>TZ</i>	Zeitzone (z.B. MEZ)
<i>TERM</i>	Terminaltyp (z.B. ANSI)
<i>LOGNAME</i>	Ihr Login-Name
<i>SHELL</i>	Name der aktuellen Shell
<i>BASH_VERSION</i>	Versionsnummer der bash
<i>PWD</i>	Pfad des aktuellen Verzeichnisses
<i>UID</i>	Benutzernummer
<i>EUID</i>	effektive Gruppennummer

Tabelle 4.11

Die letzten vier dieser Variablen werden direkt von der bash verwaltet.

Zwei der Shell-Variablen sollen hier wegen ihrer Bedeutung gesondert besprochen werden: *PATH* und *PS1*.



Benutzerumgebung



Angenommen, die Shell hat Ihre Befehlszeile so weit analysiert, dass alle Metazeichen von ihr entfernt worden sind. Dann wird sie das erste Wort der verbleibenden Zeile als Befehl ausführen. Das erste Wort ist für die bash, wie Sie jetzt genauer wissen, die Zeichengruppe, die durch den Zeilenanfang einerseits und das erste in der Zeile vorkommende Zeichen aus der IFS-Liste andererseits begrenzt ist.

Die Ausführung des Befehls ist allerdings noch eine recht komplexe Angelegenheit, die in ihren Grundzügen hier dargestellt werden soll.

Zuerst wird geprüft, ob das erste Wort ein Alias-Name ist. In diesem Fall erfolgt eine Ersetzung durch den entsprechenden Befehl.

Ist der resultierende Befehl ein interner Befehl, d.h. ein Unterprogramm der Shell (`cd` ist solch ein interner Befehl), so wird er unmittelbar ausgeführt.

Alle folgenden Prüfungen erfolgen nun nur noch, wenn alle vorhergehenden zu keinem positiven Ergebnis geführt haben.

Als Nächstes prüft die bash, ob das erste Wort oder der bisher gefundene Ersatz eine Funktion ist. Diese muss gegebenenfalls in ihre Bestandteile (die einzelnen Teile der Sequenz) zerlegt werden, welche dann jede für sich wie eine gewöhnliche Befehlszeile abgearbeitet werden müssen.

Zuletzt prüft die bash, ob in einem der Pfade, die in der Variablen `PATH`, getrennt durch `:` angegeben sind, eine Datei mit dem Namen vorhanden ist, der dem ersten Wort (oder seinem Ersatz) der Befehlszeile entspricht.

In diesem Fall beauftragt die Shell den Linux-Kernel, das entsprechende Programm auszuführen. Der Kernel seinerseits kann aber feststellen, dass die fragliche Datei kein ausführbares Programm ist. Dies meldet er der Shell.

Diese wird dann in einem (puh!) letzten Versuch die Datei öffnen, in der Erwartung, dass es ein Shell-Skript ist, um es als solches abzuarbeiten.

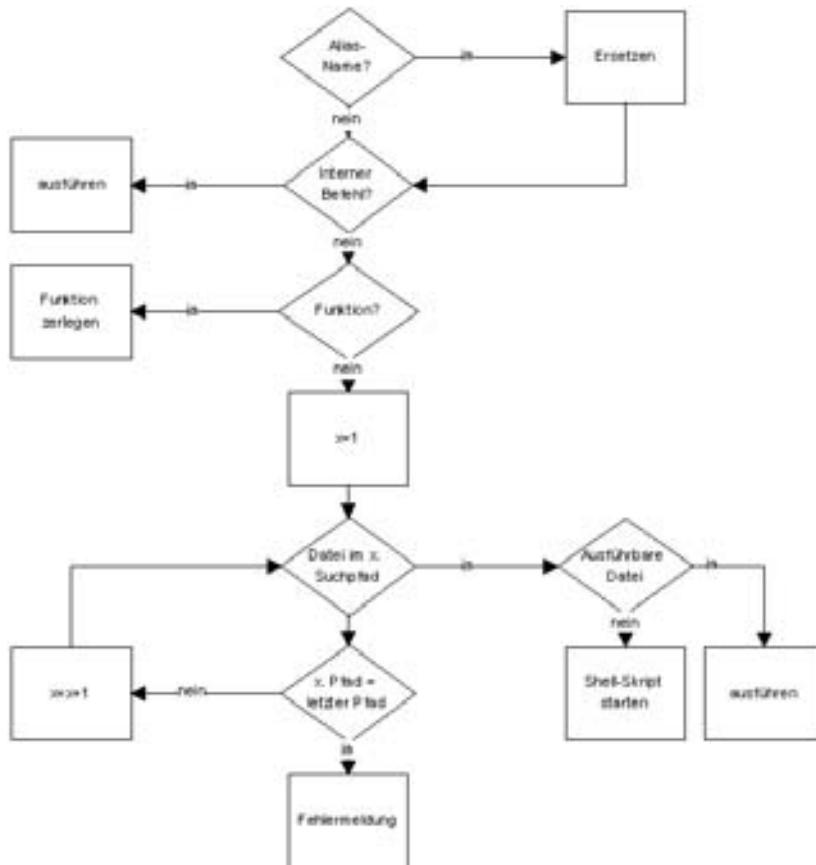
Sollten alle Bemühungen der bash scheitern, veranlasst sie das zu einer Fehlermeldung der Art:

```
bash: grr: command not found
```

Abbildung 4.4 zeigt den Gang der Prüfungen vor Ausführung eines Befehls durch die bash.



Arbeiten mit der bash



○ Abb. 44

Ausführen eines Befehls durch die bash

Die einzelnen Verzeichnisse des Suchpfades werden in der Variablen *PATH* durch : abgetrennt. Beachtenswert ist die Angabe des . im Pfad, wie Sie sich leicht durch

`echo $PATH`

überzeugen können. Gibt es keinen Eintrag . in der *PATH*-Variablen, wird das aktuelle Verzeichnis nicht nach möglichen Befehlsdateien durchsucht.

Der Punkt steht für das aktuelle Verzeichnis.

Die zweite interessante Variable ist *PS1*, der Prompt. Der Standardwert für diese Variable ist \$. Durch

`PS1=text`

kann er jede „beliebige“ Form annehmen. *text* sollte wegen möglicher Metazeichen maskiert werden. Innerhalb des Textes lassen sich eine Reihe von Sonderzeichen einfügen, die bei der Ausgabe des Prompts nicht direkt angezeigt, sondern zuvor interpretiert werden. Auf diese Weise lassen sich nützliche Informationen in den Prompt einbeziehen, wie etwa das aktuelle Verzeichnis.



Man beachte das Leerzeichen nach dem \$!



Benutzerumgebung

Eine Liste der Sonderzeichen finden Sie in *Tabelle 4.12*.

Sonderzeichen zur Gestaltung des Prompts

Zeichen	Bedeutung
\d	aktueller Systemdatum
\h	Name des Rechners (wie er auch beim Login erscheint)
\n	Zeilenumbruch
\s	Name der aktuellen Shell
\t	aktuelle Systemzeit
\u	Benutzername
\w	aktuelles Verzeichnis (mit kompletter Pfadangabe)
\W	der Name des aktuellen Verzeichnisses
\#	Nummer des aktuellen Befehls

Tabelle 4.12

- Ist das aktuelle Verzeichnis das Heimatverzeichnis, wird dies mit ~ angezeigt
- Soll der Prompt eventuell Metazeichen der Shell enthalten, so beachten Sie die richtige Maskierung.
- ```
PS1="\w> "
```
- ändert den Prompt dahingehend, dass der aktuelle Pfad gefolgt vom > und einem Leerzeichen angezeigt wird.

## Andere Shells

Die bash ist, wie schon mehrfach betont, die Standard-Shell für Linux. Sie ist die Weiterentwicklung der etwas antiquierten Bourne-Shell. Dies ist eine seit den Anfangsstagen von UNIX existierende Shell mit einer Reihe von Schwächen, die sich im Wesentlichen durch ihr Alter erklären lassen.

Durch die Weitergabe der Sourcecodes des Betriebssystems an die Universität in Berkeley wurden die dort tätigen Dozenten und Studenten ermuntert, eigene Wege bei der Weiterentwicklung des Betriebssystems UNIX zu beschreiten.

Das Ergebnis dieser Bemühungen war eines der erfolgreichsten UNIX-Systeme: BSD.



## Arbeiten mit der bash



Eine der Änderungen gegenüber den bisherigen UNIX-Varianten war eine neu entwickelte Benutzeroberfläche, die C-Shell. Sie hebt sich durch viele Neuerungen wie das *Aliasing* oder den *history*-Mechanismus von der Bourne-Shell ab.

Beide Benutzerschnittstellen, Bourne-Shell und C-Shell, bilden die Basis für die heute unter Linux verfügbaren Shells. Sehr viele Entwicklungen sind als Versuch anzusehen, Elemente der C-Shell in die Bourne-Shell zu übertragen. Die bash ist ein eindrucksvoller Beleg für diese Behauptung.

Ein anderer, schon etwas älterer Ansatz, die allgemein anerkannten Features der Bourne-Shell durch einige Goodies der C-Shell zu ergänzen, ist die so genannte „Korn-Shell“. Sie wird allerdings nicht standardmäßig mit einer Linux-Distribution ausgeliefert.

Die C-Shell ist in der Regel auch nur in einer weiterentwickelten Variante, der *tcsh* verfügbar.

Daneben gibt es noch weitere Varianten wie die *zsh*, die in ihrem Kern aber alle auf die Bourne- oder C-Shell zurückgehen.

Mit welcher der zahlreichen Shells Sie arbeiten wollen, bleibt letztlich Ihrem Geschmack und dem guten Willen des Superusers überlassen. Zwar können Sie aus der bash heraus jede andere Shell starten, es ist aber für die tägliche Arbeit von Vorteil, wenn die von Ihnen gewünschte Shell in der Datei */etc/passwd* eingetragen ist, da nur in diesem Fall beim Einloggen die Startdateien automatisch abgearbeitet werden.

Eine Shell ist eine gewöhnliche Anwendung und kann als solche vom Prompt aus gestartet werden.



### Hinweis

Sollten Sie als Superuser das System verwalten, so seien Sie gewarnt: Die Datei */etc/passwd* ist zwar eine Textdatei, eine manuelle Änderung mit einem der zahlreichen Editoren sollte aber aus Konsistenzgründen vermieden werden. Stattdessen empfiehlt sich die Benutzung der entsprechenden Tools zur Bearbeitung der Passwortdatei.

Zur Änderung des Passworts dient bekanntlich */bin/passwd*. Andere Einträge lassen sich zum Teil mittels *usermod* ändern.

`usermod -s shell user`

etwa weist dem *user* eine (neue) *shell* zu.

An dieser Stelle sollen noch kurz die wesentlichen Unterschiede zwischen bash und C-Shell dargestellt werden.



## Benutzerumgebung

Schon nach dem Einloggen fällt auf: Der Prompt ist anders. *Tabelle 4.13* zeigt die Standard-Prompts der bash und der C-Shell im Überblick. Achtung: Zum Prompt gehört standardmäßig ein Leerzeichen am Ende, um die Eingabe besser vom Prompt differenzieren zu können. Dieser lässt sich optisch nicht besonders elegant darstellen.

**Standard-Prompts von bash und C-Shell**

| Prompt | Bedeutung                                                 |
|--------|-----------------------------------------------------------|
| %      | Prompt der C-Shell für root und den gewöhnlichen Benutzer |
| \$     | Prompt der bash für den normalen Benutzer                 |
| #      | Prompt der bash für den Superuser                         |

*Tabelle 4.13*

Durch die Überarbeitung der Bourne-Shell zur bash bestehen heute glücklicherweise keine nennenswerten Unterschiede mehr zwischen den wichtigsten Features von *bash* und *tcsh*. So entsprechen sich in der Syntax:

- Dateinamensexpression
- grundlegende Datenumleitungen einschließlich *Here*-Dokumenten (die Kopplung zwischen Standardausgabe und Standardfehlerdatei erfolgt in der C-Shell allerdings mit >&)
- Pipelining
- grundlegende Befehlsverkettungen mit { ... }, ( ... ) und ;
- Befehlssubstitution
- Maskierung von Metazeichen
- Hintergrundprozesse (die Behandlung der Jobs ist von der C-Shell in die bash übertragen worden)
- Aliasing (ebenfalls in die bash übernommen)
- *history*-Mechanismus (hier gilt Gleichtes wie für das Aliasing)

Bei aller Gemeinsamkeit bleiben auch erhebliche Unterschiede (über den Prompt hinaus).

Zunächst einmal werden die Variablennamen konventionell kleingeschrieben. Der Befehl *set*, in der *tcsh* angewandt, bringt etwa folgendes Ergebnis.



## Arbeiten mit der bash

```
addsuffix
argv ()
autologout 60
cdpath /usr/spool
cwd /home/cshuser
echo_style both
edit
gid 100
history 100
home /home/cshuser
loginsh
notify
path (/usr/local/bin /bin /usr/bin /usr/X11/bin
 /usr/andrew/bin /usr/openwin/bin /usr/games .)
prompt %m:%~%#
prompt2 %R?
prompt3 CORRECT>%R (y|n|e) ?
shell /bin/tcsh
shlvl 1
status 0
tcsh 6.04.00
term con80x25
tty tty4
uid 504
user cshuser
version tcsh 6.04.00 (Cornell) 93/07/03 (i386-linux)
options 8b,d1,al
```

Sie sehen, dass sich einige der wichtigsten Variablen ausschließlich durch Groß-/Kleinschreibung unterscheiden, wie etwa *home/HOME* oder *path/PATH*. Andere wiederum haben einen etwas aussagekräftigeren Namen, *prompt1* statt *PS1*. Schließlich gibt es noch eine Reihe – für den Benutzer nicht ganz so wesentliche – Variablen, die in der bash nicht benutzt werden. Im Zweifelsfall kontrollieren Sie die Namen der Variablen mit *set*. Wollen Sie detailliertere Informationen, können Sie immer noch die *man pages* von *tcsh* zu Rate ziehen.

In der Praxis gravierender sind aber drei andere Unterschiede.

Beide Shells benutzen unterschiedliche Startdateien. Ist es bei der bash *\$HOME/.profile*, so heißt diese Datei für tcsh *\$home/.login*. Dies ist insofern logisch, als dass man bei unterschiedlichen Shells auch unterschiedliche Umgebungen haben kann.

Beide Shells benutzen eine Reihe unterschiedlicher interner Befehle. So erfolgt etwa die Zuweisung eines Wertes an eine Variable in der C-Shell mit dem Befehl *set*:

```
set home = /
```

Beachten Sie die  
Leerzeichen!



## Benutzerumgebung

---



legt / als neuen Wert für *home* fest. Die Referenzierung (Benutzung der Variablen im Befehl) mittels \$-Zeichen ist glücklicherweise in beiden Umgebungen gleich. Die unangenehmen Seiten dieses Unterschieds kommen bei der Shell-Programmierung voll zum Tragen! Lassen Sie insbesondere dort Vorsicht walten.

Die Syntax der Befehle zur Steuerung von Shell-Skripten ist absolut inkompatibel, mitunter heißen diese Befehle sogar anders. Mit anderen Worten: Shell-Skript-Programmierung muss Rücksicht auf die verschiedenen Ansätze von bash und *tcsh* nehmen. In diesem Buch wird das entsprechende Kapitel auf der Basis der bash besprochen, um so eine einheitliche Sprache zu finden und Kauderwelsch zu vermeiden.

Der gewöhnliche Benutzer, der die Möglichkeiten seiner Shell nur bedingt ausnutzt, wird diese Spitzfindigkeiten deren Nutzung oft gar nicht wahrnehmen. Im Zweifelsfalle lässt sich immer die Onlinehilfe aktivieren.

## Grafische Benutzeroberflächen

Dieser Abschnitt über die bash und ihre Verwandten soll nicht ohne einen Ausblick auf andere Möglichkeiten der Handhabung von Linux bleiben. Man muss nicht den Propheten spielen, um zu behaupten, dass Shells, wie die bisher beschriebenen, durch leistungsfähigere Systeme ersetzt worden sind.

Grafische Oberflächen wie das unter Linux schon länger eingesetzte *X Window* bestechen durch ihre einfache Handhabung. Im Zeitalter der Maus wird dieser Komfort immer mehr zum „Standard“ erhoben, ungeachtet der Tatsache, dass solche Systeme immense Ressourcen fressen. Wir, die Autoren, bedauern dies nicht, wir stellen nur fest.

Wir vertreten allerdings die Meinung, dass die Funktionalität eines Betriebssystems, das „Was passiert wie und warum?“ besser mithilfe der textorientierten Oberflächen erforscht werden kann. Die Erfahrungen, die ein Benutzer mit den spröden Meldungen einer bash machen kann, werden ihn tiefer ins System einblicken lassen als jeder noch so intelligente Mausklick.

Zudem lassen sich unter Linux gerade administrative Maßnahmen besser vom Prompt aus durchführen als von einer – die Zusammenhänge verschleiernnden – grafischen Oberfläche. Einer der Gründe dafür ist sicherlich, dass sehr viele Systemdaten in einfacher Textform vorliegen, so dass sie mit einfachsten Mitteln zu bearbeiten sind. Vor diesem Hintergrund wurden die bash und ihre Verwandten in dieser Ausführlichkeit beschrieben.

*X Window* wird immer mehr zu einem integralen Bestandteil der Benutzerumgebung gehören und die herkömmlichen Werkzeuge verdrängen, wenn nicht ersetzen. Trotzdem wird es als Beispiel einer grafischen Benutzeroberfläche an anderer Stelle vorgestellt, da eine Beschreibung hier den Rahmen einer Einführung in die Benutzerumgebung sprengen würde.





## Editoren

In den vorhergehenden Kapiteln wurde immer wieder von der einfachen Möglichkeit Gebrauch gemacht, mittels

```
cat > datei
```

auf einfache Weise eine Datei zu erstellen. An dieser Stelle sollen nun einige Hilfsmittel gezeigt werden, mit denen Sie Texte editieren können. Diese Notwendigkeit besteht immer wieder, da Änderungen der Benutzerumgebung meist nur in der Änderung eines Eintrags in einer Datei bestehen.

Es muss jedoch betont werden, dass Sie systemrelevante Dateien mit der nötigen Vorsicht behandeln sollten. Falls vorhanden, nehmen Sie zu deren Bearbeitung die dafür vorgesehenen anderen Tools, zwei *passwd* oder *usermod* für die Datei */etc/passwd*. Ist das nicht möglich, so erstellen Sie vor dem Editieren in jedem Fall eine Sicherungskopie. Eine beliebte Methode ist:

```
cp datei datei.old
```

Sie werden solche *old*-Dateien sicherlich schon in Ihrem System gefunden haben.

Für erste Übungen müssen es ja auch nicht gerade die wichtigen Dateien sein, die Sie bearbeiten. Eine erste Übung zum Erlernen eines Editors ist einfach die, mit ihm einen neuen Text zu erstellen.

Die folgenden Darstellungen sollen sich auf die Standardeditoren *ed* und *vi* sowie den recht handlichen *joe* beschränken. Der an *emacs* interessierte Leser sei an dieser Stelle ausnahmsweise nicht einfach an die *man pages* verwiesen, *emacs* gehört zu den Tools, für die es eine eigene Literatur gibt.

### ***ed***

Mit *ed* haben sie einen einfachen, zeilenorientierten Editor zur Verfügung, der sehr stabil arbeitet, aber auch einige Ansprüche an den Benutzer stellt. Seine Handhabung erfordert einige Übung.

Seine uneingeschränkten Vorteile sind:

- Er ist immer verfügbar,
- er arbeitet unabhängig von den jeweiligen Terminaleneinstellungen.

*ed* unterscheidet zwei Arbeitsmodi, den Eingabe- und den Befehlsmodus. *Eingabe-modus* heißt, Sie können eine einzelne Zeile bearbeiten, indem Sie den gewünschten Text tippen und gegebenenfalls mit Fehler löschen. Die Benutzung der Cursortasten ist tabu.



## Benutzerumgebung

Die Bearbeitung der jeweiligen Zeile wird durch ↵ bestätigt, man gelangt danach zur Bearbeitung der nächsten Zeile.

Im *Befehlsmodus* lässt sich der vorhandene Text mithilfe verschiedener Befehle bearbeiten und nicht zuletzt speichern. Die Befehle sind Buchstabenkürzel, die eventuell durch Bereichsangaben (der Art Zeile 1 bis 10) oder Zusatzinformationen wie den Dateinamen beim Speichern ergänzt werden können.

Alte MS-DOS-Hasen erinnern sich vielleicht noch an *edlin*

...

Aus dem Eingabemodus zurück in den Befehlsmodus gelangt man durch eine neue Zeile, die man mit einem . beginnen lässt.

Schließlich, wie verlässt man den Editor ohne zu speichern? Der Befehl lautet *Q* (quit), großgeschrieben!

Mit diesen notwendigsten Informationen versehen, soll eine kleine Sitzung mit *ed* durchgeführt werden.

```
$ ed neue.datei↵
neue.datei: No such file or directory
a↵
Dies ist ein Tets↵
mit dem Editor ed↵
Dabei ist mir in der↵
ersten Zeile↵
ein Fehler unterlaufen↵
.↵
p↵
ein Fehler unterlaufen
1,5p↵
Dies ist ein Tets
mit dem Editor ed
Dabei ist mir in der
ersten Zeile
ein Fehler unterlaufen
1↵
Dies ist ein Tets
s/ts/st/↵
Dies ist ein Test
q↵
?
w↵
93
q↵
$
```





Es beginnt mit dem Aufruf von *ed* in der Form

```
ed [dateiname]
```

Falls Sie *ed* ohne Dateinamen starten, müssen Sie beim Speichern (Befehl *w*) einen Dateinamen angeben. *ed* meldet sich mit der Feststellung, dass die angegebene Datei noch nicht existiert, merkt sich den Namen aber schon für den Fall, dass später gespeichert werden soll.

Mit dem Befehl *a* wird in den Eingabemodus gewechselt, um dann einige Zeilen Text zu schreiben. Die mit dem Punkt beginnende Zeile sorgt schließlich für die Rückkehr in den Befehlsmodus.

Mit *p* (print) wird zunächst die aktuelle Zeile angezeigt. Da dies in der Regel zu wenig ist – Sie wollen doch sicher den gesamten Text noch einmal sehen –, wiederholen Sie den letzten Befehl unter Angabe der Zeilennummern.

Für Befehle, die der Bearbeitung des Textes dienen, gilt hierbei eine recht einfache Regelung: Ohne Angabe einer Zeilennummer bezieht sich der Befehl auf die aktuelle Zeile, das ist im Normalfall die zuletzt eingegebene. Die aktuelle Zeile kann im Befehlsmodus durch Eingabe der Zeilennummer bestimmt werden. Die Eingabe von bedeutet im Befehlsmodus: Mache die nächste Zeile zur aktuellen.

Stellt man einem Bearbeitungsbefehl eine Zahl voran, betrifft der Befehl die Zeile mit der entsprechenden Zeilennummer. Zwei durch Komma getrennte vorangestellte Zahlen beschreiben einen Bereich. So bewirkt *1,5p* die Ausgabe der ersten fünf Zeilen. Die letzte Zeile eines Textes kann man durch *\$* ansprechen. *1,\$n* gibt den gesamten Text mit vorangestellten Zeilennummern aus (*number*).

Doch zurück zur zuvor beschriebenen kleinen Sitzung. Nach der Anzeige der fünf Zeilen soll Zeile 1, in der ein typischer Editierfehler unterlaufen ist, bearbeitet werden. Freundlicherweise zeigt *ed* den Inhalt der Zeile gleich an. Es folgt die Anwendung einer der raffinierteren Methoden, die *ed* bietet: ein Suchen und Ersetzen in der aktuellen Zeile (keine vorangestellten Zeilennummern!). *s* (substitute) tauscht die erste auftretende Zeichenfolge „*ts*“ gegen „*st*“ aus.

Natürlich lassen sich auch längere Suchmuster in größeren Bereichen auf diese Weise bearbeiten. Sie müssen nur daran denken, dass das *erste* Auftreten der zu suchenden Zeichenfolge von *s* berücksichtigt wird.

Angenommen, Sie wollen in der dritten Zeile (*Dabei ist mir in der*) das Wort *in* durch *an* ersetzen, so geschieht das mit

```
3s/in/an/
```

nicht aber mit *3s/i/a/!*

Nach dem erfolgreichen Reparaturversuch (*ed* zeigt die geänderte Zeile an) soll der Test mit *q* (quit – kleingeschrieben!) beendet werden. Aber *ed* weigert sich und zeigt stattdessen ein Fragezeichen an.



## Benutzerumgebung

---



Normalerweise kann man an dieser Stelle mit *h* (help) nähere Informationen zu dieser Meldung anfordern, aber Sie wissen sicher schon was da passiert sein muss: Sie haben noch nicht gespeichert. Sobald *ed* feststellt, dass sich der Text in irgend-einer Form geändert hat, ist ein Verlassen des Editors mit *q* nicht mehr möglich. Man muss erst speichern.

Dies geschieht mit *w* (write). *ed* gibt daraufhin die Anzahl der geschriebenen Zeichen an und lässt Sie schließlich mit *q* die Sitzung beenden.

Bevor die wichtigsten Befehle im Überblick dargestellt werden, soll hier noch eine Möglichkeit aufgezeigt werden, wie Sie sich die Arbeit mit *ed* erleichtern können. Sie haben sicher bemerkt, dass kein Unterschied zwischen Befehls- und Eingabemodus am Bildschirm zu erkennen ist. Dies kann zu unangenehmen Problemen führen, wobei noch das harmloseste das ist, dass man Zeilen wie *1,5p* im Text wiederfindet.

Durch einen geeigneten Aufruf können Sie *ed* veranlassen, den Befehlsmodus durch einen Prompt zu kennzeichnen. Die Syntax ist:

```
ed -p prompttext [dateiname]
```

Achten Sie darauf, dass im *prompttext* enthaltene Metazeichen beim Aufruf zu maskieren sind.

```
$ ed -p \> neue.datei<
93
>a<
<
>q<
$
```

Dass die Datei bereits existiert, quittiert *ed* mit der Angabe der gelesenen Zeichen. Als Prompt steht Ihnen im Befehlsmodus das Größerzeichen zur Verfügung (in der Eingabezeile maskiert!).

*Tabelle 4.14* fasst die Kommandos zusammen. Die Befehle selbst sind fett dargestellt, Ergänzungen nicht. [...] bedeutet: Ergänzung ist optional. Gebrauchte Abkürzungen sind: *z* = Zeilennummer, *b* = Bereichsangabe (kann immer auch eine einzelne Zeile sein).

Wechsel in den Eingabemodus



## Editoren



### Befehle des Editors ed

#### Befehl

#### Bedeutung

|      |                                                                                                            |
|------|------------------------------------------------------------------------------------------------------------|
| [z]a | (append) Einfügen neuer Zeilen nach der angegebenen Zeile                                                  |
| [b]c | (change) Ersetzt den angegebenen Bereich durch den nach dem Wechsel zum Eingabemodus neu eingegebenen Text |
| [b]i | (insert) Einfügen neuer Zeilen vor der angegebenen Zeile                                                   |

#### Anzeige des Textes

|      |                                                                                                                                   |
|------|-----------------------------------------------------------------------------------------------------------------------------------|
| [b]l | (list) Anzeige der genannten Bereichs, überlange Zeilen werden umbrochen, nicht druckbare Zeichen in oktaler Notation dargestellt |
|------|-----------------------------------------------------------------------------------------------------------------------------------|

|      |                                                      |
|------|------------------------------------------------------|
| [b]n | (number) Ausgabe des Bereichs mit Zeilennummerierung |
|------|------------------------------------------------------|

|      |                                                                    |
|------|--------------------------------------------------------------------|
| [b]p | Anzeige des Bereichs, dessen letzte Zeile wird neue aktuelle Zeile |
|------|--------------------------------------------------------------------|

#### Löschen und Kopieren

|      |                                           |
|------|-------------------------------------------|
| [b]d | (delete) Löschen des angegebenen Bereichs |
|------|-------------------------------------------|

|         |                                                                                                       |
|---------|-------------------------------------------------------------------------------------------------------|
| [b]m[z] | (move) kopiert den angegebenen Bereich hinter die angegebene Zeile, der Originalbereich wird gelöscht |
|---------|-------------------------------------------------------------------------------------------------------|

|         |                                                                                                                 |
|---------|-----------------------------------------------------------------------------------------------------------------|
| [b]t[z] | (transfer) kopiert den angegebenen Bereich hinter die angegebene Zeile, der Originalbereich wird nicht gelöscht |
|---------|-----------------------------------------------------------------------------------------------------------------|

#### Markieren

|       |                                                                                                                                                    |
|-------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| [z]kc | (mark) gibt der angegebenen Zeile den Namen c. Dies muss ein Kleinbuchstabe sein, kann in Zeilen-/Bereichsanzeigen in der Form 'c' benutzt werden. |
|-------|----------------------------------------------------------------------------------------------------------------------------------------------------|

#### Suchen und Ersetzen

|                |                                                                                                                                                                   |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| g/text/befehle | (global) führt die angegebene Befehle für alle Zeilen aus, die den entsprechenden Text beinhalten.<br>g/wrkst/d löscht alle Zeilen, in denen wrkst zu finden ist. |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|

|            |                                                                                                   |
|------------|---------------------------------------------------------------------------------------------------|
| [b]G/text/ | ähnlich zu g, allerdings können die Befehle interaktiv für jede gefundene Zeile eingegeben werden |
|------------|---------------------------------------------------------------------------------------------------|



## Benutzerumgebung

### Befehle des Editors *ed* (Forts.)

| Befehl                                     | Bedeutung                                                                                                                                                          |
|--------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [b]s/alt/neu/[g]                           | ersetzt im angegebenen Bereich den ersten auftretenden Text <i>alt</i> durch den Text <i>neu</i> . Durch Anhänger von <i>g</i> erzwingt man eine globale Ersetzung |
| v/text/befehle                             | wie g, nur erfolgen die Befehle nur für Zeilen, die <i>text</i> nicht enthalten                                                                                    |
| V/text/                                    | wie V, nur erfolgen die Befehle nur für Zeilen, die <i>text</i> nicht enthalten                                                                                    |
| /text/                                     | (forward search) vorwärts suchen nach <i>text</i>                                                                                                                  |
| ?text?                                     | (backward search) rückwärts suchen nach <i>text</i>                                                                                                                |
| Datei-Handling und Verlassen von <i>ed</i> |                                                                                                                                                                    |
| e datei                                    | (edit) Laden einer neuen Datei, die alte wird aus dem Puffer entfernt (mit Warnung wie bei q)                                                                      |
| E datei                                    | wie e, aber ohne Warnung                                                                                                                                           |
| f                                          | (file) Anzeige des Namens der aktuellen Datei                                                                                                                      |
| q                                          | (quit) <i>ed</i> verlassen (nur möglich, wenn der Text seit der letzten Speicherung unverändert ist)                                                               |
| Q                                          | wie q, ohne Warnung                                                                                                                                                |
| [z]r datei                                 | liest hinter die angegebene Zeile eine Datei ein                                                                                                                   |
| [b]w [datei]                               | (write) speichert den angegebenen Bereich in die genannte Datei – ist der Dateiname dem Editor vorher schon bekannt, braucht er nicht angegeben zu werden          |
| [b]W [datei]                               | wie w, allerdings wird der Text an die Datei angehängt                                                                                                             |
| Verschiedene Befehle                       |                                                                                                                                                                    |
| h                                          | (help) Hilfe zur letzten Warnung mit ?                                                                                                                             |
| H                                          | Hilfemodus einschalten (Warnmeldungen statt Ausgabe des ?)                                                                                                         |
| [b]j                                       | (join) fügt die Zeilen des angegebenen Bereichs zu einer Zeile zusammen                                                                                            |
| u                                          | (undo) macht die letzte Änderung in der aktuellen Zeile rückgängig, nach dem Verlassen der Zeile nicht mehr möglich                                                |



## Editoren



### Befehle des Editors *ed* (Forts.)



| Befehl  | Bedeutung                                                                     |
|---------|-------------------------------------------------------------------------------|
| !befehl | führt einen beliebigen Linux-Befehl aus, kehrt anschließend zum Editor zurück |

Tabelle 4.14



Bei den Such- und Ersetzungsbefehlen können Sie Suchmuster verwenden. Diese sind so zu bilden, wie im Kapitel 3 *Linux-Dateisystem* bei der Besprechung von *grep* bereits diskutiert wurde.

## vi

Mit *vi* (das spricht man „wie ei“ aus) steht Ihnen einer der leistungsfähigsten Editoren überhaupt zur Verfügung. Er bietet eine unglaubliche Fülle an Editierbefehlen, mit deren Hilfe ein komfortables Erstellen von Text möglich ist. Zudem ist er bildschirmorientiert, d.h., er zeigt während der Arbeit eine Bildschirmseite des Textes an. Leere Zeilen werden durch eine Tilde (~) am Zeilenanfang markiert. Die Version, die den neueren Distributionen beiliegt ist die verbesserte Version des *vi* mit Namen *vim*.

Zum Wandern durch den Text sind die Cursortasten erlaubt. Im Gegensatz zu vielen Textverarbeitungsprogrammen kann man aber nicht während der Wanderung durch den Text auch Eingaben machen. Das liegt daran, dass auch *vi* Eingabe- und Befehlsmodus unterscheidet. Die Benutzung der Cursortasten ist nur im Befehlsmodus erlaubt.

Auch *vi* ist ein schon recht betagter Editor, der konzipiert wurde, um ihn an jedem beliebigen Terminal einsetzen zu können, daher ist die Cursorsteuerung auch recht rustikal. Statt ihrer gibt es einen ausgeprägten Befehlssatz für Bewegungen im Text.

Nachdem nun die prinzipiellen Vorteile genannt wurden, soll an dieser Stelle aber auch nicht verschwiegen werden, dass gerade die Fülle an Befehlen für den Neuling eine große Hürde bei der Handhabung des *vi* ist. Sollten Sie sich entschlossen haben, ihn wegen seiner Leistungsfähigkeit einerseits und seiner vielseitigen Einsatzbereitschaft andererseits zu Ihrem Standardeditor zu machen, können wir Ihnen nur raten, seinen Befehlssatz peu à peu zu erlernen. Geben Sie nicht auf: Es lohnt sich!

Sollte *vi* in Ihrer Umgebung merkwürdiges Verhalten beim Bildschirmaufbau zeigen, müssen Sie ihn eventuell Ihrem Terminal anpassen. Im einfachsten Fall müssen Sie dazu die Systemvariable *TERM* ändern. Sitzten Sie an der Konsole, das ist der Linux-Rechner, sollte der Wert der Variablen *console* sein. Notfalls lässt sich dies über



## Benutzerumgebung

TERM=console

vor dem Start des vi erledigen. Arbeiten Sie an einem Terminal in einem Netz oder sind Sie auf andere Weise über Entfernung mit der Konsole verbunden, müssen Sie als Wert für *TERM* denjenigen bestimmen, der Ihrer Terminal-Emulation am besten entspricht, sehr häufig ist das *vt100* oder *ansi*.

Verfeinerte Anpassungsmöglichkeiten bietet der Befehl `set`, auf dessen Einzelheiten hier allerdings nicht eingegangen werden soll.

Bevor Ihnen die wesentlichen Befehle des vi in Form einer Kurzreferenz vorgestellt werden, sollen an dieser Stelle zunächst die notwendigsten Handgriffe zu seiner Bedienung dargestellt werden.

Der Start erfolgt mit:

`vi[ -option] [ datei]`

An wichtigen Optionen kennt vi unter anderem die Möglichkeit, die Datei nur lesend zu öffnen (-R), unmittelbar in den Eingabemodus zu wechseln (-i) oder eines der ex-Kommandos beim Start ausführen zu lassen (+befehl). Z.B. bedeutet der Aufruf

vi +123 testdatei

dass *vi* nach dem Start sofort in die Zeile 123 springt.

Es zeigt sich zunächst etwa folgender Bildschirm:

Zu Beginn der bereits fertige Text – da er sehr kurz ist, werden entsprechend viele Leerzeilen angezeigt. Am Ende folgt die Statuszeile, die Ihnen den Namen der Datei anzeigt, eventuell, ob Sie neu angelegt wird [NEW FILE], oder aber, dass Sie vi ohne Angabe eines Dateinamens aufgerufen haben „[NO FILE]“. Zuletzt sehen Sie die Anzahl der Zeilen und Zeichen eingeblendet.



**Befehle zum Verlassen von vi**

Verlassen können Sie vi auf verschiedene Weisen, hier ein Überblick:

**Befehle zum Verlassen des vi**

| Befehl | Wirkung                                                              |
|--------|----------------------------------------------------------------------|
| :q     | Verlassen, ohne zu speichern (nur nach vorherigem Speichern möglich) |
| :q!    | Verlassen, ohne zu speichern                                         |
| :wq    | Speichern und Beenden                                                |
| :x     | wie :wq                                                              |
| ZZ     | wie :wq                                                              |

Tabelle 4.15

Sollten Sie als Anfänger versehentlich vi aufgerufen haben, empfiehlt sich das sofortige Verlassen mit :q!. Eventuell müssen Sie vorher einmal `Esc` drücken, um in den Befehlsmodus zu gelangen. Wenn Sie nicht ganz sicher sind, drücken Sie einfach vor jeder Eingabe eines Befehls einmal `Esc`.

**Arbeitsmodi des vi**

An der kleinen Befehlsauswahl wird vielleicht schon deutlich, dass vi verschiedene Befehlssätze kennt. Das liegt daran, dass er streng genommen gar kein Editor ist, sondern nur die komfortable Oberfläche des zugrunde liegenden zeilenorientierten Editors ex, der nicht nur einen dem ed sehr ähnlichen Namen hat.

Alle Befehle, die mit dem Doppelpunkt eingeleitet werden, sind solche ex-Befehle. Vielleicht ist es Ihnen schon aufgefallen: Bei Eingabe des : im Befehlsmodus wechseln Sie automatisch in die letzte Zeile, die „Statuszeile“, um dort Ihren eigentlichen Befehl einzugeben. In diesem Moment befinden Sie sich im ex-Modus.

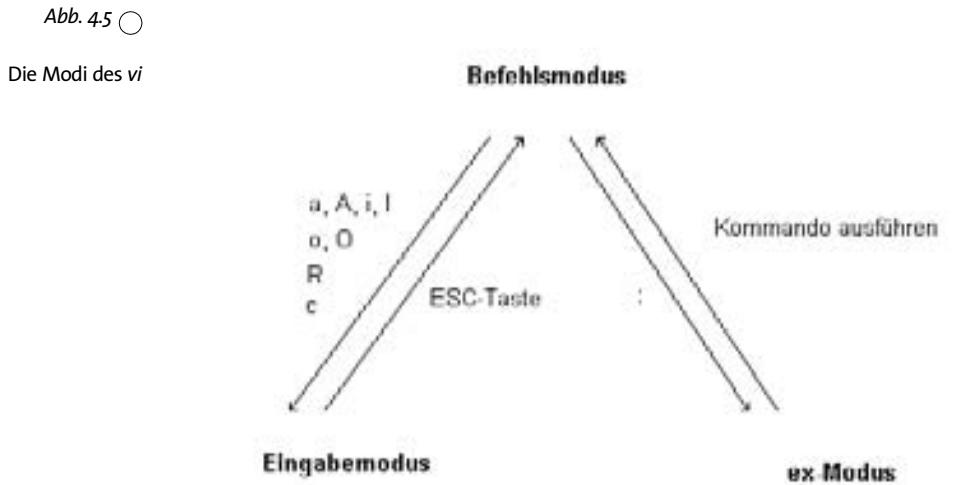
vi kennt für die Arbeit also drei Modi:

- den Eingabemodus, in dem Text geschrieben wird – je nach gewähltem Eingabemodus wird vorhandener Text überschrieben oder ergänzt,
- den Befehlsmodus,
- den ex-Modus zur Eingabe spezieller ex-Befehle.

Das Wechselspiel der drei Modi zeigt Abbildung 4.5.



## Benutzerumgebung



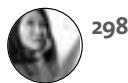
### Wechsel in den Einfügemodus

Der Wechsel vom Befehlsmodus in den ex-Modus erfolgt wie bereits erwähnt mit ;, zurück gelangt man automatisch nach Eingabe des Befehls. Zum Wechseln aus dem Eingabemodus in den Befehlsmodus genügt [Esc]. In die umgekehrte Richtung gibt es eine Reihe unterschiedlicher Varianten:

#### Befehle zum Wechseln in den Eingabemodus des vi

| Befehl | Wirkung                                                                                           |
|--------|---------------------------------------------------------------------------------------------------|
| i      | Einfügen vor der aktuellen Position                                                               |
| I      | Einfügen am Zeilenanfang                                                                          |
| a      | Einfügen nach der aktuellen Position                                                              |
| A      | Einfügen am Zeilenende                                                                            |
| o      | Einfügen einer Zeile nach der aktuellen, Wechsel in die Zeile                                     |
| O      | Einfügen einer Zeile vor der aktuellen, Wechsel in die Zeile                                      |
| R      | Wechsel in den Überschreibmodus                                                                   |
| r      | Das nächste eingegebene Zeichen löscht das Zeichen am Cursor, sofortige Rückkehr zum Befehlsmodus |

Tabelle 4.16



## Editoren



cobjekt Wechsel in den Eingabemodus unter Löschung von objekt. Mögliche objekte sind Wörter, Zeilen, Zeilenanfang oder -ende und viele mehr. Eine genauere Aufstellung entnehmen Sie der folgenden *Tabelle 4.17*.

### Objekte

Die Angabe von Objekten bei einem Befehl, wie es erstmals beim Befehl *c* erwähnt wurde, ist für sehr viele Befehle charakteristisch, deshalb soll nun eine Liste der möglichen Befehle zusammengestellt werden. Es ist dabei unerheblich, ob Sie löschen oder kopieren, sich bewegen oder etwas verschieben wollen: Der Beginn des Objekts wird immer von der aktuellen Cursorposition gerechnet.

#### Mögliche Objekte für Befehle des vi

Kürzel      Objekt

|                  |                                    |
|------------------|------------------------------------|
| <i>Leertaste</i> | einzelnes Zeichen                  |
| <i>w</i>         | Wort                               |
| <i>W</i>         | Wort mit Sonderzeichen             |
| <i>b</i>         | Wort rückwärts                     |
| <i>B</i>         | Wort rückwärts (mit Sonderzeichen) |
| <i>G</i>         | Text bis zum Ende                  |
| <i>^</i>         | Text bis zum Zeilenanfang          |
| <i>\$</i>        | Text bis zum Zeilenende            |
| <i>(</i>         | Text bis zum Satzanfang            |
| <i>)</i>         | Text bis zum Satzende              |
| <i>{</i>         | Text bis zum Anfang des Absatzes   |
| <i>}</i>         | Text bis zum Ende des Absatzes     |
| <i>[[</i>        | Text bis zum Abschnittbeginn       |
| <i>]]</i>        | Text bis zum Abschnittende         |

*Tabelle 4.17*

Als Beispiel für die Benutzung von Objekten betrachten Sie noch einmal den Befehl *c*. Mit *cb* wechselt man in den Eingabemodus (wegen *c*), dabei wird das Wort vor der Cursorposition (wegen *b*) gelöscht (wegen *c*). Gibt man zusätzlich vor dem Objekt eine Zahl an, werden entsprechend viele Objekte bearbeitet. So löscht *c10w* beim Übergang in den Eingabemodus die zehn folgenden Wörter.



## Benutzerumgebung



### Bewegen im Text

Bisher haben Sie sich nur mithilfe der Cursortasten im Text bewegt. Im folgenden lernen Sie eine Reihe weiterer Möglichkeiten kennen, die Sie unabhängig von den Cursortasten machen und die außerdem ein schnelleres Bewegen im Text erlauben.

#### Befehle zum Bewegen im Text



| Befehl    | Wirkung                                                                                                                           |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------|
| k         | Zeile nach oben (wie )                                                                                                            |
| j         | Zeile nach unten (wie )                                                                                                           |
| h         | Zeichen nach links (wie )                                                                                                         |
| l         | Zeichen nach rechts (wie ). Orientieren Sie sich nicht am Kennbuchstaben, sondern an der Tastatur, das  ist links vom  zu finden. |
| o (Null!) | zum Zeilenanfang (erstes Zeichen)                                                                                                 |
| ^         | zum Zeilenanfang (erstes sichtbares Zeichen)                                                                                      |
| \$        | zum Zeilenende                                                                                                                    |
| w         | zum nächsten Wort in der Zeile                                                                                                    |
| W         | zum nächsten durch Leerzeichen begrenzten Wort in der Zeile                                                                       |
| b         | zum vorherigen Wort in der Zeile                                                                                                  |
| B         | zum vorherigen durch Leerzeichen begrenzten Wort in der Zeile                                                                     |
| G         | zum Dateiende                                                                                                                     |
| nrG       | zur Zeile nr                                                                                                                      |
| H         | zum Anfang des Bildschirms                                                                                                        |
| M         | zur Mitte des Bildschirms                                                                                                         |
| L         | zur letzten Bildschirmzeile                                                                                                       |
| (         | zum vorherigen Satz                                                                                                               |
| )         | zum nächsten Satz                                                                                                                 |
| {         | zum vorhergehenden Absatz                                                                                                         |
| }         | zum nächsten Absatz                                                                                                               |
| [[        | zum vorhergehenden Abschnitt                                                                                                      |



## Editoren



### Befehle zum Bewegen im Text (Forts.)

| Befehl                | Wirkung                                                                                 |
|-----------------------|-----------------------------------------------------------------------------------------|
| <code>JJ</code>       | zum nächsten Abschnitt                                                                  |
| <code>fzeichen</code> | zur nächsten Position von zeichen (vorwärts)                                            |
| <code>Fzeichen</code> | zur nächsten Position von zeichen (rückwärts)                                           |
| <code>tzeichen</code> | zur nächsten Position von zeichen (vorwärts), Cursor vor dem Zeichen positionieren      |
| <code>Tzeichen</code> | zur nächsten Position von zeichen (rückwärts), Cursor vor dem Zeichen positionieren     |
| <code>;</code>        | Befehl <code>f</code> , <code>F</code> , <code>t</code> oder <code>T</code> wiederholen |
| <code>Strg+F</code>   | Seite vorwärts blättern                                                                 |
| <code>Strg+B</code>   | Seite rückwärts blättern                                                                |
| <code>Strg+D</code>   | halbe Seite vorwärts blättern                                                           |
| <code>Strg+U</code>   | halbe Seite rückwärts blättern                                                          |

Tabelle 4.18



Wie Sie sehen, wiederholen sich eine ganze Reihe von Begriffen. Die Objekte, die vorher genannt wurden, lassen sich ausnahmslos auch direkt zum Bewegen im Text benutzen. Wenn Sie die vielfältigen Möglichkeiten der Cursorbewegung verinnerlicht haben, werden Ihnen viele Befehle viel einfacher von der Hand gehen.

### Löschen und Kopieren

Auf der Basis von Objekten sind auch einige Lösch- und Kopierbefehle aufgebaut. Sie sind der Tabelle 4.19 zu entnehmen.

### Befehle zum Löschen und Kopieren



| Befehl               | Wirkung                                                                                               |
|----------------------|-------------------------------------------------------------------------------------------------------|
| <code>x</code>       | löscht das Zeichen am Cursor in einen Puffer                                                          |
| <code>X</code>       | löscht das Zeichen vor dem Cursor in einen Puffer                                                     |
| <code>dobjekt</code> | löscht das objekt in einen Puffer, z.B. <code>dw</code> löscht das folgende Wort, besondere Variante: |
| <code>dd</code>      | löscht ganze Zeile                                                                                    |
| <code>yobjekt</code> | kopiert das Objekt in einen Puffer, besondere Variante:                                               |



## Benutzerumgebung

### Befehle zum Löschen und Kopieren (Forts.)

| Befehl | Wirkung                                    |
|--------|--------------------------------------------|
| yy     | kopiert ganze Zeile                        |
| p      | fügt Pufferinhalt nach aktueller Zeile ein |
| P      | fügt Pufferinhalt vor aktueller Zeile ein  |

Tabelle 4.19

Alle genannten Befehle benutzen einen Puffer, über den sich so sehr geschickt Texte verschieben lassen. Neben dem Standardpuffer können Sie noch selbst definierte Puffer benutzen. Durch Angeben eines Kleinbuchstabens vor dem Befehl wird der entsprechende Puffer angesprochen. Um eine Verwechslung mit einem der Befehle zu vermeiden, sind einfache Anführungszeichen voranzustellen.

"ayy

löscht die aktuelle Zeile in den Puffer namens *a* und

"ap

kopiert den Inhalt von Puffer *a* in den Text zurück, und zwar nach der aktuellen Zeile.

### Suchen und Ersetzen

Eine der interessanteren Möglichkeiten eines Editors sind die Funktionen zum Suchen und Ersetzen von Text. Sie sind denen des *ed* sehr ähnlich und lassen sich der folgenden Tabelle entnehmen.

### Befehle zum Löschen und Kopieren

| Befehl  | Wirkung                                                                                    |
|---------|--------------------------------------------------------------------------------------------|
| /muster | sucht nach muster im Text (vorwärts)                                                       |
| ?muster | sucht nach muster im Text (rückwärts)                                                      |
| n       | Wiederholen der letzten Suche                                                              |
| N       | Wiederholen der letzten Suche, aber in umgekehrter Richtung                                |
| %       | Suche nach der zugehörigen Klammer (wichtiges Hilfsmittel zum Editieren von C-Quelltexten) |

## Editoren



### Befehle zum Löschen und Kopieren (Forts.)



| Befehl                | Wirkung                                                                                                                                            |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| :[b]s/text1/text2/[g] | ex-Befehl, ersetzt <i>text1</i> durch <i>text2</i> im genannten Bereich <i>b</i> . Syntax und Arbeitsweise entsprechen der des <i>ed</i> -Befehls. |

Tabelle 4.20



Bei der Suche nach Text lassen sich Suchmuster verwenden, wie Sie sie schon bei *grep* kennen gelernt haben. Eine Liste der Metazeichen, mit denen diese Suchmuster gebildet werden können, ist in Tabelle 4.21 enthalten.

### Metazeichen für Suchroutinen in Editoren und Suchprogramme



| Metazeichen | Steht für                                                                                                                                             |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| .           | ein beliebiges Zeichen                                                                                                                                |
| *           | eine beliebige Wiederholung des vorhergehenden Zeichens, es ist auch „null“-malige Wiederholung gemeint. So steht .* für eine beliebige Zeichenfolge. |
| [...]       | genau eines der Zeichen in der Klammer, z.B. [aA] meint, dass der gesuchte Text an dieser Stelle ein <i>a</i> oder ein <i>A</i> enthalten muss        |
| [...-...]   | genau eines der Zeichen aus der Liste in der Klammer, z.B. meint [a-z] alle Kleinbuchstaben                                                           |
| [^...]      | genau eines der Zeichen, das nicht in der Klammer ist, z.B. steht [^123] für alle Zeichen außer 1, 2 und 3                                            |
| [^...-...]  | genau eines der Zeichen, das nicht in der Liste in der Klammer ist, z.B. steht [^0-9] für alle Zeichen, die keine Ziffern sind                        |
| ^           | Zeilenanfang                                                                                                                                          |
| \$          | Zeilenende                                                                                                                                            |
| ^\$         | leere Zeile                                                                                                                                           |
| ^text\$     | Zeile, die genau den angegebenen text enthält                                                                                                         |
| \<          | Wortanfang                                                                                                                                            |
| \>          | Wortende                                                                                                                                              |



## Benutzerumgebung

### Metazeichen für Suchroutinen in Editoren und Suchprogramme (Forts.)

| Metazeichen | Steht für                                                                                                                                                                                                                                                                                                                   |
|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| \           | hebt die Bedeutung des folgenden Zeichens als Metazeichen auf. Will man tatsächlich nach einem Text suchen, der einen Punkt enthält, muss verhindert werden, dass <i>grep</i> ihn als Ersatz für ein beliebiges Zeichen interpretiert. Mit \. hebt man die Bedeutung des Punktes als Metazeichen für diesen einen Fall auf. |

Tabelle 4.21

### Weitere Dateibefehle

Nachdem Sie die wesentlichen Befehle zur Bearbeitung des Textes im Überblick kennen gelernt haben, geht es nun noch einmal zurück zum Datei-Handling. Neben den Befehlen zum Verlassen von *vi* gibt es noch Möglichkeiten, Dateien miteinander zu verbinden oder aber Ergebnisse eines Befehls in einen Text einzubinden. Auch hier ein kurzer Überblick.

### Befehle zum Datei-Handling

| Befehl        | Wirkung                                                                                 |
|---------------|-----------------------------------------------------------------------------------------|
| :[b]w [datei] | speichert den genannten Bereich in <i>datei</i> .                                       |
| :e datei      | laden der angegebenen Datei zum Editieren, die bisherige wird aus dem Speicher entfernt |
| :r datei      | Einfügen des Inhalts der angegebenen Datei in den aktuellen Text                        |
| :r!befehl     | Einfügen der Ergebnisse des (beliebigen Linux-Befehls) in den aktuellen Text            |

Tabelle 4.22

### Diverse Befehle

Zum Abschluss dieser „Tour de Force“ durch den *vi* sollen noch einige Befehle zusammengefasst werden, die sich nicht direkt einem der bisher besprochenen Bereiche zuordnen lassen.



## Editoren



### Diverse Befehle des vi

| Befehl                  | Wirkung                                                                                                                                                                                                   |
|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>u</i>                | hebt die letzte Änderung des Puffers auf (durch <i>d</i> , <i>x</i> , <i>X</i> oder <i>y</i> )                                                                                                            |
| <i>U</i>                | hebt die letzte Änderung der aktuellen Zeile auf, nicht mehr möglich nach Verlassen der Zeile                                                                                                             |
| <i>mzeichen</i>         | markiert die aktuelle Position, zeichen ist ein Kleinbuchstabe. Eine Verwendung finden solche Marken bei Befehlen, die Objekte bearbeiten, z.B. <i>y</i> <i>a</i> kopiert den Text bis zur Marke <i>a</i> |
| <i>~</i>                | ändert Groß-/Kleinschreibung des aktuellen Zeichens                                                                                                                                                       |
| <i>J</i>                | aktuelle Zeile mit der folgenden verbinden                                                                                                                                                                |
| <b>[Strg+R]</b>         | Auffrischen des Bildschirms                                                                                                                                                                               |
| <i>!befehl</i>          | Ausführen eines (beliebigen) Linux-Befehls                                                                                                                                                                |
| <i>:set option wert</i> | Ändern einer vi-internen Einstellung                                                                                                                                                                      |
| <i>:map kürzel text</i> | Definieren eines Makros (Tastaturkürzel)                                                                                                                                                                  |
| <i>:unmap kürzel</i>    | Aufheben der Makrodefinition                                                                                                                                                                              |

Tabelle 4.23



Diese kurze Zusammenstellung hat die wichtigsten Befehle des *vi* vorgestellt. Mit diesem Basiswissen sollten Sie in der Lage sein, die Funktionalität des Editors zu erforschen.

## joe

Auch wenn UNIX eine Reihe von Hochleistungsseditoren zur Verfügung stellt, die von ihrem Leistungsumfang meist gar nicht mehr als einfache Editoren bezeichnet werden können, sondern die eher mit komplexen Textverarbeitungs- und Programmierumgebungen zu vergleichen sind, so wünscht man sich (gerade auch als Anfänger, der sich an die hohe Komplexität vieler UNIX-Produkte erst noch gewöhnen muss) manchmal doch kleine Hilfswerzeuge, deren Handhabung nicht die Kenntnis des Inhalts von Hunderten von Handbuchseiten erforderlich macht.

Auch würde die Verwendung eines Editors wie *emacs* zur Erstellung eines kurzen Vierzeilers doch eher dem Schießen mit Kanonen auf Spatzen gleichkommen, als effektives Arbeiten bedeuten.

Ein hervorragendes Beispiel für ein solches kleines, aber dennoch feines Hilfswerkzeug ist der Editor *joe*.



## Benutzerumgebung

---



Dieser von Joseph H. Allen (*joe* ist ein unendlich rekursives Akronym für Joe's own editor – dt. Joes eigener Editor) entwickelte Editor ist ein leistungsfähiger ASCII-Editor, der bezüglich seiner Handhabung speziell denjenigen sehr stark entgegenkommt, die noch mit Micro-Pros Textverarbeitungssystem WordStar gearbeitet haben oder die mit den Editoren der Programmiersprachen zu Borlands Turbo-Sprachen (Turbo Pascal, Turbo C usw.) vertraut sind, denn *joe* verwendet WordStar-kompatible Steuerungssequenzen, die u.a. auch von diesen Editoren verwendet werden.

Bei Steuerungssequenzen handelt es sich um Tastenkombinationen zur Manipulation von Textelementen, zum Speichern, zum Laden usw. (manche dieser Sequenzen finden sich sogar noch bei Editoren wie Microsofts *edit*, wie zum Beispiel die Sequenz zum Löschen einer Zeile: **Strg+Y**).

Genau betrachtet ist *joe* nicht nur ein Editor, sondern fünf Editoren in einem, denn abhängig vom Aufruf emuliert *joe* vier weitere Editoren:

- *jrico* ist stark an die Handhabung des Editors des Pine-Mailingssystems, Pico, angelehnt.
- *jmacs* ist eine Emulation des GNU Emacs-Editors.
- *jstar* ist eine gut gelungene Imitation des WordStar-Editors.
- *rjoe* ist eine Version von *joe*, die es ausschließlich erlaubt, die in der Befehlszeile übergebenen Dateien zu editieren.

Mit *joe* verhält es sich wie mit vielen in diesem Buch vorgestellten Programmen: Eine vollständige Darstellung sämtlicher Optionen würde den Rahmen einer Einführung in dieses Programm sicherlich sprengen, so dass eine Einführung in die wesentlichen Funktionen an dieser Stelle ausreichen soll (muss).

### Syntax

Rufen Sie *joe* wie folgt auf:

```
joe [Option] [Datei1] [Datei2] ...
```

Wie Sie dieser Syntaxdarstellung entnehmen können, ist *joe* (wie ja die meisten UNIX-Programme) dazu in der Lage, gleichzeitig mehr als eine Datei zu bearbeiten; der Bildschirm wird dann in entsprechende Bildschirmbereiche geteilt, zwischen denen Sie beliebig wechseln können; auch ist beispielsweise ein Kopieren, Verschieben etc. zwischen den verschiedenen Bildschirmbereichen möglich.

Wenn Sie *joe* ohne Dateinamen als Befehlszeilenargument aufrufen, wird eine neue Datei erstellt. Das Gleiche geschieht, wenn Sie *joe* mit einem Dateinamen, den es noch nicht gibt, als Befehlszeilenargument aufrufen: Es wird dann eine neue Datei mit diesem Namen angelegt. Andernfalls lädt *joe* die von Ihnen spezifizierte(n) Datei(en), so dass sie zur weiteren Bearbeitung zur Verfügung stehen.



## Editoren



Nach dem Aufruf gibt sich *joe* zunächst bezüglich seines äußeren Erscheinungsbildes sehr spartanisch: Es werden keine großartigen Menüs, Statuszeilen, Rollbalken oder anderen Steuer- und Kontrollelemente angezeigt.

Um nun Einblick in die Funktionen von *joe* zu erhalten, können Sie mit der Steuersequenz **Strg+K, H** (halten Sie hierzu die Taste **Strg** gedrückt und geben Sie ein **K** ein, lassen diese beiden Tasten los und drücken **H**) Zugriff auf die Onlinehilfe erhalten. Diese gibt Ihnen Auskunft über weitere zur Verfügung stehende Sequenzen.

Abb. 4.6

Das Hilfenfenster von *joe*

```
Help Screen turn off with ':3h' more help with ESC, ('E,)

CURSOR GO TO BLOCK DELETE MISC EXIT
^B left ^F right ^U prev. screen ^K begin ^D char. ^X reformat ^KX save
^P up ^N down ^V next screen ^K end ^Y line ^I options ^C abort
^Z previous word ^A beg. of line ^W word ^R refresh ^KZ shell
^K next word ^E end of line ^KC copy ^O word(^@ insert FILE
SEARCH ^KU top of file ^KW File ^J >line SPELL ^KE edit
^KF find text ^KV end of file ^KY delete ^Z undo ^EH word ^KR insert
^KL find next ^KL to line No. ^K/ filter ^R redo ^EL file ^KD save
IH Unnamed Row 1 Col 1 9:49 Ctrl-K H for help
```

Beachten Sie, dass die Funktionsfähigkeit der Steuersequenzen entschieden von der von Ihnen verwendeten Terminal-Emulation abhängt!



Hinweis

Tabelle 4.24 gibt Ihnen einen Überblick über die zur Verfügung stehenden Steuersequenzen:

Die Basisfunktionen von *joe*



| Tastenkombination                 | Funktion                                                  |
|-----------------------------------|-----------------------------------------------------------|
| Sequenzen für die Cursorsteuerung |                                                           |
| <b>Strg+B</b>                     | bewegt den Cursor um jeweils eine Position nach links     |
| <b>Strg+F</b>                     | bewegt den Cursor um jeweils eine Position nach rechts    |
| <b>Strg+P</b>                     | bewegt den Cursor um jeweils eine Position nach oben      |
| <b>Strg+N</b>                     | bewegt den Cursor um jeweils eine Position nach unten     |
| <b>Strg+Z</b>                     | bewegt den Cursor an den Anfang des vorhergehenden Wortes |
| <b>Strg+X</b>                     | bewegt den Cursor an den Anfang des folgenden Wortes      |



## Benutzerumgebung

### Die Basisfunktionen von joe (Forts.)

| Tastenkombination   | Funktion                                                                                                                                        |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| Suchbefehle         |                                                                                                                                                 |
| <b>Strg+K, F</b>    | aktiviert die Suchfunktion (mehr dazu weiter unten)                                                                                             |
| <b>Strg+L</b>       | wiederholt die Suche                                                                                                                            |
| Blätterfunktion     |                                                                                                                                                 |
| <b>Strg+U</b>       | blättert zum vorherigen Bildschirminhalt zurück                                                                                                 |
| <b>Strg+V</b>       | blättert zum nächsten Bildschirminhalt                                                                                                          |
| <b>Strg+A</b>       | geht an den Anfang der Zeile                                                                                                                    |
| <b>Strg+E</b>       | geht an das Ende der Zeile                                                                                                                      |
| <b>Strg+K, U</b>    | geht an den Anfang der Datei                                                                                                                    |
| <b>Strg+K, V</b>    | geht an das Ende der Datei                                                                                                                      |
| <b>Strg+K, L</b>    | geht zu der von Ihnen zu spezifizierenden Zeile                                                                                                 |
| Blockfunktionen     |                                                                                                                                                 |
| <b>Strg+K, B</b>    | markiert den Anfang eines Blocks                                                                                                                |
| <b>Strg+K, K</b>    | markiert das Ende eines Blocks                                                                                                                  |
| <b>Strg+K, M</b>    | verschiebt den markierten Block an die aktuelle Cursorposition                                                                                  |
| <b>Strg+K, C</b>    | kopiert den markierten Block an die aktuelle Cursorposition                                                                                     |
| <b>Strg+K, W</b>    | kopiert den markierten Block in eine Datei                                                                                                      |
| <b>Strg+K, Y</b>    | löscht den markierten Block                                                                                                                     |
| <b>Strg+AltGr+B</b> | ermöglicht es, den Inhalt der Datei durch einen UNIX-Filter zu schicken, um das Ergebnis dann beispielsweise in einer neuen Datei abzuspeichern |
| Löschenfunktionen   |                                                                                                                                                 |
| <b>Strg+D</b>       | löscht das aktuelle Zeichen                                                                                                                     |
| <b>Strg+Y</b>       | löscht die aktuelle Zeile                                                                                                                       |
| <b>Strg+W</b>       | Wenn der Cursor auf dem ersten Zeichen eines Wortes steht, wird dieses Wort gelöscht                                                            |



## Editoren



### Die Basisfunktionen von joe (Forts.)

#### Tastenkombination

#### Funktion

**[Strg]+O** Wenn der Cursor auf dem letzten Zeichen eines Wortes steht, wird dieses Wort gelöscht

**[Strg]+J** löscht von der aktuellen Cursorposition bis zum Ende der Zeile

**[Strg]+[↑]+[-]** macht den letzten Befehl rückgängig

**[Strg]+[?]** wiederholt den letzten Befehl

#### Verschiedenes

**[Strg]+[K], [J]** macht eine Formatierung rückgängig

**[Strg]+[T]** aktiviert das Optionsmenü; hier stehen zur Verfügung:

*Overtime ON/OFF* – schaltet zwischen dem Einfüge- und dem Überschreibmodus hin und her

*Autoindent ON/OFF* – behält Einrückungen für die nächste Zeile bei (*ON* – nützlich für Programmierer) oder schaltet diese

aus (*OFF*)

*Word wrap ON/OFF* – bricht die Zeile automatisch bei Erreichen des rechten Bildschirmrandes um (*ON*)

**[Strg]+[R]** baut den Bildschirm neu auf

**[Strg]+[AltGr]+[Q]** fügt im Überschreibmodus ein Leerzeichen ein

#### Rechtschreibfunktionen

**[Esc]+[N]** überprüft die (englische ...) Rechtschreibung des aktuellen Wortes

**[Esc]+[L]** überprüft die (englische ...) Rechtschreibung der Datei

joe verlassen

**[Strg]+[K], [X]** speichert die aktuelle Datei und schließt das aktuelle Fenster; wenn nur eine Datei editiert wurde, ist dies gleichbedeutend mit dem Verlassen des Editors

## Benutzerumgebung

### Die Basisfunktionen von joe (Forts.)

| Tastenkombination       | Funktion                                                                                                                                                                                                                                             |
|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Strg+C</b>           | bricht den Editievorgang ab; Sie werden aufgefordert festzulegen, ob die von Ihnen vorgenommenen Änderungen verloren gehen sollen ( <b>Y</b> ), ob sie gespeichert werden sollen ( <b>N</b> ) oder ob Sie weiter editieren möchten ( <b>Strg+C</b> ) |
| <b>Strg+K, Z</b>        | kehrt temporär zur Shell zurück. Um wieder mit joe arbeiten zu können, können Sie diesen Prozess mit dem Befehl <i>fg</i> wieder in den Vordergrund verschieben (siehe „Job-Control“)                                                                |
| Dateifunktionen         |                                                                                                                                                                                                                                                      |
| <b>Strg+K, E</b>        | lädt eine weitere Datei in den Editor                                                                                                                                                                                                                |
| <b>Strg+K, R</b>        | fügt den Inhalt einer weiteren Datei an der aktuellen Cursorposition ein                                                                                                                                                                             |
| <b>Strg+K, D</b>        | speichert die aktuelle Datei; wenn die Datei noch keinen Namen besitzt, werden Sie an dieser Stelle aufgefordert, einen Dateinamen anzugeben                                                                                                         |
| Weitere Hilfefunktionen |                                                                                                                                                                                                                                                      |
| <b>Esc+.</b>            | zeigt die nächste Hilfeseite an                                                                                                                                                                                                                      |
| <b>Esc+,</b>            | zeigt die vorige Hilfeseite an                                                                                                                                                                                                                       |

Tabelle 4.24

### Die Suchfunktion **Strg+K, F**

Beachten Sie bei Ihrer Eingabe die von Ihnen verwendete Schreibweise!

Wie in Tabelle 4.24 angedeutet, wird joes Suchfunktion im Folgenden noch ein wenig detaillierter betrachtet:

Nach dem Aufruf dieses Befehls werden Sie zunächst aufgefordert, den Suchbegriff einzugeben:

Find (^C to abort) :

Nach Eingabe des Suchbegriffs werden Sie aufgefordert, das Suchverfahren genauer zu definieren:

(I)gnore (R)eplace (B)ackwards Bloc(K) NNN (^C to abort) :



## Editoren



Leider sind dem guten Mr. Allen die zur Verfügung stehenden Suchoptionen sprachlich ein wenig daneben gegangen:

- (l)gnore – ignoriert nicht etwa Ihren Suchbegriff, sondern platziert den Cursor nach erfolgreicher Suche hinter dem Begriff.
- (R)replace – ermöglicht die Eingabe einer Zeichenkette, die den Suchbegriff ersetzen soll. Wenn Sie diese mit bestätigt haben, stehen Ihnen weitere Optionen zur Verfügung, sofern der Suchbegriff gefunden werden konnte:  
*Replace (Y)es (N)o (R)est (B)ackup (^C to abort)?*  
(Y)es – ersetzt den aktuell gefundenen Begriff und sucht dann weiter.  
(N)o – ersetzt den aktuell gefundenen Begriff nicht und sucht dann weiter.  
(R)est – ersetzt sämtliche in der Datei auftretenden Entsprechungen des Suchbegriffs.  
(B)ackup – diese Option scheint zumindest in den von uns verwendeten Versionen von joe nicht zu funktionieren.
- (B)ackwards – sucht rückwärts in der Datei.
- Bloc(K) – sucht nur innerhalb des aktuellen Textblocks (ein Textblock wird mit abgeschlossen).

### Zusammenfassung

Wie bereits eingangs erwähnt, soll auf eine vollständige Darstellung der Funktionen von joe verzichtet werden, so dass eine detaillierte Darstellung der weiteren Hilfefunktionen (mit einer Ausnahme) hier nicht erfolgt:

Wenn Sie mehrere Dateien gleichzeitig in joe geöffnet haben, können Sie das aktive Fenster wechseln, indem Sie eine der beiden folgenden Tastenkombinationen verwenden:

- +, – wechselt in das nächste Editierfenster.
- +, – wechselt in das vorhergehende Editierfenster.

Mit den in diesem Abschnitt aufgeführten Funktionen sollten Sie das entsprechende Rüstzeug besitzen, eigenständig (Text-)Dateien zu erstellen und zu verändern.

Dies ist speziell für den dritten Teil des Buches (Programmierung) unerlässlich, so dass Sie sich ein wenig Zeit nehmen sollten, sich mit diesem (oder natürlich auch einem beliebigen anderen Editor, wenn Ihnen joe absolut nicht gefällt) vertraut zu machen.

Die Autoren sind jedoch der Meinung, dass dieser Editor für eine Großzahl der anfallenden Editieraufgaben ausreichend ist.

## Benutzerumgebung



### ***sed (Stream-Editor)***

Der letzte der Editoren, der Ihnen hier vorgestellt werden soll, arbeitet im Gegensatz zu den vorgenannten nicht interaktiv, d.h., Sie können nach dem Befehlsaufruf nicht mehr in den Editievorgang eingreifen. Dies scheint zunächst ein Widerspruch in sich zu sein. Eine Datei zu editieren, ohne sie dabei anschauen zu können, ist eine ungewohnte Vorstellung.

In der Tat, der Einsatz von *sed* beschränkt sich auf Fälle, in denen der Inhalt der Datei (mehr oder weniger) bekannt ist. Doch täuschen Sie sich nicht: Daraus ergeben sich eine Reihe von Anwendungsmöglichkeiten, die man sich in anderen Betriebssystemumgebungen nur wünschen kann.

- 
- Stream lässt sich frei mit „Datenstrom“ übersetzen
- Insbesondere bei Aufgabenstellungen, die eine Automatisierung beinhalten, zeigt der Stream-Editor *sed* seine Stärken:
    - gleichzeitige Bearbeitung mehrerer Dateien,
    - Einsatz in Pipelines,
    - Einsatz in nicht interaktiven Teilen von Shell-Skripten,
    - Bearbeitung besonders großer Dateien (hier war unserem Forscherdrang ein gewisser Riegel vorgeschoben).

Um Ihnen ein Bild von der Leistungsfähigkeit des *sed* zu geben, stellen Sie sich vor, Sie haben eine Reihe von Textdateien, die ein bestimmtes Programm erzeugt hat, und müssen nun diese Daten konvertieren, da ein zweites Programm sie verarbeiten soll. Da beide aber nicht den gleichen Zeichensatz verwenden, müssen eine Reihe von Zeichen in allen Dateien verändert werden.

Nun ja, Suchen und Ersetzen als Funktion ist in so vielen Programmen bekannt, dass ... doch halt, es sind jeweils vielleicht 10 Zeichen in 40 Dateien. Machen Sie sich ein Bild von dieser abwechslungsreichen Arbeit, und Sie werden begreifen, dass eine Konvertierung mindestens doppelt so lange dauert.

Mit einer einmalig angelegten, geschickt vorbereiteten Hilfsdatei, welche die Befehle zur Ersetzung der einzelnen Zeichen enthält, lässt sich dieser Vorgang zuverlässig und schnell durch ein einfaches Shell-Skript unter Verwendung des *sed* erledigen.

Der Aufruf des *sed* erfolgt mit:

```
sed[-option] [sed-Befehle] [Dateiliste]
```

Die *sed*-Befehle sind denen des *ed* äußerst ähnlich, auch wenn ihre Syntax aufgrund der unterschiedlichen Situation voneinander abweicht. Insbesondere die so genannten „Mnemos“, d.h. die Buchstabenkürzel, sind nahezu identisch.





Einige dieser Befehle werden in den folgenden Abschnitten als Beispiel besprochen. Wenn Sie die Struktur der Befehlsgebung erkannt haben, sollten Sie mithilfe der Informationen zu *ed* die nicht dargestellten elementaren Funktionen des *sed* trotzdem leicht erlernen können.

### Funktionsprinzip und Optionen von *sed*

Die Arbeitsweise von *sed* ist nun leicht erklärt. Der Editor liest die Daten der Dateien zeilenweise. Sind keine Dateien als Argumente angegeben, liest *sed* die Standardeingabe. Jede gelesene Zeile wird mit den genannten *sed*-Befehlen bearbeitet und in einen Puffer geschrieben. Der Inhalt dieses Puffers wird zuletzt am Bildschirm angezeigt.

Sollen sich die Befehle nicht auf alle gelesenen Zeilen beziehen, kann man ihnen eine Bereichsangabe voranstellen, so dass nur Zeilen bearbeitet werden, die sich in diesem Bereich befinden. Dies entspricht in Syntax und Verfahrensweise dem interaktiven Vorgehen in *ed*.

Hier lassen sich auch die beiden wichtigsten Optionen des *sed* erklären: Mit *-n* unterdrückt man die Standardausgabe (dies ist interessant in Verbindung mit dem Befehl *w*), mit *-f dateiname* kann man auf die Angabe von *sed*-Befehlen in der Eingabezeile verzichten. *sed* liest dann die Befehle aus der Datei *dateiname*.

### Ein einfaches Beispiel, der Befehl *a*

Für das folgende (zugegebenermaßen) einfache Beispiel soll eine simple Textdatei als Anschauungsmodell herangezogen werden.

```
$ cat > sedtest
Dies ist die erste Zeile
Dies ist die zweite Zeile
Dies ist die dritte Zeile
Dies ist die vierte Zeile
Dies ist die fünftete Zeile
Dies ist die sechste Zeile
[Strg]+D
$
```

Angenommen, Sie wollen zwischen die Zeilen 3 und 4 eine neue Zeile einfügen mit dem Inhalt:

Dies ist eine Zwischenzeile

Dann erinnern Sie sich an den *ed*-Befehl *a*, den man innerhalb dieses Editors dazu benutzt, einen Text nach der genannten (oder aktuellen) Zeile einzufügen.



## Benutzerumgebung

---



Der Ablauf wäre gewesen:

```
3a[←]
Dies ist eine Zwischenzeile[←]
[←]
```

Nicht viel anders stellt sich der Ablauf mit *sed* dar. Natürlich muss hier berücksichtigt werden, dass Sie nicht interaktiv arbeiten, sondern diesen Befehl beim Aufruf des *sed* angeben.

Eine Konsequenz dieser Tatsache ist, dass der Stream-Editor ergänzende Angaben zu einem Befehl meist in einer Extrazeile erwartet. Tatsächlich muss *sed* die beiden Teile des Befehls, *a* und *Dies ist eine Zwischenzeile*, durch einen Zeilenvorschub (*[←]*) getrennt erhalten.

Eine Besonderheit des *sed* ist noch, dass er selbst den Zeilenvorschub maskiert erhalten muss, d.h., vor jedem Zeilenumbruch – ob in der Befehlszeile oder in der durch *-f* spezifizierten Datei – muss ein \ stehen.

Da all die genannten Zeichen Metazeichen der Shell sind, müssen sie maskiert werden. Es empfiehlt sich immer, *sed*-Befehle mittels " ... " vor dem Zugriff durch die Shell zu schützen.

Aus dem soeben Dargestellten ergibt sich folgender Programmaufruf für die Erledigung der oben genannten Aufgabe:

```
$ sed "3a\[←]
> Dies ist eine Zwischenzeile[←]
> "_sedtest[←]
Dies ist die erste Zeile
Dies ist die zweite Zeile
Dies ist die dritte Zeile
Dies ist eine Zwischenzeile
Dies ist die vierte Zeile
Dies ist die fünftete Zeile
Dies ist die sechste Zeile
$
```

Der Aufruf erfolgt ohne Option, d.h., die *sed* unmittelbar folgende Angabe ist ein *sed*-Befehl, nämlich der Befehl *a*. Zunächst erfolgt die Maskierung mittels ", dann folgt der Befehl mit vorangestellter Bereichsangabe (Zeile 3).

---

○  
Testen Sie das Beispiel doch einmal ohne diesen Zeilenumbruch

Wie oben bereits dargestellt, muss nun ein maskierter Zeilenumbruch folgen. In der Befehlszeile hat dies die Konsequenz, dass die Shell den zweiten Prompt ausgibt, da sie das abschließende " vermisst. Sie geben also unverzagt den Rest des *sed*-Befehls ein, den einzufügenden Text. Der abschließende Zeilenumbruch ist in diesem Fall Bestandteil des Einfügetextes!





Die Maskierung muss schließlich abgeschlossen und ein Name der Quelldatei (hier *sedtest*) angegeben werden. Nach Bestätigung dieser komplexen Eingabezeile erledigt *sed* seine Arbeit und gibt den geänderten Text am Bildschirm aus.

### Kommandodateien für *sed*

Will man die *sed*-Befehle, etwa weil es sehr viele sind, in einer Datei zusammenfassen, um dann direkt den Befehl anzuwenden, sind die schon geschilderten Randbedingungen zu beachten.

Gut, Ergänzungen zu Befehlen sind in einer Extrazeile unterzubringen. Wichtig ist aber auch, dass die Zeile, die den notwendigen Zeilenumbruch enthält, durch \ abgeschlossen ist. Erweitern Sie das Beispiel des letzten Abschnitts, um das Problem besser verstehen zu können.

Ziel ist es zunächst also, in die ursprüngliche Datei eine Ergänzungszeile nach Zeile 3 einzufügen. Zusätzlich soll nun die Zeile 2 gelöscht werden.

Der Löschbefehl des *ed* ist bekanntlich *d* (*delete*). Was liegt also näher, als folgende Befehlsfolge für *sed* festzulegen:

```
$ cat > sfile←
3a\←
Dies ist eine Zwischenzeile←
2d←
Strg+D
$
```

In der ersten Zeile der Datei steht der Befehl zur Ergänzung der Quelldatei, es folgt der Ergänzungstext. Dann schließt sich noch der Befehl zum Löschen der zweiten Zeile an.

Nun lässt sich mithilfe dieser Datei der Aufruf wie folgt vereinfachen:

```
sed -f sfile sedtest
```

Der Editor liest jetzt seine Befehle aus der Datei *sfile*, die konstruiert wurde, wie oben beschrieben. Die Quelle ist die Datei *sedtest*; die Bildschirmausgabe des Befehls wird sein:

```
Dies ist die erste Zeile
Dies ist die dritte Zeile
Dies ist eine Zwischenzeile
Dies ist die vierte Zeile
Dies ist die fünftete Zeile
Dies ist die sechste Zeile
```

Die zweite Zeile ist gelöscht, eine neue Zeile nach der bisher dritten eingefügt. Das ist das gewünschte Ergebnis.

## Benutzerumgebung



### Überblick über sed-Befehle

Entnehmen Sie Tabelle 4.25 die wichtigsten Befehle des *sed*. Mögliche Bereichsangaben sind durch [b] gekennzeichnet. Ohne Bereichsangabe ist jede einzelne Zeile betroffen.

#### Einige Befehle des *sed*

| Befehl        | Bedeutung                                                                                                                                     |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| [b]a\Ütext    | fügt nach der Zeile text ein                                                                                                                  |
| [b]i\Ütext    | fügt vor der Zeile text ein                                                                                                                   |
| [b]c\Ütext    | ersetzt den Bereich durch text                                                                                                                |
| [b]d          | löscht den Bereich                                                                                                                            |
| [b]q          | Ausgabe der Datei bis zum Beginn des Bereichs                                                                                                 |
| [b]w datei    | speichert den angegebenen Bereich in der Datei                                                                                                |
| [b]!befehl    | Befehl wird nur auf Zeilen angewandt, die nicht dem Bereich angehören                                                                         |
| [b]=          | Ausgabe einer Zeilennummer vor der jeweiligen Zeile                                                                                           |
| [b]p          | Ausgabe des Bereichs am Bildschirm                                                                                                            |
| [b]h          | Text der jeweiligen Zeile des Bereichs wird in einen Puffer geschrieben – jeder Schreibvorgang löscht dabei den bisherigen Inhalt des Puffers |
| [b]g          | ersetzt den genannten Bereich durch den Inhalt des Puffers                                                                                    |
| [b]H          | Text des Bereichs wird an den Inhalt des Puffers angehängt                                                                                    |
| [b]G          | Inhalt des Puffers wird an das Ende des Bereichs angehängt                                                                                    |
| [b]y/t1/t2/   | tauscht im genannten Bereich den Text t1 gegen den Text t2 aus                                                                                |
| [b]x          | Austausch des Pufferinhalts mit dem genannten Bereich                                                                                         |
| [b]r datei    | Inhalt der datei wird nach dem Bereich in die Ausgabe von <i>sed</i> eingefügt                                                                |
| {Übf1Ü..bfnÜ} | Befehlsverkettung der Befehle bf1 bis bfn                                                                                                     |



## Linux-Werkzeuge



### Einige Befehle des sed (Forts.)



| Befehl        | Bedeutung                                                                                                                                                                                                             |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [b]s/t1/t2/bf | ersetzt im genannten Bereich den Text <i>t1</i> gegen <i>t2</i> unter Beachtung des Befehls <i>bf</i> (z.B. <i>g</i> für globale Ersetzung, <i>nr</i> für das nr. Vorkommen), Suchmuster mit Metazeichen sind erlaubt |

Tabelle 4.25



## Linux-Werkzeuge

Neben den Editoren verfügt Linux über eine ganze Reihe leistungsstarker Tools, welche die Arbeit sowohl des gewöhnlichen Benutzers als auch des Systemverwalters erleichtern können. Beispielhaft sollen hier der Midnight-Commander sowie die beiden traditionellen Tools *awk* und *bc* vorgestellt werden. Darüber hinaus erhalten Sie einen – mit Sicherheit unvollständigen – Überblick über weitere Linux-Werkzeuge.

Das Vorhandensein dieser Tools hängt entscheidend von der gewählten Distribution ab.



Hinweis

### Der Midnight Commander

Der Midnight Commander *mc* ist ein Datei-/Verzeichnismanager für UNIX-ähnliche Betriebssysteme. Beziiglich Funktionalität und Gestaltung orientiert er sich stark an dem in der Win/DOS-Welt sehr populären Norton Commander:

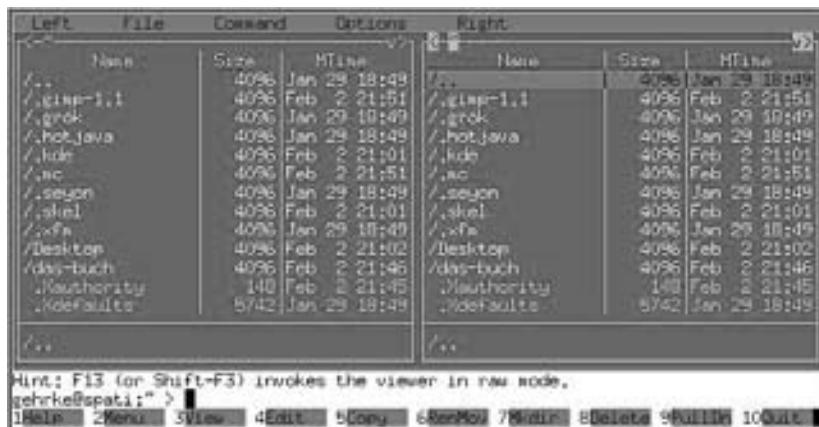


Abb. 4.7

Der Midnight Commander



## Benutzerumgebung

---



Eine vollständige Darstellung dieses Programms mit all seinen Leistungsmerkmalen würde hier sicherlich zu weit führen (die *man pages* zum *mc* sind schon 46 Seiten lang ...), doch sollen Ihnen die wichtigsten Funktionen kurz beschrieben werden:

Wie Sie der *Abbildung 4.7* entnehmen können, besitzt der *mc* standardmäßig im Wesentlichen fünf Bildschirmelemente (wenn Sie sich näher mit diesem Programm beschäftigen werden, werden Sie sehr schnell die Möglichkeiten der individuellen Anpassung kennen lernen):

- eine Menüleiste,
- zwei Fenster für die Darstellung von Verzeichnisinhalten,
- eine Eingabezeile,
- eine Zeile mit Hilfsinformationen.

Diese sollen (unterschiedlich detailliert) besprochen werden.

### Die Eingabezeile

Sie können die Eingabezeile (fast) wie die „normale“ bash-Eingabezeile verwenden, d.h. wie gewohnt Befehle eingeben und mit bestätigen.

Tipp



Um den *mc* kurzfristig auszublenden (wenn Sie beispielsweise die Bildschirmmeldung zu einer von Ihnen eingegebenen Befehlszeile anschauen möchten), ohne ihn jedoch vollständig zu verlassen, können Sie die Tastenkombination + drücken. Mit derselben Tastenkombination stellen Sie den *mc* auch wieder in den Vordergrund.

Warum aber heißt es, Sie könnten diese Eingabezeile „fast“ so verwenden, wie Sie es gewohnt sind? Nun: Wie Sie in dem Abschnitt über „Job-Control“ gelesen haben, startet ein Prozess einen Kindprozess. Wenn Sie nun aus dem Midnight Commander heraus beispielsweise eine Umgebungsvariable, zum Beispiel *TERM*, mit dem Befehl

```
TERM=vt220
```

Ändern (mit *TERM* definieren Sie Ihre Terminal-Emulation), so gilt der Wert dieser neuen Umgebungsvariablen im *Kindprozess* des *mc*. Lassen Sie sich nun die Umgebungsvariable *TERM* Ihrer aktuellen Shell mit

```
echo $TERM
```

anzeigen (stellen Sie, um die Bildschirmausgabe betrachten zu können, den *mc* mit der oben beschriebenen Methode in den Hintergrund), werden Sie feststellen, dass sich daran nichts geändert hat (was in der Regel auch so gewollt ist – aber behalten Sie diese Problematik im Hinterkopf ...).





### Die Verzeichnisfenster

Diese Fenster dienen dazu, vereinfacht Verzeichnisse sowie Verzeichnisinhalte und Dateien zu manipulieren. Folgende *Tabelle 4.26* zeigt die wichtigsten Tasten mit den ihnen zugeordneten Funktionen zur Navigation innerhalb der Fenster und zwischen den Fenstern sowie zur Markierung von Objekten.

**Tasten zur Navigation und zur Markierung**

| Taste  | Funktion                                                                                                                                                                     |
|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ↓      | bewegt den Cursor um einen Verzeichniseintrag nach unten                                                                                                                     |
| ↑      | bewegt den Cursor um einen Verzeichniseintrag nach oben                                                                                                                      |
| Bild ↓ | bewegt den Cursor um eine Fensterseite nach unten                                                                                                                            |
| Bild ↑ | bewegt den Cursor um eine Fensterseite nach oben                                                                                                                             |
| Pos 1  | bewegt den Cursor an den Anfang der Verzeichnisliste                                                                                                                         |
| Ende   | bewegt den Cursor an das Ende der Verzeichnisliste                                                                                                                           |
| ↹      | wechselt zwischen den beiden Fenstern                                                                                                                                        |
| Strg+U | wie ⏪                                                                                                                                                                        |
| ←      | wechselt in ein Verzeichnis                                                                                                                                                  |
| Einfg  | markiert einen Verzeichniseintrag                                                                                                                                            |
| Entf   | hebt die Markierung eines Verzeichniseintrags wieder auf                                                                                                                     |
| +      | ermöglicht es, einen Filter zur Auswahl von Verzeichniseinträgen zu definieren (die Standardeinstellung ist *, d.h., alle Elemente werden markiert)                          |
| -      | ermöglicht es, einen Filter zur Aufhebung der Auswahl von Verzeichniseinträgen zu definieren (die Standardeinstellung ist *, d.h., alle Markierungen werden wieder entfernt) |

*Tabelle 4.26*

### Die Hilfsinformationen

Am unteren Bildschirmrand sehen Sie eine Übersicht der zehn Funktionstasten mit den ihnen zugewiesenen Funktionen.



## Benutzerumgebung

### Die Funktionstastenbelegung des Midnight Commanders

| Taste | Funktion                                                                                                                                                                                                     |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [F1]  | aktiviert die Onlinehilfe                                                                                                                                                                                    |
| [F2]  | öffnet ein (vom Benutzer anpassbares) Menü                                                                                                                                                                   |
| [F3]  | zeigt den Inhalt der aktuellen Datei an bzw. wechselt in diese, wenn es sich um ein Verzeichnis handelt. Die näheren Funktionen werden weiter unten beschrieben.                                             |
| [F4]  | öffnet die aktuelle Datei zum Editieren. Hierzu wird standardmäßig vi aufgerufen. Möchten Sie einen anderen Editor als Standardeditor verwenden, können Sie diesen z.B. in Ihrer Datei .profile definieren.  |
| [F5]  | kopiert die aktuelle Datei bzw. die markierten Dateien des aktiven Fensters in das aktuelle Verzeichnis des zweiten Fensters. Die näheren Funktionen werden weiter unten beschrieben.                        |
| [F6]  | verschiebt die aktuelle Datei bzw. die markierten Dateien des aktiven Fensters in das aktuelle Verzeichnis des zweiten Fensters. Die näheren Funktionen werden weiter unten beschrieben.                     |
| [F7]  | legt im Verzeichnis des aktiven Fensters ein neues Unterverzeichnis an                                                                                                                                       |
| [F8]  | löscht die aktuelle Datei (das aktuelle Verzeichnis) bzw. die markierten Dateien (Verzeichnisse) aus dem aktuellen Verzeichnis des aktiven Fensters. Die näheren Funktionen werden weiter unten beschrieben. |
| [F9]  | öffnet die Menüzeile                                                                                                                                                                                         |
| [F10] | beendet den Midnight Commander                                                                                                                                                                               |

Tabelle 4.27

Auf einige dieser Funktionstasten soll hier noch ein wenig detaillierter eingegangen werden.

#### Die Funktionstaste [F3]

Wenn Sie die Funktionstaste [F3] drücken, wird der Inhalt der aktuellen Datei des aktiven Fensters in einen (internen) Dateibetrachter (Viewer) des Midnight Commanders geladen.

#### Hinweis

Beachten Sie, dass es Ihnen diese Funktion nicht erlaubt, die Datei zu editieren – hierzu müssen Sie mit [F4] den Editor öffnen.



## Linux-Werkzeuge



Dieser bietet die in *Tabelle 4.28* beschriebenen weiteren Funktionen:

### Die Funktionstastenbelegung des internen Dateibetrachters

Taste      Funktion

|       |                                                                                                                                                                                                                                                                                         |
|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [F1]  | zeigt die Onlinehilfe des Viewers an                                                                                                                                                                                                                                                    |
| .     | schaltet (im ASCII-Modus; siehe [F4]) Umbrüche ein/aus                                                                                                                                                                                                                                  |
| [F3]  | beendet den Viewer                                                                                                                                                                                                                                                                      |
| [F4]  | schaltet zwischen hexadezimaler und ASCII-Anzeige hin und her                                                                                                                                                                                                                           |
| [F5]  | ermöglicht einen direkten „Sprung“ in die von Ihnen angegebene Zeile                                                                                                                                                                                                                    |
| [F6]  | ermöglicht (im ASCII-Modus; siehe [F4]) die Suche nach einem regulären Ausdruck                                                                                                                                                                                                         |
| [F7]  | schaltet die Suchfunktion nach Zeichenketten ein/aus. In Abhängigkeit davon, ob Sie im ASCII- oder hexadezimalen Modus arbeiten, können Sie hier einen String oder ein Hex-Muster eingeben.                                                                                             |
| [F8]  | schaltet zwischen dem „Raw“- und dem „Parsed“-Modus hin und her. Wenn in der Datei <i>mc.ext</i> ein Filterprogramm definiert wurde, wird die Datei entsprechend im „Parsed“-Modus von diesem Programm gefiltert und in ihrem eigenen Format angezeigt.                                 |
| [F9]  | schaltet zwischen formatierter und unformatierter Ausgabe hin und her. Wenn Sie im „Format“-Modus arbeiten, können bestimmte Zeichensequenzen als die Formatierungsmerkmale „fett“ und „unterstrichen“ oder als farbige Hervorhebungen interpretiert und entsprechend angezeigt werden. |
| [F10] | wie [F3]                                                                                                                                                                                                                                                                                |

*Tabelle 4.28*



## Benutzerumgebung

Abb. 4.8

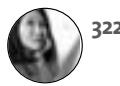
Dialogfenster der Funktion Kopieren/Verschieben



Entnehmen Sie die Bedeutung der einzelnen Dialogelemente der Tabelle 4.29.

### Die Dialogelemente der Kopier- und Verschiebefunktion des Midnight Commanders

| Dialogelement                           | Bedeutung                                                                                                                                                                                                                                                                |
|-----------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Copy xxx files with source mask:</i> | symbolisiert die markierte(n) Datei(en)                                                                                                                                                                                                                                  |
| <i>to:</i>                              | enthält den Namen des Zielverzeichnisses (dieses ist standardmäßig das Verzeichnis des Fensters, welches nicht aktiv ist)                                                                                                                                                |
| <i>using shell Patterns</i>             | Wenn diese Option eingeschaltet ist, können Sie weitere individuelle Einstellungen in der Maskierung der Quelldateien vornehmen (verwenden Sie hierzu, wie auf Befehlszeilebene, die Jokerzeichen * und ?)                                                               |
| <i>follow symlinks</i>                  | Diese Option erlaubt es, festzulegen, ob symbolische Links im Zielverzeichnis übernommen werden sollen                                                                                                                                                                   |
| <i>Dive into subdir if exists</i>       | Wenn das Zielverzeichnis bereits existiert, wird ein weiteres Verzeichnis gleichen Namens unterhalb dieses Verzeichnisses angelegt (Standard). Schalten Sie diese Option aus, wird der Inhalt des Quellverzeichnisses in das bereits bestehende Zielverzeichnis kopiert. |



## Linux-Werkzeuge



### Die Dialogelemente der Kopier- und Verschiebefunktion des Midnight Commanders (Forts.)



| Dialogelement             | Bedeutung                                                                                                                                                                                                                                                           |
|---------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>preserve UIDs/GIDs</i> | Wenn Sie als root arbeiten, können Sie durch Einschalten dieser Option festlegen, ob die User-/Gruppenrechte für die kopierten Dateien nach dem Kopieren erhalten bleiben sollen. Andernfalls fallen sie in den Rechtebereich des Besitzers des Zielverzeichnisses. |

Tabelle 4.29

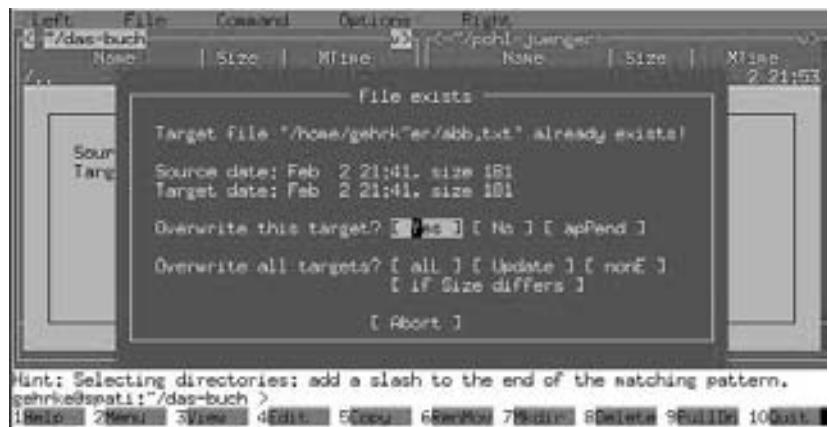


Sie können zwischen den einzelnen Dialogelementen mit der Taste und mit den Cursortasten wechseln; eine Aktivierung/Deaktivierung von Optionen erfolgt mit der Taste .

Sollte eine zu kopierende/zu verschiebende Datei bereits im Zielverzeichnis vorhanden sein, wird das Dialogfenster aus Abbildung 4.9 angezeigt.

Abb. 4.9

Die Sicherheitsabfrage  
der Funktion Kopieren/  
Verschieben



Entnehmen Sie die Bedeutung der einzelnen Elemente Tabelle 4.30.

### Die Sicherheitsabfrage der Funktion Kopieren/Verschieben



| Dialogelement                 | Bedeutung                                   |
|-------------------------------|---------------------------------------------|
| <i>Overwrite this target?</i> | Diese Datei überschreiben?                  |
| <i>Yes</i>                    | überschreibt die bereits existierende Datei |



## Benutzerumgebung

### Die Sicherheitsabfrage der Funktion Kopieren/Verschieben (Forts.)

| Dialogelement          | Bedeutung                                                 |
|------------------------|-----------------------------------------------------------|
| No                     | überschreibt die bereits existierende Datei nicht         |
| apPend                 | hängt die Quelldatei an die bereits existierende Datei an |
| Overwrite all targets? | Alle Dateien überschreiben?                               |
| All                    | überschreibt alle bereits existierenden Dateien           |
| Update                 | überschreibt nur ältere Versionen                         |
| nonE                   | bricht Kopier-/Verschiebevorgang ab                       |

Tabelle 4.30

### Die Menüleiste

Auf sämtliche Funktionen der Menüleiste einzugehen, würde eindeutig den Rahmen dieser Einführung in die Handhabung des Commanders sprengen. Viele der in ihr enthaltenen Funktionen sind ohnehin durch die Beschreibungen der (Funktions-)Tasten bereits erklärt, so dass (mit einer Ausnahme) an dieser Stelle eine bloße Kurzbeschreibung der Menüpunkte erfolgen soll und Sie im Weiteren auf die Erklärungen der Onlinehilfe zurückgreifen sollten:

Das Öffnen der Menüs mit **[F9]** bietet folgende Menüpunkte:

- Left – enthält Optionen zur Anpassung des linken Verzeichnisfensters.
- File – bietet Optionen zur Datei-/Verzeichnismanipulation. Viele dieser Optionen können Sie durch Betätigen der oben beschriebenen Funktionstasten direkt nutzen.
- Command – enthält Steuerungs- und weitere Dateibefehle.
- Option – bietet Optionen zur Konfiguration des Commanders.
- Right – enthält Optionen zur Anpassung des rechten Verzeichnis-Fensters.

Die Ausnahme, die hier gemacht werden soll, ist, auch wenn damit an dieser Stelle ein wenig vorgegriffen wird, eine kurze Darstellung der Option **FTP link** in den Menüs **Left** und **Right**.

#### Hinweis



Lesen Sie bitte zunächst die *Kapitel 13 Linux im Netzwerk* und *Kapitel 14 Mit Linux ins Internet*, wenn Sie diese Funktion nutzen möchten, Ihr Rechner aber noch keinen Netzwerzugang hat und Ihnen FTP noch nichts sagt!





### FTP mit dem Midnight Commander

Sie können den Midnight Commander dazu verwenden, komfortabel einen FTP-Datentransfer von einem entfernten Host auf Ihre lokale Maschine durchzuführen.

Öffnen Sie mit [F9] das Befehlsmenü, wählen Sie den Menüpunkt Left oder Right (in Abhängigkeit davon, in welchem Fenster der Inhalt des entfernten Hosts angezeigt werden soll), und wählen Sie dort die Option FTP link.

Im folgenden Dialogfenster werden Sie aufgefordert, den Namen des entfernten Hosts einzugeben. Verwenden Sie hierbei die Syntax:

Username@Host

Geben Sie also z. B. das ein, was Abbildung 4.10 zeigt.

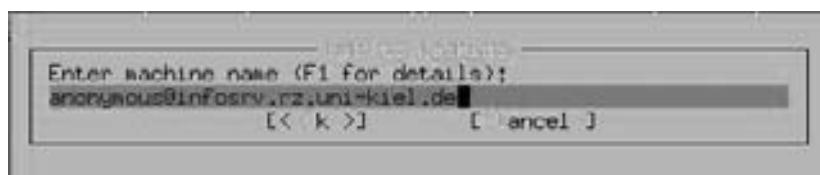


Abb. 4.10

Aufbau einer Verbindung mit einem FTP-Server

Nach der Eingabe eines Passworts steht Ihnen nun der entfernte Host im Midnight Commander zur Verfügung, als sei dieser Bestandteil des Verzeichnisbaums Ihres Linux-Rechners, so dass Sie von ihm Dateien übertragen können, als wenn es sich um ganz normale Kopiervorgänge innerhalb Ihres Dateibaums handeln würde.

Es muss allerdings darauf hingewiesen werden, dass sich diese Funktion des Midnight Commanders, auch wenn sie sich an Komfort kaum noch überbieten lässt, leider nicht durch hundertprozentige Stabilität auszeichnet, so dass es häufiger einmal zu Hängern des Commanders kommt; reproduzierbare Gründe hierfür konnten wir leider bisher nicht isolieren, so dass wir Ihnen nur den Tipp auf den Weg geben können: Versuch macht kluch ...

### awk – ein Mustererkennungsprogramm

*awk* ist ein Mustererkennungsprogramm, das in seiner einfachsten Form ähnlich wie *grep* arbeitet. So liefern die beiden Eingaben

```
who | grep root
```

und

```
who | awk "/root/"
```

dasselbe Ergebnis: eine Liste aller Terminals, an denen *root* eingeloggt ist.

Die allgemeine Syntax ist ähnlich der des Stream-Editors *sed*:

## Benutzerumgebung

---



```
awk [-optionen] [awk-aktionen] [dateiliste]
```

Eine der wichtigsten Optionen ist dabei *-f dateiname*, wodurch Sie die *awk-aktionen* von der Befehlszeile in eine Datei verlagern können. Auch hier also eine Analogie zu *sed*.

*awk* ist dahingehend eine Erweiterung von *grep*, dass dieses Programm wesentlich mehr Operationen ausführen kann als die Ausgabe der Zeilen, die ein angegebenes Textmuster enthalten. Basis ist aber immer die Suche nach eben solchen Mustern.

Die Möglichkeit, einzelne „Felder“ einer Zeile zu erkennen, nummerische Felder zu bearbeiten und Ausgaben vor und nach der Bearbeitung der Datei zu machen, veranlasst eine Reihe von Autoren, *awk* als „Listengenerator“ zu bezeichnen.

Der Name des Programms ist im Übrigen aus den Anfangsbuchstaben der Namen seiner Programmierer gebildet – ein nicht ganz ungewöhnliches Verfahren in der UNIX-Gemeinde. Es sind dies **Aho**, **Weinberger** und **Kernighan**, Letzterer einer der Entwickler von UNIX und vor allem der Programmiersprache C. Dass Brian Kernighan an der Entwicklung von *awk* maßgeblichen Anteil hat, schlägt sich nieder in der Tatsache, dass viele der Elemente von *awk* sehr stark an die Programmiersprache C erinnern.

*awk* bzw. die GNU-Variante *gawk* gehören standardmäßig zu einer Linux-Distribution.

### Suchmuster

Die *awk-aktionen* setzen sich in der Regel aus einem Suchmuster und der eigentlichen Aktion zusammen, die *awk* durchführen soll, falls es eine Zeile findet, in der das Suchmuster enthalten ist.

Da Suchmuster und Aktion zusammen häufig sehr lang sind und sich somit über mehrere Zeilen erstrecken und da außerdem immer wieder Shell-Metazeichen auftreten, empfiehlt es sich, wenn man diese Angaben in der Befehlszeile macht, sie durch einfache Hochkommata „...“ zu maskieren.

Die einfachste Aktion ist die Ausgabe der Zeile. Dies geschieht über das Schlüsselwort *print*. Ein recht einfaches Beispiel soll Ihnen den Gebrauch von *awk* nun demonstrieren.

```
awk "/root/ {print}" /etc/passwd
```

sucht in der Datei */etc/passwd* nach Zeilen, die den Text *root* enthalten, und gibt diese aus.

Wird die eigentliche Aktion nicht angegeben, so nimmt *awk* an, die gesamte Zeile soll gedruckt werden. In diesem Sinne entspricht

```
awk "/root/" /etc/passwd
```





dem vorhergehenden Beispiel. Das erste Beispiel dieses Abschnitts müsste andererseits vollständig lauten:

```
who | awk "/root/ {print}"
```

Wird die Aktion angegeben, muss sie – in guter C-Tradition – in geschweifte Klammern gesetzt werden.

Das Suchmuster selbst ist in / ... / einzuschließen, was Sie ja schon von den Suchroutinen der verschiedenen Editoren kennen. Natürlich lassen sich auch bei *awk* Suchmuster mithilfe von Metazeichen bilden. In *Tabelle 4.21* weiter oben in diesem Kapitel finden Sie die wichtigsten dieser Metazeichen zur Bildung regulärer Ausdrücke. Hier einige Beispiele:

```
awk "/bash$/" /etc/passwd
```

sucht in der Passworddatei nach Benutzern, welche die bash als Standard-Shell haben (die Standard-Shell ist der letzte Eintrag der jeweiligen Zeile!).

```
awk "/^s/" /etc/passwd
```

sucht nach Benutzern, deren Benutzername (erster Eintrag und damit Beginn der Zeile) mit s beginnt.

```
ls -l | awk "/^d/"
```

zeigt alle Unterverzeichnisse des aktuellen Verzeichnisses an (Zeile beginnt mit d).

Wird im Übrigen kein Suchmuster angegeben, wird die genannte Aktion für alle Zeilen der Datei vorgenommen.



Hier ersetzt wieder einmal die Standardeingabe den Dateinamen!

### Felder einer Zeile

Die vorhergehenden Beispiele lassen sich ausnahmslos mithilfe von *grep* lösen. Ein erster entscheidender Vorteil von *awk* ist die Möglichkeit, die einzelnen Felder einer Zeile zu analysieren. Als Beispieldatei dient im Folgenden eine einfache Telefonliste (angenommen, die Datei heißt *telefon*):

|                  |           |        |
|------------------|-----------|--------|
| Christine Maier  | Kiel      | 786543 |
| Ernst Regenbogen | Dortmund  | 556677 |
| Dora Halber      | Hannover  | 667213 |
| Ernst Maier      | Dortmund  | 972307 |
| Vera Meier       | Kiel      | 767676 |
| Mathias Breiter  | Stuttgart | 613456 |
| Carlo Traub      | Kiel      | 743211 |
| Erni Bert        | Kiel      | 840931 |

Jede Zeile dieser Datei enthält vier Felder: den Vornamen, den Namen, den Ort und die Telefonnummer. Als Trennzeichen für die einzelnen Felder akzeptiert *awk* das Leer- und das Tabulatorzeichen.



## Benutzerumgebung

---



Die Felder lassen sich ähnlich zu den Positionsparametern von Shell-Skripten mittels `$1`, `$2`, `$3` usw. ansprechen. Eine Beschränkung auf neun Parameter existiert nicht. Mit über 200 dieser Parameter sollte sich normalerweise jede Zeile einer Datei in ihre einzelnen Bestandteile zerlegen ...

`$1` steht dabei für das erste Feld der Zeile, in diesem Beispiel den Vornamen, `$2` für das zweite, den Namen, `$3` für das dritte, den Ort, `$4` für das vierte, die Telefonnummer, und so fort.

So liefert

```
awk "/Kiel/ {print $1 $2 $4}" telefon
```

als Ergebnis eine Liste der Kieler Personen des Telefonverzeichnisses, ohne aber noch einmal den Ort in der Liste anzugeben.

```
Christine Maier 786543
Vera Meier 767676
Carlo Traub 743211
Erni Bert 840931
```

Aber auch im Zusammenhang mit den Suchmustern lassen sich die Parameter geschickt einsetzen. Mithilfe der beiden Operatoren `~` und `!~` lassen sich gezielt Suchmuster für Felder festlegen. So bedeutet zum Beispiel

```
$1~/a/
```

dass nach Zeilen gesucht werden soll, in denen das erste Feld den Buchstaben *a* enthält.

```
$1!~/a/
```

bedeutet Suche nach solchen Zeilen, in denen das erste Feld *nicht* den Buchstaben *a* enthält.

Das Ausrufezeichen `!` steht wie in der C-Programmierung für die Negation.

### Hinweis



Ein zusammenhängendes Beispiel:

```
awk "$2~/^M/" telefon
```

sucht nach allen Personen, deren Nachnamen (zweites Feld) mit *M* beginnen.

## Formatierte Ausgabe mit `printf`

Anstelle der Aktion `print` lässt sich die Ausgabe mit `printf` durchführen. `printf` lässt für die Ausgabe wesentlich mehr Möglichkeiten zu als `print`. Sie ist sehr stark an die gleichnamige Funktion aus der C-Standardbibliothek angelehnt.





Die Syntax von *printf* lautet:

```
printf "formatstring", ausgabeliste
```

*ausgabeliste* ist eine durch Kommata getrennte Liste von Objekten, die ausgegeben werden soll, in der Regel Text, Zahlen oder Positionsparameter.

Der *formatstring* enthält neben normalem Text Platzhalter für die einzelnen Elemente der Ausgabeliste. Dabei kann über die Platzhalter eine Ausgabeformatierung erfolgen. Das folgende Beispiel soll das Wirkungsprinzip von *printf* erläutern:

```
$ awk "/Hannover/ {printf "Name: %s Telefonnummer: %s",
$2, $4}" telefon
Name: Halber Telefonnummer: 667213
$
```

*awk* sucht nach Personen aus Hannover. Ausgegeben werden sollen der Name (zweites Feld der Zeile) und die Telefonnummer (viertes Feld der Zeile). Dabei wird dem jeweiligen Feld ein entsprechender Text vorangestellt. Platzhalter innerhalb von *formatstring* ist dabei jeweils *%s*.

Ausgegeben wird also pro gefundener Zeile der *formatstring*, in den die Elemente der *ausgabeliste* integriert werden, indem jedem Platzhalter ein Element zugewiesen wird. Der erste Platzhalter nimmt das 2. Feld der Zeile (*\$2*) auf, der zweite Platzhalter das vierte Feld (*\$4*).

*%s* reserviert Platz für Textobjekte (*s* = String). Darüber hinaus lassen sich unter anderem die Platzhalter aus *Tabelle 4.31* benutzen.

### Die wichtigsten Platzhalter im Formatstring von *printf*



Platzhalter für

|               |                                                                                     |
|---------------|-------------------------------------------------------------------------------------|
| <i>%d, %i</i> | Werte vom Datentyp <i>int</i> , d.h. ganze Zahlen                                   |
| <i>%u</i>     | Werte vom Datentyp <i>unsigned int</i> , d.h. ganze Zahlen ohne Vorzeichen          |
| <i>%x, %X</i> | Werte vom Datentyp <i>int</i> bzw. <i>unsigned int</i> in hexadezimaler Darstellung |
| <i>%o</i>     | Werte vom Datentyp <i>int</i> bzw. <i>unsigned int</i> in oktaler Darstellung       |
| <i>%f</i>     | Werte vom Datentyp <i>float</i> , d.h. Gleitkommazahlen (Dezimalzahlen)             |
| <i>%e</i>     | Werte vom Datentyp <i>float</i> in exponentieller Darstellung                       |
| <i>%g</i>     | wählt das kürzere Ergebnis der Formate <i>%e</i> und <i>%f</i>                      |



## Benutzerumgebung

### Die wichtigsten Platzhalter im Formatstring von *printf* (Forts.)

| Platzhalter | für                                              |
|-------------|--------------------------------------------------|
| %s          | Texte                                            |
| %c          | einzelne Zeichen, Werte vom Datentyp <i>char</i> |

Tabelle 4.31

Der Clou dieser Platzhalter ist es aber nun, dass sich über sie die Ausgabe der jeweiligen Elemente genau steuern lässt. Dies hat entscheidende Vorteile bei der Erstellung von Tabellen. Nach dem %-Zeichen lassen sich verschiedene Angaben für diese Gestaltung unterbringen.

Eine ganze Zahl bedeutet, dass *printf* auf dem Bildschirm (als Standardausgabegerät) für die Ausgabe des betreffenden Objekts entsprechend viele Zeichen reserviert und es rechtsbündig hineinschreibt.

```
printf " |%10d| ", 12345
```

bewirkt die Ausgabe:

```
| 12345 |
```

Für die Zahl 12345 werden 10 Stellen für die Ausgabe reserviert. Da sie rechtsbündig in diesen reservierten Bereich geschrieben wird, gehen ihr bei der Ausgabe also fünf Leerzeichen voran. Um dies einigermaßen deutlich zu machen, wird hier vor und nach der Ausgabe der Zahl jeweils das Zeichen | gezeigt.

Eine Gleitkommazahl ist – vereinfacht ausgedrückt – eine Zahl mit Nachkommastellen, eine nicht ganze Zahl also.

Wird der Zahl beim Platzhalter ein Minuszeichen vorangestellt, wird der Wert linksbündig in den reservierten Platz geschrieben. Lässt man der Zahl, getrennt durch einen Punkt, eine zweite folgen und ist der auszugebende Wert eine Gleitkommazahl, so gibt diese zweite Zahl an, wie viele Nachkommastellen des fraglichen Wertes ausgegeben werden sollen.

Innerhalb von *formatstring*, aber auch innerhalb jedes anderen Textes, den *printf* oder *print* ausgeben soll, lassen sich auch eine Reihe von Sonderzeichen durch einfache Ersatzdarstellungen unterbringen.

Es ist offensichtlich, dass man den Zeilenvorschub in der Ausgabe nicht dadurch erzwingen kann, dass man ↵ innerhalb der Befehlszeile eintippt. ↵ beendet schließlich die Eingabe. Dieses und ähnliche Probleme lassen sich mithilfe der genannten Ersatzdarstellungen lösen. Die Tabelle 4.32 zeigt eine Liste der wichtigsten dieser Ersatzdarstellungen.





### Ersatzdarstellung für Sonderzeichen

| Ersatzdarstellung | Bedeutung                                                                                                                                                                                                                          |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| \n                | Zeilenvorschub                                                                                                                                                                                                                     |
| \b                | Backspace                                                                                                                                                                                                                          |
| \t                | Tabulator                                                                                                                                                                                                                          |
| \a                | akustisches Signal (ASCII-Zeichen 7)                                                                                                                                                                                               |
| \\                | Backslash: \                                                                                                                                                                                                                       |
| \“                | Anführungszeichen                                                                                                                                                                                                                  |
| \”                | Hochkomma                                                                                                                                                                                                                          |
| \ooo              | beliebiges ASCII-Zeichen in dreistelliger oktaler Notation<br>(jedes o repräsentiert eine oktale Ziffer, so ist z.B. \o33 eine Ersatzdarstellung für Esc; denn o33 oktal entspricht 27 dezimal, das ASCII-Zeichen 27 ist aber Esc) |

Tabelle 4.32



### BEGIN und END

Mit den Schlüsselwörtern *BEGIN* und *END* (beachten Sie die Großschreibung) lassen sich Aktionen definieren, die jeweils nur einmal, nämlich vor Analyse der ersten und nach Analyse der letzten Zeile der Datei, ausgeführt werden. *BEGIN* definiert in der Regel eine Überschriftenzeile und *END* eine Zusammenfassung des „Berichts“.

Nehmen Sie wieder die Telefonliste als Ausgangsbeispiel, und nehmen Sie an, Sie wollen sie mit einer erläuternden Kopfzeile ausgeben. Dann kann dies etwa wie folgt geschehen:

```
$ awk "BEGIN {printf "%-20s%-20s", "Name:", "
"Telefon:"}\n
> /Kiel/ {printf "%-20s%-20s", $2, $4}" telefon
Name: Telefon:
Maier 786543
Meier 767676
Traub 743211
Bert 840931
$
```

Die durch *BEGIN* eingeleitete Aktion wird nur einmal durchgeführt, nämlich vor der Auswertung der Datei *telefon*. Die auf das Suchmuster „Kiel“ bezogene Aktion erfolgt für jede Zeile der Datei, die das Suchmuster enthält.



## Benutzerumgebung



Auf der Grundlage der soeben gezeigten Mittel lässt sich auf einfachste Weise eines der berühmtesten Programmierbeispiele mit *awk* nachvollziehen. Lautet das „einfachste“ C-Programm in der Bibel der C-Programmierer

Wir zitieren Kernighan und Ritchie mittels Aho, Weinberger und Kernighan.

```
main()
{
 printf("Hello world\n");
}
```

so kann man sein Pendant in *awk* zum Beispiel wie folgt eingeben:

```
awk "BEGIN {printf \"Hello world\\n\"}" /dev/null
```

Die Angabe eines Dateinamens in der Befehlssyntax zwingt zu ein wenig Phantasie. */dev/null* ist eine Datei ohne Inhalt, wodurch *awk* nicht zu irgendwelchen Aktionen bei der Analyse der Datei gezwungen ist.

Wegen des Schlüsselwortes *BEGIN* erfolgt die Ausgabe des „welt“-bewegenden Textes nur einmal, wie bei seinem großen Vorbild.

### Variablen

Wie in einer guten Programmiersprache üblich, lassen sich auch mit *awk* Variablen benutzen. Bei der ersten Benutzung der Variablen wird deren Wert, falls nicht anders vereinbart, auf *0* gesetzt.

Der Name einer Variablen muss mit einem Buchstaben beginnen, dem wiederum Buchstaben und Ziffern folgen können. Nicht nur aus ökonomischen Gründen (Schreibarbeit!) sollten Sie kurze Namen wählen. Die Namenslänge ist auf etwa 30 Zeichen beschränkt.

#### Hinweis



Zu den Buchstaben zählt auch der Unterstrich *\_*.

Mit diesen Variablen kann man die in der Programmierung üblichen Manipulationen durchführen. Ein einfaches Beispiel mag dies illustrieren:

```
$ ls -l | awk "/^d/ {anzahl=anzahl + 1} \n"
> END {printf "Anzahl der Verzeichnisse: %d", anzahl}"
telefon\n
Anzahl der Verzeichnisse: 6
$
```

*awk* prüft die von *ls -l* übergebenen Daten, d.h. die ausführliche Liste der Dateien des aktuellen Verzeichnisses. Beginnt eine der Zeilen mit dem Buchstaben *d*, d.h., ist die betreffende Datei ein Verzeichnis, wird eine Variable namens *anzahl* um den Wert 1 erhöht (*anzahl = anzahl + 1*). Beim ersten Auftreten eines Verzeichnisses wird





diese Variable im Übrigen initialisiert, d.h. auf *0* gesetzt. Da sie dann unmittelbar danach um *1* erhöht wird, beinhaltet sie am Ende der Prüfung tatsächlich die Anzahl der im Verzeichnis enthaltenen Unterverzeichnisse.

Dieser ermittelte Wert wird zuletzt nach der Dateianalyse über die *END*-Aktion ausgegeben.

### Vordefinierte Variablen

Neben den Variablen, die man selbst deklarieren kann, gibt es einige nützliche vordefinierte Variablen, deren Bedeutung Sie der *Tabelle 4.33* entnehmen können.

#### Vordefinierte Variablen

| Variable        | Bedeutung                             |
|-----------------|---------------------------------------|
| <i>NR</i>       | aktuelle Zeilennummer                 |
| <i>NF</i>       | Anzahl der Felder der aktuellen Zeile |
| <i>FILENAME</i> | Name der bearbeiteten Datei           |

*Tabelle 4.33*

`awk "{print $NR}" telefon`  
gibt das jeweils letzte Feld einer Zeile der Datei *telefon* aus.

### Andere Trennzeichen

Während in gewöhnlichen Textdateien meist das Leer- oder das Tabulatorzeichen als Trennzeichen benutzt werden, gilt dies für viele andere Dateien nicht. Viele Datenbanken benutzen das Semikolon als Trennzeichen, und Linux selbst benutzt als Trennzeichen bei datenbankähnlichen Dateien meistens den Doppelpunkt.

Im letzten Abschnitt wurde anhand eines Beispiels demonstriert, wie man sich das jeweils letzte Feld einer Textdatei anzeigen lassen kann. Interessieren Sie sich aber für den letzten Eintrag einer systemrelevanten Datei, wie etwa */etc/passwd*, nämlich die Standard-Shell der Benutzer, müssen Sie *awk* veranlassen, den Doppelpunkt als Trennzeichen für Felder zu interpretieren. Dies können Sie, indem Sie mit der Option *-F* ein anderes Trennzeichen definieren:

```
awk -F: "$NR~/bash/" /etc/passwd
```

Dieses sicherlich nicht sehr elegante Beispiel gibt alle Zeilen der Datei */etc/passwd* aus, deren letztes Feld – der Name der Standard-Shell – den Textteil „bash“ enthält. Es wird also geprüft, welcher Benutzer die bash als Standard-Shell verwendet.



## Benutzerumgebung

---



### Kontrollstrukturen

In den bisherigen Beispielen gab es immer nur eine Aktion, die von *awk* ausgeführt werden sollte, falls eine Zeile in der Datei gefunden wurde, die dem Suchmuster entsprach. Für einen solchen Fall lassen sich aber auch mehrere Aktionen festlegen, die dann, durch ein Semikolon getrennt, anzugeben sind. Auch dies geschieht in Anlehnung an die C-Programmierung.

Angenommen, Sie wollten eine Liste aller Personen Ihrer Telefonliste, die in Kiel wohnen, sowie eine Angabe über deren Anzahl. Gut, gut! Bei dieser kleinen Datei lässt sich das Problem noch durch Hingucken lösen, was aber ist, wenn die Telefonliste mehrere hundert Personen enthält? Mit folgendem Aufruf von *awk* erhalten Sie die gewünschten Informationen:

```
awk "$3~/Kiel/ {print; summe=summe + 1}\nEND {printf \"Anzahl: %d\\n\", summe}" telefon
```

Neben der Ausgabe jeder gefundenen Zeile wird eine Variable um 1 erhöht. Ihr Wert wird nach dem Ende der Dateianalyse ausgegeben.

Je gefundener Zeile werden also zwei Aktionen ausgeführt: die Ausgabe der Zeile und die Neuberechnung von *summe*. Diese Möglichkeit der Bildung von Befehlssequenzen und die Art und Weise der zugrunde liegenden Syntax lassen die enge Verwandtschaft von *awk* und C erkennen.

Diese Verwandtschaft geht aber noch weiter. Natürlich werden die einzelnen Aktionen in der Reihenfolge ausgeführt, in der sie innerhalb der Befehlszeile (oder der Kommandodatei) zu finden sind.

Mittels Programmkontrollstrukturen lässt sich allerdings – wie in gewöhnlichen Programmierumgebungen auch – dieser Ablauf beeinflussen. In Anlehnung an die Syntax in C gibt es Schleifen zur Programmwiederholung und Abfragen zur bedingten Ausführung von Programmteilen.

Als Beispiel für Abfragen soll hier kurz *if* vorgestellt werden. Seine Syntax lautet

```
if (bedingung) { aktionen} [else { aktionen}]
```

Der *else*-Zweig ist optional, wie die Klammern [...] andeuten, kann also entfallen. Das Funktionsprinzip von *if* ist denkbar einfach: Ist die Bedingung erfüllt, werden die ihr folgenden Aktionen ausgeführt, ansonsten, falls vorhanden, die Aktionen, die *else* folgen. Ein einfaches Beispiel mag dies illustrieren:

```
awk "{if ($3~/Kiel/) { print $1, $2, $3} \nelse { print $1, $2, \"wohnt nicht in Kiel\"}}"\ntelefon
```

Es wird die Telefonliste nach allen Kieler Personen durchsucht. Von diesen wird Name und Telefonnummer ausgegeben, von den nicht in Kiel wohnenden der Name sowie der Hinweis, dass sie nicht in Kiel wohnen.

Als Beispiel für die Programmwiederholung soll eine der beiden Syntaxvarianten der *for*-Schleife dienen:





```
for (ausdr1; ausdr2; ausdr3) {aktionen}
```

Diese Syntax entspricht wieder exakt derjenigen der *for*-Schleife in C. Die *aktionen* werden so lange wiederholt, wie *ausdr2* wahr ist. *ausdr1* ist normalerweise eine Aktion, die einmal, nämlich unmittelbar vor Beginn der Schleife, ausgeführt wird. *ausdr3* ist eine Aktion, die nach jeder Wiederholung einmal ausgeführt wird.

Auch hierfür ein einfaches Beispiel:

```
awk "BEGIN {for (i=1; i<=10; i++) {print i}}" \ /dev/null
```

Vor Beginn der Analyse der Datei wird eine Schleife durchlaufen, also eine Aktion mehrfach ausgeführt. Die Aktion ist die Ausgabe des Inhalts einer Variablen, nämlich *i*. Zunächst, vor Beginn der Schleife, wird die Variable *i* mit dem Wert 1 belegt (*i=1*). So lange diese Variable einen Wert hat, der 10 nicht übersteigt (*i<=10*), wird die genannte Aktion – Ausgabe von *i* – ausgeführt. Nach jedem Schleifendurchlauf – jeder Wiederholung – wird der Wert der Variablen *i* um 1 erhöht. Dies bewirkt der Operator *++*.

Ergebnis dieser *for*-Schleife ist also die Ausgabe der Zahlen von 1 bis 10 einschließlich. Die zu analysierende Datei ist */dev/null*. Da diese keinen (zu analysierenden) Inhalt hat, wird *awk* keine weiteren Aktionen durchführen.

### Operatoren

Für die Durchführung von Vergleichen und deren logische Verknüpfung stellt *awk* ebenso eine Reihe von Operatoren zur Verfügung wie für die Berechnung von Werten. Diese orientieren sich ebenfalls sehr stark an der Programmiersprache C, so dass diese Kurzbeschreibung des *awk* auch als eine theoretische Einführung in die Programmiersprache C verstanden werden kann.

#### Operatoren des *awk*



| Operator                 | Bedeutung                                       |
|--------------------------|-------------------------------------------------|
| Vergleichsoperatoren     |                                                 |
| <                        | kleiner als                                     |
| >                        | größer als                                      |
| <=                       | kleiner oder gleich                             |
| >=                       | größer oder gleich                              |
| !=                       | ungleich                                        |
| ==                       | gleich (beachten Sie: zwei Gleichheitszeichen!) |
| Arithmetische Operatoren |                                                 |
| +                        | Summenbildung                                   |



## Benutzerumgebung

### Operatorien des *awk* (Forts.)

| Operator                | Bedeutung                                                                                                                                                       |
|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -                       | Differenzbildung (Abziehen)                                                                                                                                     |
| *                       | Produktbildung (Malnehmen)                                                                                                                                      |
| /                       | Quotientenbildung (Teilen)                                                                                                                                      |
| %                       | Modulo (Rest bei der Division ganzer Zahlen;<br>7 % 5 ist 2; denn teilt man 7 durch 5, so ist das<br>ganzzahlige Ergebnis 1, es bleibt aber ein Rest<br>von 2!) |
| Zuweisungsoperatoren    |                                                                                                                                                                 |
| =                       | Die Variable (stets links vom Operator) erhält<br>den neuen Wert (stets rechts vom Operator)<br>zugewiesen                                                      |
| +=                      | Die Variable erhöht sich um den angegebenen<br>Wert                                                                                                             |
| -=                      | Die Variable vermindert sich um den angege-<br>benen Wert                                                                                                       |
| *=                      | Die Variable wird mit dem angegebenen Wert<br>multipliziert und erhält den neuen Wert                                                                           |
| /=                      | Die Variable wird durch den angegebenen<br>Wert dividiert und erhält den neuen Wert                                                                             |
| %=                      | analog zu /=, allerdings wird der Rest der Divi-<br>sion als Zwischenwert benutzt                                                                               |
| Inkrement und Dekrement |                                                                                                                                                                 |
| ++                      | erhöht den Wert einer Variablen um 1 (hat<br>etwa x den Wert 5, so hat es nach x++ den<br>Wert 6)                                                               |
| -                       | vermindert den Wert einer Variablen um 1                                                                                                                        |
| Logische Operatoren     |                                                                                                                                                                 |
| &&                      | logisches UND                                                                                                                                                   |
|                         | logisches ODER                                                                                                                                                  |
| !                       | Negation (logisches NICHT)                                                                                                                                      |

Tabelle 4.34





Im Rahmen der in dieser Linux-Einführung gebotenen Kürze lässt sich *awk* sicher nicht erschöpfend beschreiben. Wir hoffen trotzdem, Sie auf dieses vielseitige Tool neugierig gemacht zu haben, und verweisen den wissbegierigen Leser auf die vielfältige Spezialliteratur, die es zu *awk* gibt.

### **bc – ein einfacher Taschenrechner**

*bc* ist ein einfacher Taschenrechner, in seiner Handhabung recht rustikal – er ist befehlszeilenorientiert, aber in seinen Möglichkeiten recht beeindruckend.

Ruft man *bc* auf, gelangt man in eine ähnliche Situation wie beim Aufruf von *cat* ohne Angabe von Argumenten. Der Prompt verschwindet, das Programm wartet in einer absolut leeren Zeile auf die Eingabe des Benutzers.

Um Sie vor Frust zu bewahren: *bc* wird durch Eingabe von

`quit`

oder

`Strg+D`

beendet.

Falscheingaben werden mit einigermaßen verständlichen Fehlermeldungen beantwortet. Die Eingabe von Rechenoperationen erfolgt allerdings in einer natürlich anmutenden Form.

Will man etwa das Produkt aus 5 und 5,25 berechnen, so erreicht man dies durch Eingabe von

`5*5.25`   
26.25

Beachten Sie, dass das Dezimalzeichen, wie so häufig in der EDV, ein Punkt ist.

### **Variable**

Will man beispielsweise Rechnungen mit immer derselben Zahl ausführen, kann man diese als Variable deklarieren. Dies geschieht einfach in Form einer Zuweisung der Art:

`a=100; b=5`

Mit dieser Eingabezeile wurden zwei Variablen deklariert, d.h. zwei Speicherplätze reserviert, und diese Variablen wurden mit Werten versehen. Die Variable *a* hat zurzeit den Wert 100, *b* den Wert 5.

Beachten Sie wieder, dass die beiden Angaben durch Semikolon getrennt werden.

## Benutzerumgebung



Eine Ausgabe erfolgt im Übrigen nicht, *bc* ist nicht sehr „gesprächig“!

Die Variablen lassen sich nun während der laufenden Sitzung wie Zahlen in Berechnungen nutzen:

```
a+b; a*b \leftarrow
105
500
```

Das Beispiel basiert auf dem vorhergehenden. Es zeigt zusätzlich, dass mehrere Berechnungen, durch Semikolon getrennt, in einer Eingabezeile erfolgen können. Auch dies ein Hinweis darauf, wie sehr Programmierung unter UNIX/Linux C-Programmierung ist.

Zulässige Variablennamen sind in der Standardversion des *bc* die (Klein!-)Buchstaben *a* bis *z*, das ergibt maximal 26 Variablen. In der GNU-Variante können mehr Zeichen zur Namensvergabe benutzt werden. Die Namensregel entspricht in diesem Fall in etwa den Konventionen der Programmiersprache C.

### Arithmetische Ausdrücke, Operatoren

Für Berechnungen stellt *bc* die bekannten Operatoren +, -, \* und / zur Verfügung. Darüber hinaus berechnet man mit % den (ganzzahligen) Rest einer Ganzahldivision („Modulo“):

```
7%3 \leftarrow
1
```

Teilt man 7 durch 3 (ganzzahlig), so ergibt das 2 Rest 1. Eben dieser Rest ist das Ergebnis des Operators %.

C-untypisch ist der Operator ^: Er berechnet die Potenz.  $a^n$  entspricht in der normalen Schreibweise an. So ergibt  $2^3$  als Ergebnis 8; denn  $2^3$  ist natürlich nichts anderes als  $2 \cdot 2 \cdot 2$ .

Das Minuszeichen wird im Übrigen als Vorzeichen akzeptiert:

```
a=3 \leftarrow
7 + -a \leftarrow
4
```

Ganzzahlige Division ergibt ganzzahlige Ergebnisse, ganz im Sinne des Modulo-Operators %. Somit:

```
50/20 \leftarrow
2
```

Als Besonderheit bietet *bc* die Möglichkeit der direkten Berechnung einer Quadratwurzel. Dazu dient die eingebaute Funktion *sqrt()*.





sqrt ist die Abkürzung für Squareroot (engl. für Quadratwurzel).

```
sqrt(191) ↵
13
x=sqrt(361) ↵
x ↵
19
```



Hinweis

### Rechnen mit anderen Zahlensystemen

Üblicherweise rechnet bc im dezimalen, d. h. dem auf der Basis 10 beruhenden Zahlensystem. Sowohl für die Ein- als auch für die Ausgabe lässt sich ein – prinzipiell – beliebiges Zahlensystem festlegen.

Mit

```
ibase=wert
```

legt man fest, dass bc ab sofort alle Zahleneingaben so interpretiert, dass die Basis des zugrunde liegenden Zahlensystems wert ist. Z.B. bewirkt

```
ibase=16
```

dass ab sofort alle Eingaben als hexadezimal interpretiert werden. Demnach:

```
ibase=16 ↵
F ↵
15
```

Dieses Beispiel zeigt nun, warum für die Namen von Variablen auf Groß-/Kleinschreibung geachtet werden muss. F ist der hexadezimale Wert für 15. Ein Variablenname F ist nicht möglich, da er zu Zweideutigkeiten führen würde.

Will man in der obigen Situation die Eingabe wieder auf dezimal zurücksetzen, gibt es eine offenkundige Schwierigkeit.

```
ibase=10
```

wird natürlich jetzt hexadezimal interpretiert. Die hexadezimale Zahl 10 entspricht aber der dezimalen 16, d.h., ibase=10 bewirkt keine Änderung der Interpretation von Eingaben durch bc. Ein korrektes Zurücksetzen auf die Basis 10 müsste lauten:

```
ibase=A
```

Während mittels ibase die Interpretation der Eingabe beeinflusst werden kann, legt man mit

```
obase=wert
```

fest, in welchem Zahlensystem die Ausgaben erfolgen sollen.



## Benutzerumgebung

---



`obase=2`

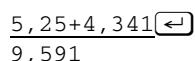
legt etwa das duale Zahlensystem für Ausgaben fest. Das Zurücksetzen von `obase` ist denkbar einfach, da nicht die Eingabe manipuliert wurde.

`obase=10`

stellt den ursprünglichen Zustand wieder her, sofern nicht zusätzlich `ibase` verändert wurde!

### Dezimalstellen

Rechnet man mit Gleitkommazahlen, ermittelt `bc` die Nachkommastellen für die Ausgabe anhand der Nachkommastellen der verwendeten Zahlen. Bei der Addition ist dies besonders einfach: `bc` benutzt die Anzahl an Nachkommastellen, die bei den zu addierenden Werten die größere ist.

  
5,25+4,341  
9,591

Die größte Anzahl an Nachkommastellen weist der zweite Summand auf (drei Stellen). Dies legt fest, dass das Ergebnis ebenfalls mit drei Nachkommastellen angezeigt wird.

Die Anzahl der Nachkommastellen lässt sich aber auch explizit durch

`scale=wert`

festlegen, wobei `wert` eine Zahl zwischen 0 und 99 ist.

`scale=3`

bewirkt also, dass Ergebnisse mit drei Nachkommastellen angezeigt werden.

### Operatoren

Zur Berechnung von Zahlen aus vorgegebenen Werten, für Vergleiche und logische Operationen stehen ihnen dieselben Operatoren zur Verfügung, die auch `awk` kennt. (In *Tabelle 4.34* sind diese Operatoren zusammengefasst.)

### Funktionen

Analog zur Funktion `sqrt()` kann der Benutzer des `bc` bis zu 26 eigene Funktionen definieren, nämlich `a()` bis `z()`. Die Definition innerhalb des `bc` erfolgt in starker Anlehnung an die Syntax der Programmiersprache C. Die allgemeine Syntax lautet:

```
define name(parameterliste) {
 anweisungen
}
```





Als einfaches Beispiel dient noch einmal die Quadratwurzel. Will man die vierte Wurzel einer Zahl berechnen, so lässt sich diese mittels einer einfachen Funktion auf das Ziehen einer Quadratwurzel zurückführen:

```
define v(x){
auto y
auto z
y=sqrt(x)
z=sqrt(y)
return (z)
}
v(81)
4
```

Die Funktion selbst heißt *v()*. Ihr wird beim Aufruf ein Parameter in der Deklaration der Funktion als *x* übergeben, nämlich die Zahl, aus der die vierte Wurzel berechnet werden soll. So berechnet *v(81)* die vierte Wurzel aus 81, was aber 3 ergibt.

Mit *auto y* und *auto x* werden zwei interne Variablen deklariert. In einem ersten Rechenschritt wird *y* der Wert der Quadratwurzel aus *x* zugewiesen. Aus dem Ergebnis wird erneut die Wurzel gezogen. Das Ergebnis (zwischengespeichert in der Variablen *z*) wird mittels *return* an *bc* zurückgegeben. *bc* selbst zeigt das Ergebnis an.

### Kontrollstrukturen

Bei der Definition von Funktionen lassen sich selbstverständlich Kontrollstrukturen verwenden, wie sie aus der Programmiersprache C bekannt sind.

Die allgemeine Syntax der Abfrage *if* lautet:

```
if (bedingung) anweisung
```

Die *bedingung* lässt sich mithilfe der gleichen Operatoren formulieren, wie Sie sie im Abschnitt über *awk* kennen gelernt haben. So bedeutet etwa

```
if (a == 0) return (0)
```

dass, falls die Variable *a* den Wert 0 hat, die Funktion verlassen wird (*return*) und als Ergebniswert die Zahl 0 liefert.

Die Syntax der *for*-Schleife ist prinzipiell aus dem letzten Abschnitt bekannt.

```
for (ausdr1; ausdr2; ausdr3) anweisung
```

*ausdr1* wird wieder vor Beginn der Schleife einmal ausgeführt, die Schleife, d.h. in diesem Fall *anweisung*, wird so lange wiederholt, wie *ausdr2* wahr ist, und am Ende eines jeden Schleifendurchlaufs wird *ausdr3* ausgeführt.

Ein schon recht anspruchsvolles Beispiel zeigt das folgende Listing:

## Benutzerumgebung

---



```
define f(x) {
auto i
auto s
s=1
for (i=1; i<=x; i++) s = s * i
return (s)
}
f(5)
120
```

Eine Variable *s* erhält den Wert 1 zugewiesen. In einer Schleife wird *s* so lange mit einem Wert *i* multipliziert, bis dieser Wert dem durch den Aufruf vorgegebenen Wert – in diesem Beispiel 5 – entspricht. *i* ist dabei zunächst 1 und wird bei jedem Schleifendurchlauf um 1 erhöht. Mithin wird 1 zunächst mit 1, dann mit 2, dann mit 3, dann mit 4 und schließlich mit 5 multipliziert. Das Ergebnis ist 120. Die gesamte Operation ist als Bildung der Fakultät einer Zahl bekannt.

Sie kennen vielleicht die Geschichte, in der ein ganz schlauer Fuchs für die von ihm entrichteten Dienste damit „entlohnt“ werden wollte, dass der ihn zu bezahlende Herrscher auf das erste Feld eines Schachbretts ein Reiskorn zur Entlohnung legen sollte, auf das zweite zwei Körner, auf das dritte vier usw. Hätte der arme Herrscher doch nur schon den *bc* zur Verfügung gehabt und im oben beschriebenen Listing die Zuweisung

*s = s \* 1*

durch

*s = 2 \* 2*

ersetzt ...

*bc* ist, wie unter anderem das letzte Beispiel zeigt, ein programmierbarer Taschenrechner. Dass die Programmierung im C-Stil erfolgt, sollte nicht überraschen. Eine sorgfältige Analyse der Syntax des *bc* fördert sicherlich das Verständnis für die Programmiersprache C.

Die beiden letzten Abschnitte, für *awk* lässt sich an dieser Stelle dasselbe sagen wie für *bc*, sollten aber nicht als Einführung in die Programmiersprache C missverstanden werden. Arbeiten mit Linux-Tools kann jedoch auf die C-Programmierung vorbereiten, dadurch, dass es hilft, die Syntaxelemente von C zu verstehen.

Bei der Auswahl der Tools, die Ihnen hier etwas näher vorgestellt wurden, sind wurde sicherlich sehr einseitig vorgegangen, weil in der Regel die Programmierung im Vordergrund gestanden hat. Das Bemühen war es bei aller Einseitigkeit allerdings auch, Ihnen zu zeigen, dass selbst bei so unterschiedlichen Dienstprogrammen, wie es *awk* und *bc* nun einmal sind, immer wieder ganz bestimmte Strukturen auftauchen, die diese Programme kennzeichnen.





Das Erkennen solcher Strukturen sollte für Sie ein Leitmotiv bei der Arbeit mit Linux sein. Es sind diese immer wiederkehrenden Dinge, die dem Benutzer auf längere Zeit die Arbeit mit dem System zu einer angenehmen Aufgabe werden lassen.

### **Xcdroast**

Die Datenmengen auf unseren Rechnern werden immer umfangreicher, und Multimedia ist in aller Munde. Zwei Gründe mindestens, sich in diesem Kapitel mit einem Hilfsprogramm zu beschäftigen, das es ermöglicht, CDs unter Linux zu brennen.

War das Selbstbrennen von CDs vor zwei, drei Jahren noch ein Luxus, den sich aufgrund der Preise für CD-Brenner und -Rohlinge nur Wenige leisten konnten oder wollten, so bekommt man die entsprechende Hardware heute schon fast standardmäßig mit einem „PC von der Stange“ in fast jedem Supermarkt für wenig Geld angeboten.

Um diese jedoch nutzen zu können, musste man sich lange kryptischer, befehlszeilenorientierter Software wie *cdrecord* bedienen, die zwar bezüglich ihrer Funktionalität überzeugten, bezüglich ihrer Benutzerergonomie jedoch vieles zu wünschen übrig ließen.

An dieser Stelle setzen Tools wie *xcdroast* an, das hier kurz vorgestellt werden soll, ohne jedoch auf sämtliche Konfigurationsmöglichkeiten einzugehen.

Dieses Programm ist leider nicht Bestandteil der dem Buch beiliegenden SuSE-Distribution. Die jeweils aktuelle Version des (auch speziell für SuSE-Linux angepassten) Programmpakets findet sich unter <http://www.fh-muenchen.de/rz/xcdroast> (der „Heimatadresse“ des Xcdroast-Projekts).

Wer die jeweils neueste Version, für die es noch keine Binärpakete gibt, jedoch selbst übersetzen möchte, sei an dieser Stelle gewarnt: Da es sich bei *xcdroast* um einen Mix aus C- und Tcl-Programmen handelt, müssen für ein erfolgreiches Komplizieren die entsprechenden Libraries auf dem Rechner vorhanden sein. Hinweise hierzu findet man ebenfalls unter der zuvor genannten Adresse.

Zur erstmaligen Konfiguration muss das Programm als Benutzer *root* gestartet werden, kann dann aber ebenfalls von anderen Usern verwendet werden, wenn es durch Setzen des *setuid*-Bits für alle Benutzer zugänglich gemacht wurde oder – und dies ist ein Leistungsmerkmal, welches erst seit der Version 0.96 zur Verfügung steht – mit folgender Befehlssequenz aufgerufen wird:

```
xcdroast -nonroot
```

Der Startbildschirm präsentiert sich beim Erstaufruf, wie es Abbildung 4.11 zeigt.



## Benutzerumgebung

Abb. 4.11 ○  
Startbildschirm von  
*xcdroast*



Bei Betätigung der Schaltfläche *Setup* finden Sie Informationen über die vorhandenen IDE- und SCSI-Geräte, die recht zuverlässig erkannt werden.

Für jeden das Programm nutzenden Anwender wird eine Konfigurationsdatei *.xcdroast* im Heimatverzeichnis des jeweiligen Anwenders angelegt.

Zu den Einstellungsmöglichkeiten gehören u.a. der Typ des verwendeten CD-Brenners, die Schreibgeschwindigkeit und so weiter.

Nach dem bedauerlichen Ableben unseres treuen Doublespeed-HP-Brenners schafften wir uns einen 8-fach-Writer von Teac an, für den leider noch keine Treiber für das dem Front-End *xcdroast* zugrunde liegende *cdrecord* zur Verfügung stand. Die Funktion Autodetect identifizierte dieses jedoch als einen 6-fach-Brenner, mit dessen Einstellung das Laufwerk problemlos arbeiten konnte.

In diesem Zusammenhang sei ohnehin vor allzu schnellen Brennern gewarnt:

Abgesehen von der Tatsache, dass bisher nur wenige Hersteller CD-Rohlinge anbieten, die für das Brennen mit sehr hohen Geschwindigkeiten ausgelegt sind und die im Regelfall auch noch deutlich teurer sind, hat sich beim Brennen von Audio-CDs herausgestellt, dass diese auch von neueren Audio-CD-Playern oftmals nicht problemlos verwendet werden konnten: Sie hatten Schwierigkeiten bei der Positionierung auf den hinteren Spuren, so dass hier ohnehin als Vorsichtsmaßnahme nur mit einer maximal 4-fachen Geschwindigkeit gebrannt werden sollte.

Als Gerät zum Einlesen von Audiodaten lässt *xcdroast* nur SCSI-Geräte oder IDE-Devices mit SCSI-Emulation zu. Um diese Emulation zu unterstützen, müssen Sie gegebenenfalls den von Ihnen verwendeten Kernel neu übersetzen (beachten Sie hierzu auch Kapitel 8 *Konfiguration und Kompilierung des Kernels*).





Grundlegende Einstellungen müssen Sie unter HD-Setup vornehmen. Hier geben Sie an, wo das Programm die so genannten „Image-Dateien“, die anschließend auf die CD kopiert werden, ablegen soll.

Sie können bis zu zwei komplette Partitionen definieren, die dann allerdings vollständig für *xcdroast* reserviert werden. Dies birgt jedoch die Gefahr, dass bei Auswahl der „falschen“ Partition unter Umständen wichtige Bereiche Ihres Dateisystems überschrieben werden könnten, da das Programm mit root-Rechten ausgeführt wird. Aus diesem Grund ist es vorzuziehen, einen Image Data Path (ein „Eingangsverzeichnis“) auf einer bereits bestehenden Partition festzulegen, unterhalb dessen dann die Images abgelegt werden.

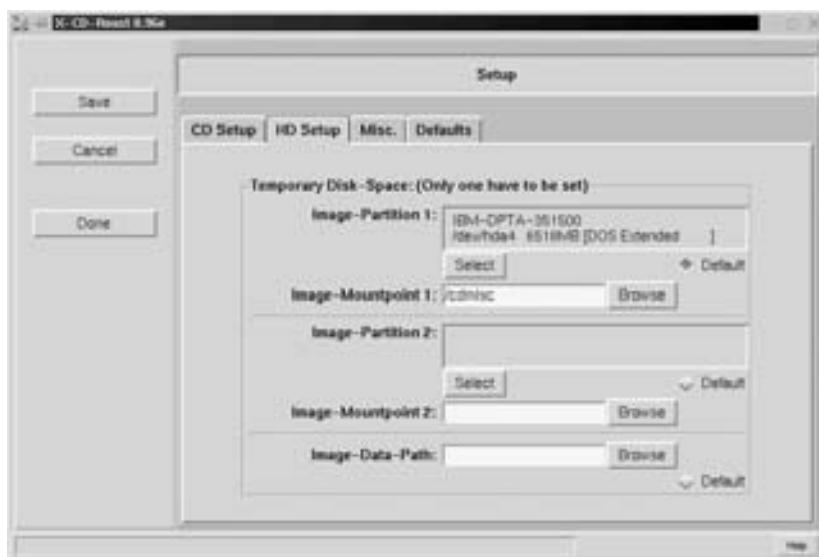


Abb. 4.12

Für *xcdroast* reservierte Partitionen für Images

Die Software ist dazu in der Lage, reine Daten-CDs, Mixed-Mode-CDs wie auch Audio-CDs zu erstellen. Bei der Erstellung von Daten-CDs kann beim Mastering zwischen unterschiedlichen Formaten wie dem Standardformat ISO9660 bis hin zu proprietären Formaten wie dem Joliet-Format gewählt werden (siehe Abbildung 4.13).

Mit Erscheinen dieses Buches soll die Software dazu in der Lage sein, so genannte „Multisession-CDs“ (hierbei werden nacheinander in mehreren Sitzungen Daten auf die CD gebrannt) und bootfähige CDs zu erstellen. Ein Blick auf die Homepage der Entwickler lohnt sich also in jedem Fall.

Die Bedienung des Programms erfolgt weitestgehend intuitiv, braucht an dieser Stelle also nicht detailliert erläutert zu werden.



## Benutzerumgebung

Abb. 4.13

Auswahl des gewünschten CD-Formats



Auf ein wesentliches Leistungsmerkmal dieser Software sei jedoch noch hingewiesen: Ein so genannter „Buffer-Underrun“ (dieser tritt ein, wenn die Festplatte die Daten nicht schnell genug an den Brenner liefern kann) führt dazu, dass ein Brennvorgang nicht ordnungsgemäß abgeschlossen werden kann. Die hierbei produzierte CD kann nur noch als Glasuntersetzer verwendet werden ... Ein simuliertes Schreiben der CD kann dieser Verschwendungen vorbeugen. Wenn der „Buffer-Underrun“ auftreten sollte, versuchen Sie, dem zu begegnen, indem Sie die Brenngeschwindigkeit sukzessive herabsetzen!!!

## Überblick über weitere Werkzeuge

Der Midnight Commander, der im vorhergehenden Abschnitt exemplarisch vorgestellt wurde, ist eines der vielen Werkzeuge, die schon fast standardmäßig zu einer Linux-Distribution gehören. Zu einer Vielzahl dieser Tools existiert bereits eigene Literatur. Um den Rahmen dieses Werks nicht zu sprengen, wird an dieser Stelle auf die Vorstellung einzelner Tools verzichtet.

Schon der Überblick über diese Tools ist eine eher subjektive Auswahl aus den Programmen, die Ihnen mit Linux (automatisch) zur Verfügung stehen. Wir hoffen dennoch, Ihnen einige Anregungen geben zu können, den Einsatz des einen oder anderen Tools zumindest zu testen. Für den Power-User verweisen wir auf die in den meisten Fällen vorhandene Spezialliteratur.





### **bison**

Eine GNU-Variante des UNIX-Veteranen *yacc* (steht für **y**et **a**nother **c**ompiler **c**ompi**le**r). Mit seiner Hilfe lassen sich so genannte „Parser“ erzeugen, die Texte nach gewissen Grammatikregeln überprüfen. Dies ist auch eine der grundlegenden Funktionen eines Compilers.

### **elvis**

Eine Variante des *vi*.

### **emacs**

Ein enorm leistungsfähiger, bildschirmorientierter Editor. Sollte Bestandteil Ihrer Distribution sein.

### **fax**

Eine Faxsoftware, die dem neuestem EIA-Standard genügt.

### **finger**

Eine Erweiterung des einfachen *finger*-Befehls. Erlaubt Auskünfte über User, auch wenn Sie nicht am gleichen Rechner eingeloggt sind, sondern nur Fernkontakt zu ihnen besteht z.B. über ein Netzwerk. Das Programm ist deshalb berüchtigt, weil durch seine Benutzung einige System-immanente Sicherheitslücken von UNIX-basierten Systemen bekannt wurden (der berüchtigte Internet-Wurm).

### **gas**

Ein GNU-Assembler.

### **gcc**

Der C-Compiler von GNU (beinhaltet heute auch einen C++-Compiler).

### **gdb**

Ein Debugger (eine Programmierhilfe – Fehlersuche in Programmen).

### **ghostscript und ghostview**

Programme zur Bearbeitung und Visualisierung von PostScript-Dateien.

### **gzip**

Ein Archivierungsprogramm zur Kompression (Verkleinerung) und Dekompression großer Dateien.

## **Benutzerumgebung**

---



### ***ispell***

Eine interaktive Rechtschreibhilfe.

### ***make***

Ein Hilfswerkzeug für Programmierer zur Automatisierung der Programmentwicklung.

### ***mtools***

Ein Programmpaket, das den Zugriff auf Dateien von auf DOS basierenden Systemen erlaubt. Es enthält Programme wie *mattrib*, *mcd*, *mcopy*, *mdel* oder *mdir*, die in ihrer Funktionsweise den (beinahe) gleichnamigen DOS-Befehlen entsprechen.

### ***rcs***

(revision control system) Ein Paket, das bei der Pflege verschiedener Versionen von Softwareprojekten hilft.

### ***smalltalk***

Die GNU-Variante der objektorientierten Entwicklungsumgebung von Xerox.

### ***tar***

Ein Archivierungs-Tool. Gehört zum UNIX-Standard und sollte deshalb in Ihrer Distribution zu Verfügung stehen.

### ***uucp***

(unix-to-unix-copy) Ein Programmpaket zum Aufbau einer einfachen Verbindung zwischen UNIX-Systemen. Gehört ebenfalls zum UNIX-Standard.

Alle zuvor genannten Programme (Programmpakete) zeichnen sich in ihrem Bereich durch hohe Funktionalität und meist auch Stabilität aus, sie gehören aber alle nicht zu den Bestandteilen eines Linux-Systems, die zu dessen Funktionieren unbedingt notwendig sind. Zum Betrieb von Linux werden sie also nicht unbedingt benötigt, sie sind „lediglich“ zusätzliche Goodies für den Anwender.





# Systemadministration – der Bootvorgang



|                          |     |
|--------------------------|-----|
| Der Bootvorgang          | 351 |
| Warum Bootparameter?     | 351 |
| Bootparameter            | 353 |
| Allgemeine Einstellungen | 353 |
| SCSI                     | 355 |
| Proprietäre CD-ROMs      | 358 |
| Diskettenlaufwerke       | 360 |
| IDE/EIDE/ATAPI           | 363 |
| Ethernet                 | 365 |
| Sonstige Geräte          | 366 |
| Weitere Informationen    | 366 |



5

## Systemadministration – der Bootvorgang

---



In den vorhergehenden Kapiteln haben Sie sich überwiegend mit den Funktionen von Linux beschäftigt, mit denen auch viele „Normal-User“ umgehen.

Dieses Kapitel ist dem Thema Systemadministration gewidmet. Hierzu gehören die Bereiche:

- Bootvorgang
- Userverwaltung
- Systemverwaltung
- Prozessverwaltung
- Daemons
- Hardwareintegration
- Emulatoren
- Datensicherung
- X-Konfiguration
- Netzwerke
- Auch das Thema Internet wird erst in diesem Kapitel behandelt, da doch einiges an Konfigurationswissen und -tätigkeiten von Ihnen gefordert wird, bevor hier eine sinnvolle Nutzung möglich ist.

Diese Bereiche müssen also detailliert verstanden werden, wenn Sie das Linux-System in seiner ganzen Pracht und Leistungsfähigkeit voll ausnutzen wollen bzw. wenn Sie auf Probleme im Alltagsbetrieb stoßen.

Nach Abarbeitung dieses Kapitels haben Sie dann tatsächlich Ihr Gesellenstück zum „Power-User“ geleistet und dürfen sich dann vielleicht sogar mit Recht *Super-user* nennen, so dass dies nicht nur ein anderes Wort für *root* ist.

Zunächst werden Parameter vorgestellt, die beim Erststart des Linux-Rechners unter Umständen angegeben werden müssen, da der Kernel nicht in jedem Fall die von Ihnen verwendete Hardware selbstständig identifizieren kann.

Im schlimmsten Fall kann dies zu einer Unterbrechung des Bootvorgangs führen, so dass das System gar nicht erst geladen wird.

Sinnvollerweise wird man dann natürlich später, wenn man seine optimale Bootkonfiguration gefunden hat, diese in seinen Kernel einbinden, was im *Kapitel 8 Konfiguration und Kompilierung des Kernels* ausführlich dargestellt wird.

## Warum Bootparameter?



Verstehen Sie diesen Abschnitt mehr als Referenz denn als vergnüglichen Lesestoff – bei auftretenden Problemen mit von Ihnen verwendeten Hardwarekomponenten können Ihnen die hier enthaltenen Informationen jedoch zur gezielten Suche nach Möglichkeiten der Problembehebung nützlich sein.

## Der Bootvorgang

Linux kann prinzipiell auf drei verschiedene Arten gebootet werden:

- von Diskette
- von DOS mit dem Startprogramm LOADLIN
- von Festplatte mit dem Linux-Loader LILO

Nebst einer einfachen Diskette, die nur ein Abbild des Kernels trägt, können auch Disketten mit LILO ausgestattet werden, so dass in allen drei Fällen Bootparameter eingegeben werden können.

Bei dem „Urladen“ von Linux von Diskette müssen die Bootparameter von Hand eingegeben werden, außerdem muss hier noch als erster Parameter *mount* oder *ramdisk* eingegeben werden.

Falls LILO konfiguriert wird, werden die Parameter dort abgelegt. Das bedeutet aber auch, dass LILO nach jeder Änderung des Kernels neu konfiguriert und installiert werden muss, wenn sich die Bootparameter geändert haben.

LOADLIN erwartet die Bootparameter als Befehlszeilenargumente am DOS-Prompt, um sie an den Kernel weiterzureichen. Da dies leicht in Befehlszeilenakrobatik ausfert kann, schreibt man sich hierfür praktischerweise eine Batchdatei.

## Warum Bootparameter?

Peripheriegeräte des PCs wie serielle und parallele Schnittstellen CD-ROM-Laufwerke oder Netzwerkkarten müssen beim Start des Betriebssystems initialisiert werden, um arbeiten zu können. Die beiden wichtigsten Werte sind der *Interrupt (IRQ)* und die *E/A-Adresse*, oft auch „Port“ genannt. Für Geräte, die standardmäßig an jedem Rechner angeschlossen sind, wie z.B. Diskettenlaufwerke, Tastatur oder serielle und parallele Schnittstellen, hat sich ein Quasi-Standard etabliert.

Geräte, die dann zusätzlich eingebaut werden, wie z.B. Netzwerkkarten, CD-ROM-Laufwerke oder Soundkarten, konkurrieren jetzt um die verbleibenden Interrupts, von denen es im PC nur 16 (0-15) gibt.



## Systemadministration – der Bootvorgang

---



Wenn der Rechner nun in einem Kleinbetrieb oder einer Abteilung Teil eines Netzwerks ist, so tut der Systemadministrator gut daran, allen Netzwerkkarten den gleichen Interrupt zuzuweisen, so dass sie im Falle von defekter Hardware oder „fest verdrahteter“ Software jederzeit ohne neue Konfiguration ausgetauscht werden können.

Die Wahl der freien Interrupts für weitere Geräte wird hiermit also wieder ein Stück eingeschränkt, und Geräte, die vom Hersteller werkmäßig z.B. auf den IRQ 5 eingestellt worden sind, müssten demnach per Hardware (*Jumper*) oder Software (Konfigurationsprogramm) umkonfiguriert werden, wenn an diesem Interrupt die Netzwerkkarte betrieben wird.

Es kann aber noch härter kommen: Bestimmte Billighardware, wie z.B. einige Soundkarten, erlauben nicht die freie Auswahl aller Interrupts, sondern nur eine Untermenge.

Zusätzliche Geräte können also in verschiedenen Rechnern auf verschiedenen Interrupts und Adressen liegen. Dies ist der Grund, warum vor dem Hochfahren von Linux diverse Parameter eingegeben werden müssen. Linux selbst versucht nun, diesen Aufwand zu reduzieren und hat in viele Treiber ein so genanntes „Autoprobing“ eingebaut, d.h., der Treiber versucht eigenständig alle IRQs und Adressen zu durchlaufen, um jeweils eine Antwort von dem Gerät zu erhalten. Zum einen dauert dies eine gewisse Zeit, zum anderen kann es zu Konflikten zwischen Geräten kommen. Zum Beispiel hängen sich einige SCSI-Treiber auf, wenn sie während des Autoprobings auf eine ältere Netzwerkkarte treffen. Hier müssen dann auch Bootparameter eingegeben werden, um die Adresse explizit mitzuteilen.

Der andere Teil der Bootparameter beschäftigt sich mit defekter Hardware – und zwar defekt in dem Sinne, dass sie defekt ab Werk geliefert wird.

Da nicht nur die Softwareindustrie, sondern auch mehr und mehr Hardwarefirmen dazu übergehen, den Benutzer als Tester zu missbrauchen, kommt leider immer mehr Hardware mit Designfehlern auf den Markt. Wie weiter unten zu sehen ist, geht es um fehlerhafte Chipsätze und Co-Prozessoren, um nicht funktionierende Prozessorinstruktionen und Diskettenlaufwerke in Laptops, die nicht dem Standard entsprechen.

Dies muss den Treibern mitgeteilt werden, damit sie um derartige Fehler herum auf „Umgehungsstraßen“ arbeiten oder sie im schlimmsten Fall abschalten.



### Bootparameter

Die Bootparameter werden dem Kernel in folgender Form übergeben:

`Bezeichner [=Wert1] [,Wert2]....`

*Bezeichner* ist eines der in diesem Abschnitt aufgeführten Schlüsselwörter, denen meistens – nicht immer – ein oder mehrere Werte folgen, die durch Kommata getrennt werden. Maximal können elf Werte pro Bezeichner eingegeben werden.

Zwischen diesen Werten und links und rechts vom Gleichheitszeichen dürfen keine Leerzeichen auftreten, da Leerzeichen verschiedene Parameter voneinander trennen.

Eine Eingabezeile für Bootparameter könnte wie folgt aussehen:

```
root=/dev/hdb1 mcd=0x390,10 ether=5,0x320,eth0
```

Hier wird nacheinander eine Festplatte als *Root*-Verzeichnis initialisiert, ein Mitzumi-CD-ROM-Laufwerk aktiviert und eine Ethernet-Karte angesprochen.

Adressen werden in aller Regel hexadezimal eingegeben, was durch ein vorangestelltes *ox* kenntlich gemacht wird. Falls auch dezimale Eingaben erlaubt sind, ist dies im Folgenden erwähnt.

Interrupts werden dezimal angegeben.

## Allgemeine Einstellungen

### Das Root-Laufwerk

Bei dem *Root*-Laufwerk handelt es sich um die Festplatte(npartition), auf der sich das *root*-Verzeichnis befindet.

```
root=Gerätename
```

Für *Gerätename* können die im Verzeichnis */dev* vorhandenen Bezeichner für die entsprechenden Hardwarekomponenten eingesetzt werden, z.B. */dev/fdo* für das erste Diskettenlaufwerk oder */dev/hdb1* für die erste Partition der zweiten Festplatte.

Dem Gerätenamen kann *ro* für *read only*, also nur zum Lesen, oder *rw* für *read/write*, also Lesen und Schreiben, nachgestellt werden.



## Systemadministration – der Bootvorgang

---



### Kein Autoprobing bestimmter Speicherbereiche

In manchen Fällen muss ein Treiber vom Autoprobing abgehalten werden, da irgendwelche Unverträglichkeiten auftreten. Dies geschieht mit der Befehlszeile:

```
reserve=Adresse,Adressbreite
```

Mit *reserve* wird ein Speicherbereich vor einem Zugriff geschützt, der unverträglich ist. Falls der Kernel sich an einer Ethernet-Karte festbeißt, die an der hexadezimalen Adresse 320 liegt, muss Folgendes eingegeben werden:

```
reserve=0x320,32
```

Die Ethernet-Karte selbst kann danach ganz normal z.B. mit *ether=5,0x320,eth0* (s.u.) initialisiert werden.

Bei *reserve* können mehrere Paare Adresse/Adressbreite eingegeben werden.

### Defekter Co-Prozessor

Einige frühe Co-Prozessoren zeigen ein fehlerhaftes Verhalten, wenn sie in den Protected Mode geschaltet werden.

Die Option

```
no387
```

deaktiviert den Co-Prozessor, sollte einer vorhanden sein. Natürlich muss dann die Softwareunterstützung der Fließkommaberechnung im Kernel eingestellt werden.

### Defekte CPU-Instruktion *hlt*

Einige i486 DX-100-Prozessoren haben Probleme mit der Instruktion *hlt* – nachdem der Prozessor schlafen gelegt worden ist, wacht er nie wieder auf. Die Option *no-hlt* belässt Linux in einer Endlosschleife, wenn es nichts zu tun gibt, anstatt den Prozessor anzuhalten:

```
no-hlt
```

### RAM > 64 MByte

Aufgrund seines schlechten Designs kann das BIOS keine Werte für den Arbeitsspeicher halten, die größer als 64 MByte sind. Wenn jemand also einen „richtigen“ Rechner hat, muss er Linux beim Start die Speichergöße mitteilen, damit der Speicherbereich oberhalb von 64 MByte genutzt werden kann. Hierzu dient der Befehl:

```
mem=Speichergröße
```





Beispiel:

`mem=97920k`

Es sind sowohl dezimale als auch hexadezimale (0x) Eingaben für die Speichergröße erlaubt. Der Zahl kann ein kleines k für 1.024 Byte oder ein großes M für 1.024 \* 1.024 Byte nachgestellt werden.

Diese Angaben der Größe werden von Linux nicht überprüft, also Vorsicht! Außerdem benutzen manche Rechner den obersten Speicherbereich für den Cache oder andere Aufgaben, so dass nicht gesamte Bereiche frei sein können.

Auch dies muss von Menschenhand und -hirn berücksichtigt werden.

## **SCSI**

Es gibt eine Reihe von SCSI-Treibern, die keine Bootparameter akzeptieren. Falls trotzdem ungewöhnlicherweise Standardwerte geändert werden müssen, müssen die entsprechenden Werte im Treiber direkt geändert und mit diesem Treiber ein neuer Kernel erstellt werden.

Karten, deren Treiber keine Bootparameter benötigen, sind:

- Always IN 2000
- Adaptec AHA-1740
- EATA-DMA
- EATA-PIO
- NCR 53C7XX
- NCR 53C8XX
- Qlogic
- Ultrastor
- Western Digital WD 7000

### **Adaptec-Host-Adapter AHA-1505/1520/1510/1515/1522 und Soundblaster 16-SCSI**

Neben den angegebenen Karten können auch viele kompatible Controller mit dem folgenden Treiber gesteuert werden, insbesondere viele Soundkarten mit SCSI-Controller.

## Systemadministration – der Bootvorgang

---



```
aha152x=Adresse[,IRQ[,SCSI-ID[,Reconnect[,Parity]]]]
```

Für die Soundkarte Pro Audio Spectrum ist ein eigener Treiber mit eigenen Parametern zuständig (s.u.).

### Adaptec-Host-Adapter AHA-1540/1542

Für den Treiber des AHA-1540/1542-Host-Adapters sind folgende Einstellungen möglich:

```
aha1542=Adresse[,Buson,Busoff[,DMAspeed]]
```

Beispiel:

```
aha1542=0x330,12,3,5
```

Die optionalen Parameter *Buson* und *Busoff* geben die „Fairness“ gegenüber anderen Geräten auf dem Bus an; die Werte werden in s angegeben. *Buson* (1...15) gibt an, wie lange der Bus belegt wird, *Busoff* (1...64), wie lange er unbelegt bleibt.

Die *DMAspeed* wird in MB/s angegeben und kann einen Wert zwischen 5 und 10 annehmen. Auf älteren Karten wird dieser Wert gejumpt, neuere bieten die Möglichkeit der softwaremäßigen Einstellung. Bei Werten über 5 muss sichergestellt sein, dass das Motherboard diese Transferraten verarbeiten kann.

### Adaptec-Host-Adapter AHA-274x, AHA-284x und AHA-294

Die Host-Adapter AHA-274x, AHA-284x und AHA-294 können wie folgt initialisiert werden:

```
aha7xxx=Extended,No_reset
```

Falls *Extended* ungleich Null ist, wird die „Extended Translation“ für große Festplatten aktiviert. Falls *No\_reset* ungleich Null ist, wird kein Reset des SCSI-Bus während des Bootens ausgelöst.

### BusLogic-SCSI Host-Adapter

Der BusLogic-SCSI-Host-Adapter wird wie folgt gesteuert:

```
BusLogic=Adresse
```

Der BusLogic Hostadapter benötigt als Parameter nur die Adresse, die einen der Werte *0x130*, *0x134*, *0x230*, *0x234*, *0x330* oder *0x334* besitzen muss.





## Future Domain TMC-8xx und TMC-950

Falls die Karte nicht automatisch während des Bootens erfasst wird, müssen folgende Parameter angegeben werden:

```
tmc8xx=Adresse, IRQ
```

Folgende Werte für die Adresse sind erlaubt: *0xc8000, 0xca000, 0xcc000, 0xce000, 0xdco00 und 0xde000*.

## Pro Audio Spectrum

Ältere Modelle dieses Typs werden noch mit Jumpern konfiguriert. Geben Sie diese Werte wie folgt beim Aufruf an:

```
pas16=Adresse, IRQ
```

Als Besonderheit kann der *IRQ* hier den Wert *255* annehmen, der den Treiber anweist, ohne Interrupts zu arbeiten, was aber geringere Geschwindigkeiten nach sich zieht. Die *Adresse* lautet normalerweise *0x388*.

## Seagate ST-01/02

Nehmen Sie folgende Einstellungen für den Seagate ST-01/02-Adapter vor:

```
st0x=Adresse, IRQ
```

Die *Adresse* kann einen der folgenden Werte annehmen: *0xc8000, 0xca000, 0xcc000, 0xce000, 0xdco00, 0xde000*.

## Trantor T128

Folgende Angaben sind für die Trantor T128 möglich:

```
t128=Adresse, IRQ
```

Gültige Werte für die *Adresse* sind *0xc8000, 0xcc000, 0xd8000 und 0xdco00*.

## SCSI-Bandlaufwerke

SCSI-Bandlaufwerke können wie folgt konfiguriert werden:

```
st=Puffergröße[, Schwellwert[, Max_Puffer]]
```

Die *Puffergröße* wird als erster Wert, gemessen in Kilobyte, angegeben und hat den interessanten Maximalwert von *16.384 KByte*.



## Systemadministration – der Bootvorgang

---



Die Voreinstellung ist 32 KByte. Der *Schwellwert* ist der Wert, ab dem der gefüllte Puffer auf Band geschrieben wird, er sollte daher natürlich immer etwas kleiner als die Puffergröße sein. Die Voreinstellung ist 30 KByte.

Der optionale Parameter *Max\_Puffer* gibt die maximale Anzahl der verwendeten Puffer an, die Standardeinstellung ist 2.

## Proprietäre CD-ROMs

In diesem Abschnitt werden Parameter von CD-ROM-Laufwerken beschrieben, die nicht dem IDE-/ATAPI- oder SCSI-Standard entsprechen.

In der Anfangszeit der CD-ROM-Laufwerke versuchten viele Hersteller, die Lesegeschwindigkeit durch die Entwicklung von eigenen Controllern zu erhöhen. Da diese Laufwerke ziemlich unverwüstlich sind, sind viele immer noch im Einsatz.

### Aztech

Konfigurieren Sie den Treiber für Aztech-Laufwerke wie folgt:

```
aztcd=Adresse [,0x79]
```

Der Treiber unterstützt die Laufwerke Aztech CDA 268–01A, Orchid CD-3110, Okano oder Wearnes CDD 110, Conrad TXC, CyCDROM CD 520 und CR 540 und verlangt als Parameter die *Adresse*.

Bei den kompatiblen Modellen mit eigener Firmware muss unter Umständen auch noch der „magische“ Parameter *0x79* gesetzt sein.

### Sony CDU-31A und CDU-33A

Die Sony-Laufwerke CDU-31A und CDU-33A können wie folgt für den Kernel verfügbar gemacht werden:

```
cdu31a=Adresse, [IRQ[,PAS]]
```

Beispiel:

```
cdu31a=0x340,10
```

Neben den beiden angegebenen Laufwerken ist diese Schnittstelle auch auf einigen Modellen der Soundkarte Pro Audio Spectrum zu finden.

Der Parameter *PAS* muss als Zeichenkette für diese Karten eingegeben werden. Einige dieser Karten unterstützen keinen Interrupt, so dass für den IRQ eine Null eingegeben werden muss.





### Sony CDU-531 und CDU-535

Die Sony-CD-ROM-Laufwerke CDU-531 und CDU-535 erlauben folgende Einstellungen:

```
sonycd535=Adresse [, IRQ]
```

Falls der *IRQ* spezifiziert und die richtige *Adresse* automatisch identifiziert wird, kann eine Null für die Adresse eingegeben werden.

### GoldStar

Endlich mal etwas Einfaches: die Konfiguration der GoldStar-Laufwerke:

```
gscd=Adresse
```

### Mitsumi

Weit verbreitet sind Laufwerke des Herstellers Mitsumi. Machen Sie diese wie folgt für Ihren Linux-Kernel kenntlich:

```
mcd=Adresse, [IRQ[, Wartezeit]]
```

Diese Parameter gelten für die älteren Mitsumi-Laufwerke LU-005, FX-001 und FX001D.

Mitsumi baut inzwischen auch ATAPI-Laufwerke, für die diese Parameter nicht gültig sind!

Da manche Mitsumi-Laufwerke lange Anlaufzeiten haben, kann es zu einer Meldung des Treibers (*timeout*) kommen. In diesem Fall sollte man als weiteren Parameter auch die *Wartezeit* (in Sekunden) angeben. Erfahrungsgemäß sind zwei Sekunden immer ausreichend. Die Standardwerte des Treibers sind *0x300,11,0*.

### Mitsumi XA/MultiSession

Falls der neuere Treiber für die Mitsumi-Laufwerke, der auch Photo-CDs lesen kann, installiert werden soll, müssen folgende Parameter eingegeben werden:

```
mcdx=Adresse, [IRQ]
```

Der Treiber ist für die gleiche Hardware – wie zuvor unter *mcd* aufgeführt – zuständig.

## Systemadministration – der Bootvorgang

---



### Optics Storage

Die folgende Option ist für das Laufwerk Optics Storage Dolphin 8000 AT sowie für das Lasermate CR 328A zuständig:

`optcd=Adresse`

Das Optics Storage 8001 hingegen ist ein ATAPI-Laufwerk.

### Philips CM206

Für die Verwendung des Philips CM206 sind folgende Angaben notwendig:

`cm206=[Adresse] [, IRQ]`

Der Treiber akzeptiert Werte zwischen `3` und `11` für den `IRQ` und hexadezimale Zahlen zwischen `0x300` und `0x370` für die `Adresse`. Es kann einer der beiden Werte oder es können beide in beliebiger Reihenfolge eingegeben werden.

Falls keine Werte bekannt sind, kann mit folgender Option auch das Autoprobing eingeschaltet werden:

`cm206=auto`

### Sanyo H 94 A

`sjcd=Adresse[, IRQ[, DMA-Kanal]]`

### SoundBlaster Pro

Sind Sie (stolzer?) Besitzer des SoundBlaster-Pro-Pakets mit SoundBlaster-eigenem CD-ROM-Laufwerk? Dann geben Sie folgende Parameter an:

`sbpscd=Adresse, Typ`

Als `Adresse` muss natürlich die der CD-ROM-Schnittstelle eingegeben werden, nicht die der „Krach-Abteilung“.

Als `Typ` muss eine der drei folgenden Zeichenketten in korrekter Groß- und Klein-schreibung eingegeben werden: `SoundBlaster`, `LaserMate` oder `SPEA`.

### Diskettenlaufwerke

Auch wenn wir, die Autoren, keine Probleme mit Diskettenlaufwerken hatten, so scheint es bei manchen Hardwarekonfigurationen dennoch welche zu geben.

## Diskettenlaufwerke



Gesonderte Behandlung erfordern vor allen Dingen wieder einmal Laptops und bestimmte PCI-Boards.

### IBM Thinkpad

`floppy=thinkpad`

Notwendig für die Diskettenlaufwerke des IBM Thinkpads, da sich IBM hier nicht an seine selbst gesetzten PC-Standards hält. Das Gegenteil wird mit `floppy=o,thinkpad` mitgeteilt.

### HP Omnibook

`floppy=omibook` bzw. `floppy=nodma`

Diese beiden Optionen sind für die Benutzung von Diskettenlaufwerken der Omni-book-Laptops von Hewlett Packard gedacht. Die gegenteilige Einstellung wird erreicht durch:

`floppy=dma`

Ist ein arbeitender DMA-Kanal für das Diskettenlaufwerk vorhanden, so ist dies die Standardeinstellung.

### IBM L 40 SX

`floppy=L40SX` bzw. `floppy=no_unexpected_interrupts`

In bestimmten Videomodi des Laptops L40SX von IBM scheint es zu Konflikten zwischen Video und Diskettenlaufwerk zu kommen. Dann können nicht erwartete Interrupts (*unexpected interrupts*) mit entsprechenden störenden Fehlermeldungen die Folge sein.

Diese Option verhindert die oben beschriebene Maßnahme. Die Standardeinstellung lautet:

`floppy=unexpected_interrupts`

### PCI-Boards

`floppy=asus_pc`

Nur die ersten beiden Diskettenlaufwerke werden hierdurch zugelassen. Wenn unabsichtlich auf das dritte oder vierte Laufwerk zugegriffen wird, hängen sich manche Rechner mit PCI-Board komplett auf.

## Systemadministration – der Bootvorgang

---



### Schnell oder sicher?

Die Angabe von

`floppy=daring`

teilt dem Treiber mit, dass die Floppy fehlerfrei läuft und die Performance erhöht werden soll. Funktioniert nicht auf allen Diskettenlaufwerken.

`floppy=0,daring`

hingegen teilt dem Treiber mit, dass alle Diskettenoperationen mit verminderterem Tempo, aber größerer Sicherheit ausgeführt werden sollen.

### Nur ein Floppycontroller

`floppy=one_fdc`

Die Standardeinstellung: ein Controller mit maximal zwei Laufwerken.

### Zwei Floppycontroller

`floppy=[Adresse,]two_fdc`

Für die Benutzung von mehr als einem Controller. Falls sich der zweite Controller an einer anderen Adresse als ox370 befindet, muss die *Adresse* angegeben werden.

### Veraltete Optionen

Der Vollständigkeit halber sind die beiden nächsten (veralteten) Optionen aufgeführt.

`floppy=[Maske],allowed_drive_mask`

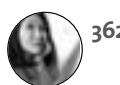
gibt die erlaubten Diskettenlaufwerke an. Statt dieser Option soll die Option *cmos* (s.u.) benutzt werden.

`floppy=all_drives`

zum Aktivieren von mehr als zwei Laufwerken. Veraltet, stattdessen soll die Option *cmos* (s.u.) benutzt werden.

### Mehr als zwei Laufwerke

`floppy=[Laufwerk],[Typ],cmos`



## IDE/EIDE/ATAPI



Notwendige Option, wenn mehr als zwei Laufwerke betrieben werden sollen, da das CMOS nur zwei verwalten kann. Kann auch für den Betrieb eines CMOS genutzt werden, das nicht dem Standard entspricht.

Die Typen sind:

- 0: Benutzung der CMOS-Werte
- 1: 5 DD, also 360 KByte
- 2: 5 HD, also 1,2 MByte
- 3: 3 DD, also 720 KByte
- 4: 3 HD, also 1,44 MByte
- 5: 3 ED, also 2,88 MByte
- 6: 3 ED, also 2,88 MByte
- 16: unbekannt, also Autoprobing

Ursprünglich war Typ 5 für Bandlaufwerke am Floppycontroller gedacht und Typ 6 für Diskettenlaufwerke selbst.

Das AMI-BIOS ignorierte dies und benutzte Typ 5. Also wurde der Treiber derart geändert, dass er beide Möglichkeiten verwalten kann.

## IDE/EIDE/ATAPI

### Standard-IDE-Festplatten

Falls der Kernel nicht selbstständig die Geometrie der Festplatte erkennt, können ihm Parameter mit auf den Weg gegeben werden.

Dies geschieht auf zwei ähnliche Arten, die aber nicht gemischt verwendet werden sollten.

Die erste Möglichkeit ist

`hd=Zylinder, Köpfe, Sektoren`

Da hier nicht die jeweilige Festplatte explizit angegeben wird, gilt das erste Auftreten dieses Parameters für die erste Festplatte. Für eventuelle zweite oder weitere Festplatten muss diese Option dann mit den jeweiligen Werten in der richtigen Reihenfolge auftreten.



## Systemadministration – der Bootvorgang

---



Die zweite Möglichkeit hat folgende Syntax:

```
hdx=Zylinder, Köpfe, Sektoren[, wpcom[, IRQ]]
```

wobei x einen Wert entsprechend der Linux-Syntax zwischen a und h besitzt.

Die optionale „write precompensation“ *wpcom* wird für IDE-Laufwerke nicht genutzt, und der optional einzugebende *IRQ* ist hier etwas deplatziert, da er zum Controller gehört und nicht zur Platte selbst.

Falls der Kernel die Festplatte nicht erkennt und die Geometrie explizit angegeben wird, sollte auch das Autoprobing ausgeschaltet werden:

```
hdx=noprobe
```

## Chipsatzunterstützung

Einige EIDE-Controller haben leider Designfehler im Chipsatz und arbeiten nicht korrekt, insbesondere, wenn ein zweiter Controller angeschlossen worden ist.

Dafür gibt es inzwischen Unterstützungen im Kernel, wobei die jeweilige Option natürlich im Kernel eingebunden sein muss.

Auch beim Controller sollte kein Autoprobing durchgeführt werden. Schalten Sie es deshalb mit folgenden Befehlen aus:

```
ide=noprobe
```

Der Chipsatz 640 von CMD Technologies gehört zu diesen traurigen Kandidaten. Falls ein Gerät am zweiten Controller angeschlossen ist und beide Geräte gleichzeitig arbeiten, kann es zu Datenverlusten kommen. Die Option

```
ide0=serialize
```

verhindert diesen parallelen Zugriff. Die Kernel-Unterstützung des CMD 640 wird auf einem PCI-Bus automatisch erkannt. Wenn allerdings ein VESA Local Bus auf dem System vorhanden ist, muss noch folgender Parameter eingegeben werden:

```
ide0=cmd640_vlb
```

Der Treiber für den DTC-2278 kann standardmäßig nur einen Controller ansteuern. Durch die Eingabe von

```
ide0=dtc2278
```

kann auch der zweite Controller betrieben werden.

## Ethernet



Auch der Treiber für die Holtek-Karte HT 6560 B kann standardmäßig nur einen Controller ansteuern. Der Bootparameter

`ide0=ht6560b`

ermöglicht den Betrieb des zweiten Controllers und erhöht außerdem den I/O-Durchsatz.

Auch für den Chipsatz Promise DC 4030 gibt es die entsprechende Kernel-Unterstützung für den zweiten Controller, wobei diese Option gleichzeitig die Cache-Fähigkeiten der Karten einschaltet.

Die Parameter lauten:

`ide0=dc4030`

Der spezielle Treiber für den Chipsatz QDI QD 6580 erlaubt höhere I/O-Geschwindigkeiten. Als Kernel-Parameter muss Folgendes eingegeben werden:

`ide0=qd6580`

Für den Chipsatz UMC 8672 gibt es eine Kernel-Unterstützung, die den Betrieb des zweiten Controllers und außerdem höhere I/O-Raten ermöglicht. Der Parameter lautet:

`ide0=umc8672`

## ATAPI-CD-ROM-Laufwerke

Falls statt einer Festplatte ein ATAPI-CD-ROM-Laufwerk am Controller angeschlossen ist, kann eingegeben werden:

`hdx=cdrom`

Normalerweise stellt der Treiber die Existenz eines CD-ROM-Laufwerks eigenständig fest, falls dies jedoch nicht der Fall sein sollte, kann dieser Parameter verwendet werden, wobei *x* die Werte von *a* bis *d* annehmen kann.

## Ethernet

Die Ethernet-Karten werden beim Kernel folgendermaßen angemeldet:

`ether=IRQ,Adresse,[Parameter1,Parameter2,]Gerätename`

Die Parameter *Parameter1* und *Parameter2* werden selten genutzt und haben zudem für jeden Ethernet-Kartentyp unterschiedliche Bedeutung.



## Systemadministration – der Bootvorgang

---



Meistens werden sie als Start- und Endadresse des gemeinsam genutzten Arbeitsspeichers (*shared memory*) benötigt. Falls der *IRQ* auf Null gesetzt ist, versucht der Treiber ein Autoprobing.

Dasselbe gilt für die *Adresse*. Der Gerätename lautet *eth0* für die erste Karte, *eth1* für die zweite etc.

Falls der Rechner während des Hochfahrens stehen bleibt und eine Netzwerkkarte vorhanden ist, sollten Bootparameter für die Netzwerkkarte angegeben werden; oft hilft das.

## Sonstige Geräte

### Logitech Busmaus

Die Logitech Busmaus benötigt als Parameter nur den *IRQ*:

`bmouse=IRQ`

### XT-Festplatte

Wer diesen Dinosaurier unbedingt unter Linux betreiben will, der muss den entsprechenden Treiber dafür in seinen Kernel aufgenommen haben. Normalerweise sollte die Festplatte dann selbstständig erkannt werden; falls nicht, müssen folgende Parameter eingegeben werden:

`xt=Typ, IRQ, Adresse, DMA-Kanal`

Der *Typ* bezeichnet den Hersteller wie folgt: 0=generisch; 1=DTC; 2, 3, 4=Western Digital; 5, 6, 7=Seagate; 8=OMTI.

## Weitere Informationen

Die von Linux durch Treiber abgedeckte Hardware umfasst inzwischen, wie Sie gesehen haben, eine sehr große Anzahl der gängigen Komponenten.

Allerdings können natürlich bei der Vielfalt an PC-Steckkarten nicht alle Möglichkeiten berücksichtigt sein.

Exotische Karten, die nur in kleiner Stückzahl in Umlauf gekommen sind, können erfahrungsgemäß nur selten unterstützt werden.

Neue Komponenten, die neue Standards setzen oder alte nicht korrekt erfüllen, können auch nicht sofort laufen, da erst neue Treiber geschrieben oder alte geändert werden müssen.



## Weitere Informationen

---



Es ist also nicht sinnvoll, Linux die allerneueste Hardware zu bieten, es sei denn, man fühlt sich befähigt, selbst einen Treiber zu schreiben und diesen der weltweiten Linux-Gemeinde zur Verfügung zu stellen.

Wessen Wissen noch nicht gestillt ist oder wer komplexe Probleme mit exotischer Hardware hat, der sei auf das Verzeichnis */usr/doc/faq/howto* verwiesen, das jede Menge so genannte „HOWTO-Dateien“ bereithält, u.a. zu den Themen Bootdisk, Busmaus, PCI, Ethernet, SCSI und CD-ROM.







# Userverwaltung



Modifikation von Benutzern und  
Gruppen **378**



**6**

## Userverwaltung

---



Stellen Sie sich doch bitte einmal Folgendes vor: Sie starten Ihr System, der Rechner fährt hoch, und es erscheint nichts weiter auf dem Bildschirm als das Wort „Loginname“.

Gut, Sie wissen schließlich, wie Sie heißen. Also: Sie geben – kein Problem – *Ernie* ein und drücken die Taste . Das Betriebssystem reagiert damit, dass es Sie auffordert, ein Passwort am Prompt einzugeben. Sie fragen sich: „Passwort?“ ... „Wofür brauche ich ein Passwort, schließlich arbeite ich ja allein an diesem Rechner, und ansonsten passe ich schon auf, dass hier niemand in die Wohnung kommt.“

So geht das nicht!

UNIX und somit auch Linux sind standardmäßig als potenzielle Mehrbenutzersysteme konzipiert:

- Sie stellen Ihnen, wie Sie am Anfang dieses Buches auch schon erfahren haben, mehrere virtuelle Konsolen zur Verfügung.
- Sie können sich von weit her über ein Modem ins System einwählen, auch wenn dort schon jemand am System arbeitet.
- Oder Ihr Rechner kann mit anderen Rechnern zu einem Netzwerk verbunden sein. Dies bedeutet schließlich, dass zur gleichen Zeit an verschiedenen Orten mehrere Benutzer auf Ihrem Linux-System arbeiten können.

Sie können sich ebenfalls auf Ihrer ersten Konsole, wie von Ihnen gewünscht, als *Ernie*, auf der zweiten z.B. als *Hein Blöd* (Grüße an die Sesamstraße) und an der dritten unter Ihrem richtigen Namen beim System anmelden.

Sollten Sie dann noch Veränderungen an Ihrem System vornehmen wollen, stehen Ihnen darüber hinaus die Konsolen vier bis sechs zur Verfügung (die Konsolen sieben und acht sind anderweitig reserviert).

Loggen Sie sich doch an einer dieser Konsolen als Systemadministrator ein. Geben Sie dafür am Prompt den Namen

`root`

ein und anschließend das Passwort, welches Sie bei der Installation des Systems für `root` gewählt haben.

## User-Account

Gehen Sie jetzt einmal davon aus, dass Sie sich weder als *Ernie* noch als *Hein Blöd* haben einloggen können, da Sie diesen beiden Benutzern bei der Erstinstallation von Linux noch keinen so genannten „User-Account“ (dt. „Benutzerkonto“) eingeräumt haben. Die Eingabe solcher Accounts ist Ihnen nur als `root` möglich.



## Userverwaltung



Nur der Systemverwalter *root* besitzt das Recht,

- einen neuen Benutzer anzulegen,
- einem neuen Benutzer ein Verzeichnis einzurichten (*/home/benutzer*),
- Benutzerinformationen zu verändern,
- einen Benutzer zu löschen,
- Gruppen anzulegen,
- Gruppeninformationen zu verändern,
- Gruppen zu löschen.



Nur *root* hat das  
Recht der Benutzer-  
verwaltung!

Um nun einen neuen Benutzer anzulegen, steht Ihnen in den meisten Linux-Distributionen ein Skript zur Verfügung, das durch den jeweiligen Installationsmanager abgearbeitet wird und mit dem Sie in Menüform einen neuen Benutzer auf Ihrem System einrichten können.

### ***useradd***

Ist dies nicht der Fall, ist es möglich, mit dem Befehl *useradd* (in manchen Systemen auch *adduser*) „von Hand“ einen neuen Account einzurichten.

#### **Syntax**

```
useradd [-option] benutzername
```

Wenn Sie den Befehl *useradd* mit dem Namen des Users als einzigem Befehlszeilenargument aufrufen, wird eine manuelle Nachbearbeitung der Datei */etc/passwd* nötig sein. So ergibt zum Beispiel die Eingabe der Befehlszeile

```
useradd axel
```

folgende Ergänzung von */etc/passwd*:

```
tgripp:x:510:100::/home/tgripp:/home/gehrke/bin/sperre
inges:x:511:100::/home/inges:/bin/bash
jutta:x:512:100::/fsoe/jutta:/bin/bash
axel:*:513:100::/home/axel:
```

Zwar hat der Befehl die User-ID korrekt hochgezählt, aber dem armen Axel würde keine Shell zugewiesen werden. Um den Befehl also einigermaßen effektiv einzusetzen, bedarf es einer näheren Erläuterung der Optionen.



## Userverwaltung

Entnehmen Sie die wichtigsten Optionen der folgenden *Tabelle 6.1.*:

**Die (wichtigsten) Optionen des Befehls *useradd***

| Option                   | Bedeutung                                                                                                                                                                                                                           |
|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>-c Kommentar</i>      | Kommentar zum <i>User</i> ; hier können Sie beispielsweise den richtigen Namen, die Telefonnummer oder Ähnliches eintragen. Wenn der Kommentar z.B. Leerzeichen enthalten soll, müssen Sie ihn in Anführungszeichen „einschließen“. |
| <i>-d Home</i>           | Heimatverzeichnis <i>Home</i> des Users                                                                                                                                                                                             |
| <i>-e Tage</i>           | Anzahl der Tage, nach denen das Passwort geändert werden muss                                                                                                                                                                       |
| <i>-g GID</i>            | Gruppensatz <i>GID</i> des Users                                                                                                                                                                                                    |
| <i>-m</i>                | legt das <i>Home</i> -Verzeichnis des Users an, wenn es dieses nicht schon gibt                                                                                                                                                     |
| <i>-s Shell</i>          | Standard-Shell des Users                                                                                                                                                                                                            |
| <i>-u UID</i>            | UID des Users                                                                                                                                                                                                                       |
| <i>-G GID1, GID2 ...</i> | eine Liste weiterer Gruppen, denen der User angehören soll                                                                                                                                                                          |

*Tabelle 6.1*

Die Eingabe der folgenden Befehlszeile

```
useradd -c A.Reusch -d /fsoe/axel -e 30 -g 200 -s
/bin/bash -m axel
```

würde also die Datei */etc/passwd* wie folgt ergänzen:

```
axel:*:513:200:A.Reusch:/fsoe/axel:/bin/bash
```

Folgende Punkte sind damit „erledigt“:

- Vergabe des Usernamens
- Die UID wurde wieder automatisch hochgezählt, da diese nicht angegeben wurde.
- Der User gehört nicht der Standard-Usergruppe 100 an, sondern der Gruppe 200.
- Der Realname wurde aufgenommen.



## Userverwaltung

- Als Home-Verzeichnis wurde `/fsoe/axel` spezifiziert und auch angelegt (Option `-m`).
- Dem User wurde `/bin/bash` als Standard-Shell zugewiesen.
- Sein Passwort verfällt in 30 Tagen.

Zentrale Dateien der Benutzerverwaltung sind (wie bereits in *Kapitel 4 Benutzerumgebung* angedeutet und wie Sie gerade gesehen haben) die Dateien `passwd` und `group`.

### ***/etc/passwd***

In dieser Datei hat jeder Benutzer des Systems einen Eintrag in der Form:

`benutzername:passwort:uid:gid:textfeld:homeverz:shell`

Sie können die Userverwaltung hier auch „manuell“ durchführen, indem Sie die Datei editieren.

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/bin/bash
daemon:x:2:2:daemon:/sbin:/bin/bash
lp:x:4:7:lp daemon:/var/spool/lpd:/bin/bash
news:x:9:13::/usr/lib/news:/bin/bash
uucp:x:10:14::/var/spool/uucp:/bin/bash
games:x:12:100::/tmp:/bin/bash
man:x:13:2::/var/catman:/bin/bash
at:x:25:25::/var/spool/atjobs:/bin/bash
postgres:x:26:2::/usr/lib/postgres:/bin/bash
lnx:x:27:27:LNX
Datenbankadministrator:/usr/lib/lnx:/bin/bash
mdom:x:28:28:Mailing list
agent:/usr/lib/majordomo:/bin/bash
yard:x:29:29:YARD
Datenbankadministrator:/usr/lib/YARD:/bin/bash
wwwrun:x:30:-2:Daemon user for apache:/tmp:/bin/bash
squid:x:31:-2:WWW proxy squid:/var/squid:/bin/bash
ftp:x:40:2:ftp account:/local/ftp:/bin/bash
nobody:x:-2:-2:nobody:/tmp:/bin/bash
zasi:x:501:100:Karsten Heyn:/home/zasi:/bin/bash
ernie:x:502:100:blabla:/home/ernie:/bin/bash
```

○ Beachten Sie, dass die einzelnen Elemente jeweils durch Doppelpunkte voneinander getrennt sind

Der *Benutzername* ist dabei ein eindeutiger String, der den Zugang bezeichnet. Hierbei handelt es sich bei persönlichen Zugängen (es gibt auch Programme, die einen Account besitzen, z.B. der Proxy-Server SQUID) um einen eindeutigen Namen, der den Benutzer bezeichnet.

Geben Sie hier z.B. *Ernie* ein (dies entspricht dem als Letztes übergebenen Befehlszeilenparameter beim Befehl `useradd`).

## Userverwaltung

---



Der nächste Eintrag betrifft das Passwort des Benutzers. Dieses wird nicht von Hand eingetragen. Hierfür steht das Programm *passwd* zur Verfügung. Lassen Sie dieses Feld bitte leer.

Ein *-\** an dieser Stelle würde jeden Zugriff auf das System für *Ernie* unmöglich machen.

Die *UID* (*User- bzw. Benutzer-ID*) ist eine Ganzzahl im Bereich von 0 bis 65.535, mit der das System den Account bei der Verwaltung der Prozesse und Dateiberechtigungen intern verwaltet. Diese zu vergebende Zahl darf bisher noch nicht benutzt sein (dieser Eintrag wird von *useradd* mit der Option *-u* manipuliert).

Die *GID* (*Group- bzw. Gruppen-ID*) ist ebenfalls eine Ganzzahl im Bereich von 0 bis 65.535, mit der das System die primäre Gruppe identifiziert. Hierbei muss es sich um eine Gruppe aus der Datei */etc/group* handeln (es ist jedoch möglich, dass der Benutzer *Ernie* über diese Gruppe hinaus weiteren Gruppen angehört; dazu später mehr). Die Manipulation dieses Feldes erfolgt bei *useradd* durch die Option *-g*.

In *Textfeld* wird gewöhnlich der volle Name des Benutzers angegeben. Zusätzlich können die Telefonnummer sowie ein zusätzlicher beliebiger Hinweis eingetragen werden (dieses Feld wird von *useradd* durch *-c* manipuliert).

Im Bereich *homeverz.* wird dem Benutzer (in der Regel) ein eigenes Verzeichnis unterhalb des Verzeichnisses */home* zugeordnet. Hier hat er dann die Möglichkeit, all die Dateien abzulegen, mit denen er arbeiten möchte. Nur er und *root* haben vollen Zugriff auf dieses Verzeichnis (entspricht der Option *-d*).

Anschließend wird mit dem Eintrag *shell* noch eine individuelle Benutzer-shell festgelegt. Dies kann eine beliebige auf dem System zur Verfügung stehende Shell sein (unter Linux ist das meist die *bash*). Der Befehl *useradd* manipuliert dieses Feld mit der Option *-s*.

Wenn Sie auf diese Arbeiten neuen User eingerichtet haben, müssen Sie nach dem Abspeichern von */etc/passwd* noch ein Verzeichnis für ihn anlegen, z.B. mit:

```
mkdir /home/ernie
```

Bei Verwendung des Befehls *useradd* durch die Option *-m* passierte dies automatisch.

Sie müssen dieses Verzeichnis noch für den User *Ernie* zugänglich machen, indem Sie die Besitzrechte an ihn abtreten

```
chown ernie /home/ernie
```

und gegebenenfalls die Gruppenzugehörigkeit des Verzeichnisses mit *chgrp* ändern.

In beiden Fällen (ob Sie den neuen User nun mit *useradd* oder wie zuvor beschrieben angelegt haben) sollten Sie ihm unverzüglich ein Passwort zuweisen.

Dies geschieht mit dem Befehl *passwd*.





### **passwd**

Die Syntax für diesen Befehl lautet:

```
passwd [-f | -s] [name]
passwd [-x max] [-n min] [-w Tage] [-i Tage] name
passwd [-g] [-r|R] Gruppe
passwd { -l | -d | -S } name
```

Wenn Sie den Usernamen nicht angeben, bezieht sich das Programm auf Sie als *root*, was ja in diesem Fall nicht gewünscht ist.

Die Optionen für diesen Befehl entnehmen Sie bitte der folgenden *Tabelle 6.2.*:

**Die Optionen des Befehls *passwd***

| Option         | Bedeutung                                                                                                                                                                                         |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>-f</i>      | ändert die User-spezifischen Einträge der Datei <i>passwd</i> , die durch den Befehl <i>finger</i> ausgegeben werden. Beachten Sie hierzu die nähere Beschreibung weiter unten in diesem Kapitel. |
| <i>-s</i>      | ändert die Festlegung der Login-Shell für den User. Beachten Sie hierzu die nähere Beschreibung weiter unten in diesem Kapitel.                                                                   |
| <i>-x max</i>  | gibt an, nach maximal wie vielen Tagen das Passwort vom User geändert werden muss                                                                                                                 |
| <i>-n min</i>  | gibt an, wie viele Tage minimal vergangen sein müssen, bevor das Passwort geändert werden kann                                                                                                    |
| <i>-w Tage</i> | gibt an, wie viele Tage vor Ablauf des Passworts der User gewarnt werden soll, dass dieses geändert werden muss                                                                                   |
| <i>-i Tage</i> | gibt an, nach wie vielen Tagen nach Ablauf eines Passworts der dazugehörige Account deaktiviert werden soll                                                                                       |
| <i>-l</i>      | sperrt einen Account durch Änderung des Passworts                                                                                                                                                 |
| <i>-u</i>      | hebt die Sperre durch Wiederherstellung des Passworts wieder auf                                                                                                                                  |
| <i>-S</i>      | gibt Auskunft über den Status eines Accounts                                                                                                                                                      |
| <i>-g</i>      | ändert das Gruppenpasswort. Dies kann nur durch <i>root</i> oder durch einen Gruppenadministrator geschehen.                                                                                      |
| <i>-r</i>      | wird mit der Option <i>-g</i> verwendet, um ein Passwort einer Gruppe wieder zu entfernen. Dies erlaubt allen Mitgliedern Zugang zu dieser Gruppe.                                                |
| <i>-R</i>      | Diese Option wird mit der Option <i>-g</i> verwendet, um die genannte Gruppe für alle User zu sperren                                                                                             |

*Tabelle 6.2*

## Userverwaltung

---



### Der Dialog bei Verwendung der Option **-f**

Wenn Sie die Option **f** mit dem Befehl *passwd* verwendet haben, werden Sie interaktiv durch die Veränderung der User-spezifischen Informationen geleitet (die Werte in den eckigen Klammern sind die voreingestellten Einträge der Datei *passwd*):

```
Linserv:/>passwd -f axel
Changing the user information for axel
Enter the new value, or press return for the default

Full Name [A.Reusch]: Axel Reusch
Room Number []: 12
Work Phone []: 1112
Home Phone []: 12345
Other []: -
```

### Der Dialog bei Verwendung der Option **-s**

Wenn Sie die Option **s** mit dem Befehl *passwd* verwendet haben, werden Sie interaktiv durch die Veränderung der Login-Shell des Users geleitet (die Werte in den eckigen Klammern sind die voreingestellten Einträge der Datei *passwd*):

```
linserv:/ > passwd -s axel
Changing the login shell for axel
Enter the new value, or press return for the default

Login Shell [/bin/bash]: /bin/ksh
```

Sie sollten nun dem guten Ernie ein Passwort zuweisen:

```
passwd Ernie
```

Dieses müssen Sie nach der ersten Eingabe durch nochmaliges Eingeben der Zeichenkette bestätigen.

Anschließend – nachdem Sie Ernie sein „Geheimnis“ anvertraut haben – ist dieser in der Lage, sich beim ersten Einloggen und dann jederzeit (mit den oben beschriebenen Einschränkungen) eine neue, noch geheimere Zeichenkette auszudenken und diese nach erneutem Aufruf des Programms *passwd* als sein Passwort festzulegen.

Dabei ist *root* – und hoffentlich nur dieser (Ernie haben Sie bei der Eingabe auch hinter sich geschaut?) – allerdings in der Lage, Ernies Passwort zu ändern. Was *root* allerdings nicht kann, ist, das Passwort von Ernie herauszufinden.

Haben Sie Dateien für Ernie bereitgestellt, so müssen Sie ihm diese auch noch zugänglich machen. Noch sind Sie ja der Eigentümer der Dateien. Dies geschieht mithilfe des Befehls *chown*. Schauen Sie sich für die Syntax bitte das *Kapitel 4 Benutzerumgebung* in diesem Buch an.





### **/etc/group**

Es handelt sich bei dieser Datei ebenfalls um eine ASCII-Datei, in der definiert wird, welchen Gruppen die einzelnen User angehören, wobei die einzelnen Einträge folgendem Muster folgen:

```
gruppenname:passwort:GID:benutzer-Liste
```

Der Eintrag *gruppenname* definiert den Namen der Benutzergruppe. Das *passwort*, das es ermöglicht, mit dem Befehl *newgrp* einen Gruppenwechsel zu vollziehen, wurde mit dem Befehl

```
passwd -g Gruppe
```

von *root* oder vom Gruppenadministrator festgelegt.

Die Gruppen-ID ist (wie bereits im *Kapitel 4 Benutzerumgebung* beschrieben) ebenfalls ein numerischer Wert. Diesem folgt eine (durch Kommata separierte) Liste der User, die dieser Gruppe angehören.

Eine typische Gruppdatei könnte so aussehen:

```
root:x:0:roo
bin:x:1:root,bin,daemon
daemon:x:2:
tty:x:5:
lp:x:7:
kmem:x:9:
wheel:x:10:
mail:x:12:
news:x:13:news
uucp:x:14:uucp
at:x:25:at
lnx:x:27
yard:x:29
users:x:100:
nogroup:x:-2:root
shop:x:200:
```

Statt diese Datei nun manuell zu editieren, steht auch hier wieder ein (dem Befehl *useradd* analoger) Befehl zur Verfügung:

### **groupadd**

Der Befehl *groupadd* fügt dem System eine neue Gruppe hinzu.



## Userverwaltung



### Syntax

```
groupadd [-g gid [-o]] Gruppe
```

Der Befehl besitzt (wie aus der Syntaxdarstellung ersichtlich ist) die beiden Optionen *-g* und *-o*, deren Bedeutung Sie der *Tabelle 6.3* entnehmen können.

Die Optionen des Befehls *grpadd*

| Option        | Bedeutung                                                                                                                                                                                                                                                            |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>-g gid</i> | definiert den nummerischen Wert der Gruppen-ID. Dieser muss eindeutig sein, es sei denn, die Option <i>-o</i> wird verwendet. Die Werte zwischen 0 und 99 sind im Allgemeinen für das System reserviert, Sie sollten also nur Werte größer 100 bis 65.535 verwenden. |
| <i>-o</i>     | ermöglicht die doppelte Verwendung einer <i>gid</i>                                                                                                                                                                                                                  |

Tabelle 6.3



### *groups*

Mit dem Befehl *groups* können Sie schließlich herausfinden, welchen Gruppen Sie angehören. Für den Systemadministrator *root* würde (basierend auf der zuvor dargestellten Gruppendatei) auf dem Bildschirm folgende Meldung erscheinen:

```
root bin nogroup
```

da *root* keiner eigenen Gruppe (wie Ernie als Mitglied der Gruppe *user*) angehört.

## Modifikation von Benutzern und Gruppen

In der Praxis ergibt sich immer wieder die Notwendigkeit, Benutzer- und Gruppen-eigenschaften zu modifizieren, bis hin zum Entfernen von Benutzern und Gruppen. Auch für diese Aufgaben steht eine Reihe von Befehlen zur Verfügung.

### Modifikationsbefehle

#### **usermod**

Der Befehl *usermod* dient der Modifikation eines Benutzerkontos.



## Modifikation von Benutzern und Gruppen



### Syntax

```
usermod [-u UID [-o]] [-g Gruppe]
 [-G Gruppe,...] [-d Home [-m]]
 [-s Shell] [-c Kommentar]
 [-l Neuname] [-f Tage]
 [-e Verfallsdatum] Username
```

Der Befehl modifiziert die System-Account-Dateien in Abhängigkeit von den ihm übergebenen Befehlszeilenargumenten. Hierbei ist es nahe liegend, dass die ihm zur Verfügung stehenden Optionen an die des Befehls *useradd* erinnern.

### Die Optionen des Befehls *usermod*

| Option                  | Funktion                                                                                                                                                                                                                                       |
|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>-c Kommentar</i>     | Die zu modifizierenden Einträge des Kommentarfeldes der Datei <i>/etc/passwd</i> . Beachten Sie hierzu die weiter oben beschriebenen Konventionen oder die Erklärungen zum Befehl <i>chfn</i> .                                                |
| <i>-d home</i>          | Das neue Heimatverzeichnis des Users. Wenn die Option <i>-m</i> verwendet wird, werden die Inhalte des aktuellen Heimatverzeichnisses in das neue Heimatverzeichnis verschoben. Sollte dieses noch nicht vorhanden sein, wird es neu angelegt. |
| <i>-e Verfallsdatum</i> | spezifiziert das „Verfallsdatum“, an dem der Account ausläuft. Das Datumsformat lautet MM/TT/JJ.                                                                                                                                               |
| <i>-f Tage</i>          | die Anzahl der Tage, nach denen ein Account verfällt, wenn ein Passwort verfallen ist. Der Wert <i>0</i> deaktiviert den Account unverzüglich nach Verfall des Passworts. Der Standardwert <i>-1</i> schaltet diese Funktionalität aus.        |
| <i>-g Gruppe</i>        | die neue Gruppe, welcher der User anfänglich nach dem Einloggen angehört                                                                                                                                                                       |
| <i>-G Gruppe,...</i>    | eine Liste weiterer Gruppen, denen der User ebenfalls angehört. Diese Liste ist Komma-separiert, wobei keine Leerzeichen verwendet werden dürfen.                                                                                              |
| <i>-l Neuname</i>       | Der Username wird in <i>Neuname</i> abgeändert                                                                                                                                                                                                 |
| <i>-m</i>               | siehe Option <i>-d</i>                                                                                                                                                                                                                         |
| <i>-o</i>               | nur in Verbindung mit der Option <i>-u</i> . Ermöglicht es, mehreren Usern die gleiche UID zu zuweisen.                                                                                                                                        |

## Userverwaltung

### Die Optionen des Befehls *usermod* (Forts.)

| Option          | Funktion                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>-s Shell</i> | spezifiziert die neue Shell, mit welcher der User standardmäßig arbeitet. Wenn keine Angabe gemacht wird, wird die Standard-Login-Shell verwendet.                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <i>-u UID</i>   | die nummerische UID. Diese muss eindeutig sein, es sei denn, die Option <i>-o</i> wird verwendet. Der Wert darf nicht negativ sein. Die Standardeinstellung ist, einen Wert zu verwenden, der mindestens größer 99 oder der bisher größten vergebenen User-ID ist. Die Werte 0 bis 99 sind typischerweise für System-Accounts reserviert. Wird die UID geändert, wirkt sich diese Änderung auf alle Dateien unterhalb des Heimatverzeichnisses des User aus. Dateien in anderen Dateisystemen müssen manuell geändert werden (beachten Sie hierzu auch den Befehl <i>find</i> ). |

Tabelle 6.4

Der Befehl *usermod* erlaubt keine Modifizierungen von Usern, die aktuell eingeloggt sind. Darüber hinaus müssen Sie sich vergewissern, dass bei einem User, dessen UID geändert werden soll, aktuell keine Prozesse laufen.

#### Hinweis



Besitzer von *crontab*-Dateien und von *at*-Jobs müssen manuell geändert werden. Wurden Änderungen bezüglich NIS vorgenommen, müssen diese ebenfalls auf dem NIS-Server ausgeführt werden.

#### Tipp



Eine sehr bequeme Methode, die Eintragungen der Kommentarspalte in der Datei */etc/passwd* zu modifizieren, bietet der Befehl *chfn*.

#### **chfn**

Der Befehl *chfn* dient der Modifikation des Kommentarfeldes der Datei */etc/passwd*.

#### Syntax

```
chfn [-f VollerName] [-n Zimmernr] [-w Durchwahl] [-h
Privatnr] [-o weiteres]
```

Der Befehl *chfn* dient der Modifikation des vollen Namens des Users, der Zimmernummer, der Telefonnummer am Arbeitsplatz, der privaten Telefonnummer und weiterer Informationen. Es handelt sich hierbei um die Art Informationen, die beispielsweise von dem Befehl *finger* ausgegeben werden.



## Modifikation von Benutzern und Gruppen



Nur der Superuser ist dazu berechtigt, die Informationen anderer User zu ändern, ansonsten darf dieser Befehl mit Ausnahme der Option `-o`, die wiederum nur dem Superuser vorbehalten ist, nur von einem User zur Änderung seiner eigenen „personengebundenen Daten“ verwendet werden.

Prinzipiell dürfen alle Zeichen für die Modifikation dieser Einträge verwendet werden – mit Ausnahme von Steuerzeichen, des Doppelpunkts und des Gleichheitszeichens, wobei diese Einschränkungen für das Feld *Weiteres* nicht zutreffen. Die hier vorgenommenen Einträge enthalten Verwaltungsinformationen, die von weiteren Applikationen verwendet werden.

### Die Optionen des Befehls `chfn`

| Option                     | Funktion                                                                               |
|----------------------------|----------------------------------------------------------------------------------------|
| <code>-f VollerName</code> | dient der Modifikation des realen Namens des Users                                     |
| <code>-n Zimmernr</code>   | dient der Modifikation der Zimmernummer des Users                                      |
| <code>-w Durchwahl</code>  | dient der Modifikation der telefonischen Durchwahl des Users                           |
| <code>-h Privatnr</code>   | dient der Modifikation der privaten Telefonnummer des Users                            |
| <code>-o weiteres</code>   | dient der Modifikation weiterer Informationen, die applikationsabhängig genutzt werden |

Tabelle 6.5

Wenn keine der Optionen in der Befehlszeile angegeben wird, funktioniert der Befehl `chfn` interaktiv: Die aktuellen Werte werden in eckigen Klammern angezeigt und können durch Drücken der Taste `←` übernommen werden. Andernfalls geben Sie den jeweils neu gewünschten Eintrag ein und bestätigen mit `→`.

## groupmod

Der Befehl `groupmod` modifiziert die System-Account-Dateien in Abhängigkeit von den ihm übergebenen Befehlszeilenargumenten.

### Syntax

```
groupmod [-g GID [-o]] [-n Gruppenname] Gruppe
```

Der Befehl stellt folgende Optionen zur Verfügung:



## Userverwaltung

### Die Optionen des Befehls *groupmod*

| Option                      | Funktion                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>-u GID</code>         | die nummerische Gruppenkennung <i>GID</i> . Diese muss eindeutig sein, es sei denn, die Option <code>-o</code> wird verwendet. Der Wert darf nicht negativ sein. Die Standardeinstellung ist, einen Wert zu verwenden, der mindestens größer 99 oder der bisher größten vergebenen Gruppen-ID ist. Die Werte 0 bis 99 sind typischerweise für System-Accounts reserviert. Die GIDs von Dateien müssen manuell geändert werden (beachten Sie hierzu auch den Befehl <i>find</i> im Kapitel 3 <i>Linux-Dateisystem</i> ). |
| <code>-n Gruppenname</code> | der neue Gruppenname für die angegebene Gruppe                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <code>-o</code>             | ermöglicht eine „Doppelvergabe“ von GIDs                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |

Tabelle 6.6

Um zum Beispiel der Gruppe *tkie* die neue GID 200 zu zuweisen, geben Sie ein:

```
groupmod -g 200 tkie
```

## Löschenbefehle

Die im Folgenden beschriebenen Befehle dienen dazu, User bzw. Gruppen endgültig aus dem System zu entfernen.

### ***userdel***

Wenn ein User komplett aus dem System entfernt werden soll, können Sie den Befehl *userdel* verwenden. Er löscht einen Account und die damit zusammenhängenden Dateien.

### Syntax

```
userdel [-r] Username
```

Der Befehl modifiziert die Account-Dateien des Systems dahingehend, dass alle Einträge entfernt werden, die *Username* betreffen.



## Modifikation von Benutzern und Gruppen



### Die Optionen des Befehls `userdel`

#### Option

#### Funktion

-r

löscht das Heimatverzeichnis des Users und die in ihm enthaltenen Dateien, die diesem User gehören, aber in anderen Dateisystemen abgelegt sind, müssen manuell gesucht und gelöscht werden (siehe hierzu auch den Befehl `find` im Kapitel 3 *Linux-Dateisystem*).

Tabelle 6.7



### **groupdel**

Der Befehl `groupdel` löscht eine Gruppe.

#### Syntax

```
groupdel Gruppe
```

Der Befehl `groupdel` modifiziert die System-Account-Dateien in der Form, dass alle die Gruppe betreffenden Einträge gelöscht werden. Sie müssen alle Dateisysteme manuell dahingehend überprüfen, dass keine Dateien mit der GID der angegebenen Gruppe mehr vorhanden sind (beachten Sie hierzu auch den Befehl `find -group` im Kapitel 3 *Linux-Dateisystem*).

Eine Gruppe kann nicht gelöscht werden, wenn sie die Primärgruppe eines Users ist. Löschen Sie zunächst den User oder weisen Sie ihn einer anderen Benutzergruppe zu.



Hinweis

Um die Gruppe `tkie` zu löschen, geben Sie Folgendes ein:

```
groupdel dbgroup
```

Nachdem Sie in diesem Kapitel einige Möglichkeiten der Benutzerverwaltung kennen gelernt haben, werden Sie sich im folgenden Kapitel 7 *Systemverwaltung* einigen weiteren allgemeinen Möglichkeiten der Systemverwaltung zuwenden.







# Systemverwaltung



|                         |            |
|-------------------------|------------|
| Prozesse                | <b>386</b> |
| <i>mount und umount</i> | <b>391</b> |
| Dämonen                 | <b>393</b> |



**7**

## Systemverwaltung



In diesem Kapitel des Buches sollen Sie einige wichtige Hintergrundinformationen zur Verwaltung Ihres Linux-Systems und zur Steuerung von Prozessen erhalten.

### Prozesse

Linux kann mehrere Prozesse, auch „Tasks“ genannt, gleichzeitig verarbeiten. Die Rechenzeit des Prozessors wird auf mehrere Benutzer aufgeteilt: Linux ist also ein Multiuser-System. Je-der dieser Benutzer (in Linux sind dies häufig, wenn der Rechner nicht als Server dient, die sechs virtuellen Konsolen) kann wiederum mehrere Prozesse starten: Linux ist also auch ein Multitasking-System.

Welche Prozesse gerade arbeiten, kann man sich über `ps -u` anzeigen lassen. Es wird eine Liste ausgegeben, die etwa wie in Abbildung 7.1 aussieht.

Abb. 7.1

`ps` zeigt alle im Arbeitsspeicher befindlichen Prozesse

| USER  | PID | RCPU | XMEM | VSS  | RSS | TTY | STRT | START | TIME | COMMAND           |
|-------|-----|------|------|------|-----|-----|------|-------|------|-------------------|
| root  | 1   | 0,0  | 0,1  | 344  | 64  | ?   | S    | Mar23 | 0:06 | init [2]          |
| root  | 2   | 0,0  | 0,0  | 0    | 0   | ?   | SM   | Mar23 | 0:06 | [kflushd]         |
| root  | 3   | 0,0  | 0,0  | 0    | 0   | ?   | SM   | Mar23 | 0:02 | [kupdate]         |
| root  | 4   | 0,0  | 0,0  | 0    | 0   | ?   | SM   | Mar23 | 0:00 | [kpiod]           |
| root  | 5   | 0,0  | 0,0  | 0    | 0   | ?   | SM   | Mar23 | 0:41 | [kswapd]          |
| root  | 6   | 0,0  | 0,0  | 0    | 0   | ?   | SM   | Mar23 | 0:00 | [md_thread]       |
| root  | 63  | 0,0  | 1,0  | 4584 | 384 | ?   | S    | Mar23 | 0:10 | /sbin/imsilog -F  |
| root  | 95  | 0,0  | 0,9  | 1704 | 364 | ?   | S    | Mar23 | 0:02 | /sbin/ippdd pidfi |
| bin   | 109 | 0,0  | 0,0  | 1060 | 0   | ?   | SM   | Mar23 | 0:00 | [portmap]         |
| root  | 110 | 0,0  | 0,5  | 1096 | 200 | ?   | S    | Mar23 | 0:02 | /usr/sbin/suzlogd |
| root  | 122 | 0,0  | 0,4  | 1424 | 184 | ?   | S    | Mar23 | 0:01 | /usr/sbin/klogd - |
| named | 146 | 0,0  | 1,9  | 2676 | 740 | ?   | S    | Mar23 | 0:01 | /usr/sbin/named - |
| root  | 163 | 0,0  | 0,1  | 1140 | 60  | ?   | S    | Mar23 | 0:00 | /usr/sbin/automou |
| root  | 175 | 0,0  | 0,1  | 1140 | 60  | ?   | S    | Mar23 | 0:00 | /usr/sbin/automou |
| root  | 188 | 0,0  | 0,5  | 1140 | 228 | ?   | S    | Mar23 | 0:00 | /usr/sbin/automou |
| root  | 196 | 0,0  | 0,0  | 0    | 0   | ?   | SM   | Mar23 | 0:00 | [lockd]           |
| root  | 197 | 0,0  | 0,0  | 0    | 0   | ?   | SM   | Mar23 | 0:00 | [rpciod]          |
| root  | 201 | 0,0  | 0,0  | 1080 | 0   | ?   | SM   | Mar23 | 0:00 | [rpc,kstatd]      |
| root  | 205 | 0,0  | 0,0  | 1172 | 0   | ?   | SM   | Mar23 | 0:00 | [rpc,umountd]     |
| root  | 208 | 0,0  | 0,0  | 0    | 0   | ?   | SM   | Mar23 | 0:04 | [nfed]            |
| root  | 209 | 0,0  | 0,0  | 0    | 0   | ?   | SM   | Mar23 | 0:04 | [nfed]            |
| root  | 210 | 0,0  | 0,0  | 0    | 0   | ?   | SM   | Mar23 | 0:04 | [nfed]            |
| root  | 211 | 0,0  | 0,0  | 0    | 0   | ?   | SM   | Mar23 | 0:04 | [nfed]            |

Es werden die verschiedensten Details aufgeführt. Wie man sieht, erhält jeder Prozess seine eigene, eindeutige Identifikationsnummer, die *Prozess-ID (PID)*, die beim Start vergeben wird und mit der man sich später auf den jeweiligen Prozess beziehen kann.

Prozesse können sich dabei in verschiedenen Zuständen befinden:

- Ein Prozess kann sich im Vordergrund befinden und aktiv sein – dies ist der Normalfall.
- Auch kann sich ein Prozess im Hintergrund befinden und aktiv sein. Dies ist sehr praktisch, um z.B. Disketten oder Bänder zu formatieren, um größere Datenmengen zu sortieren oder zu drucken. Ein Prozess wird im Hintergrund





gestartet, indem man dem Programmnamen einen Ampersand (&) nachstellt. Mit *fg Programmname* kann der Prozess dann wieder in den Vordergrund geholt werden. Ganz einfach lassen sich mehrere Prozesse ausführen, indem man sie in verschiedenen Terminals unter X Window ausführt.



Programme können durch **Strg+Z** „schlafen“ gelegt werden. Durch ein Prozentzeichen, das dem Programmnamen vorangestellt wird, lassen sie sich wieder aktivieren.

In den meisten Fällen laufen all diese Prozesse auf einer CPU, obwohl Linux inzwischen mehrprozessorfähig ist. Die Rechenzeit wird unter den verschiedenen Prozessen geteilt – fast immer für den Benutzer unmerklich .

Erst wenn im Hintergrund oder auf den anderen virtuellen Konsolen Prozesse laufen, die viel Rechenzeit kosten und nur wenig auf die Hardware warten müssen, kann es – je nach Rechner-ausstattung – sein, dass man merkt, dass die Performance auf der Konsole sinkt. Wenn man beispielsweise auf einer virtuellen Konsole einen neuen Kernel kompiliert, fällt bei langsameren Rechnern auf, dass z.B. der Editor anfängt zu „humpeln“.

Falls man als Systemverwalter *root* eingeloggt ist, kann man Prozessen eine geringere Rechen-zeit zuweisen, damit sie *nice* zu den anderen Prozessen sind. Durch die Eingabe von

```
nice Prozessname
```

wird die Priorität des Prozesses verringert, d.h., der prozentuale Anteil dieses Prozesses an der Rechenzeit wird kleiner, er arbeitet langsamer.

Mit dem Befehl *top* kann überprüft werden, welches Programm wie viel Rechenpo-wer verbraucht.

### Zeitversetzte Prozesse

Prozesse müssen nicht sofort ausgeführt werden. Das kann aus vielen Gründen sinnvoll sein. Zum einen wegen begrenzter Hardweareressourcen:

Ausdrucke von großen Datenmengen, wie z.B. Statistiken, Kundenlisten oder den Kernel-Quelltexten, können den Abteilungsdrucker für längere Zeit beschäftigen, und der wird dringend von den Kollegen benötigt. Also könnte man das auch nachts erledigen.

Eine andere Möglichkeit ist die Nutzung des Nachttarifs der Telekom von 2.00 bis 5.00 Uhr: Faxe oder Dateien werden erst ab 2.00 Uhr übertragen, um Telefon-gebühren zu sparen. Außerdem gibt es wie-derkehrende Arbeiten, die das System in bestimmten Intervallen automatisch erledigen soll. Z.B. soll jede Nacht auto-matisch eine Archivdatei aller neuen und veränderten Dateien angelegt wer-den, oder für alle Kunden der Firma, die in zwei Tagen Geburtstag haben, soll der Stan-dardglück-wunschbrief ausgedruckt werden.

Es gibt drei Möglichkeiten, Prozesse zeitversetzt zu starten: *at*, *batch* und *cron*.

## Systemverwaltung

---



### ***at, batch, cron***

Während *cron* zur Ausführung von in bestimmten Abständen immer wieder auftretenden Jobs zu bestimmten Zeitpunkten dient, ist *at* zur Ausführung von bestimmten Jobs gedacht, die einmalig oder selten zu einem bestimmten Zeitpunkt ausgeführt werden sollen.

*batch* hingegen ist für Aufgaben gedacht, deren Erledigung keinen genauen Zeitpunkt erfordert, sondern die dann abgearbeitet werden können, wenn Linux feststellt, dass die Systembelastung unter einen bestimmten Schwellwert fällt.

#### ***at***

*at* verlangt als Argument die Uhrzeit. Nach dem Drücken der Taste erwartet *at* dann die zeilenweise Eingabe der Jobs, die zur angegebenen Uhrzeit auszuführen sind. Die Eingabe wird mit beendet, *at* meldet dann noch die vergebene Jobnummer.

Wenn z.B. nachts um zwei Uhr zuerst eine Abonnentenliste sortiert und danach ausgedruckt werden soll, könnte das Ganze wie folgt aussehen:

```
at 2:00
sort /usr/abos/Aboliste >/usr/abos/Aboliste.sort
lp /usr/abos/Aboliste.sort
```

Da erfahrungsgemäß immer wieder gleiche Arbeiten auftreten, die in unregelmäßigen Abständen erledigt werden müssen, kann man sich diese Jobs mit einem Editor in eine Datei schreiben und *at* diese Datei abarbeiten lassen. Man legt also eine Datei mit dem Namen *Abos.Drucken* an, deren Inhalt das obige Beispiel ohne die erste Zeile ist. Dann gibt man ein:

```
at 2:00 < /usr/abos/Abos.Drucken
```

So unkompliziert Ihnen *at* bis hierher entgegengetreten ist, so vielfältig sind die Möglichkeiten zur Eingabe der Uhrzeit und des Datums. Die Uhrzeit kann in den Formaten *HH:MM*, aber auch *HHMM* eingegeben werden. Weiterhin kann das Zwölf-Stunden-Format mit einem nachgestellten *am* bzw. *pm* benutzt werden.

Neben dem amerikanischen Datumsformat *MM/TT/JJ* wird auch das deutsche *TT.MM.JJ* unterstützt. Wer nicht weiß, welches Datum gerade ist, aber in vier Tagen um 22 Uhr seinen Job laufen lassen will, der gibt ein:

```
at 22:00 + 4 days
```

Und wer morgen Nachmittag um 16 Uhr seine Arbeit erledigt sehen möchte, gibt ein:

```
at 16:00 tomorrow
```





Für die Uhrzeit sind außerdem die Werte *midnight* und *teatime* (16 Uhr) gültig – very british ...

*at* neigt auch an anderen Stellen zum Scherzen: Wenn eine Uhrzeit ohne Datum eingegeben wird und diese bereits überschritten ist, bricht *at* lapidar mit folgender Meldung ab:

*at*: Trying to travel back in time

Die Liste der vorhandenen Jobs, die per *at* eingegeben wurden, können mit *atq* angezeigt werden.

Der Systemadministrator sieht alle Jobs, sonstige Benutzer sehen nur ihre eigenen. Die Jobs sind nummeriert; diese Nummernangabe benötigt man, wenn man einen Job löschen möchte.

*atrm* 23

löscht den Job 23 aus der Liste.

### batch

*at*, *atq*, *atrm* und *batch* gehören einer Programmgruppe an. So ist die Vorgehensweise mit *batch* fast identisch mit der von *at*; es wird nur kein Zeitpunkt zum Ausführen der Jobs eingegeben. *batch* arbeitet die Jobs ab, sobald die Systemauslastung unter einen bestimmten Schwellwert gefallen ist.

Die mit *batch* eingegebenen Jobs befinden sich in der gleichen Warteschlange wie die von *at*. Sie werden also auch von *atq* angezeigt bzw. von *atrm* gelöscht.

### cron

Wenn bestimmte Jobs in fest definierten Zeitintervallen abgearbeitet werden sollen, ist *cron* die richtige Wahl. Jobs dieser Art gibt es viele: So kann z.B. jede Nacht eine Archivdatei geschrieben werden, welche die veränderten und neu erzeugten Dateien enthält, oder es sollen am 1. und 15. des Monats Mahnungen gedruckt werden, oder montags morgens sollen alle Dateien, die in der letzten Woche zu Testzwecken erzeugt wurden und die mit *bubu* beginnen, gelöscht werden.

*cron* besteht aus zwei Programmen: Zum einen wird der Dämon *cron* beim Start von Linux geladen und bleibt unsichtbar im Hintergrund, um jede Minute nachzusehen, ob es Arbeit gibt.

*cron* wird bei der SuSe-Slackware-Installation automatisch gestartet, so dass der Benutzer hier keine weiteren Installationen vornehmen muss.

Zum anderen gibt es das Programm *crontab*, mit dem die Daten eingerichtet werden können. *cron* -e ruft den Editor *vi* auf, damit Einträge hinzugefügt, geändert oder gelöscht werden können.



## Systemverwaltung

---



`cronab -l` gibt eine Liste der aktuellen Einträge aus, und `cronab -r` löscht die Datei, die diese Daten enthält. Man muss die Liste nicht zwangsläufig mit der Option `-e` erstellen, sie kann auch in eine beliebige Datei geschrieben werden, die dann `cronab` übergeben wird, das mit diesen Daten seine Umgebung automatisch einrichtet.

Die Einträge in der `cron`-Datei umfassen jeweils eine Zeile und enthalten sechs Parameter. Davon sind fünf für die Spezifikation des Zeitpunkts reserviert. Der sechste Parameter ist der Befehl selbst, der ausgeführt werden soll. Ein `cron`-Eintrag besitzt also folgende Syntax:

`m s T M W Befehl`

Wichtig ist, dass zwischen den Zeitangaben und dem Befehl jeweils ein Leerzeichen steht (siehe auch unten stehendes Beispiel). Die Parameter und ihre erlaubten Werte sind folgende:

- die Minute *m* von 0 bis 59
- die Stunde *s* von 0 bis 23
- der Tag *T* von 1 bis 31
- der Monat *M* von 1 bis 12
- der Wochentag *W* von 0 bis 6, wobei Sonntag durch 0 und Sonnabend durch 6 dargestellt wird

Der Stern `*` steht bei allen fünf Parametern für alle jeweils möglichen Werte.

Wenn also jede Nacht um zwei Uhr die `tar`-Datei `/usr/Sicherung` mit allen am Tage erstellten oder veränderten Dateien auf den neuesten Stand gebracht werden soll, muss der Eintrag folgendermaßen aussehen:

`0 2 * * * tar -u -f /usr/Sicherung /*`

An jedem Tag des Monats, in jedem Monat und an jedem Wochentag wird demnach die Sicherung um zwei Uhr und null Minuten ausgeführt.

Dass der arme Rechner unnötigerweise sonnabends und sonntags arbeitet, ist ein Schönheitsfehler, der behoben werden kann. `cronab` gestattet für jeden Eintrag mehrere Werte, die durch Kommata getrennt werden müssen. Für die Arbeit an fünf Werktagen lautet der Eintrag jetzt:

`0 2 * * 1,2,3,4,5 tar -u -f /usr/Sicherung /*`

Wenn am 1. und 15. jedes Monats nachts um Viertel nach drei jeweils Mahnungen ausgedruckt werden sollen, kann der Eintrag wie folgt aussehen:

`15 3 1,15 * * 1p /usr/abos/Mahnungen`



## **mount und umount**



Wer die Liste der zu erledigenden Arbeiten und ihre Zeitpunkte mit *crontab -e* angelegt hat, muss jetzt speichern und den Editor verlassen. *crontab* richtet automatisch seine Umgebung ein.

Wer diese Informationen in einer Datei abgelegt hat, muss *crontab* sie übergeben:

```
crontab Dateiname
```

Hierbei überschreibt *crontab* ältere Befehlslisten. Die neue Liste wird von *crond* sofort – also spätestens in einer Minute – überprüft, es müssen keinerlei weitere Initialisierungsvorgänge vor-genommen werden.

## **mount und umount**

Da das Linux-Dateisystem datei-, und nicht geräteorientiert ist, müssen alle Arten von Datenträgern des Rechners mit *mount* angemeldet werden.

Für die Linux-Partitionen wird dieses automatisch beim Hochfahren des Systems erledigt. Die notwendigen Informationen hat das System zum einen durch die Einstellungen des Setup-Programms von Slackware erhalten, zum anderen wird die Root-Partition von LOADLIN oder LILO übergeben.

Wenn man aber Laufwerke mit wechselbaren Medien wie CD-ROM-Laufwerke, Disketten-laufwerke oder Wechselplatten besitzt, müssen diese bei jedem Wechsel des Mediums abge-trennt und wieder angebunden werden. Dieses geschieht mit *umount* und *mount*.

Auch muss die DOS- oder OS/2-Partition manuell gemountet werden, wenn das automatische Mounten im Setup verneint wurde.

*mount* hat für Geräte, die dem System bekannt sind, eine schlichte Syntax: Es muss nur der Mount-Point angegeben werden, also das Verzeichnis, in dem sich die Dateien des betreffenden Geräts auffinden lassen sollen.

Wenn man im Setup angegeben hat, dass die DOS-Partition im Verzeichnis */dos* gemountet werden soll, so muss nur Folgendes eingegeben werden:

```
mount /dos
```

Der umgekehrte Weg funktioniert genauso unproblematisch. Die DOS-Partition wird wie folgt vom Dateisystem abgetrennt:

```
umount /dos
```

Leider unterläuft der Slackware-Installation eine Nachlässigkeit: Obwohl sie Linux vom CD-ROM-Laufwerk installiert, wird nicht eingetragen, dass ein solches Laufwerk im System vorhanden ist. Eine evtuell eingelegte CD wird zwar beim Hochfah-



## Systemverwaltung

---



ren des Systems erkannt, man kann sie auch mit *umount* abmelden, beim nächsten Mount-Versuch mit *mount /cdrom* meldet *mount* jedoch lakonisch, dass es keinen Eintrag für das CD-ROM-Laufwerk findet ...

Die Laufwerke, die gemountet werden können, sowie ihre speziellen Daten werden in der Datei */etc/fstab* gehalten. Hier muss mit einem Editor folgende Zeile eingegeben werden:

```
/dev/mcd /cdrom iso9660 ro 0 0
```

Dabei stellt */dev/mcd* den „special file“ für das Mitsumi-Laufwerk dar; für andere Laufwerke muss die entsprechende Datei eingesetzt werden, also für das Goldstar-Laufwerk */dev/gscd0*, für das Sony CDU 535 */dev/cdu535* etc.

Danach lässt sich das CD-ROM-Laufwerk mit *mount /cdrom* anbinden.

Man kann aber auch die erweiterte Syntax von *mount* benutzen, bei der Treiber, Dateisystem und weitere Optionen angegeben werden.

```
mount Treiber Optionen [-t Dateisystem] Mount-Point
```

Für (wiederum) das Mitsumi-Laufwerk würde die erweiterte Syntax folgendermaßen aussehen:

```
mount /dev/mcd -r -t iso9660 /cdrom
```

Die wichtigsten Optionen sind *-r* wie hier für *read only* oder *-w* für *read/write*.

Die wichtigsten Werte für die Angabe des Dateisystems sind:

- *iso9660* für CD-ROM-Laufwerke,
- *msdos* für eben dieses,
- *htfs* für das Dateisystem von OS/2,
- *ext2* für das Linux-Standarddateisystem,
- *vfat* für Windows mit Dateinamenserweiterungen.

Durch die Eingabe von

```
mount -a
```

werden schließlich alle Laufwerke gemountet, die in der Datei */etc/fstab* angegeben sind.





## Dämonen

Dämonen (engl. *daemons*) sind weder furchterregend noch peinigend, sondern eher eine sehr praktische Einrichtung der UNIX- und folglich auch der Linux-Welt: In aller Regel werden diese Programme beim Hochfahren von Linux gestartet, und bis auf die Meldung „Starting daemons“ merkt der Benutzer kaum etwas von ihrer Existenz.

Wenn aber plötzlich die Festplatte anfängt zu arbeiten, ohne dass man eine entsprechende Eingabe getätigter hat, ist einer dieser Belzebuben am Werk.

Dämonen arbeiten im Orkus – kein Benutzer kann sie direkt sehen. Deshalb sind sie auch nicht im Besitz einer Konsole wie normale Programme. Bei der Eingabe von `ps -xu` wird deshalb in der Spalte *TTY* auch ein Fragezeichen ausgegeben.

*crond* (s.o.) ist ein typischer Dämon: Er schlafst den Großteil der Zeit und sieht jede Minute nach, ob die Zeit gekommen ist, eine Arbeit aus der *crontab*-Liste zu verrichten. Falls ja, fängt mit einiger Wahrscheinlichkeit die Festplatte an, aktiv zu werden, und der Benutzer wundert sich. Falls nicht, so legt *crond* sich wieder schlafen.

Im Folgenden sind die wichtigsten Dämonen aufgeführt, von denen es eine große Anzahl für die verschiedensten Zwecke gibt.

### ***mountd***

Falls ein Linux-Rechner als NFS-Server arbeitet, arbeitet *mountd* als NFS-Mount-Dämon. Er exportiert seine Verzeichnisse an die Clients, so dass diese auf das Dateisystem des Servers zugreifen können.

### ***nfsd***

Auch dieser Dämon ist für den Betrieb eines Linux-Rechners als NFS-Server gedacht. Wenn Clients eine bestimmte Datei anfordern, stellt *nfsd* diese zur Verfügung.

### ***crond***

*crond* erledigt vom Benutzer definierte Arbeiten in frei wählbaren Zeitabständen. *Dieser Dämon* ist weiter oben in diesem Kapitel beschrieben.

### ***syslogd***

*syslog* führt Buch über Systemmeldungen. Gesteuert wird er über die Datei */etc/syslog.conf*. Dort wird auch angegeben, wo *syslogd* seine Meldungen ablegt; die Voreinstellung ist das Verzeichnis */usr/adm*, die Dateinamen lauten *messages*, *debug* und *syslog*.

## Systemverwaltung

---



### ***klogd***

*klogd* war in früheren Versionen ein Bestandteil von *syslogd*. Um die Kernel-Meldungen klar von anderen Systemmeldungen zu trennen, wurde *klogd* als eigenständiger Dämon entwickelt.

### ***inetd***

*inetd* ist ein Dämon, der an den Internetsockets lauscht. Wenn eine neue Verbindung vorge-funden wird, entscheidet *inetd*, welcher weitere Dämon gestartet werden muss, um den ent-sprechenden Dienst bereitzustellen. So muss für die verschiedenen Netzdienste nur ein einzi-ger Dämon laufen, nämlich *inetd*.

### ***lpd***

*lpd* ist die Abkürzung für **l**ine **p**rinter **d**aemon. Beim Start entnimmt *lpd* der Datei */etc/print-cap* Informationen über die angeschlossenen Drucker.

Wenn eine Anforderung an *lpd* vorliegt, eine Datei zu drucken, überträgt *er* diese Datei in den Spool-Bereich, um sie von dort an den Drucker weiterzureichen.

### ***kerneld***

Seit einiger Zeit brauchen Teile des Kernels nicht mehr „fest verdrahtet“ zu werden, sondern sie können als Modul hinzugeladen werden. Dies ist gedacht, um zum einen den Kernel relativ klein zu halten, auf der anderen Seite aber nicht auf selten benutzte Geräte oder Merkmale verzichten zu müssen.

So sind die Treiber für CD-ROM- oder Bandlaufwerke typische Beispiele für Module. Module müssen von Hand ge- und entladen werden, wenn nicht *kerneld* seinen Dienst versieht.

Wenn *kerneld* aber seinen Dienst versieht, geschieht dies automatisch: Falls das gewünschte Modul nicht im Arbeitsspeicher vorhanden ist, lädt *kerneld* es bei einer Anfrage. Wenn das Modul eine bestimmte Zeit lang nicht aktiv war – die Standardeinstellung beträgt eine Minute –, wird das Modul wieder entladen. Der Benutzer bemerkt davon nichts.





# Konfiguration und Kompilierung des Kernels



|                      |            |
|----------------------|------------|
| Konfiguration        | <b>396</b> |
| Das Kompilieren      | <b>397</b> |
| Kernel mit Modulen   | <b>398</b> |
| Große Kernel         | <b>398</b> |
| Die Kernel-Parameter | <b>399</b> |



**8**

## Konfiguration und Kompilierung des Kernels

---



Im folgenden Abschnitt werden Ihnen die für die Neukonfiguration und -kompilierung des Kernels notwendigen Schritte und die Optionen, die hierfür zur Verfügung stehen, vorgestellt. Dieses zugegebenermaßen etwas trockene Kapitel werden Sie sicherlich nicht von vorne bis hinten studieren. Sie werden aber, wenn Sie aus einem der im folgenden aufgeführten Gründe einmal in die Situation kommen, Ihren Kernel rekonfigurieren zu wollen (oder gar zu müssen ...), die hier enthaltenen Informationen als eine unersetzliche Fundgrube an Informationen schätzen lernen. Die verschiedenen mitgelieferten Kernel reichen in den meisten Fällen aus, um ein Linux-System zu installieren. Wer jedoch außer Standardkomponenten weitere Hardware im Rechner benutzen will, wie z.B. Netzwerkkarten, Soundkarten, Bandlaufwerke oder ISDN-Karten, der muss die entsprechenden Treiber in seinen Kernel einbinden. Es wird eine riesige Auswahl an Treibern geboten, so dass der Kernel genau passend auf die individuellen Hardware-Anforderungen maßgeschneidert werden kann. Das kann ein vorkonfigurierter Kernel der CD-ROM niemals leisten. Die Situation innerhalb der vielfältigen Softwareumgebungen, in denen Linux sich befinden kann, ist ähnlich: Auf vielen Rechnern sollen sich neben Linux auch noch DOS, Windows 9x, NT oder OS/2 befinden, und die entsprechenden Festplatten sollen unterstützt werden. Der Linux-Rechner kann sich innerhalb eines NOVELL- oder APPLE-Netzes befinden und dort als Client oder Server arbeiten. Auch kann der Linux-Rechner als Firewall oder Server im WWW (siehe *Kapitel 13 Linux im Netzwerk*) genutzt werden. Diese Fähigkeiten erlangt Linux aber erst, nachdem ein Kernel neu konfiguriert und danach kompiliert wurde.

## Konfiguration

Es gibt drei Programme, die der Konfiguration des Kernels dienen:

*config*: Zeigt Ihnen stur einen Kernel-Parameter mit Kurzbeschreibung nach dem anderen an, wobei schlicht mit **[Y]** oder **[N]** geantwortet wird. Oft kann auch mit **[M]** geantwortet werden: Der Treiber wird dann als Modul geladen, d.h., er wird nicht fest im Kernel verankert, sondern bei Bedarf automatisch hinzugeladen bzw. nach einer bestimmten Zeit ohne Nutzung automatisch wieder entfernt. Module eignen sich gut für bestimmte Bestandteile, die man nicht missen möchte, die aber nicht permanent benötigt werden. *config* eignet sich zum Neuaufbau eines Kernels und für den Anfänger. Alle Möglichkeiten werden angeboten, es kann nichts vergessen werden.

*menuconfig*: Bietet in den bekannten semi-grafischen Fenstern (wie z.B. in YaST) menügesteuert die gleichen Optionen. Die jeweilige Wahl wird angezeigt und kann jederzeit geändert werden, fehlerhafte Eingaben können im Gegensatz zu *config* überschrieben werden. Verschiedene Konfigurationsdateien können geladen werden, so dass verschiedene Kernel konfiguriert werden können. *menuconfig* eignet sich zum einen, wenn nur geringe Änderungen im Kernel vorgenommen werden sollen, zum anderen wendet es sich an den erfahrenen Benutzer, da es die Abfrage aller Parameter nicht erzwingt. Falls Anfänger mit diesem Programm arbeiten, sollten sie alle Menüpunkte nacheinander abarbeiten, so dass nichts vergessen wird.



## Das Kompilieren



*xconfig*: arbeitet unter X Window und leistet das gleiche wie *menuconfig* unter der bekannten grafischen Oberfläche. Alle Aktionen, die die Konfiguration und die Kompilierung des Kernels betreffen, finden in dem Verzeichnis */usr/src/linux* statt, in das man als Erstes wechseln sollte. Dort muss dann Folgendes eingegeben werden:

```
make config
```

oder

```
make menuconfig
```

oder unter X Window

```
make xconfig
```

Informationen zu den einzelnen Kernel-Parametern finden Sie in aller Ausführlichkeit auf den nächsten Seiten. Nachdem man sich durch die einzelnen Fragen bzw. Menüpunkte „durchgetastet“ hat, werden die ausgewählten Parameter in die Datei *.config* geschrieben, die zur Steuerung der Kompilierung dient.

## Das Kompilieren

Das Kompilieren des Kernels erfolgt in drei Schritten. Zuerst werden alle alten temporären Dateien des letzten Kompilationslaufs gelöscht:

```
make clean
```

Als Nächstes werden die einzelnen Abhängigkeiten neu berechnet:

```
make dep
```

Diese beiden Schritte benötigen jeweils nur kurze Zeit. Der eigentliche Vorgang des Kompilierens dauert länger und wird wie folgt gestartet:

```
make zImage
```

Hier kann man sich nun freuen, wenn man neuere Hardware besitzt: Das Kompilieren eines Kernels auf einem Rechner mit neuerem Pentium dauert nur wenige Minuten, auf einem älteren 100-MHz-Pentium-II immerhin schon ca. zehn Minuten. Auf einem 486er mit 50 MHz benötigte man seinerzeit ca. 35 Minuten. Von 386ern mit wenig RAM wird von stundenlangen Läufen berichtet, was wir aber (glücklicherweise) nicht überprüft haben. Um die Abarbeitung jedes dieser einzelnen Schritte nicht überwachen und den jeweils nächsten Schritt „manuell“ einleiten zu müssen, können Sie diese drei Befehlszeilen wie im Kapitel 4 Benutzerumgebung (*Job-Control*) beschrieben auch wie folgt aufrufen:

```
make clean && make dep && make zImage
```



## Konfiguration und Kompilierung des Kernels

---



Selbstverständlich können Sie aber auch (statt einen Kaffee trinken zu gehen, während diese drei Schritte abgearbeitet werden) auf einer anderen virtuellen Konsole weiterarbeiten. Auf jeden Fall sollten diese Hinweise genügen, um bei der Konfiguration des Kernels sorgfältig vorzugehen, damit dann nach dem Kompilieren nicht festgestellt werden muss, dass man etwas vergessen hat. Nach dem Vorgang findet sich der neue Kernel unter dem Namen `zImage` in dem Verzeichnis `/usr/src/linux/arch/i386/boot`. Dies ist aber nur die „Standardablage“, nachdem der Kernel neu erstellt wurde. Falls man mit LILO arbeitet, liegt der Kernel normalerweise im Verzeichnis `/boot` unter dem Namen `vmlinuz`. Wenn mit LOADLIN gearbeitet wird, liegt der Kernel in dem entsprechenden DOS-Verzeichnis, also z.B. `C:\LOADLIN`. In beiden Fällen sollte zuerst der alte Kernel umbenannt werden, denn es besteht immer die Möglichkeit, dass der neue nicht laufen will. Erst dann sollte das gerade erstellte `zImage` in das gewünschte Verzeichnis (also in `/boot` bzw. in das Verzeichnis `LOADLIN`) kopiert und evtuell umbenannt werden. Wenn Sie mit LILO arbeiten, ist noch ein sehr wichtiger Schritt durchzuführen: Der LILO muss durch den Aufruf des Befehls `make zlilo` neu installiert werden. Unter bestimmten Umständen, die aufgrund der Vielzahl der Kernel-Parameter, die Auswirkungen auf den Bootvorgang haben und daher für eine genaue Analyse und somit Darstellung an dieser Stelle zu unspezifisch sind, startet das Betriebssystem gegebenenfalls nicht mehr.

## Kernel mit Modulen

Nach Überprüfung der Lauffähigkeit des neuen Kernels können Sie, falls Bestandteile des Kernels als Modul mit der Option `[M]` in `config` bzw. `menuconfig` definiert worden sind, jetzt noch diese Module erzeugen bzw. installieren. Dies wird mit den folgenden beiden Befehlszeilen erledigt:

```
make modules
make modules_install
```

Der DOS-Emulator und einige andere Module benötigen die Datei `System.map` im Wurzelverzeichnis, die sich nach dem Kompilieren im Verzeichnis `/usr/src/linux` befindet:

```
cp /usr/src/linux/System.map /
```

Jetzt sollte sich der neue Kernel mittels LILO oder LOADLIN starten lassen.

## Große Kernel

Wer – aus welchen Gründen auch immer – nicht mit Modulen arbeiten will oder sehr viele Treiber in seinem Kernel benötigt, läuft Gefahr, dass das Kompilieren nicht regulär beendet wird, sondern sich das System mit einer Meldung wie etwa „System is too big“ oder „Kernel too big“ meldet und den Vorgang abbricht. Hier ist Abhilfe geschaffen worden. Statt `make zImage` gibt man ein:

```
make bzImage
```

Die anderen Vorgänge bleiben unverändert. Durch `make bzImage` erhält der Kernel u.a. auch eine Kennung, dass er *high* geladen wird.





## Die Kernel-Parameter

Wer ein Buch schreibt, der hat nicht nur mit dem Objekt seiner Begierde – in diesem Falle Linux – zu kämpfen, sondern auch mit anderen Umgebungsvariablen. Eine davon ist die Kalkulation des Verlags und – logischerweise damit zusammenhängend – die Seitenzahl dieses Buches. Da sich an sehr vielen Fronten bezüglich Linux viele Neuerungen und Erweiterungen ergeben habe, die gerne von uns beschrieben worden wären, drohte das Buch aus allen Nähten zu platzen. Wir haben uns deshalb entschlossen die Kernel-Parameter als Kapitel zwar weiterhin zu pflegen, diese aber nur als „electronic matter“ in Form einer PDF-Datei auf der zweiten beiliegenden CD auszuliefern. Diese Entscheidung macht das Buch um ca. 200 Seiten schlanker und entsprechend preiswerter. Um eine PDF-Datei zu lesen, benötigt man den Acrobat Reader, der den meisten Linux-Distributionen beiliegt, aber auch auf anderen Plattformen zur Verfügung steht. Vorteilhaft ist außerdem, dass man damit nicht nur – wie in einem Buch – blättern, sondern auch gezielt nach Begriffen suchen kann, was gerade für dieses Kapitel von unschätzbarem Wert ist.

Um dem Leser, der gerade keinen Rechner zur Hand hat, einen kurzen Überblick über die teils gravierenden Änderungen und Verbesserungen des Kernel 2.4.x zu geben, führen wir sie hier noch einmal kurz Stichwortartig auf.

### Prozessoren

Die Leistungsmerkmale neuerer Prozessoren werden gezielt unterstützt. Das **SMP** (**Symmetric Multi Processing**) ist grundlegend überarbeitet und verbessert worden. Ein Wehrmutstropfen bleibt: Der 386er wird per Standardeinstellung nicht mehr unterstützt.

### Laptops

Laptops sind inzwischen aus ihrer exotischen Ecke entrückt und haben sich als Standardwerkzeug ihren Platz erobert. Zum einen werden deshalb die verschiedensten PCMCIA-Kartentypen unterstützt, zum anderen auch fast alle einschlägigen Dienste über diese kleinen Kärtchen. Außerdem ist das Powermanagement für Laptops deutlich erweitert worden.

### USB

Während sich Linux anfänglich mit einigen schnellen Hacks in die USB-Welt begab, ist inzwischen der Treiber grundlegend und für alle denkbaren Geräte überarbeitet worden. USB stellt kein Problem mehr dar.



## Konfiguration und Kompilierung des Kernels

---



### Sound

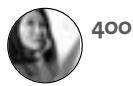
Nachdem Atari noch seine allerletzte Hochburg in Musikerkreisen besaß, scheint dieses Terrain langsam aber sicher durch Linux besetzt zu werden. Entsprechend ist der Zuwachs an Treibern für Geräte aller Art.

### Video for Linux

Seinen Linux-Rechner als Videorecorder zu benutzen, funktioniert bereits seit langerem. Aber erst die leistungsstarken Prozessoren und Grafikkarten der letzten Zeit ermöglichen einen ruckelfreien Genuss, so dass sich dies zu einer Standardanwendung entwickelt.

### ... und, und, und

Außerdem gibt es Treiber für Embedded Systems, für Hochgeschwindigkeitsnetze wie Ethernet 1000 Mbit oder ATM, Treiber für das Telefonieren übers Internet, IrDA-Treiber für den Infrarotschluss und Unterstützung für den Fire-Wire-Anschluss u.a. von Videokameras am Linux-Rechner. Es fehlt aber noch das Interface für meinen Toaster ...





# **Windows für Linux? – Emulatoren**



Warum werden Emulatoren  
benutzt? **402**  
VMware **404**



**9**

## Windows für Linux? – Emulatoren

---



In einigen Situationen sieht sich auch der eingefleischte Linux-Fan gezwungen, auf andere Betriebssysteme zurückzugreifen, um spezielle Software einzusetzen zu können. Programme werden nämlich immer für bestimmte Systemplattformen entwickelt.

### Warum werden Emulatoren benutzt?

Auch wenn sich heute immer mehr Softwareschmieden darum bemühen, ihre Produkte für verschiedene Betriebssysteme anzubieten, so muss doch bedacht werden, dass dann für jede Plattform ein eigenes Produkt angeboten wird. Dieses teilt mit den anderen die Funktionalität, ist aber innerhalb der Produktfamilie das einzige, das auf dem entsprechenden Betriebssystem lauffähig ist. Bedenken Sie bitte, dass eine der wesentlichsten Aufgaben eines Betriebssystems die ist, Programme auf der vorhandenen Hardware ablaufen zu lassen. Umgekehrt fordert dies von den Anwendungsprogrammen, dass sie die Vorgaben des Betriebssystems erfüllen. Diese bestehen darin, für spezielle Aufgaben, wie den Zugriff auf Festplatte, Grafikkarte/Bildschirm, Speicher oder Drucker, entsprechende, vom Betriebssystem bereitgestellte Systemaufrufe zu benutzen.

Diese Schnittstelle zwischen Anwendungsprogrammen und Hardware ist aber bei den diversen Betriebssystemen, die für den PC verfügbar sind, unterschiedlich ausgelegt. Dies hat zum Teil historische Gründe – UNIX als Urahn von Linux ist nicht für PCs entwickelt worden, DOS als Urahn aller Redmondschen Betriebssysteme dagegen ist das ursprüngliche PC-Betriebssystem. Auf der anderen Seite werden an die verschiedenen Betriebssysteme unterschiedliche Anforderungen gestellt. Linux etwa ist eine Serverplattform par excellence – ein Bereich, in dem Microsoft noch erheblichen Nachholbedarf hat. Linux stellt für diese Funktionalität Systemaufrufe zur Verfügung, die in anderen Umgebungen nicht zu finden sind, da sie dort nicht gebraucht werden.

Nun stehen Sie bisweilen aber vor der Situation, dass unter Linux Programme einsetzen wollen, die (noch) nicht in einer Linux-Version existieren. Dies kann zum Beispiel der Fall sein, wenn man

- eines der vielen noch nicht portierten Spiele spielen will,
- eine Spezialanwendung, etwa ein CAD-Programm, benutzen will, dessen Funktionalität man in langjähriger Praxis erlernt hat und das nur in einer DOS- oder Windows-Version vorliegt (hier einen Umstieg auf ein äquivalentes Linux-Produkt vorzunehmen, ist wegen der Komplexität solcher Programme oft zu aufwändig),
- Dokumente bearbeiten muss, die in einer anderen Umgebung mit einem ungewöhnlichen Dateiformat gespeichert wurden (auch die unter Linux zur Verfügung stehende Standardsoftware wie StarOffice, ApplixWare oder WordPerfect ist damit überfordert, die Texte aller mehr oder weniger bekannten Textverarbeitungen zu lesen – denken Sie nur an die vielen *Probleme*, die



## Warum werden Emulatoren benutzt?



entstehen, wenn ein Standardprogramm unter Windows in einer neuen Version Daten lesen soll, die mit dem gleichen Programm, aber in einer älteren Version erstellt wurden).

Zur Lösung solcher Probleme gibt es prinzipiell zwei Möglichkeiten. Entweder reservieren Sie bei der Installation auf der Festplatte Platz für das Betriebssystem (oder die Betriebssysteme!), unter dem die gewünschten Programme laufen, oder Sie benutzen so genannte „Emulatoren“. Die erste Variante hat den entscheidenden Nachteil, dass der Rechner neu gebootet werden muss, wann immer Sie zwischen Linux und diesem Programm wechseln wollen. Auf einem PC kann (jedenfalls bis heute) nun mal immer nur je ein Betriebssystem zur Zeit laufen, dieses verwaltet schließlich die Hardware.

Dieser Nachteil lässt sich mithilfe eines Emulators ausschalten.

Die meisten Emulatoren, wie der DOS-Emulator, arbeiten nach dem gleichen Prinzip: Die Anwendungsprogramme laufen unter der Aufsicht des Emulators. Systemaufrufe der Anwendung werden von diesem abgefangen und in einen oder mehrere Systemaufrufe umgewandelt, die Linux benutzt und die zum gleichen Ergebnis führen wie der ursprüngliche Systemaufruf unter dem ursprünglichen Betriebssystem.

Dies ist eine prinzipiell lösbarer Aufgabe, wenn die Anwendungssoftware „sauber“ programmiert worden ist, d.h. die vorgesehenen Systemaufrufe auch wirklich benutzt. Sie können sich aber sicher vorstellen, dass dies eine recht aufwändige Aufgabe ist, und so nimmt es nicht Wunder, dass etwa WINE, ein Emulator für Windows 95, zurzeit über ein Alpha-Stadium noch nicht hinauskommt.

Prinzipielle Probleme dagegen hat der DOS-Emulator. Gerade unter DOS haben sich die Anwendungsprogrammierer sehr viele Freiheiten herausgenommen und ihre Programme am Betriebssystem vorbei auf die Hardware zugreifen lassen (was unter diesem Single-Tasking-System deshalb möglich war, weil im Prinzip immer nur ein Programm die Systemressourcen beanspruchen konnte).

Dies betrifft insbesondere eine Reihe von DOS-Spielen. Auch wenn der DOSEMU als sehr stabil bekannt ist – er leidet darunter, dass gerade diese hardwarenah programmierten Spiele unter ihm nicht laufen.

Nicht nur Spiele wurden unter DOS derart am System vorbei programmiert, auch Standardsoftware. Microsoft hatte in den späten 80er Jahren erhebliche Schwierigkeiten, Word für DOS nach OS/2 zu portieren (was damals auch noch von Microsoft entwickelt wurde!). So wurde etwa der Bildschirmaufbau durch direktes Ansprechen des Bildschirmspeichers erledigt. Unter einem Multitasking-System sind solche Tricks aber nicht verfügbar, eine Nutzung der bekannten Systemaufrufe brachte aber einen erheblichen Performance-Einbruch, was die damaligen User noch nicht so klaglos hingenommen haben.



Hinweis



## Windows für Linux? – Emulatoren

---



### VMware

Seit zirka zwei Jahren sorgt nun ein Emulator für Furore, der einen anderen Ansatz verfolgt, als die Systemaufrufe abzufangen und umzusetzen. Er wurde von der Firma VMware aus Palo Alto in den USA entwickelt und trägt den gleichen Namen: VMware.

Gerüchten zufolge haben hier ein Informatik-Professor und einige seiner Studenten eine Idee in die Tat umgesetzt und vor rund einem Jahr damit begonnen, diese auch zu vermarkten. Dies ist natürlich ein entsprechender Nachteil: VMware ist ein kommerzielles Produkt. Es kostet als VMware Express 99 Dollar. Man kann aber auch tiefer in die Tasche greifen und eine „Luxus“-Version für 299 Dollar erstehen. Trotz der Tatsache, dass es sich bei VMware um ein (teures) kommerzielles Produkt handelt, wird es in diesem Buch beschrieben,

- weil die Art der Emulation ein mehr als interessanter Ansatz ist,
- weil wir uns von der Stabilität im professionellen Einsatz überzeugen konnten (mehrere Arbeitsplätze in der Firma, in der zwei der Autoren ihre Brötchen verdienen, sind als Linux-Rechner mit VMware ausgestattet, um den dortigen Anwendern Zugang auf Windows-Software zu ermöglichen – dies läuft seit nunmehr eineinhalb Jahren zur Zufriedenheit aller Beteiligten),
- vor allem aber, weil VMware für prinzipiell alle auf PC verfügbaren Betriebssysteme einsetzbar ist, ohne dabei speziellen Einschränkungen unterworfen zu sein, wie der DOSEMU (auch unseren Lieblingsflipper – ein Programm aus alten DOS-Tagen – konnten wir so wieder zum Leben erwecken).

Mit besonderem Interesse verfolgen die Autoren im Übrigen die Entwicklung einer freien Software mit dem gleichen Ansatz, der FreeMWare. Interessenten seien auf die Homepage dieses Projekts verwiesen: <http://www.plex86.org>.

### Funktionsprinzip von VMware

Während Emulatoren wie WABI, WINE oder der DOSEMU der unter ihrer Regie laufenden Anwendung das jeweils benötigte Betriebssystem vorgaukeln (emulieren), greift VMware das Problem an der Wurzel an und emuliert die Hardware oder – präziser formuliert – einen PC.

Grundsätzlich nutzt es hier ein Feature des Intel-Prozessors, das mit dessen Version 386 eingeführt wurde, den so genannten „Virtual Mode“, der es einem Betriebssystem gestattet, mehrere virtuelle Maschinen bereitzustellen. Unter Windows 3.x und 9x wird dies etwa zur Bereitstellung von DOS-Fenstern genutzt. Diese laufen auf einem eigenen, virtuellen PC.





Während sich allerdings solche DOS-Fenster mit virtueller CPU und virtuellem Speicher begnügen müssen (die Festplatte ist nicht virtuell, sondern real), geht VMware einen Schritt weiter und emuliert weitere Hardware, wie etwa die Festplatte. Dies ist darin begründet, dass jedes Betriebssystem seine eigene Dateiverwaltung verwendet und somit beim Öffnen und Speichern von Daten anders auf die Platte zugreift. Während Linux sehr wohl in der Lage ist, Daten von einer Windows-Platte zu lesen, versteht Windows das Dateisystem von Linux nicht.

Windows NT ist nicht in der Lage, das Dateisystem von bestimmten Windows-9x-Versionen zu lesen, umgekehrt können diese nicht mit NT-Partitionen umgehen!

**Hinweis**

Wenn Sie nun VMware unter Linux starten (es ist zunächst einmal eine Linux-Anwendung wie etwa der Gimp), steht Ihnen ein kompletter PC zur Verfügung, auf dem Sie ein beliebiges genauer: ein auf dem PC lauffähiges Betriebssystem installieren können. Unter diesem Betriebssystem können Sie dann alle dafür konzipierten Programme laufen lassen, sofern die Systemvoraussetzungen des virtuellen PCs dies zulassen. (Im Rahmen der durch den realen PC gegebenen Voraussetzungen wie Hauptspeicher und Plattenkapazität lassen sich diese aber in den meisten Fällen anpassen.)

Dieser PC läuft nun – und das ist das Besondere an der Situation – als Task unter Linux. Sie können also parallel z.B. Windows- und Linux-Anwendungen benutzen. Dieser Abschnitt des Buches ist so entstanden, dass in einem Fenster VMware mit Windows 98 lief und getestet wurde, in einem anderen Fenster mit StarOffice unter Linux der vorliegende Text geschrieben wurde. Ein anderes Beispiel: Die Screenshots von der Installation wurden mithilfe des Gimp erzeugt. Dabei wurde die Installation in einem Fenster mit VMware durchgeführt! Schließlich ist auch Linux ein auf dem PC lauffähiges Betriebssystem und kann somit auf einem virtuellen PC installiert werden.

Windows- oder DOS-Anwendungen können also mithilfe von VMware eingesetzt werden. Das läuft im Großen und Ganzen wie folgt ab:

- VMware emuliert einen PC,
- auf diesem läuft das geeignete Betriebssystem
- und darunter die Anwendung.

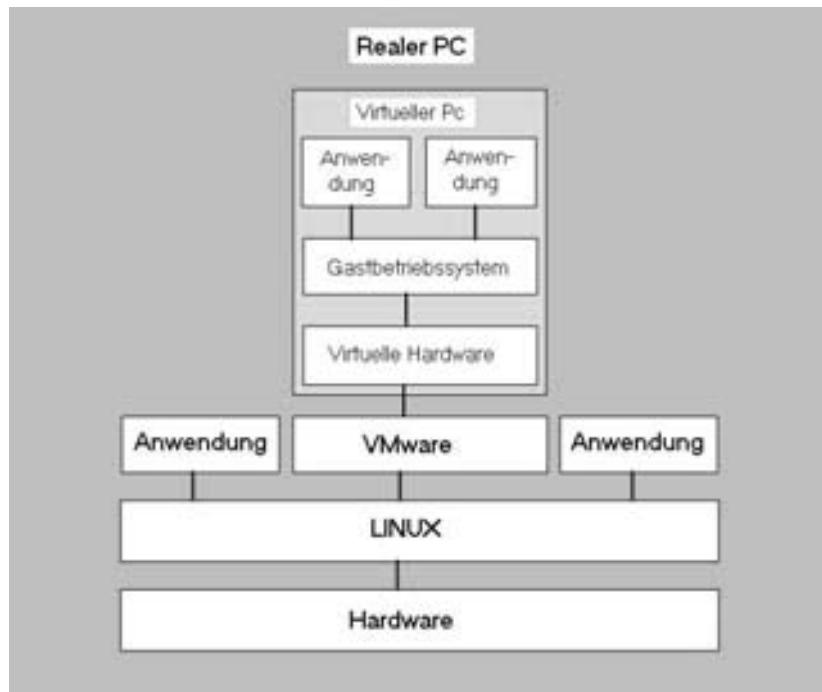
Deren Systemaufrufe wenden sich nun an das Original-Betriebssystem, welches sie an den virtuellen PC weiterleitet. Unter Regie von VMware und letztlich von Linux werden dann Vorgänge wie Speichern, Öffnen oder Drucken von Dateien auf der realen Hardware ausgeführt.

## Windows für Linux? – Emulatoren



Abb. 9.1 ○

Funktionsprinzip von  
VMware



### Systemvoraussetzungen

So vielfältig die Möglichkeiten von VMware auch sind, das Produkt setzt doch einiges an Hardware voraus, damit es zufriedenstellend läuft. Der Hersteller nennt als Minimum:

- einen Intel-Pentium-II-Prozessor mit mindestens 266 MHz Taktfrequenz
- 96 MB Hauptspeicher (besser 128 MByte)

Wir haben das Produkt auch auf Notebooks mit geringerer Ausstattung getestet und waren angenehm überrascht. Allerdings sollte zumindest die Taktfrequenz nicht unterschritten werden, dann beginnt VMware tatsächlich zu lahmen.

Zur Steigerung der Performance stellt Ihnen VMware im Übrigen noch die *vmtools* mit einem speziellen X-Server zur Verfügung. Dieser spielt seine Vorzüge vor allem dann aus, wenn Sie VMware im Full-Screen-Modus laufen lassen.

Wichtig ist natürlich auch noch der Platzbedarf auf der Festplatte. Hier schluckt das Programm selbst nur etwa 11 MByte. Für jeden virtuellen PC müssen Sie jedoch Plattenplatz im Verhältnis 1:1 einplanen. Wollen Sie einem solchen virtuellen PC eine virtuelle Festplatte von 2 GByte Kapazität gönnen, dann müssen Sie diese Kapazität





von der Linux-Partition abziehen, auf der Sie installieren. Darauf wird nämlich eine Datei entsprechender Größe angelegt, in die hinein alle Daten geschrieben werden, die Sie unter dem Gast-Betriebssystem auf die virtuelle Festplatte schreiben.

Zusätzlich zu den genannten Hardwarevoraussetzungen müssen Sie als Betriebssystem Linux sowie eine grafische Benutzeroberfläche eingerichtet haben, einen X-Server mit einem Window-Manager wie KDE, Gnome oder dem herkömmlichen fvwm also.

Es gibt seit einiger Zeit auch eine Version für Windows 9x und Windows NT, so dass Nutzer dieser Systeme auch in den Genuss kommen können, Linux auf ihrem Rechnern einzusetzen, ohne sie vorher neu installieren zu müssen.

**Hinweis**

Eine weitere wichtige Voraussetzung soll hier nicht verschwiegen werden: Wollen Sie ein Betriebssystem einsetzen, das nicht frei verfügbar ist, benötigen Sie „selbstverständlich“ auch eine gültige Lizenz dafür.

### **Wann setzt man VMware ein?**

Aus den vorgenannten Überlegungen lässt sich leicht entnehmen, dass einem Einsatz von VMware gewisse (Hardware-)Grenzen gesetzt sind. Man sollte also schon genau überlegen, wann sich sein Einsatz lohnt. Im privaten Bereich kann man bei gegebenen Hardwarevoraussetzungen sicher einen Einsatz erwägen, wenn man DOS- oder Windows-Spiele spielen will. Im ersten Fall ist aber auch zu prüfen, ob nicht der DOSEMU einsetzbar ist.

Gleiches gilt für den Einsatz bestimmter Anwendungssoftware, für die es unter Linux keinen Ersatz gibt oder deren Gebrauch man nicht missen möchte.

Im professionellen Bereich lassen sich auf diese Weise Anwender unterstützen, die mit Windows-Umgebungen vertraut sind und nur allmählich auf Linux-Produkte umgestellt werden sollen.

Entwickler, die plattformübergreifend programmieren, erleichtern sich Ihre Arbeit, indem sie parallel – und ohne zwischendurch ihr System jeweils neu zu booten – auf den verschiedenen Plattformen kodieren, übersetzen und testen.

Auch beim plattformübergreifenden Support zahlt sich diese Konstruktion aus, indem sie verhindert, dass bei Anfragen zu unterschiedlichen Systemen der Rechner neu gebootet werden muss oder aber mehrere Rechner mit unterschiedlichen Systemen parallel laufen müssen.

Schließlich eignet sich VMware – privat und professionell – auch als Sandkasten zum Testen von Software. Anstatt einen Rechner neu zu installieren, um eine neue Systemsoftware zu testen, kann dies nun auf einem virtuellen PC geschehen.

Die Autoren dieses Buches sind aus verständlichen Gründen daran interessiert, unterschiedliche Linux-Distributionen einzusetzen und zu testen. Jedoch werden Sie, die Leser, nicht erwarten, dass wir, die Autoren, jedes Mal, wenn eine neue Red-



## Windows für Linux? – Emulatoren



Hat-, SuSE-, Slackware- oder sonstige Distribution auf dem Markt erscheint, unsere Rechner „platt machen“, um diese neue Version in Ihrem Interesse zu testen. Dies geschieht dann eben in einer durch VMware zur Verfügung gestellten „Sandbox“.

### Bezugsquellen

Wie gelangen Sie nun zu einer solchen Eier legenden Emulations-Wollmilchsau? Besitzer einer SuSE-Distribution (ab Version 6.3) können eine Laufzeitbeschränkte Demoversion verwenden. Diese gilt für einen Monat und ist in der Serie *pay* als Paket *vmware* zu finden. Mithilfe von YaST lässt es sich einfach installieren.

Falls vorhanden, installieren Sie gleich auch das Paket *vmtools* mit.



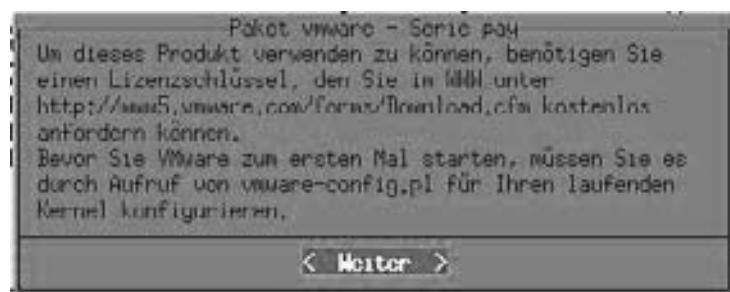
Abb. 9.2

vmware in der Paketauswahl des YaST

Wenn Sie das Paket in der Paketauswahl *vmware* aktivieren, erhalten Sie den Hinweis, dass Sie sich noch eine (kostenlose) Lizenz via Internet bei VMware besorgen müssen (vgl. Abbildung 9.3).

Abb. 9.3

Installationshinweis im  
YaST





Die geschieht am einfachsten über die Internetadresse <http://www.vmware.com/suse>. Hier füllen Sie ein kleines Formblatt mit Ihrem Namen, Ihrer E-Mail-Adresse und einem so genannten „PROMOCODE“ aus. Diesen erhalten Sie, wenn Sie VMware konfigurieren, was weiter unten genauer beschrieben wird.

Innerhalb kürzester Zeit wird Ihnen via E-Mail eine Lizenzdatei zugesandt, die Sie im Heimatverzeichnis des Users, der das Tool einsetzen wird, ins Verzeichnis `.vmware` kopieren müssen.

Falls Sie mit einer anderen Linux-Distribution arbeiten, können Sie natürlich auch eine auf einen Monat Laufzeit beschränkte Demoversion von VMware erhalten. Diese können Sie direkt unter der Internetadresse <http://www5.vmware.com/forms/Download.cfm> abrufen (die 5 nach www ist kein Druckfehler!). Auch für diese Version brauchen Sie einen Lizenzschlüssel.

Wir empfehlen Ihnen im Übrigen, die Homepage der Firma <http://www.vmware.com> etwas genauer zu studieren, da sie sehr informativ ist und eine Reihe von Informationen rund um die Installation und den Einsatz von VMware bietet. Natürlich können Sie auf diesem Wege auch eine Vollversion erwerben.

## Paketinstallation

Falls Sie Ihre Version nicht mit YaST installieren können, weil Sie zum Beispiel keine SuSE-Distribution einsetzen, folgen hier einige Hinweise zur Paketinstallation.

Das Paket liegt als gepackte `tar`-Datei und als `rpm`-Datei vor. Wir gehen davon aus, dass Ihnen der RedHat Package Manager (`rpm`) zur Verfügung steht und empfehlen Ihnen den Download der entsprechenden Version. Liegt es als `tar`-Datei vor, sollten Sie sich zunächst mit den Installationsanleitungen auf der VMware-Homepage vertraut machen. Unter anderem werden Sie die Datei mit

```
tar xvzf dateiname
```

bzw.

```
tar xvf dateiname
```

entpacken müssen. Außerdem sind die entpackten Dateien in geeigneten Verzeichnissen unterzubringen.

Einfacher gestaltet sich die Installation der Pakete mit `rpm`. Arbeiten Sie unter einer grafischen Oberfläche wie KDE, dann öffnen Sie nach erfolgreichem Download des Pakets zunächst ein Terminalfenster. Wechseln Sie dort in das Verzeichnis, in das Sie das Paket kopiert haben. Wechseln Sie Ihre Benutzeridentität und werden Sie mittels

```
su
```



## Windows für Linux? – Emulatoren

---



zum Superuser. Geben Sie dann ein:

```
rpm -Uhv paketname
```

Dies ist im Prinzip genau das, was YaST auch für Sie erledigt, wenn Sie mit diesem Tool arbeiten. Nach erfolgter Installation können Sie den Superuser-Status natürlich wieder verlassen, es empfiehlt sich aber in jedem Fall, zunächst noch die Software zu konfigurieren, da dafür auch *root*-Rechte erforderlich sind.

### Konfiguration des Softwarepakets

Nachdem Sie das Packet installiert haben, müssen Sie es nun noch konfigurieren. Wechseln Sie dazu in das Verzeichnis */etc/vmware*. Hier finden Sie ein Perl-Skript namens *vmware-config.pl*. Dieses können Sie allerdings nur als Superuser ausführen. Wechseln Sie also zunächst mit

```
su
```

Ihre Benutzeridentität, und starten Sie dann das Skript durch Eingabe von:

```
. /vmware-config.pl
```

Beachten Sie, dass der Superuser (User *root*) aus Sicherheitsgründen normalerweise das aktuelle Verzeichnis nicht im Suchpfad hat. Daher sollten Sie es ausdrücklich im Aufruf des Programms angeben (*./*).

#### Hinweis



Die Reihenfolge der nun folgenden Abfragen kann in Abhängigkeit von der Vmware-Version, der Hardwareausstattung (Netzwerkkarte!) und der jeweils gewählten Option variieren.

Das Skript ist interaktiv und erwartet einige Angaben, die im Folgenden kurz besprochen werden. Zunächst fordert es Sie auf, das *EULA* (End User License Agreement) von VMware Inc. zu lesen. Dies geschieht mithilfe eines herkömmlichen Pagers wie *more* oder *less* und kann durch Eingabe von

```
q
```

beendet werden.

Danach will das Skript wissen, ob Sie Netzwerkfunktionalität benutzen wollen. Damit kann Ihr virtueller Rechner in ein Netzwerk eingebunden werden. Dies kann aber auch von Nutzen sein, wenn Sie eine Kommunikation zwischen Gast- und Wirt-Betriebssystem (hier Linux) planen. Eine Netzwerkkarte in Ihrem Rechner ist dafür allerdings Voraussetzung.

```
Do you want to be able to use the network in your Virtual
Machines? [yes]
```





Wie bei den folgenden Abfragen stellt das Skript an dieser Stelle eine Standardantwort in eckigen Klammern zur Verfügung. Diese (hier *yes*) kann durch Drücken von Ü gewählt werden. Wollen Sie keine Netzwerkunterstützung, geben Sie hier einfach

n Ü

ein. Wollen Sie aber Netzwerkunterstützung, kann es sein, dass die vorhandenen Netzwerkmodule von VMware vom aktuellen Kernel nicht akzeptiert werden. Das Skript will dann versuchen, ein geeignetes Modul neu zu erstellen.

```
None of VMware's pre-built vmnet modules is suitable for
your running kernel. Do you want this script to try to
build the vmnet module for your system (you need to have a
C compiler installed on your system)? [yes]
```

Wie Sie der Frage entnehmen können, ist dazu die Existenz eines C-Compilers notwendig. Wenn Sie diesen bei der Linux-Installation nicht ausdrücklich abgewählt haben, sollte er vorhanden sein. Allerdings müssen auch die *Include*-Dateien (ein Teil der Kernel-Quellen) zugänglich sein.

```
What is the location of the directory of C header files
that match your running kernel? [/usr/src/linux/include]
```

Das vorgegebene Verzeichnis entspricht demjenigen, in dem diese Dateien normalerweise bei einer SuSE-Installation zu finden sein sollten. Sollte Ihnen das Skript melden, dass Ihr System ein „*single processor system*“ sei, die *Include*-Dateien jedoch zu einem „*multi processor system*“ gehören, so könnte dies daran liegen, dass Sie die SuSE-Kernel-Quellen installiert haben, VMware diese aber nicht akzeptiert. Installieren Sie in diesem Falle mittels YaST die Standard-Kernel-Quellen nach. Diese finden Sie im YaST in der Serie *d (Development)* als Paket *linclude* bzw. *linux*.

Falls Sie Netzwerkunterstützung angefordert haben, können Sie nun festlegen, ob Sie *host-only networking* nutzen wollen oder nicht.

```
Do you want to be able to use host-only networking in your
Virtual Machines?
[yes]
```

Dies beinhaltet die Möglichkeit, über eine Netzwerkkarte, aber ohne Nutzung eines realen Netzwerks zwischen dem Gastbetriebssystem und Linux zu kommunizieren. Dies erleichtert natürlich den Datenaustausch zwischen den Systemen erheblich.

Netzwerkkommunikation findet in einer UNIX-Umgebung gewöhnlich via TCP/IP statt. Haben Sie also Netzwerkunterstützung gewählt, werden Sie nun die IP-Adressen eingeben müssen, die für die Kommunikation verwendet werden sollen. Der an Einzelheiten interessierte Leser sei hier auf das Kapitel 13 *Linux im Netzwerk* verwiesen. Zunächst versucht das Skript aber, Ihnen die Arbeit abzunehmen, indem es nach einem unbenutzten Subnetz suchen will.



## Windows für Linux? – Emulatoren

---



```
Do you want this script to probe for an unused private
subnet? (yes/no/help)
[yes]
```

In der Regel werden Sie diese Frage jedoch verneinen und die Adressen per Hand eingeben. Falls Sie nur das host-only networking verwenden, können Sie hier ohnehin kaum einen Fehler machen. Die Eingabe der Adressen erfolgt, falls nötig, in zwei Schritten. Zunächst müssen Sie die IP-Adresse angeben

```
What will be the IP address of your host on the private
network?
```

und danach die Netzwerkmaske

```
What will be the netmask of your private network?
```

Beides sind jeweils vier durch Punkt getrennte Zahlen zwischen 0 und 255 (Einzelheiten entnehmen Sie bitte dem *Kapitel 13 Linux im Netzwerk*). Eine korrekte Eingabe für die IP-Adresse könnte etwa lauten

```
192.6.1.182
```

und für die Netzwerkmaske

```
255.255.255.0
```

Im Falle des host-only networking können Sie (mit wenigen Einschränkungen) beliebige Zahlenkombinationen verwenden. Planen Sie die Anbindung an ein reales Netzwerk, sollten Sie sich bezüglich der Adressen mit dem Verwalter dieses Netzes in Verbindung setzen, der Sie dann mit den notwendigen Informationen versorgt.

Haben Sie Netzwerkunterstützung gewählt, müssen Sie sich zum Abschluss dieses Installationsteils noch die Copyright-Meldung für den *DHCP-Server* des Internet Software Consortiums (ISC) durchlesen.

Auch die *DHCP server copyright information* ist so abzuschließen.

### Hinweis



Zu guter Letzt gibt Ihnen das Skript Ihren PROMOCODE bekannt:

```
Please visit http://www.vmware.com/suse to register as a
VMware user and obtain a valid license, which you will
need
to run this evaluation version.
--- Your promotional code is 119905DE001 ---
This text is included in the file |`PROMOCODE" for your
convenience.
```

In der aktuellsten, der SuSE Distribution 7.1 beiliegenden Version wird kein PROMOCODE mehr benutzt.

### Hinweis





Diesen brauchen Sie gegebenenfalls, um via Internet einen Lizenzschlüssel zu erhalten. Den Lizenzschlüssel können Sie über die Adresse <http://www.vmware.com> beziehen, wie weiter oben beschrieben. Das Lizenz-File, das Sie dann als Attachment zu einer E-Mail erhalten, müssen Sie dann noch in das Unterverzeichnis `.vmware` des Heimatverzeichnisses desjenigen Users kopieren, der VMware einsetzen soll. Der Name der Datei muss dort `license` lauten.

Angenommen, das Verzeichnis, das die ursprüngliche Lizenzdatei mit dem Namen `license-123-456` enthält, heißt `/home/all`. Der besagte Benutzer heiße *rattle* und sein Heimatverzeichnis `/home/rattle`. Dann nehmen Sie mit

```
su rattle
```

die Identität dieses Benutzers an und wechseln mit

```
cd /home/rattle
```

in sein Heimatverzeichnis (alternativ loggen Sie sich als User *rattle* ein).

Legen Sie hier mit

```
mkdir .vmware
```

ein neues Verzeichnis an, und kopieren Sie schließlich die Lizenzdatei mit

```
cp /home/all/license-123 -- 456 .vmware/license
```

in das neue Verzeichnis. Beachten Sie noch einmal, dass die Datei den Namen `license` haben muss, benennen Sie sie gegebenenfalls um.

Mit diesem Schritt ist die Konfiguration abgeschlossen.

## Installieren von *vmtools*

Zur Steigerung der Performance stellt VMware einen eigenen X-Server zur Verfügung. Zusammen mit kleineren Hilfsprogrammen steht er in der Regel als *VMware Tools* zur Verfügung. Man kann diese Tools gegebenenfalls über Internet nachträglich downloaden. Die Installation erfolgt dann über die Option *VMware Tools Install ...* im Menü *Settings* der VMware. Rufen Sie es erst auf, nachdem Sie das Betriebssystem im virtuellen PC installiert und diesen neu gebootet haben. In der Onlinehilfe von VMware sind die einzelnen Schritte zur Installation ausführlich beschrieben.

In älteren Versionen der VMware sind diese Tools im Paket *vmtools* enthalten. Gewöhnlich finden Sie diese dann nach der Installation der Pakete unter SuSE im Verzeichnis `/etc/vmware` im Unterverzeichnis `vmware-linux-tools`. Wechseln Sie dort hinein, und werden Sie mittels

```
su
```

## Windows für Linux? – Emulatoren

---



zum User *root*. Die Installation erfolgt durch Aufruf eines Installationsskripts (wie-  
der ein Perl-Skript) namens *install.pl*. Da Sie als Superuser üblicherweise keinen  
direkten Zugriff auf ausführbare Dateien im aktuellen Verzeichnis haben, geben Sie  
zu seiner Aktivierung ein

```
./install.pl
```

Folgen Sie den (wenigen) Anweisungen am Bildschirm. Danach stehen Ihnen die  
*vmtools* und damit der spezielle X-Server zur Verfügung. Seine volle Leistung ent-  
wickelt er, wenn Sie VMware im Full-Screen-Modus einsetzen.

### Einrichten eines virtuellen PCs

Nach diesen Vorbereitungen ist es nun an der Zeit, einen virtuellen PC einzurichten.  
Wie schon weiter oben erwähnt, benötigen Sie dazu eine gültige Lizenz des  
Betriebssystems, das Sie auf diesem PC einzusetzen gedenken. Wollen Sie auf die-  
sem Wege eine spezielle Linux-Version erproben, ist diese Voraussetzung ohne wei-  
teres erfüllt. In der Regel sollte einem Besitzer eines PCs aber auch eine Windows-Li-  
zenz zur Verfügung stehen, die man ja heutzutage immer noch mit dem Kauf  
eines PCs erwirbt, wenn man sie nicht ausdrücklich ablehnt.

Sie sollten sich zunächst als der Benutzer einloggen, der VMware benutzen wird.  
Stellen Sie sicher, dass sich die Lizenzdatei im Verzeichnis *.vmware* befindet (verglei-  
chen Sie dazu auch den Abschnitt „Konfiguration des Softwarepaketes“). Starten Sie  
dann zunächst Ihren X-Server wie gewohnt, etwa durch Eingabe von

```
startx
```

Möglicherweise arbeiten Sie bereits mit dem grafischen Login, dann ist dieser  
Schritt natürlich nicht mehr nötig. Öffnen Sie nun ein Terminal, und geben Sie dort  
am Prompt ein:

```
/usr/bin/vmware
```

Wahrscheinlich befindet sich */usr/bin* bereits in Ihrem Suchpfad, dann genügt  
natürlich die Eingabe von:

```
vmware
```

Sollten Sie eine Fehlermeldung der Art erhalten, dass der Virtual Machine Monitor  
nicht geladen ist, dann können Sie dies ganz einfach nachholen, indem Sie als User  
*root* (*su!*) eingeben:

```
insmod vmmon
```

Verlassen Sie danach den Superuser-Modus wieder, und versuchen Sie den Start  
von VMware erneut.

Hinweis





Das (Nach-)Laden von *vmmon* ist nach einem Neustart von Linux nicht mehr notwendig, da es durch die Konfigurationsroutine in die Startautomatik von Linux eingebunden wird.



Hinweis

Nach einem Monitortest meldet sich VMware so, wie in Abbildung 9.4 zu sehen.

Abb. 9.4

Startbild von VMware



Gleichzeitig öffnet sich ein Dialog, der Ihnen die Möglichkeit gibt, eine neue Konfiguration (dies ist ein neuer virtueller PC, nicht eine neue Konfiguration der VMware) festzulegen oder eine vorhandene zu laden. Das Laden erfolgt über einen üblichen Dialog zum Öffnen von Dateien.

Solche Konfigurationen werden standardgemäß im Heimatverzeichnis unter *vmware/config/config.cfg* gespeichert, wobei *config* der Name der gewählten Konfiguration ist. Wählen Sie etwa Windows 98 als Betriebssystem, wird die zugehörige Konfiguration (und später auch die virtuelle Platte – allerdings unter dem Namen *wing8.dsk*) als *wing8.cfg* im Verzeichnis *\$HOME/vmware/wing8* gespeichert (*\$HOME* bezeichnet hier das Heimatverzeichnis). Eine bereits existierende Konfiguration können Sie natürlich auch beim Start von VMware angeben. So starten Sie aus Ihrem Heimatverzeichnis mit

```
/usr/bin/vmware vmware/win98/win98.cfg
```

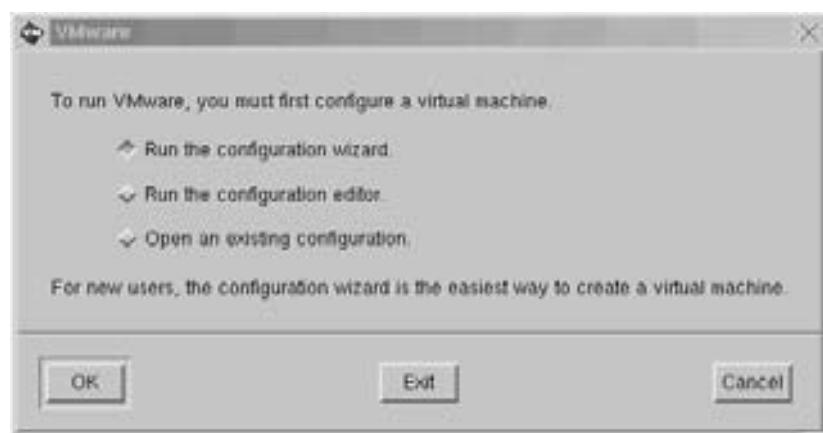


## Windows für Linux? – Emulatoren



Ihr Tool mit der für Windows 98 festgelegten Konfiguration. Diese Konfiguration bezieht sich auf die Eckdaten für die zur Verfügung stehende Hardware, wie RAM, Plattspeicher oder Nutzung von Floppy und CD-ROM, nicht aber auf die Konfiguration des eingesetzten Betriebssystems. Unter einer Wing8-Konfiguration können Sie sehr wohl Linux installieren, es werden Ihnen halt die Ressourcen zur Verfügung gestellt, die in der Konfiguration festgelegt wurden, wie etwa Größe von RAM oder Festplatte.

Abb. 9.5



Der Einfachheit halber sollten Sie bei Ihrem ersten Versuch den *Configuration wizard* wählen. Dieser führt Sie in mehreren Schritten zur Erzeugung einer (hoffentlich) geeigneten Konfiguration. Der Installationsassistent sollte keine großen Schwierigkeiten für einen halbwegs interessierten Anwender darstellen. Wer aber nur 100 MB für die Installation eines Windows-gx-Systems zur Verfügung stellt, sollte sich nicht wundern, wenn es nicht nach seinen Vorstellungen funktioniert oder sich erst gar nicht installieren lässt.

Nach den üblichen Eingangsinformationen erscheint der Dialog *Guest Operating System*.

Hier offenbart VMware eine seiner entscheidenden Stärken. Sie finden in der Liste praktisch jedes Betriebssystem, das für PCs eine Bedeutung hat oder hatte.

Im nächsten Schritt legen Sie das Verzeichnis fest, in dem die für den virtuellen PC notwendigen Dateien gespeichert werden. Wie schon gesagt, ist dies für Windows 98 im Standardfall \$HOME/vmware/wing8. Für andere Systeme gilt Entsprechendes.





Abb. 9.6

Wahl des Guest-Betriebssystems

Danach legen Sie den Typ der virtuellen Festplatte fest. Normalerweise ist dies *Virtual Disk*. Dabei wird die Linux-Partition benutzt, in der Sie Ihren virtuellen PC installieren. Darin wird eine Datei erzeugt (namens *config.dsk*), in der die Daten der virtuellen Festplatte abgelegt werden.

VMware erlaubt es Ihnen aber auch – ein weiterer Pluspunkt für diese Software –, eine vorhandene Partition mit dem darauf installierten System einzubinden. Haben Sie etwa auf Ihrer Festplatte eine funktionierende Windows-Installation, können Sie darauf zurückgreifen, indem Sie in diesem Dialog festlegen, dass Sie auf eine vorhandene Partition zurückgreifen wollen.

Voraussetzung für diese Möglichkeit ist allerdings, dass Ihr Kernel so konfiguriert ist, dass er auf eine solche Partition zugreifen kann. Sollten Sie im Zweifel sein, können Sie in dieser Situation auf die ausgezeichnete Hilfefunktion zurückgreifen, die im verfügbaren Browser (z.B. Netscape) dargestellt wird. Diese wird Sie, falls möglich und notwendig, mit der Homepage der Firma verbinden.

## Windows für Linux? – Emulatoren

Abb. 9.7

Hilfefunktion des Konfigurationsassistenten



Haben Sie als Typ *Virtual Disk* festgelegt, können Sie im Dialog *Virtual Disk Size Setting* die Größe der virtuellen Harddisk festlegen. Bedenken Sie an dieser Stelle, dass deren Größe von der Speicherkapazität der Linux-Partition abzuziehen ist, in der Sie den virtuellen PC installieren (in der Regel ist dies die Partition, welche die Heimatverzeichnisse enthält).

In den beiden folgenden Dialogen legen Sie fest, ob Ihr virtueller PC auf die reale CD-ROM und das reale Diskettenlaufwerk zugreifen können soll. Dazu benötigt VMware die Namen der entsprechenden Devices. Sollten Sie nicht sicher sein, hilft in der Regel ein Blick in die Datei */etc/fstab*. Dort sollte sich für das CD-ROM-Laufwerk folgender Eintrag finden lassen für das Diskettenlaufwerk suchen Sie nach einem Eintrag in Zusammenhang mit */floppy*):

```
/dev/hdc /cdrom iso9660 ro,noauto,user 0 0
```





In diesem Fall sollten Sie als Device für Ihr CD-ROM-Laufwerk `/dev/hdc` eintragen. Bei der Wahl des Floppy-Laufwerks ist normalerweise `/dev/fdo` einzutragen. Sie sollten in jedem Fall beachten, dass Linux und das Gastsystem nicht gleichzeitig auf Diskette und CD-ROM zugreifen können. Hier reißt VMware alle Ressourcen an sich. Ein Versuch, die Floppy zu mounten, während VMware läuft, wird gewöhnlich mit der Meldung „mount: /dev/fdo already mounted or /floppy busy“ beantwortet. Die Benutzung von CD-ROM und/oder Floppy kann allerdings unter VMware im Menü *Settings->Removable Devices* zurückgesetzt werden. Klickt man dort `/dev/fdo` oder `/dev/hdc` (die Namen sind abhängig von der vorgenommenen Konfiguration) an, werden Floppy- bzw. CD-ROM-Laufwerk wieder für andere Linux-Anwendungen verfügbar.

Im nächsten Schritt werden Sie aufgefordert festzulegen, welche Art der Netzwerkunterstützung Sie wollen (diese Auswahl hängt natürlich auch davon ab, wie Sie VMware konfiguriert haben). Als privater User ohne Zugang zu einem lokalen Netzwerk werden Sie die Option *No Networking* wählen. Wollen Sie Ihren virtuellen PC in ein reales Netzwerk einbinden, dann wählen Sie *Bridged Networking*. Wollen Sie die (vorhandene) Netzwerkkarte (nur) nutzen, um eine Kommunikation zwischen Linux und dem Gastsystem zu ermöglichen, wählen Sie *Host-only Networking*.



Abb. 9.8

Wahl des Netzwerktyps



## Windows für Linux? – Emulatoren



Zum Abschluss dieser Konfiguration stellt Ihnen VMware die getroffenen Einstellungen im Dialog *Confirmation* noch einmal dar. Sie können sie bestätigen (*Done*), verwerfen (*Cancel*) oder zu den Auswahldialogen zurückkehren (*Prev*).



Abb. 9.9

Bestätigung der getroffenen Einstellungen

Falls Sie Ihre Wahl mit *Done* bestätigen, haben Sie einen neuen virtuellen PC eingerichtet, der nun auf die Installation eines (Gast-)Betriebssystems wartet.

### Installation des Gast-Betriebssystems

Für die nun folgende Installation benötigen Sie eine (lizenzierte) Version der Systemsoftware, die Sie auf dem virtuellen PC einrichten wollen, d.h., Sie benötigen die Software (auf Diskette oder CD zum Beispiel) und, falls es sich um eine proprietäre (Microsoft-)Software handelt, den Lizenzschlüssel, der bei dieser Firmensoftware gewöhnlich irgendwo auf der Verpackung zu finden ist.





Sie sollten sich darüber im Klaren sein, dass die folgenden Schritte Ihren virtuellen PC betreffen. Wenn Ihnen also gemeldet wird, dass Sie Ihren Computer nun ausschalten können, dann ist nicht der Linux-PC gemeint! Was immer Sie an Ihrem virtuellen PC machen, schalten Sie den realen Rechner nicht aus und rebooten Sie ihn auch nicht! Versuchen Sie in dieser Situation, Probleme mithilfe von Linux zu lösen!

Bevor Sie nun auf Ihrem virtuellen PC eine Systemsoftware installieren, versichern Sie sich, dass im Floppy-Laufwerk keine Linux-Installationsdisketten und im CD-ROM-Laufwerk keine Linux-Installations-CD liegt. Sorgen Sie stattdessen dafür, das Installations-CD oder -Diskette des Gast-Betriebssystems im entsprechenden Laufwerk eingelegt sind. Das andere Laufwerk sollte leer sein.

Der nun folgende Vorgang entspricht einer gewöhnlichen Installation des entsprechenden Betriebssystems mit der einzigen Ausnahme, dass sie in einem Fenster unter KDE oder einem anderen Fenstermanager erfolgt.

Zunächst können Sie den Bootvorgang eines Rechners bestaunen, auch wenn Ihr realer PC bereits gebootet ist.



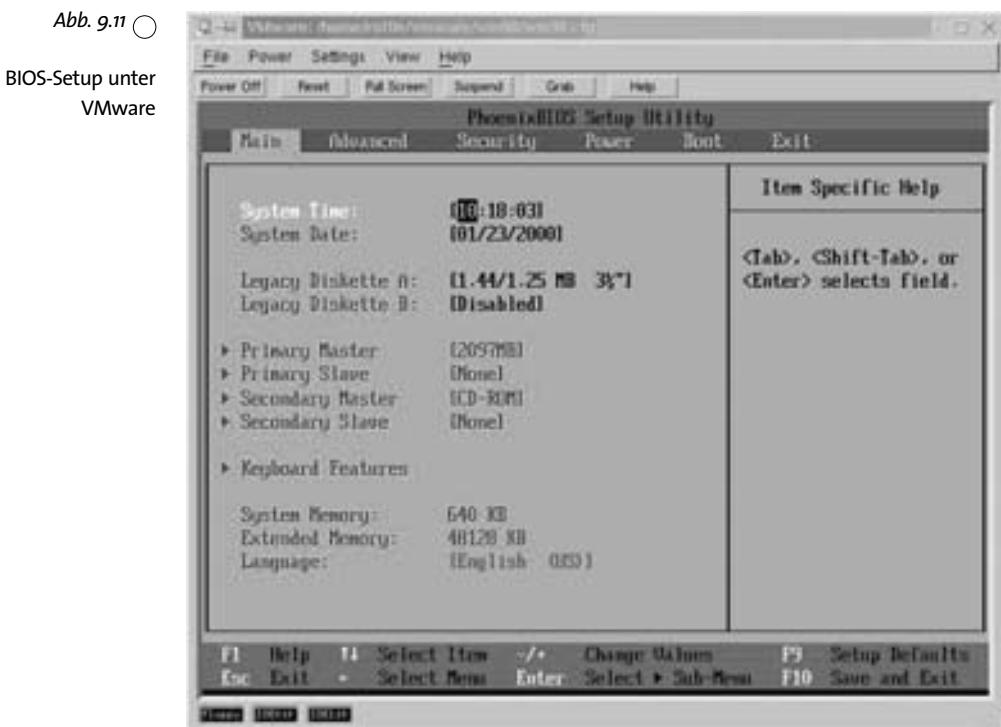
Abb. 9.10

Bootvorgang unter  
VMware

Wer genügend Erfahrung mit der Konfigurierung von PC-Hardware hat, kann an dieser Stelle die Taste **F2** drücken, um ins *BIOS-Setup* zu gelangen.



## Windows für Linux? – Emulatoren



Eine genauere Analyse offenbart, dass sich hier nicht die Daten des realen PCs finden lassen, sondern die des virtuellen. Beachten Sie insbesondere die Daten zu *Primary Master* (die Festplatte des Gastsystems) und *Extended Memory*.

Haben Sie eine Bootdiskette oder -CD für die Installation von Windows 98 eingelegt, dann sollte sich Ihnen im VMware-Fenster folgende Situation darstellen:

An dieser Stelle soll nun nicht die Installation eines Windows-Betriebssystems beschrieben werden. Folgen Sie den Anweisungen der Installationsroutine einschließlich der Aufforderungen, die Disketten- oder CD-ROM-Laufwerk betreffen (beispielsweise, wenn der virtuelle PC neu gestartet werden muss). Dies geschieht in der Regel nach dem bekannten Muster

- Partitionieren der Platte (dies ist die virtuelle Platte, deren gesamten Platz Sie für das Gastsystem einplanen können),
- Formatieren der Partition,
- Kopieren der Dateien und
- Konfigurieren (z.B. Wahl der Zeitzone und ähnliche Aufgaben).

## VMware

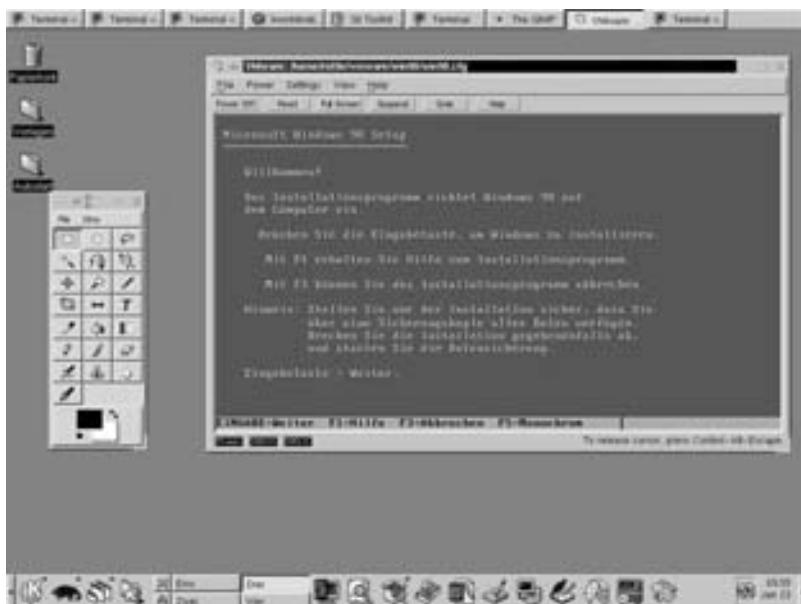


Abb. 9.12

Windows 98 Setup  
unter KDE

Die wahrscheinlich aufregendste Situation während der Installation ist sicherlich die, wenn Ihnen die Installationsroutine meldet, dass nun die Festplatte formatiert wird. Sollten Sie auf dem Bildschirm die Meldung aus Abbildung 9.13 sehen, ist diese Arbeit bereits erledigt. Doch es besteht kein Grund zur Panik, die Formatierung ist im „Sandkasten“ und daher auch recht fix erfolgt.

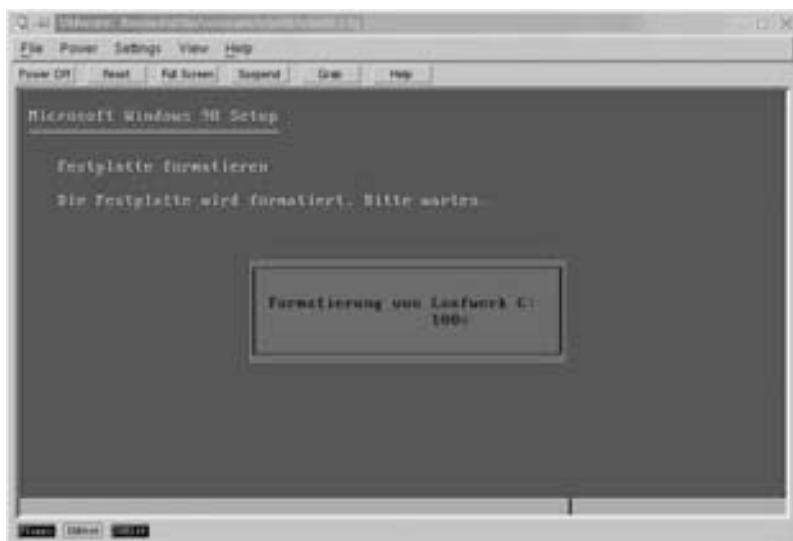


Abb. 9.13

Eine merkwürdige  
Situation: Eine virtuelle  
Festplatte wird forma-  
tiert



## Windows für Linux? – Emulatoren



Beim Kopieren der Dateien werden Ihnen gewöhnlich verschiedene Optionen angeboten. Trauen Sie sich ruhig an die Variante, die Ihnen eine Auswahl der zu kopierenden Pakete erlaubt. Im Regelfall wird Ihnen in dieser benutzerdefinierten Version eine Vorauswahl an Dateien angeboten, die der Standardvariante entspricht, nur können Sie hier prüfen, ob auch die für Sie interessanten Zusatz-Tools mit installiert werden.

Nach dem Kopiervorgang verlangen Installationsroutinen wie die von Windows 98 die Eingabe eines Lizenzschlüssels (hier „Produkt ID“ genannt). Auch wenn Sie über den Einsatz von Linux frei verfügen dürfen (Weitergabe von Linux ist ausdrücklich erlaubt), gilt dies nicht für alle Software, die Sie unter Linux einsetzen wollen. So müssen Sie etwa für VMware, aber natürlich auch für Microsoft-Produkte eine gültige Lizenz besitzen, um diese Software einsetzen zu dürfen.



Abb. 9.14

Die Eingabe des Lizenzschlüssels. Beachten Sie, dass Sie Windows 98 nur einsetzen dürfen, wenn Sie über eine gültige Lizenz verfügen!

Nachdem Sie im weiteren Verlauf der Installation noch einige Konfigurationen vorgenommen haben, wird schließlich Ihr virtueller Rechner neu booten und mit Windows 98 als Betriebssystem starten. Sie werden danach auf dem Bildschirm gleichzeitig die Aktivitäten zweier Betriebssysteme beobachten können.



## VMware

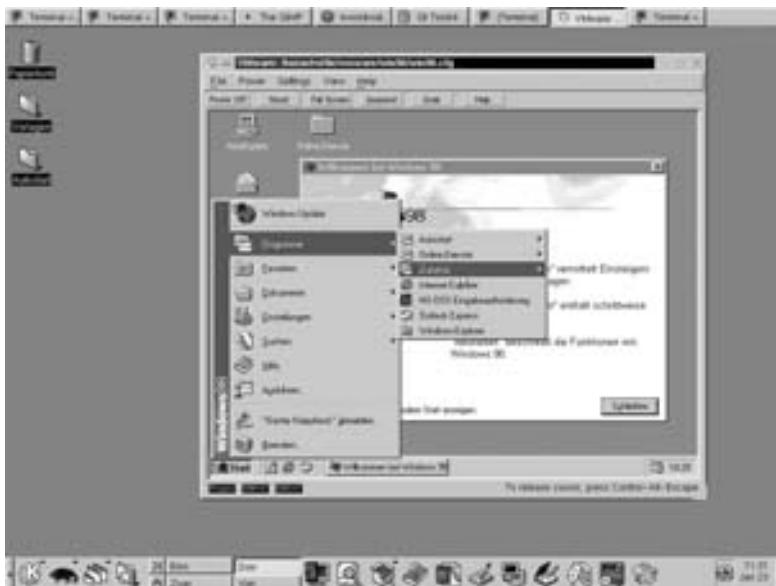


Abb. 9.15

Windows 98 im VMwa-  
re-Fenster unter KDE

Wie schon mehrfach erwähnt, als Systemsoftware lässt sich prinzipiell jedes unter Intel-basierten Rechnern lauffähige Betriebssystem verwenden, also auch DOS, OS/2 oder eben eine (andere) Linux-Distribution.

Abbildung 9.16 zeigt den Beginn einer SuSE-Installation im Fenster des virtuellen PCs.

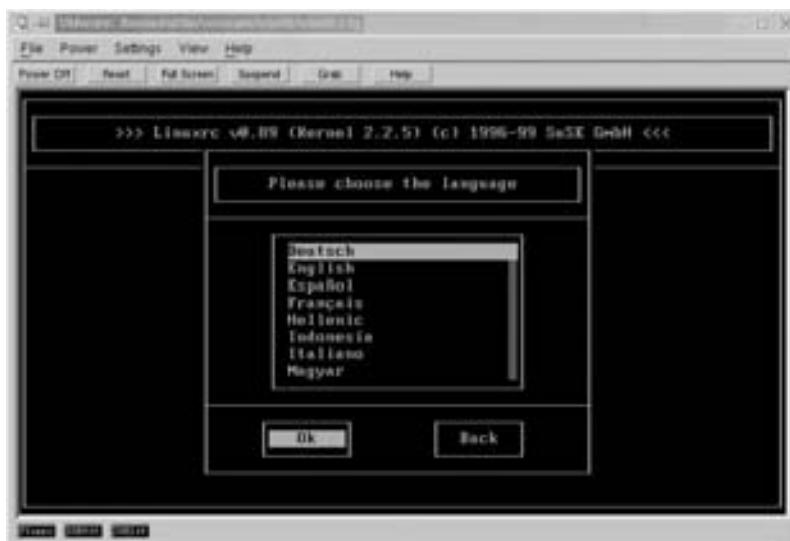


Abb. 9.16

Installation von Linux  
auf einem virtuellen PC



## Windows für Linux? – Emulatoren

---



### Arbeiten mit VMware

Haben Sie Ihren virtuellen PC erfolgreich mit einem Betriebssystem ausgestattet, können Sie unter diesem nun in der dort üblichen Form Software installieren. Häufig wird es also genügen, die Installations-CD ins Laufwerk einzulegen und den virtuellen Rechner zu starten.

Die Vorbereitung dazu erfolgt, indem Sie (falls noch nicht geschehen) unter Linux den X-Server starten und im Window Manager *VMware* aufrufen (unter KDE können Sie dieses Tool ins Menü einbinden oder mit einem Icon auf dem Desktop verknüpfen). Üblicherweise erscheint nun der Dialog zur Erstellung oder Auswahl einer vorhandenen Konfiguration (siehe Abbildung 9.5). Sie wählen hier natürlich *Open an existing configuration* und gelangen in einen Dialog zum Öffnen einer Datei.

Konfigurationsdateien sollten sich üblicherweise im Pfad `$HOME/vmware/config` befinden und *config.cfg* heißen. Hierbei bezeichnet `$HOME` das Heimatverzeichnis des Benutzers und *config* die vorhandene Konfiguration (für Windows 98 standardgemäß *wing8*).

Sie können diesen auf die Dauer lästigen Dialog vermeiden, indem Sie VMware beim Start den Namen der Konfigurationsdatei unter Angabe des Pfades als Parameter übergeben. Auf der Befehlszeile könnte ein Aufruf dann wie folgt aussehen:

```
vmware /home/rattle/vmware/win98/win98.cfg
```

Entsprechend können Sie auch den Eintrag ins Menü oder das Icon unter KDE konfigurieren.

Das Öffnen der Konfigurationsdatei legt nun fest, welchen virtuellen PC Sie benutzen – nicht, welches Betriebssystem darauf. Die Konfigurationsdatei enthält unter anderem die Angaben über die Datei, die VMware als virtuelle Platte nutzen wird (Standardname *config.dsk*).

Die Wahl des Betriebssystems erfolgt mit dem eigentlichen Start des virtuellen PCs. Dies geschieht durch Anklicken des Buttons *Power on* oder des gleichnamigen Eintrags im Menü *Power*.

Je nachdem, wie Sie den virtuellen PC konfiguriert haben (Nutzung von CD-ROM- und Diskettenlaufwerk, Einstellungen des BIOS), wird nun das erste verfügbare Betriebssystem gestartet. Bisweilen liegt im CD-ROM-Laufwerk von der letzten Benutzung des YaST noch die erste CD der Linux-Distribution.

Ist diese bootfähig, dann landen Sie in der Regel in der Linux-Installationsroutine. Diese würde, wenn Sie sie ausführen, Ihren virtuellen Rechner zu einem Linux-Rechner machen. Üblicherweise wollen Sie aber das auf der virtuellen Festplatte installierte System booten. Sorgen Sie also zunächst einmal dafür, dass weder Diskette noch CD-ROM in den entsprechenden Laufwerken liegen, wenn Sie den virtuellen PC starten.





Für die weitere Arbeit mit VMware sollten Sie sich nun eine wichtige Tastenkombination merken:

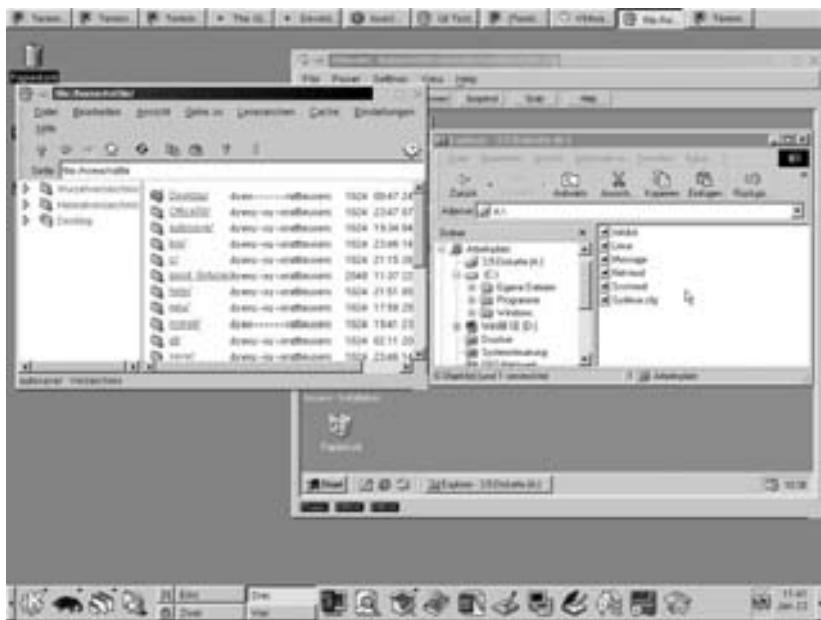
[Strg] + [Alt] + [Esc]

VMware fängt, wenn Sie so wollen, den Cursor ein, wenn Sie in das Fenster klicken. Hier arbeiten Sie dann unter den Gegebenheiten, die unter dem Gast-Betriebssystem üblich sind. Der Clou der Situation ist aber der, dass dieses Gast-Betriebssystem in einem Task von Linux läuft, Sie also weiterhin Linux nutzen können. Wollen Sie nun einen anderen Linux-Task per Mausklick in den Vordergrund holen, etwa Gimp, um einen Screenshot zu machen, müssen Sie zunächst mit der genannten Tastenkombination den Cursor aus seiner „Gefangenschaft“ befreien und wieder unter die Regie von KDE (oder den Window-Manager Ihrer Wahl) stellen.

Gleichermaßen gilt auch, wenn Sie das Fenster in den Vollbildmodus schalten, was aus Performance-Gründen zu empfehlen ist. Dieses verlassen Sie ebenfalls über diese Tastenkombination. Ansonsten können Sie nun, wie Abbildung 9.17 zeigt, Linux-Anwendungen und Anwendungen des Gast-Betriebssystems parallel laufen lassen.

Abb. 9.17

kfm und Windows-Explorer nebeneinander



Was die Arbeit mit Ihrem Gast-Betriebssystem angeht, so können wir Ihnen keinen weiteren Tipp geben. Bisweilen ist es aber notwendig, die Konfiguration des virtuellen PCs nachträglich zu ändern. Dies ist – mit einigen Einschränkungen (versuchen Sie doch einmal, 256 MByte RAM virtuell aus 128 MB RAM real zu erzeugen) – jederzeit über den Eintrag *Configuration Editor* im Menü *Settings* möglich.



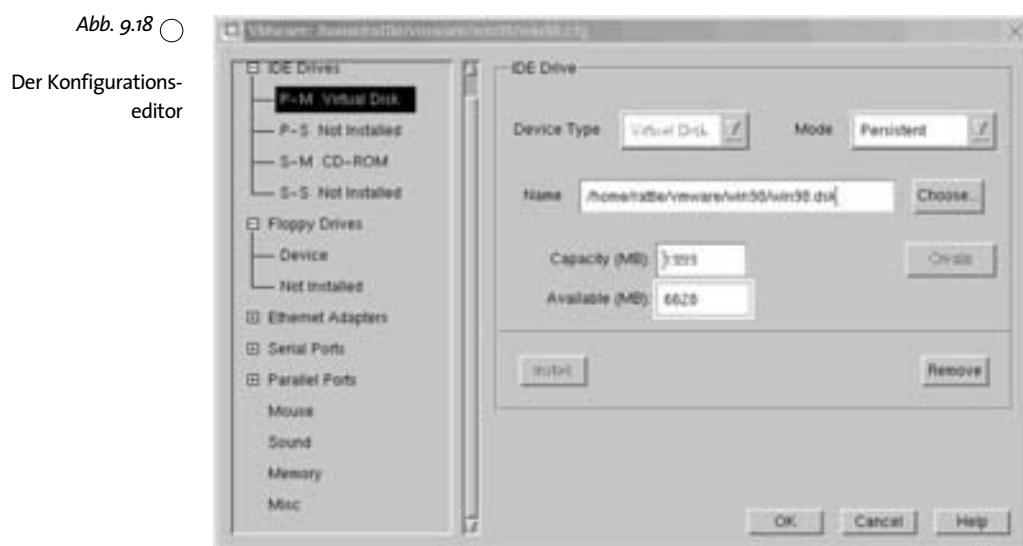
## Windows für Linux? – Emulatoren



Hier können Sie Einstellungen zu

- IDE-Laufwerken,
- Floppy-Laufwerken,
- Netzwerkkarte,
- seriellen Schnittstellen,
- Parallelports,
- Sound,
- Maus und
- Speicher

verändern. Abbildung 9.18 zeigt Ihnen beispielsweise, wie Sie im Konfigurationseditor die Größe der virtuellen Platte ändern können.



Für weiter reichende Fragen, die in dieser knappen Einführung naturgemäß nicht beantwortet werden können, verweisen wir Sie gerne (noch einmal) auf die informative Homepage der Firma, <http://www.vmware.com>.





# Sicherheit ist Trumpf: Datensicherung



Das Archivierungsprogramm *tar* **432**  
Beispiele für die Arbeit mit *tar* **438**



**10**

## Sicherheit ist Trumpf: Datensicherung

---



Datensicherung ist ein leidiges Thema: Sie kostet Zeit, Geld und Arbeit und zeigt keine sichtbaren Ergebnisse, solange Hardware, Software und Benutzer keine gravierenden Fehler machen. Doch an dem legendären Tag X verweigert die Festplatte den Dienst und lässt sich auch von Fachleuten nicht mehr reanimieren. Oder man ist gezwungen, die neueste Version der Textverarbeitung eines namhaften Softwareherstellers zu benutzen, die leider nicht immer das tut, was in den Menüs steht.

Stunden oder Tage an Arbeit sind verschwunden – so mindestens zweimal im Dezember 96 in Kiel geschehen. Oder der berühmte Satz nach durchhackter Nacht: „Wir müssen jetzt nur noch schnell ...“, einige unüberlegte Handgriffe, und das falsche Verzeichnis ist gelöscht oder die falsche Partition formatiert.

Alle diese Fälle sind ärgerlich und kommen leider öfter, vor als einem lieb sein kann. Noch ärgerlicher sind diese Situationen allerdings, wenn man keinerlei Sicherheitskopien der Daten besitzt – Tage an Arbeit müssen eventuell wiederholt werden, und nichts ist langweiliger, als eine Arbeit zu wiederholen, die eigentlich schon fast fertig war.

Wohl dem, der seine Daten täglich sichert und in dieser Situation zu einem Tape, einem Stapel Disketten oder seiner Wechselplatte greifen kann.

Womit wir bei den verschiedenen Möglichkeiten der Hardware zur Datensicherung wären:

### Disketten

Früher war die Sicherung von Festplatten auf Disketten die einzige preiswerte Möglichkeit für einen PC-Benutzer. Allerdings waren die Größenverhältnisse zwischen Festplatte und Diskette auch andere: Während eine Standardfestplatte vor 15 Jahren 10 MByte umfasste, hatte eine Diskette 360 KByte Fassungsvermögen.

Mit knapp 30 Disketten konnte man seine gesamte Festplatte sichern, was sich durch den Einsatz von Komprimierungsprogrammen auf ca. die Hälfte reduzierte. Der Zeitaufwand für die Datensicherung war noch gerade erträglich.

Während aber das Fassungsvermögen von Disketten bis heute nur um den Faktor vier gestiegen ist, enthält die heutige Gigabyte-Festplatte hundertmal mehr Daten als ihr Kollege vor 15 Jahren. Mit Hunderten von Disketten als Backup zu hantieren, ist schlicht unpraktikabel.

Trotzdem sollte man die Idee der Sicherung auf Disketten nicht verwerfen, falls einem keine anderen Möglichkeiten zur Verfügung stehen. Falls man nicht mit großen Datenbanken oder mit großen Mengen von grafischen Daten – beides Speicherfresser – arbeitet, bleibt die Datenmenge, die sich verändert, relativ gering.

Zum einen passt z.B. das Wochenpensum der Arbeit eines Programmierers auf eine oder wenige Disketten.



## Sicherheit ist Trumpf: Datensicherung



Zum anderen werden große Teile des Linux-Systems nur vom Benutzer gelesen und nicht verändert, wie z.B. Dokumentationen, Programme oder Skripte. Diese brauchen heutzutage aber nicht gesichert zu werden – man besitzt sie auf CD-ROM.

Weiter unten werden im Zusammenhang mit dem Archivprogramm *tar* einige Beispiele angeführt, wie man Programmgruppen oder Zeitspannen spezifizieren kann und die zugehörigen Daten dann auf Disketten sichert.

### Wechselplatten

Wechselplatten haben nicht nur den Vorteil, dass man sie in der Praxis fast wie eine Festplatte handhaben kann. Sie bieten außerdem den schnellen Zugriff auf eine Datei, während Bandlaufwerke sich erst sequenziell zu dieser Datei vorarbeiten müssen – sie spulen.

Da bestimmte Typen, wie z.B. Syquest- oder die Iomega-ZIP-Laufwerke, recht verbreitet sind, kann man außerdem recht praktisch größere Mengen Daten zu anderen Rechnern transportieren. Dies kann beispielsweise in der Zusammenarbeit mit einem Belichtungsservice erforderlich sein: Eine grafisch aufwändig gestaltete Seite – z.B. ein Buch- oder Zeitschriftentitel – kann als vierfarbige PostScript-Datei durchaus zwischen 10 und 40 MByte Umfang besitzen.

Wechselplatten werden in letzter Zeit zusehends günstiger angeboten, und für das Iomega-ZIP-Laufwerk, das am parallelen Port angeschlossen wird, gibt es unter Linux sogar eine direkte Kernel-Unterstützung.

### Bandlaufwerke

Bänder sind klassische Sicherungsmedien für Computerdaten – früher waren sie die einzige Möglichkeit, Daten zu sichern, wenn man von Lochkartenstapeln einmal absieht.

Die Vielzahl von Normen, Aufzeichnungsformaten, Kassetten- und Cartridge-Bauweisen ist kaum zu überblicken, obwohl sich zwei Quasi-Standards herauskristallisiert haben: QIC-02 und QIC-80.

Laufwerke dieses Standards gibt es inzwischen recht günstig zu dem Preis einer Festplatte. Linux bietet für beide Normen Treiber, die in den Kernel eingebunden werden können. Außerdem bietet Linux für SCSI-Streamer, die eigene Bootparameter benötigen, eine Kernel-Unterstützung.

Das Sichern von größeren Datenmengen geht inzwischen auf solchen Streamern relativ zügig vonstatten: Ein Gigabyte pro Stunde kann als Praxiswert für einen PC angenommen werden. Relativ langsam wird das Ganze, wenn wenige bestimmte Dateien vom Band zurückkopiert werden sollen: Da die Daten sequenziell abgelegt werden, wird erst einmal ordentlich gespult, bis die Datei erreicht ist – im schlechtesten Fall fast das ganze Band.



## Sicherheit ist Trumpf: Datensicherung

---



Bänder können auch dem Datenaustausch dienen – sobald beide Parteien mit der gleichen Norm arbeiten. Dies verliert aber zusehends an Bedeutung. Filetransfer über ISDN ist einfach praktischer und oft auch billiger als Bänder, die immer noch DM 60,00 bis 90,00 pro Stück kosten.

### Festplatten

Da die Festplattenpreise rasant gefallen und gleichzeitig die Kapazitäten im Gigabyte-Bereich angekommen sind, lohnt es sich, darüber nachzudenken, ob man zur Sicherung seiner Daten eine weitere Harddisk kauft. Dies kann für ganze Abteilungen in einem Server geschehen, aber selbst bei einem Rechner, der sich nicht in einem Netz befindet, kann diese Alternative interessant sein: Der Einbau in den PC ist unproblematisch, für Linux muss keine weitere Kernel-Unterstützung hinzugefügt werden, und der Zugriff auf die gesicherten Daten kann schnell erfolgen.

Im Gegensatz zu Bändern oder Wechselplatten hat man natürlich nicht die Möglichkeit zum Datenaustausch mit anderen Rechnern.

Auch wenn man in einem Betrieb eine sehr große Datenbank derart sichern will, dass man von jedem Werktag eine Kopie zieht und mindestens einen Monat aufbewahrt, ist das Fassungsvermögen von Festplatten schnell überfordert, während man bei einem Bandlaufwerk für jeden Tag ein anderes Band benutzen kann.

Bei der entsprechenden Umgebung kann eine weitere Festplatte aber durchaus der kostengünstigste Weg zur Datensicherung sein.

## Das Archivierungsprogramm *tar*

*tar* ist DAS Werkzeug, um Dateien zu archivieren, Sicherheitskopien auf Bandlaufwerken, Disketten oder Festplatten anzulegen, zu erweitern, anzuzeigen und Dateien aus Archiven wieder zurück auf die Festplatte zu kopieren. *tar* kann dabei die Daten komprimieren bzw. dekomprimieren und auch ein Archiv auf mehreren Bändern oder Disketten anlegen.

*tar* ist die Abkürzung für **tape archive**, da das Programm zu Beginn seiner Existenz nur mit Bändern umgehen konnte.

Da *tar* ein vielseitig und oft genutztes Werkzeug ist, werden die hiermit erzeugten Archivdateien auch „tarfiles“ oder „Tar-Dateien“ genannt.

*tar* ist (fast) so alt wie UNIX und bringt die unschöne Seite der unzähligen UNIX-Derivate sehr deutlich zum Vorschein: Mit den verschiedenen Systemen wird jeweils ein *tar* geliefert, das verschiedene Erweiterungen besitzt. Im Folgenden gilt die GNU-Version von *tar*, die mit dem Linux-System geliefert wird.

Die Syntax von *tar* lautet wie folgt:



## Das Archivierungsprogramm ***tar***



***tar*** -Funktion -Optionen Dateiname

Es darf jeweils nur genau eine Funktion angegeben werden, wie beispielsweise „Erzeugen“ (-c) oder „Anzeigen“ (-t). Die Optionen können in beliebiger Anzahl angegeben werden.

*Dateiname* können eine oder mehrere Dateien sein oder auch ein oder mehrere Verzeichnisse, jeweils mit all ihren Dateien und Unterverzeichnissen. Die Angabe eines schlichten */\** kann also alle Dateien des aktuellen Linux-Filesystems mit allen gemounteten Geräten kopieren!

Falls in der folgenden Liste mehrere Funktionen bzw. Optionen für die gleiche Tätigkeit nebeneinander aufgeführt sind, führen sie auch die gleiche Tätigkeit aus. Die Redundanz der Parameter wird aus historischen Gründen weiter unterstützt.

### Funktionen

***-A, --catenate, --concatenate***

fügt einem existierenden Archiv ein weiteres an. Funktioniert nur mit Bandlaufwerken.

***-c, --create***

Erstellt ein neues Archiv. Ein vorhandenes Archiv mit gleichem Namen wird ohne weitere Fehlermeldung oder Abfrage überschrieben.

***-d, --diff, --compare***

findet die Unterschiede zwischen den Dateien in dem Archiv und dem aktuellen Dateisystem.

***--delete***

entfernt Dateien aus der Archivdatei. Da Bänder die Daten sequenziell schreiben, kann diese Aktion auf sie nicht angewendet werden.

***-r, --append***

hängt an das Archivende weitere Dateien bzw. Verzeichnisse an. Für die Arbeit mit *tar*-Dateien genutzt, bei Bändern sollte man -A verwenden

***-t, --list***

listet das Verzeichnis des Archivs auf.

***-u, --update***

hängt nur Dateien ans Ende des Archivs, wenn sie seit der letzten Archivierung neu erstellt oder verändert worden sind.



## Sicherheit ist Trumpf: Datensicherung

---



`-x, --extract, --get`

kopiert die Dateien aus dem Archiv zurück auf die Festplatte. Das Archiv bleibt dabei unverändert.

### Optionen

`-b, --block-size N`

Blockgröße für Bandlaufwerke in  $N * 512$  Byte. Die Standardeinstellung für  $N$  ist 20. Größter Faktor kann 1.024 sein. Beim Datenaustausch mit anderen UNIX-Systemen muss darauf geachtet werden, dass die Blockgrößen übereinstimmen, sonst kann das Band nicht gelesen werden.

`-C, --directory DIR`

wechselt zum Inhaltsverzeichnis  $DIR$ .

`--checkpoint`

zeigt die Namen der Inhaltsverzeichnisse an, während das Archiv gelesen wird.

`-f, --file [HOSTNAME:] F`

Diese Option ist für das Schreiben von Bändern nicht notwendig, die Grundeinstellung für das Ziel ist `/dev/rmto`. Wenn andere Treiber für das Bandlaufwerk zuständig sind, wie `/dev/ftape`, `/dev/hto` oder `/dev/nhto`, muss dies unter  $F$  angegeben werden.

Wenn auf Disketten archiviert werden soll, muss hier `/dev/fdo` bzw. `/dev/fd1` angegeben werden. Falls das Archiv in einer Datei abgelegt wird, muss  $F$  den Dateinamen mit dem Pfad enthalten.

Falls die Datei auf einem Host abgelegt werden soll, kann optional der  $HOSTNAME$  eingegeben werden.

`--force-local`

Die Archivdatei ist lokal, selbst wenn der Dateiname einen Doppelpunkt enthält, da der Doppelpunkt eigentlich zur Separation von Hostname und Dateiname benutzt wird (siehe `f`).

`-F, --info-script F --new-volume-script F`

Falls die Option *Multi-Volumes* (-M) gesetzt wurde, wird am Ende jedes Bandes (oder jeder Diskette oder jedes Wechselplattenmediums) ein Info-Skript mit dem Dateinamen  $F$  geschrieben.



## Das Archivierungsprogramm ***tar***

---



***-G, --incremental***

erzeugt oder listet oder extrahiert das alte GNU-Format.

***-g, --listed-incremental F***

erzeugt oder listet oder extrahiert das neue GNU-Format.

***-h, --dereference***

Nicht die symbolischen Links werden gespeichert, sondern die Dateien, auf die sie verweisen.

***-i, --ignore-zeros***

ignoriert einen mit Nullen gefüllten Block, der normalerweise das Ende der Datei bei einem Bandarchiv darstellt.

***--ignore-failed-read***

Falls ein Lesefehler bei einem Archiv auftritt, wird dennoch weitergelesen und nicht das Programm beendet, um so evtl. weitere Teile lesen zu können.

***-k, --keep-old-files***

Wenn beim Zurückschreiben vom Archiv gleichnamige Dateien neueren Datums vorhanden sind, werden die älteren Dateien auf der Festplatte nicht überschrieben.

***-K, --starting-file F***

beginnt die jeweilige Tätigkeit bei der Datei *F* in dem Archiv.

***-l, --one-file-system***

Wenn ein neues Archiv erzeugt wird, wird es nur im lokalen Dateisystem geschrieben.

***-L, --tape-length N***

Angabe der Bandlänge. Nach  $N * 1024$  Byte kommt die Aufforderung, das nächste Band einzulegen.

***M, --multi-volume***

Falls die *tar*-Datei nicht auf ein Band oder eine Diskette passt, muss diese Option angegeben werden. Der Benutzer wird dann aufgefordert, das nächste Medium einzulegen.

***-N, --after-date DATUM, --newer DATUM***

archiviert nur solche Dateien, deren Erstellungs- bzw. Veränderungsdatum neuer als *DATUM* ist.

## Sicherheit ist Trumpf: Datensicherung

---



**-o, --old-archive, --portability**

schreibt ein Archiv im alten Format V7, statt das ANSI-Format zu benutzen.

**-O, --to-stdout**

zeigt extrahierte Dateien an der Standardausgabe an.

**-P, --absolute-paths**

benutzt den absoluten Pfad von Dateinamen; das führende Zeichen / wird nicht – wie in der Standardeinstellung – entfernt.

**--remove-files**

löscht die Dateien auf der Festplatte, nachdem sie einem Archiv hinzugefügt wurden.

**-R, --record-number**

zeigt die Record-Nummer an, falls die Option -v gesetzt wurde.

**-s, --same-order, --preserve-order**

sortiert die Liste der zu extrahierenden Dateinamen wird sortiert.

**--same-owner**

erzeugt Dateien, die aus dem Archiv extrahiert werden (mit den gleichen Benutzerrechten).

**-T, --files-from F**

liest die Dateinamen, die aus einem Archiv extrahiert oder zu einem Archiv hinzugefügt werden sollen, aus der Datei F.

**--null**

Nur im Zusammenhang mit -T. Liest Null-terminierte Zeichenketten.

**--totals**

gibt am Ende der Erzeugung eines neuen Archivs die Gesamtzahl der geschriebenen Byte aus.

**-v, --verbose**

zeigt beim Extrahieren, Erzeugen oder Hinzufügen der gerade bearbeiteten Datei den Namen. Beim Listen des Archivs werden zusätzliche Informationen entsprechend der langen Ausgabe des List-Befehls ls -l ausgegeben.



## Das Archivierungsprogramm ***tar***



***-V, --label NAME***

Das Archiv erhält das Label *NAME*, in dem auch längere Informationen – allerdings ohne Leerzeichen – für den Benutzer abgespeichert werden können.

***--version***

gibt die Versionsnummer von *tar* aus.

***-w, --interactive, --confirmation***

fragt den Benutzer vor jedem Vorgang, ob er ausgeführt werden soll.

***-W, --verify***

überprüft das Archiv auf Integrität, nachdem es geschrieben worden ist.

***--exclude DATEINAME***

schließt die Datei mit dem spezifizierten *DATEINAMEN* von dem jeweiligen Vorgang.

***-X, --exclude-from DATEI***

schließt alle Dateinamen, die in der *DATEI* enthalten sind, von dem jeweiligen Vorgang.

***-Z, --compress, --uncompress***

komprimiert bzw. dekomprimiert das Archiv wird mit dem Programm *compress*.

***-z, --gzip, --ungzip***

komprimiert bzw. dekomprimiert das Archiv mit dem Programm *gzip*.

***--use-compress-program PROGRAMM***

komprimiert bzw. dekomprimiert das Archiv mit dem angegebenen *PROGRAMM*.

***--block-compress***

Falls erstens ein Kompressionsprogramm eingesetzt wird (-z, -Z) und zweitens die Archivdatei auf ein Band geschrieben werden soll, muss diese Option eingesetzt werden, damit die Daten korrekt in das Blockformat passen.

***-[0-7] [lmh]***

spezifiziert das Bandlaufwerk (*0-7*) und dessen Schreibdichte (*low, middle, high*).



## Sicherheit ist Trumpf: Datensicherung

---



### Beispiele für die Arbeit mit *tar*

Das Inhaltsverzeichnis */home* soll mit allen seinen Dateien und Unter-Inhaltsverzeichnissen in die Datei */dos/backup/mytar* geschrieben werden, die neu erzeugt wird.

Dies kann durchaus Sinn machen, wenn */dos* und */home* auf verschiedenen Festplatten liegen, im Falle eines Defekts also nur das Original oder die Kopie betroffen sind. Mit *-v* werden die einzelnen Dateinamen während des Archivierens angezeigt.

Ein wichtiges Beispiel ist *tar* mit den Optionen *-xzf*, da die meisten *tar*-Archive heute gezippt sind und beim Installieren auf der Festplatte extrahiert werden müssen (*-x*).

Beispiel:

```
tar -xzf Wabi-2.2B-linux.tar.gz
tar -cv -f /dos/backup/mytar /home/*
```

Den Inhalt des *tar*-Archivs *mytar* kann man sich jetzt ansehen; die Option *-v* gibt zusätzliche Informationen:

```
tar -tv -f /dos/backup/mytar
```

Man kann das Verzeichnis */home* natürlich auch auf Diskette sichern:

```
tar -cv -f /dev/fd0 /home/*
```

Wenn größere Datenmengen in dem Verzeichnis */home* vorhanden sind, können mehrere Disketten nötig sein. Hier hilft die Option *-M (Multi Volume)*:

```
tar -cvM -f /dev/fd0 /home/*
```

Nachdem jeweils eine Diskette beschrieben worden ist, meldet sich *tar* mit der Aufforderung, eine weitere Diskette einzulegen:

Prepare volume #2 for /dev/fd0 and hit return:

Auch die Sicherung von Dateien mit dem gleichen Suffix ist durchaus möglich: Wer alle C-Quelltexte seines Linux-Systems auf Disketten sichern möchte, muss folgendermaßen vorgehen:

```
find / -name *.c -print >/tmp/C-Quellen
tar -cvMT /tmp/C-Quellen -f /dev/fd0
```

Da Linux seine gesamten Quelltexte mitliefert, muss man im Falle der Slackware allerdings zwölf Disketten bereithalten.



## Beispiele für die Arbeit mit ***tar***



Wer nur die Dateien eines bestimmten Verzeichnisses sichern möchte, die in den letzten zehn Tagen erstellt oder geändert worden sind, muss folgendermaßen vorgehen:

```
find /home/glinski -atime -10 -print >die_letzten_10_Tage
tar -cvMT /tmp/die_letzten_10_Tage -f /dev/fd0
```

Wer diese Art der Sicherung für sein gesamtes Dateisystem durchführen möchte, muss statt */home/glinski* nur das Wurzelverzeichnis */* angeben. Dabei werden aber natürlich alle derzeit gemounteten Laufwerke, wie z.B. CD-ROM, durchsucht. Also trennen Sie diese Laufwerke vorher mit *umount* ab.

Wer *tar* in allen Details studieren möchte, dem sei die sehr ausführliche Info zu *tar* empfohlen: Mit *info tar* und den *man pages* zu *tar* erhält man eine erschöpfende Anzahl an Bildschirmseiten zu dem Archivierungsprogramm.







# Wenn es mal nicht sofort klappt: Troubleshooting



|                                         |     |
|-----------------------------------------|-----|
| CD-ROM                                  | 443 |
| <i>mount</i> funktioniert nicht         | 446 |
| Probleme beim Floppy-Boot               | 446 |
| Soundkarten und LILO                    | 447 |
| Probleme beim Starten mit LOADLIN       | 447 |
| Probleme mit SCSI-Konfigurationen       | 449 |
| Designfehler bei Prozessoren            | 450 |
| Probleme mit dem EIDE-Controller        | 451 |
| Probleme mit Laptop-Diskettenlaufwerken | 451 |
| Kein deutscher Tastaturtreiber          | 451 |
| LILO ist verschwunden!                  | 452 |
| Probleme in der Handhabung von Linux    | 452 |



11

## Wenn es mal nicht sofort klappt: Troubleshooting

---



So stabil und ausgefuchst Linux ist und somit die meisten darauf basierenden Distributionen sind, kommt es doch auch bei diesem System manchmal zu Problemen – sowohl während des Installationsvorgangs als auch während des laufenden Betriebs. Während der Versuche, das System weitgehend anzupassen, zu erweitern oder auch Komponenten zu entfernen, können Stolpersteine im Weg liegen, deren Beseitigung zwar (meist) recht trivial ist, die aber gerade dem Anfänger zum Teil unüberwindbar zu sein scheinen, da diesem naturgemäß die Erfahrung in der Handhabung solcher Problematiken fehlt.

Eine auch nur annähernd vollständige Darstellung sämtlicher möglicher Probleme ist sicherlich aufgrund der Vielzahl unterschiedlicher PC-Komponenten, die in den handelsüblichen Marken- und No-Name-Geräten Verwendung finden, als unmöglich zu bezeichnen. Zumal diese dann auch noch von dem einen oder anderen Anwender durch Erweiterungen „aufgemotzt“ werden, deren Funktionsfähigkeit zum Teil mehr als dubios erscheint (nachträglicher Einbau einer Soundkarte, die man vielleicht ohne jegliche Dokumentation billig auf dem Flohmarkt ersteigert hat, Scanner, deren mitgelieferte Controller zu sich selbst, aber zu sonst nichts Weiterem in der Welt kompatibel sind etc.).

Auch wenn man diese vielleicht noch unter DOS oder Windows 3.x zum Laufen gebracht hat: Zumindest unter Windows 95 (gar nicht zu reden von NT) oder OS/2 bereiten sie vielleicht schon Schwierigkeiten, so dass man von Linux auch nicht immer erwarten sollte, wirklich alles zu „schlucken“.

Aus diesem Grunde finden Sie in diesem Kapitel ausschließlich einige Tipps zur „Fehlerbehebung“ zusammengestellt, die auch im Rahmen der Erstellung dieses Buches (für das überwiegend „o8/15-PCs von der Stange“ verwendet wurden) Anwendung fanden.

Sollten die im Folgenden vorgestellten Lösungsansätze jedoch nicht fruchten, bleibt nur der Verweis auf die entsprechenden FAQ-Seiten (**Frequently Asked Questions**) und Newsgroups im Internet. Gehen Sie gegebenenfalls so vor, zunächst alle nicht unbedingt für den PC „lebenswichtigen“ Komponenten auszubauen und ihn Schritt für Schritt wieder um diese Bauteile zu erweitern, bis der „Übeltäter“ gefunden ist.

Wenn zu diesem dann wirklich keine Informationen zu finden sind, bleibt nur eines: Diese Komponente umweltgerecht zu entsorgen oder sie weiterzuverkaufen oder zu verschenken (wobei Sie natürlich so fair sein sollten, den glücklichen Neubesitzer über die eventuell mit dieser Hardware auftauchenden Probleme aufzuklären) und sich ein entsprechendes Gerät neu zuzulegen oder ganz darauf zu verzichten (wer benötigt heute eigentlich noch 5,25-Zoll-Laufwerke?).

Doch hier nun einige Tipps zu Problemen, mit denen auch die Autoren das eine oder andere Mal (entweder selbst oder im Rahmen von Hilferufen aus dem Freundes- oder Kollegenkreis) konfrontiert wurden.





## **CD-ROM**

Das CD-ROM-Laufwerk ist ein immer wiederkehrender Quell des Ärgers, sowohl während der Installation als auch während des Normalbetriebs.

Hier können folgende Probleme auftauchen.

### **Probleme während der Installation**

Damit die unterschiedlichen Erweiterungskarten problemlos miteinander arbeiten können, muss festgelegt werden, wie der Computer darauf zugreifen soll.

Dies geschieht über die Einstellung einer so genannten „Interrupt-Nummer“ (IRQ – über diese Leitung kann das jeweilige Gerät den Computer über auftretende Ereignisse, wie z.B. einen erwünschten Speicherzugriff, informieren) und einer so genannten „Ein-/Ausgabeadresse“ (I/O – über diese wiederum wird die entsprechende Hardware vom Rechner angesprochen).

Viele Geräte sind standardmäßig auf die gleichen Werte (z.B. IRQ 3 und I/O ox300) eingestellt, so dass es bei deren gleichzeitigem Betrieb zu Problemen kommen kann.

Sie müssen eine entsprechende Hardwareneukonfiguration vornehmen, indem Sie diese Werte auf einem der Geräte entweder mithilfe eines beiliegenden Setup-Programms softwareseitig ändern oder die Einstellungen hardwareseitig vornehmen (wenn das entsprechende Gerät eine solche, mittlerweile als recht antiquiert betrachtete Art der Konfiguration fordert). Hierzu stehen kleine Steckbrücken (*Jumper*) oder Kippschalter (*DIP-Schalter*, wegen ihrer Größe auch „Mäuseklavier“ genannt) zur Verfügung.

Beachten Sie zur entsprechenden Jumperung die Dokumentation Ihrer Hardware und die Beschriftung der Platine.

Nur wenn sämtliche IRQ- und Adresskonflikte beseitigt sind, ist ein reibungsloser paralleler Betrieb mehrerer Geräte gewährleistet.

○—————  
Diese Problematik von IRQ- und Adresskonflikten gilt somit auch für andere Geräte als CD-ROM-Laufwerke

### **Das CD-ROM-Laufwerk ist nach der Installation „nicht mehr vorhanden“**

Die Slackware-Distribution in den vorhergehenden Versionen scheint folgendes Problem zu haben: Obwohl die Installation von CD-ROM stattfindet, meldet das Setup-Programm das CD-ROM-Laufwerk nicht an.

Es steht dann nicht wie erwartet unter dem Verzeichnis `/cdrom` zur Verfügung.

Dies bedeutet, es muss entweder mit der Angabe des entsprechenden *Special Files* und des Dateisystems gemountet oder manuell in die Tabelle `/etc/fstab` eingetragen werden.

## Wenn es mal nicht sofort klappt: Troubleshooting

---



Das „manuelle“ Mounten geschieht durch Eingabe einer Befehlszeile, die in etwa folgendem Muster entspricht:

```
mount -t iso9660 -o ro /dev/mcd /cdrom
```

Voraussetzung hierfür ist natürlich das Vorhandensein des Mount-Points */cdrom*; auch müssen Sie ggf. den Typ des von Ihnen verwendeten Laufwerks durch Angabe des entsprechenden *Special Files* ersetzen.

Bei moderneren CD-ROM-Laufwerken, die wie eine Festplatte angesprochen werden können, sähe der entsprechende Befehl in etwa wie folgt aus:

```
mount -t iso9660 -o ro /dev/hdb /cdrom
```

Der Device-Name *hdb* (er steht für das zweite Laufwerk am IDE-Bus – das erste ist im Regelfall die erste Festplatte) muss hier natürlich ebenfalls an Ihre Hardwaresituation angepasst werden (Enhanced IDE (*EIDE*) ermöglicht auch den Einsatz der Geräte *hdc* und *hdd*).

Sollte bei diesen Mount-Versuchen eine Fehlermeldung der Art

„*wrong fs type*“

oder

„*/dev/mcd already mounted*“

oder

„*/cdrom busy, or other error*“

aufreten, haben Sie vermutlich bei der Installation einen Kernel-Typ gewählt, der diesen Laufwerkstyp nicht unterstützt; starten Sie in diesem Fall das Setup-Programm erneut, um einen anderen Kernel auszuwählen, mit dem Sie das Mounting erneut versuchen.

Der nächste Schritt wäre, sich dem *Kapitel 8 Konfiguration und Kompilierung des Kernels* zuzuwenden, in dem Sie die möglichen Schritte kennen lernen, den Kernel optimal Ihrer Hardware anzupassen. Hier sind speziell die Einstellmöglichkeiten hervorzuheben, die Ihnen das „Ausklammern“ von Hardwarekomponenten, welche nicht in Ihren PC eingebaut sind, ermöglichen, so dass explizit nur der von Ihnen gewünschte Treiber verwendet wird, der dann nicht länger mit überflüssig einge bundenen Treibern kollidiert.

Sollte das manuelle Mounten des CD-ROM-Laufwerkes erfolgreich sein, können Sie die entsprechende Eintragung auch in der Tabelle */etc/fstab* vornehmen, so dass es in jeder neuen Arbeitssitzung zur Verfügung steht.





### Der Mitsumi-Treiber meldet „mcd: init failed“

Diese Fehlermeldung kann einen der folgenden drei Gründe haben:

- Ihr Rechner verfügt nicht über ein Mitsumi- (oder kompatibles) Laufwerk; Sie müssen also einen anderen Treiber wählen.
- Ihr Mitsumi-Laufwerk ist ein ATAPI-IDE-Laufwerk, für das Sie den entsprechenden IDE-Treiber (z.B. für die Laufwerke FX200, FX300 oder FX400) verwenden müssen und nicht den *mcd*-Treiber.
- Die Mitsumi-Laufwerke vom Typ Lu005, Lu005S, FX001, FX001S und FX001D müssen auf die Standardwerte *I/O 0x300* und *IRQ 10* eingestellt sein; wurden sie von Ihnen geändert, kommt es zu Problemen.

### Timing-Probleme beim CD-ROM/RAM-Zugriff

Abhängig von dem in Ihrem PC eingebauten Prozessortyp, Mainboard und den dafür vorgenommenen BIOS-Einstellungen kann es zu Timing-Problemen beim Speicherzugriff kommen. Diese äußern sich in:

- Compiler-Fehlermeldungen (*Signal 11*)
- Programmabstürzen (*Segmentation fault*)
- Fehlermeldungen beim Auspacken von Archiven von der CD-ROM mit *tar*
- vergeblichen Zugriffen auf die CD-ROM; noch nicht einmal der Befehl *ls* funktioniert

Verschiedene Ursachen können hierfür verantwortlich sein; es handelt sich dabei um Maßnahmen, die dazu dienen, die Performance des PCs (für DOS) möglichst optimal zu nutzen. Linux hingegen kann unter Umständen mit diesen Einstellungen nicht richtig zureckkommen. Dazu gehören:

- Eine Einstellung von *o Waitstates* für das Board. Dies mag zwar unter DOS funktionieren, da DOS aufgrund seines Zugriffsverfahrens (*segmentierter Zugriff*) wesentlich langsamer auf die Hardware zugreift als Linux; damit Linux jedoch fehlerfrei arbeiten kann, sollten Sie es mit der Einstellung von einem oder zwei Waitstates versuchen.
- Manche Boards besitzen Probleme mit dem Second-Level-Cache. Stellen Sie diesen versuchsweise aus, um zu überprüfen, ob er für das oben beschriebene Problem verantwortlich ist.



## Wenn es mal nicht sofort klappt: Troubleshooting

---



### **mount funktioniert nicht**

Sollten Sie bei dem Versuch, ein Laufwerk (CD-ROM, Festplatte, Diskette) in Ihr System einzubinden, die Fehlermeldung „*cannot create lock file /etc/mstab~: file exists*“ erhalten, so wurde die Datei */etc/mstab* nach dem Installationsvorgang nicht entfernt. Erledigen Sie dies „manuell“ durch die Eingabe von

```
rm /etc/mstab~
```

### **Probleme beim Floppy-Boot**

Verschiedene Kernel-2.x-Versionen (dazu gehört auch die mit diesem Buch ausgelieferte Slackware) verweigern beim Ur-Boot ihren Dienst mit 5,25-Zoll-Laufwerken.

Dieses Phänomen äußerst sich darin, dass beim Urladen des Linux-Loaders gerade noch die beiden Buchstaben *L* und *I* angezeigt werden, das System sich aber daraufhin nicht weiter röhrt.

Zurückzuführen ist dies vermutlich auf generelle Timing-Probleme des 5,25-Zoll-Laufwerks (der Fehler war mit mehreren Exemplaren reproduzierbar, wobei fabrik-neue Disketten verwendet wurden, um die Möglichkeit unzureichenden Disketten-materials auszuschließen).

Die einzige Möglichkeit, dieses Problem in den Griff zu bekommen, scheint darin zu bestehen, das 5,25-Zoll-Laufwerk mit dem vorhandenen 3,5-Zoll-Laufwerk zu vertauschen.

Verfügen Sie über beide Laufwerkstypen, müssen Sie gegebenenfalls ein neues Floppy-Kabel (neuere Floppy-Kabel ermöglichen ein problemloses Umstecken) hinzukaufen und die beiden Laufwerke entsprechend anschließen, so dass das 3,5-Zoll-Laufwerk als Laufwerk A: (unter DOS) erkannt wird.

Besitzen Sie noch kein 3,5-Zoll-Laufwerk, bleibt Ihnen kaum etwas anderes übrig, als sich jetzt eines zuzulegen (achten Sie bitte vor dem Kauf darauf, ob Ihr Floppy-Kabel auch zwei Laufwerke „verkraftet“, und kaufen Sie dieses gegebenenfalls gleich mit, damit Sie nicht zweimal zu Ihrem Hardwarehändler müssen ...).

Sie müssen außerdem noch die entsprechende Anmeldung im BIOS Ihres PCs vornehmen, welches Laufwerk wie angesprochen werden soll (als A: und B: unter DOS; beachten Sie hierzu die Dokumentation Ihres Mainboards, wie Sie Einstellungen im BIOS-Setup vornehmen können).



## Probleme beim Starten mit LOADLIN



### Soundkarten und LILO

Einige Soundkartenhersteller legen die Spezifikationen ihrer Hardware nicht offen, so dass bestimmte Funktionalitäten dieser Karten nicht genutzt werden können, da mangels ausreichender Informationen keine Treiber dafür geschrieben werden können.

Um dennoch die Leistungsmerkmale dieser Karten nutzen zu können, darf der Rechner nicht mittels LILO hochgefahren werden, sondern Sie müssen wie folgt vorgehen:

- 1 Laden Sie zunächst das Betriebssystem DOS.
- 2 Laden Sie die spezielle Treibersoftware der Soundkarte (wenn dies nicht bereits über die beiden Konfigurationsdateien *autoexec.bat* und *config.sys* erfolgt ist).
- 3 Starten Sie nun Linux per LOADLIN.

### Probleme beim Starten mit LOADLIN

Bei der Arbeit mit LOADLIN können leicht folgende Probleme auftreten:

#### Zu wenig Speicher

LOADLIN meldet „not enough memory“ (nicht genügend freier Speicher) und bricht den Ladevorgang ab. Dies passiert bei dem Versuch, große Kernel zu laden, deren Umfang 500 MByte übersteigt.

So lange man noch mit einem der mitgelieferten Kernel arbeiten muss, müssen alle speicherresidenten Programme, die beim Hochfahren von DOS automatisch von der *autoexec.bat* und *config.sys* geladen werden, vor dem Laden von LOADLIN aus dem Speicher entfernt werden.

Dazu gehören insbesondere:

- SMARTDRV.EXE und SMARTDRV.SYS
- DOSKEY
- Netzwerktreiber, wie zum Beispiel *NETX.COM*, *IPX.COM* und *CAPTURE.EXE*
- SHARE.EXE
- Treiber für die Soundkarte
- Treiber für sonstige Peripheriegeräte wie Scanner, Streamer usw.



## Wenn es mal nicht sofort klappt: Troubleshooting

Da DOS von Haus aus kein Hilfsprogramm zum Entfernen solcher speicherresidenten Programme besitzt und nur die wenigsten über Parameter verfügen, sich selbst wieder aus dem Speicher zu entfernen, gibt es nur zwei Möglichkeiten, das Laden dieser Treiber zu verhindern:

- Kommentieren Sie sie aus den Konfigurationsdateien aus (durch den jeweils in jeder betreffenden Zeile einzutragenden Befehl *REM*), so dass sie beim Hochfahren des Rechners nicht geladen werden.
- Ab MS-DOS 6.0 hat man die Möglichkeit der Menüprogrammierung. Solche Bootmenüs erlauben, dass man für die normale Arbeit mit DOS seine Standardumgebung beibehält, aber auch eine DOS-Umgebung mit minimaler Treiberausstattung starten kann, unter der LOADLIN dann Linux hochfahren kann.

Beachten Sie zu dieser Bootmenüprogrammierung Ihre DOS-Handbücher und die Onlinehilfe; hierzu ist die Eingabe des Befehls

`help menuitem`

ein guter Ausgangspunkt, sich durch die Befehlssyntax hindurchzukämpfen.

Auch zum Thema Stapeldateien beachten Sie bitte Ihre DOS-Handbücher.

### Hinweis



Besitzer niedrigerer DOS-Versionen können behelfsweise auch wie folgt verfahren:

- Laden Sie beim Hochfahren mit DOS nur die über *config.sys* zu ladenden Treiber. Alle Hilfsprogramme, die über die *autoexec.bat* geladen werden (wie zum Beispiel *DOSKEY* oder auch der Tastaturtreiber), schreiben Sie in eine Stapeldatei, die Sie manuell aufrufen, wenn Sie nicht mit Linux, sondern mit DOS arbeiten wollen. Dies sollte Ihnen unter Umständen bereits genügend freien Arbeitsspeicher für das Laden von LOADLIN verschaffen.

### „Einfrieren“ beim Hochfahren des Kernels

Ein weiteres Phänomen beim Starten von Linux mit LOADLIN ist das folgende:

LOADLIN startet, lädt den Kernel, und dann bleibt der Rechner plötzlich „eingefroren“ stehen.

Dies tritt oft dann auf, wenn entweder speicherintensive Programme unter DOS geladen waren oder wenn vor dem Starten von Linux mit Windows 3.x gearbeitet wurde. Insbesondere, wenn darunter große Dateien in WinWord geladen waren, hinterbleiben anscheinend Spuren davon im Arbeitsspeicher.



## Probleme mit SCSI-Konfigurationen



Wie so oft bei Microsoft-Produkten bleibt hier nur der Griff zum Reset-Knopf Ihres PCs (versuchen Sie gar nicht erst einen Warmstart des Rechners – laut Murphy wird sicherlich ein Cache-Speicher Ihre Hoffnung auf Besserung der Situation zunichte machen); oder schalten Sie den Computer am besten einfach ganz aus, warten zehn Sekunden und laden dann Ihr Linux.

## Probleme mit SCSI-Konfigurationen

Oftmals sind einige Hürden zu meistern, bevor man in den Genuss der Vorteile von SCSI-Rechnern kommt.

Ein Faktor, der häufig unterschätzt wird, sind die Anschlussmechanismen dieser Technologie, aber auch unterschiedliche Standards. Folgende Liste soll Ihnen einen Überblick über mögliche „Fallstricke“ dieser Technologie geben.

### Prinzipielle Probleme mit SCSI

- Sind alle Pins intakt, und liegt die rot gekennzeichnete Leitung des Verbindungskabels auch immer auf Pin 1? Verbogene Pins sollten Sie gerade biegen und natürlich in jedem Fall die Stecker richtig aufstecken.
- Ist das Kabel mehrfach um sich selbst gewunden? Wenn ja, „entwirren“ Sie es bitte – es sollte möglichst flach gehalten werden.
- Sind die Abstände zwischen den SCSI-Geräten ausreichend? Der Abstand zwischen dem Host-Adapter und dem ersten SCSI-Gerät sollte mindestens 40 cm, der Abstand zwischen allen weiteren Geräten mindestens 30 cm betragen.
- Ist die Terminierung korrekt? Der SCSI-2-Standard legt fest, dass beide Enden des Busses passiv terminiert sein müssen, wobei einer aktiv terminiert sein sollte – befindet sich der Host-Adapter an einem Ende des Busses, sollte dieser die aktive Terminierung erledigen, da er die *Termination Power* liefert. In diesem Zusammenhang ist dann darauf zu achten, dass auch wirklich nur eine aktive Terminierung vorhanden ist!
- Bei einem NCR-Host-Adapter sollten alle Geräte folgende Einstellungen haben:  
  
Parity check enabled  
XOR disabled

Kommt es bei hohen Transferraten zu Treiberproblemen? Der Grund hierfür kann ein inkohärenter Second-Level-Cache sein, den Sie also versuchsweise ausschalten sollten. Im Gegenzug sollten Sie alle Einstellungen für die unterschiedlichen PCI-Leistungsmerkmale einschalten (*enabled*).



## Wenn es mal nicht sofort klappt: Troubleshooting



### SCSI und alte Netzwerkkarten

Ältere No-Name-Ethernet-Karten vertragen sich häufig während des Urladens nicht mit dem Autoprobing des Kernels – der Kernel versucht, den SCSI-Kartentyp zu ermitteln und stößt dabei auf eine dieser „nicht hundertprozentig kompatiblen“ Netzwerkkarten, was zur Folge haben kann, dass der Rechner stehen bleibt.

Hier gibt es im Wesentlichen zwei Möglichkeiten, den Rechner eventuell doch noch zum Hochfahren zu bewegen:

- Versuchen Sie beim Starten, explizit die Daten Ihrer Netzwerkkarte als Bootparameter anzugeben. Dies geschieht nach dem Muster

`ether=IRQ, I/O, INTERFACE`

z.B.

`ether=5,0x300,eth0`

Sollte diese Maßnahme nicht fruchten, verfahren Sie wie folgt:

- Bauen Sie die Netzwerkkarte für den Urladevorgang aus, fahren Sie Linux hoch, und erstellen Sie einen Kernel mit dem entsprechenden SCSI-Treiber und dem Treiber für die Netzwerkkarte (beachten Sie hierzu auch das Kapitel 8 *Konfiguration und Kompilierung des Kernels*).

Falls das auch nicht hilft, sollten Sie vielleicht in Erwägung ziehen, doch DM 39,90 in eine neue Netzwerkkarte zu investieren (lange genug herumgeärgert mit der alten haben Sie sich schließlich ...) – Sie können Ihre alte Karte dann ja vielleicht in Ihren Win/DOS- (95-, NT-, ...) PC einbauen, der mit dem Linux-Rechner vernetzt werden soll.

### Designfehler bei Prozessoren

Sowohl unter Zentral- als auch unter den Co-Prozessoren gab es in den letzten Jahren einige mit Designfehlern.

#### Ältere i486-DX/100-Prozessoren

Einige frühe i486-DX/100-Prozessoren haben einen Designfehler bezüglich der Instruktion *hlt* – falls der Prozessor angehalten wurde, kann man ihn nicht wieder aktivieren.

Für diese Prozessoren gibt es als Bootparameter die Option *no-hlt*: Linux bleibt bei Verwendung dieser Option in einer Endlosschleife aktiv, auch wenn es nichts für den Prozessor „zu tun“ gibt. Damit kommt der Prozessor gar nicht erst zu einem Halt, da er ja „künstlich beschäftigt“ wird. Er muss somit auch nicht wieder neu aktiviert werden.



## Kein deutscher Tastaturtreiber



### Co-Prozessoren

Einige Co-Prozessoren zeigen ein fehlerhaftes Verhalten, wenn sie in den Protected Mode geschaltet werden. Damit dies nicht zum Tragen kommt, können Sie die Nutzung des Co-Prozessors abschalten, indem Sie den Bootparameter *no387* übergeben.

Beachten Sie hierbei, dass, wenn Sie diese Option verwenden, auf jeden Fall die Softwareunterstützung für Fließkommaberechnungen im Kernel aktiviert werden muss.

### Probleme mit dem EIDE-Controller

Einige EIDE-Controller haben Designfehler ab Werk und arbeiten nicht korrekt, wenn ein zweiter Controller angeschlossen wird.

Hier gibt es für eine Anzahl der Chipsätze, die auf solchen Controllern sitzen, eine Softwareunterstützung, um dieses Verhalten zu korrigieren.

### Probleme mit Laptop-Diskettenlaufwerken

Die Diskettenlaufwerke folgender Laptops weisen Designfehler bzw. -eigenheiten auf:

- IBM Thinkpad
- IBM L40 SX
- HP Omnibook

Für diese Rechner gibt es ebenfalls eine Softwareunterstützung, die durch Bootparameter gesteuert werden kann.

### Kein deutscher Tastaturtreiber

Während des Ur-Boots ist noch kein deutscher Tastaturtreiber geladen, so dass fast alle Sonderzeichen nicht der deutschen Tastaturbelegung entsprechend aufzufinden sind.

Da nun aber die Bootparameter

- das Gleichheitszeichen
- den Slash und
- eine Reihe weiterer Sonderzeichen

benötigen, ist die Eingabe dieser Zeichen elementar.



## Wenn es mal nicht sofort klappt: Troubleshooting

---



Für Benutzer, die es nicht gewohnt sind, mit der US-Belegung zu arbeiten, ist im *Anhang A Die Tastaturbelegung* eine Tabelle mit den Sonderzeichen der US-Tastatur und ihren „deutschen Entsprechungen“ beigefügt.

### LILO ist verschwunden!

Dieses Phänomen ist bekannt und wieder einmal Microsoft zu verdanken.

Wer den Linux-Loader im **Master Boot Record (MBR)** installiert hat und danach Windows 95 oder Windows 98 auf seinem Rechner installiert, dem wird eben dieser MBR gnadenlos „übergebügelt“, ohne dass Wing95 eine Sicherheitsabfrage für nötig erachtet – LILO wird also überschrieben. (Wurde Wing95 etwa die Philosophie „Es soll kein anderes Betriebssystem geben neben dir!“ eingepflanzt?)

Hier hilft nur die erneute Installation von LILO (woraufhin beide Systeme friedlich nebeneinander auf dem gleichen PC existieren).

Starten Sie hierzu Linux über LOADLIN oder von Diskette, und bemühen Sie abermals das Setup-Programm zur Einrichtung von LILO unter

`/etc/lilo.conf`

Die Details zur Installation finden Sie im *Kapitel 1 Erstinstallation* beschrieben.

Doch auch mit der Handhabung von Linux kann es einige Probleme geben.

## Probleme in der Handhabung von Linux

### Ich kann mich nur als *root* einloggen

Entweder haben Sie bei der Einrichtung Ihres Beispiel-Users einen Fehler gemacht, so dass dieser nicht die erforderlichen Rechte besitzt, oder Ihr System verfügt über eine Datei

`/etc/nologin`

Ist Letzteres der Fall, ergänzen Sie das Skript

`/etc/rc.local`

um den Eintrag

`rm -f /etc/nologin`

Andernfalls überprüfen Sie die Rechte in Ihrer Shell.



## Probleme in der Handhabung von Linux



### **Der Bildschirm enthält nur wirre Zeichen, keine Buchstaben**

Durch irgendeinen unglücklichen Fehler haben Sie Binärdaten auf Ihren Bildschirm gesendet.

Um wieder einen lesbaren Bildschirm zu erhalten, geben Sie Folgendes ein:

```
echo [Strg]+[V], [Esc], [C], [←]
```

### **Beim Kompilieren erfolgt die Fehlermeldung „signal 11: internal compiler error“**

In 99 Prozent der Fälle deutet dies auf ein Hardwareproblem hin. Dies kann im Allgemeinen dadurch beseitigt werden, dass Sie die Anzahl der Waitstates im CMOS erhöhen.

Als „letzte Verzweiflungstat“ können Sie auch versuchen, den RAM-Cache auszuschalten, was allerdings erhebliche Performance-Verluste nach sich zieht.

### **Fehler beim Kompilieren des Kernels**

Vergewissern Sie sich, dass

```
/usr/include/linux
```

und

```
/usr/include/asm
```

keine Verzeichnisse, sondern symbolische Links auf

```
/usr/src/linux/include/linux
```

und

```
/usr/src/linux/include/asm
```

sind.

Wenn nötig, löschen Sie diese und verwenden den Befehl

```
ln -s
```

um diese Links zu erstellen.

Wenn Sie einen Patch verwenden, müssen Sie die Option

```
-p0
```

## Wenn es mal nicht sofort klappt: Troubleshooting

---



oder

-p1

benutzen, da er sonst nicht richtig angewendet wird. Beachten Sie hierzu auch die *man pages* für den Befehl *patch*.

### Als *root* erhalte ich die Fehlermeldung „command not found“

Obwohl sich das aufgerufene Programm eindeutig im aktuellen Verzeichnis befindet, erhalten Sie die Fehlermeldung „command not found“.

Dies ist ein Zeichen dafür, dass das aktuelle Verzeichnis nicht in der Umgebungsvariablen *PATH* integriert ist.

Korrigieren Sie dies, indem Sie eingeben:

```
export PATH=$PATH:.
```

Oder rufen Sie das Programm unter Angabe des aktuellen Verzeichnisses auf:

```
. /PROGNAME
```





## X-Konfiguration und KDE



|                                                       |     |
|-------------------------------------------------------|-----|
| Die Entwicklung der grafischen Betriebssystemaufsätze | 456 |
| Die Funktionsweise von X Window                       | 457 |
| Die Einrichtung der grafischen Oberfläche             | 458 |
| Sax                                                   | 459 |
| Die Konfiguration mittels <i>xf86config</i>           | 466 |
| KDE                                                   | 467 |
| Das Einrichten von Icons auf dem Desktop              | 490 |



12

## X-Konfiguration und KDE

---



Bisher ist Linux in diesem Buch dem Benutzer als System an der Befehlszeile entgegentreten, und so war dieser Modus auch lange Jahre die einzige Möglichkeit, mit dem System zu kommunizieren. Auch hierdurch besitzt Linux – neben anderen UNIX-Systemen – den Ruf, relativ kryptisch zu sein. Allerdings sollte der geneigte Leser an dieser Stelle des Buches wissen, was die Befehlszeile

```
tar -xvzf /opt/downloads/pingu/blackpenguin-0.2.tar.gz
```

bewirkt: Sie packt natürlich die komprimierte tar-Datei *blackpenguin-0.2.tar.gz* im Verzeichnis */opt/downloads/pingu/* aus. Danach noch fix durch den Compiler, und schon kann man Black Pinguin spielen, geschrieben von Holger Priebs ([www.priebs.de](http://www.priebs.de)). In einer Zeit, in der sich nicht jeder Benutzer mehr mit derartigen Befehlszeilen herumschlagen möchte, in der nicht jeder Benutzer wissen will, was wirklich passiert, sondern ungeduldigerweise Ergebnisse sehen möchte, treten die grafischen Betriebssystemaufsätze in den Vordergrund. Der Benutzer möchte nicht mehr wissen, was passiert, wenn er auf das Gaspedal tritt, sondern will, dass das Auto schneller wird. Diese Herangehensweise hilft auf der einen Seite natürlich der allgemeinen Verbreitung von Rechnersystemen, auf der anderen Seite ist dann natürlich die Existenz eines ADAC notwendig, wenn niemand mehr weiß, was wirklich passiert, wenn man das Gaspedal drückt. Hier setzen die grafischen Betriebssystemaufsätze an, die (fast) alle möglichen Aktionen des Rechners per Mausklick auslösen.

## Die Entwicklung der grafischen Betriebssystemaufsätze

Die Erfindung der wichtigsten Bestandteile einer grafischen Oberfläche fand in den Xerox-Laboratories in Palo Alto in den 70er Jahren statt – nebst vieler anderer Erfindungen, wie dem Ethernet, den Grundlagen der Seitenbeschreibungssprache und der objektorientierten Programmierung ([www.parc.xerox.com/history.html](http://www.parc.xerox.com/history.html)).

Jedenfalls wurde hier die Maus erfunden, das Fenster und das Icon. Vor diesem Hintergrund mutet der juristische Streit zwischen Apple und Microsoft, wer die grafische Oberfläche erfunden hat, natürlich wie ein Witz an.

Alle diese grafischen Oberflächen, die dem Benutzer entgegentreten, sind nichts anderes als Aufsätze auf das eigentliche Betriebssystem und sollen dem Benutzer den Umgang mit dem eigentlichen Betriebssystem vereinfachen. So setzt Windows 3.x, 9x und ME immer noch auf DOS auf, während Windows NT auf einem von Microsoft (mithilfe von gekauften Programmierern von Digital Equipment) selbst entwickeltes UNIX aufsetzt. Atari setzte auf sein Betriebssystem TOS seinerzeit die Betriebssystemoberfläche GEM auf.

Nachdem UNIX in den 80er Jahren an Marktanteilen im Serverbereich zunahm, wurden auch die ersten UNIX-Rechner für rechenintensive Aufgaben am Arbeitsplatz eingesetzt. Nun bringt die Materie es mit sich, dass rechenintensive Probleme oft mit visuellen oder gut visualisierbaren Problemen daherkommen, wie z.B. Grafik, Strömungstechnik oder Video. Das setzt also auch für UNIX-Systeme eine grafische



## Die Funktionsweise von X Window



Benutzeroberfläche voraus. Mitte der 80er wurde von einigen UNIX-Herstellern das X-Konsortium gegründet, um diese zu realisieren. 1987 wurde die erste Version der Oberfläche X Window veröffentlicht, und in der Tat konnte man X-Terminals auf der Basis eines 386er-PCs schon 1989 bewundern. Inzwischen wird die X-Oberfläche von der Open Group gepflegt. XFree86 ist die Implementierung von X-Serven für PCs, die – wie Linux – auch frei erhältlich ist.

## Die Funktionsweise von X Window

Wie so vieles in dem „Werkzeugkasten“ Linux ist auch X Window modular aufgebaut. Eine funktionierende X-Oberfläche benötigt immer zwei verschiedene Bestandteile, die miteinander kommunizieren:

- Einen X-Server, der auf der entsprechenden Hardware, also der Grafikkarte aufsetzt. In der Tat gibt es für zahlreiche verschiedene Grafikchips jeweils (mindestens) einen X-Server. Dieser Server stellt grafische Primitive zur Verfügung.
- Einen Window-Manager, der sich um die Anordnung und das Aussehen der Fenster kümmert und entsprechende Aktionen wie das Starten eines Programms auslöst. Der Window-Manager ist die Schnittstelle zu dem Benutzer.

Dieser modulare Aufbau bedeutet u.a., dass Server und Window-Manager nicht auf demselben Rechner laufen müssen, sondern dass das Display exportiert werden kann und der eventuell leistungsschwache Desktop-Rechner dargestellt wird, während die Anwendung und der X-Server auf einem leistungsstarken Server an beliebiger Stelle im lokalen Netz laufen.

### Die X-Server

Die X-Server gibt es inzwischen für jeden gängigen Grafikchip, wobei für Neuentwicklungen von Chips eine Entwicklungszeit für den Server von ca. 3 Monaten eingeräumt werden muss. Allerdings kann man sich in der Zwischenzeit mit dem SVGA-Server behelfen, der die Arbeit mit dem System ermöglicht, aber natürlich nicht alle Leistungsmerkmale der Grafikkarte ausschöpft. Für professionellen Einsatz verkaufen außerdem einige Firmen – wie z.B. SuSE – die so genannten „accelerated X-Server“ für relativ schmales Geld.

Der X-Server ist also zuständig für die primitiven Grafikbefehle wie „zeichne ein Rechteck in der Breite x und der Höhe y an der Stelle x1/y1 mit der gewählten Farbe“. Dies ist für den Bildschirmaufbau völlig ausreichend, da letztendlich alle Bildschirmelemente Rechtecke sind – wenn es auch manchmal anders aussieht.





### Der Window-Manager

Die Schnittstelle zwischen X-Server und Window-Manager ist also genormt und bekannt und relativ einfach gehalten. Hierdurch war es möglich, dass eine Vielzahl von Window-Managern programmiert wurde. Einige von ihnen entstanden zum einen aus dem professionellen Bedürfnis, für ein spezielles UNIX mit seinen speziellen Anwendungen eine Fensterverwaltung zu schreiben. Andere Window-Manager entstanden aus dem Bedürfnis, der bisherigen Oberfläche auf anderen Rechnern möglichst nahe zu kommen. Einige Beispiele für Window-Manager sind nachfolgend aufgeführt:

- *fwm*, der robuste und spartanische Standard-X-WM
- *afterstep*, ein NeXTStep-ähnlicher Window-Manager
- *amiwm*, ein Amiga-ähnlicher Window-Manager
- *fwm95*, die Weiterentwicklung von *fwm* im Look & Feel von Windows 95
- *mlvwm*, ein Macintosh-ähnlicher Window-Manager

Ein Nachteil der Window-Manager sei hier nicht verschwiegen: Es ist keine Schnittstelle für „Säg & Kleb“ (*Cut & Paste*) vorgesehen, so dass dieses gerne genutzte Leistungsmerkmal entweder komplett entfällt oder aber nur für Programme vorhanden ist, die für diesen Window-Manager speziell entwickelt wurden. Hier setzen die späteren Entwicklungen *KDE* und *gnome* an, die dem Benutzer den gleichen Komfort bieten wollen, wie ihn auch andere grafische Aufsätze bieten. Dazu aber später mehr.

### Die Einrichtung der grafischen Oberfläche

Bezogen auf die SuSE-Distribution bezogen, ist dieser Abschnitt inzwischen in den meisten Fällen nicht mehr notwendig: Das Installationswerkzeug YaST2 nimmt Ihnen dankenswerterweise bei sehr vielen Grafikkarten die Arbeit fast komplett ab, und der Benutzer kann nach der Installation sofort mit dem „*Muff*“ (**Maus- und Fenster-Firefanz**) beginnen.

Die Praxis aber zeigt, dass das schöne, buntblinkende YaST2 immer wieder einmal an den Klippen der bösen, grauen Praxis scheitert. Das ist in den wenigsten Fällen YaST2 anzukreiden, sondern hat mit den extrem kurzen Entwicklungszyklen der Hardwarehersteller zu tun, und so ist die Super-Duper-Grafikkarte von heute in sechs Monaten schon wieder ein alter Hut, bei dem der Hardwarehändler nur noch mitleidig lächelt – weil er schon wieder etwas viel „Besseres“ im Angebot hat. Nun kann es aber vorkommen, dass eben für diese nagelneue Karte noch keine X-Unterstützung vorhanden ist. Zum anderen sind Notebooks und Laptops immer wieder eine Quelle von unerquicklichen Inkompatibilitäten: In dem Konstruktoreswahn, möglichst den Stromverbrauch zu minimieren, wird laufend vergessen, dass es für





einen PC bestimmte Normen gibt, an die man sich halten sollte. Dies gilt auch gerne für einen der größten Stromfresser im tragbaren Computer, nämlich das Display.

So kann es durchaus zu der Situation kommen, dass man trotz aller Automatisierung der Installation doch wieder zu den bisherigen Werkzeugen greifen muss, und sich seinen X Server (mehr oder minder) manuell einrichten muss. Hierzu bieten sich drei Werkzeuge an:

- für die Versionsreihe 3.x der XFree86-Server das von SuSE mitgelieferte Installationswerkzeug *sax*,
- für die neue Versionsreihe 4.x der XFree86-Server das auch von SuSE mitgelieferte Installationswerkzeug *sax2*,
- für die ganz hartnäckigen Fälle, in denen auch diese beiden Programme versagen, kann man durchaus mit dem „Steinzeit-werkzeug“ *xf86config* erfolgreiche Ergebnisse erzielen, bei dem man alle notwendigen Werte an der Textkonsole eingibt.

## Framebuffer

Falls sich die Grafikkarte trotz aller Bemühungen als inkompatibel herausstellt und leider ohne X-Unterstützung ist, kann man versuchen, den Weg über das Framebuffer-Gerät zu gehen. Es wurde in die Linux-Welt von den Non-PC-Benutzern getragen, für die es oft keine andere Möglichkeit gab, Grafik darzustellen als eben durch dieses virtuelle Gerät.

Der Framebuffer ist nichts anderes als die programmtechnische Abstraktion einer Grafikkarte. Hierdurch können alle (Hardware)-Grafikkarten angesteuert werden, die Vesa-2-kompatibel sind. Die Unterstützung für den Framebuffer ist im Kernel verankert; man muss also einen Kernel besitzen, der die entsprechenden Treiber bereits einkompiliert hat, was bei den SuSE-Standardkernen der Fall ist. Bei einem selbsterzeugten Kernel muss natürlich darauf geachtet werden, diesem Treiber aufzunehmen.

Seit Version 3.2 von XFree86 gibt es für den Framebuffer einen X-Server namens *XF86\_FBDev*. Weitergehende Information findet man im Verzeichnis */usr/src/linux/Documentation/fb*; als Einstieg ist der *framebuffer.txt* zu empfehlen.

## Sax

Sax wird an der Befehlszeile eingegeben durch

`sax`

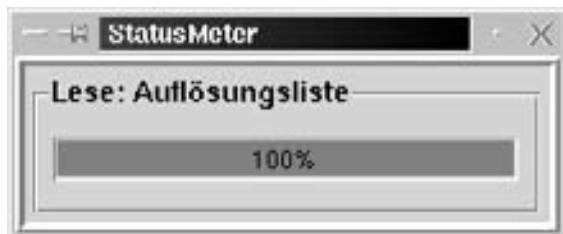
Das Installationswerkzeug liest nun diverse Daten seiner Hardwaredatenbank ein:



## X-Konfiguration und KDE

Abb. 12.1

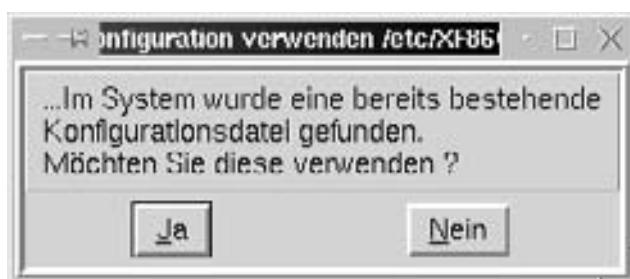
Sax liest die Hardware-datenbank ein



Falls Sie eine bestehende X-Konfiguration besitzen, fragt Sax, ob diese Konfigurationsdatei als Grundlage benutzt werden soll:

Abb. 12.2

Die bestehende Konfigurationsdatei kann benutzt werden



Danach tritt Ihnen das Programm mit seinen Registerblättern entgegen.

Abb. 12.3

Die Benutzeroberfläche von Sax





Sax hat nun die vorhandene Hardware gescannt, und in den meisten Fällen muss man die angebotenen Einstellungen nur bestätigen.

## Mauseinstellungen

Als Erstes kommt das erweiterte Dialogfeld der Mauseinstellungen zur Anwendung, das automatisch geöffnet wird.



Abb. 12.4

Die Einstellungen der Maus

Das Mausprotokoll wurde in fast allen Fällen automatisch erkannt; in den meisten Fällen wird dies entweder eine PS/2-Maus (kleiner runder Stecker) oder die serielle Maus (9-poliger-Stecker für die serielle Schnittstelle) sein.

An dieser Stelle sei noch einmal einem weit verbreiteten Vorurteil entgegengetreten, nämlich dass die Linux-Gemeinde nur aus Microsoft-Hassern besteht. Wir, die Autoren, meinen nämlich, dass die serielle Microsoft-Maus ein sehr praktisches und billiges Werkzeug ist, das sich zu Recht in sehr weiten Bereichen durchgesetzt hat und mit zu dem Besten gehört, was Microsoft je abgeliefert hat. An der zweiten Stelle unserer Lieblingsprodukte von Microsoft steht übrigens wiederum Hardware: das Windows-SoundSystem.

Auch relativ exotische Busmäuse werden von Sax erkannt. Falls nicht, kann man sie mittels der Dropdown-Liste *Protokoll* einstellen. Darunter kann man die Schnittstelle der Maus auswählen, was aber wiederum auch in den meisten Fällen bereits

## X-Konfiguration und KDE



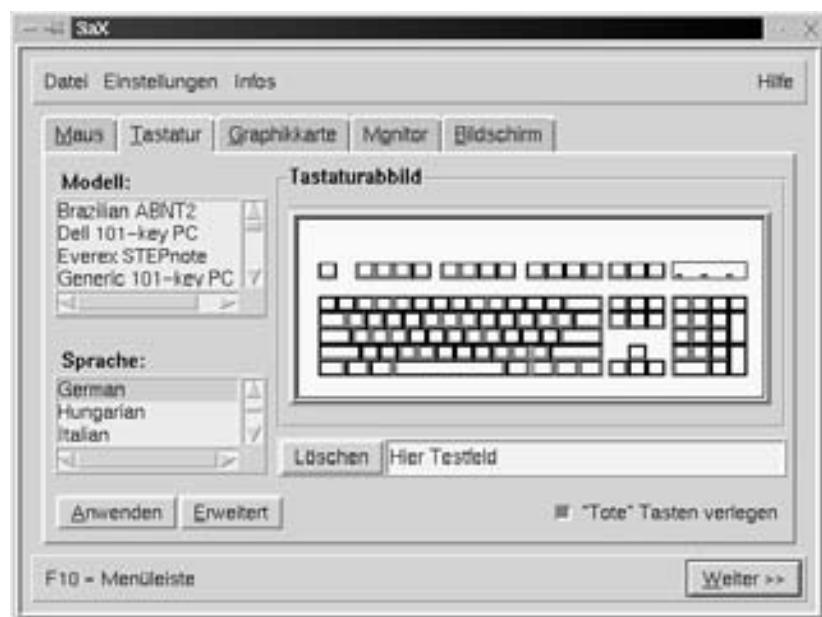
von Sax erledigt wurde. Der Neuling sei nochmals auf die Zählung der seriellen Schnittstellen hingewiesen: Sie fängt bei Null an, also ist die in der Abbildung ausgewählte Schnittstelle `/dev/ttys1` die zweite serielle Schnittstelle.

Unter *Eigenschaften* kann die Geschwindigkeit der Mausabfrage geregelt werden, falls Ihnen die Standardeinstellung zu langsam oder schnell erscheint. Wichtiger sind die Einstellungen unter *Optionen*. Fast jede Maus besitzt drei Tasten. Microsoft hat vor einiger Zeit immerhin die zweite Taste entdeckt, während unter X Window die mittlere Taste schon immer wichtige Funktionen übernommen hat. Falls man keine dritte Taste auf der Maus besitzt oder diese – wie manchmal in Billigstmäusen – überhaupt nicht angeschlossen ist, so muss man auf die Funktionalität dieser Taste nicht verzichten: Mittels der Option *3 Tasten emulieren* kann man durch gleichzeitiges Drücken von linker und rechter Taste das Signal der mittleren Taste an das System übermitteln, auch wenn diese nicht vorhanden ist. Der Effekt kann sofort auf dem Testfeld der Registerkarte *Test* überprüft werden. Hier können auch geplagte Linkshänder die Tasten vertauschen. Ein *OK* führt zur nächsten Option.

## Tastatur

Abb. 12.5

Die Einstellung der Tastatur



Auch die Tastaturerkennung findet vollautomatisch statt, und der Benutzer eines Standard-PCs findet hier die Einstellungen *Generic 104-key PC* als *Modell* sowie *German* als *Sprache*. Besitzer von Laptops mögen hier etwas anderes vorfinden, aber erstaunlicherweise klappt's auch dabei. Ein *OK* führt auch hier zur nächsten Option.





## Grafikkarte

Dies war früher ein leidiges Thema, da nicht alle Hersteller von Grafikkarten ihre Interna offen legen wollten und man entweder auf die Bemühungen von begabten Hackern angewiesen war, die trotzdem Treiber schrieben, oder sich für diese Grafikkarten mit dem SVGA-Server und 256 Farben begnügen musste. Inzwischen haben Hardwarehersteller Linux auch als Markt angenommen, und so gibt es für fast jede Grafikkarte auch einen passenden X-Server.

Falls Sax die Karte automatisch erkannt hat, findet man in der Listbox den ersten Menüpunkt —*AUTO DETECTED*— aktiviert und erhält weiter unten noch Informationen über den ausgewählten X-Server sowie über den Videospeicher. Falls eine automatische Erkennung hier nicht stattfand, so muss auf der linken Seite der Hersteller der Grafikkarte ausgewählt werden und danach auf der rechten Seite das Modell. Nach dem Anklicken des Modells teilt Sax Ihnen freiwillig mit, welcher X-Server benötigt wird.

○ Abb. 12.6

Sax teilt mit, welcher X-Server fehlt



Falls der benötigte Server noch nicht installiert ist, muss dies nachgeholt werden.

## Monitor

Als Erstes die obligatorische Warnung für alle Benutzer von (sehr) alten Festfrequenzmonitoren: Falsche Einstellungen der Monitordaten können die Hardware zerstören! Finden Sie mithilfe des Handbuchs ihre genauen Festfrequenzen heraus und stellen Sie diese penibel ein!

Glücklicherweise sind derartige Monster aus der Rechner-Frühzeit selten geworden, und wer halbwegs neue Hardware besitzt, findet hier eine große Auswahl an Herstellern und Modellen. Sax besitzt an dieser Stelle keine Möglichkeit, die Daten automatisch zu erkennen – hier ist der Benutzer gefordert. Wie bei der Einstellung der Grafikkarten befindet sich links die Liste mit den Herstellern, rechts die Liste mit den Modellen. Da sich der Monitor meistens vor Ihnen befindet, sind hier Hersteller- und Modellangaben relativ leicht zu erforschen: Sie finden diese Angaben meistens auf der Vorder-, manchmal auf der Rückseite der Hardware. Wenn sie angeklickt worden sind, erscheinen unten auf dem Registerblatt sofort die möglichen zugehörigen Frequenzen des Monitors. Falls wider Erwarten das Modell in der Tabelle von Sax nicht aufzufinden sein sollte, so ist das kein Beinbruch: Jetzt benö-



## X-Konfiguration und KDE



tigt man das Handbuch, um die möglichen Frequenzen herauszubekommen. Manche Hersteller drucken die gesuchten Werte auch auf die Rückseite des Monitors. Zum zweiten muss man die Schaltfläche *Erweitert* anklicken, um folgendes Dialogfeld zu öffnen:

Abb. 12.7

Die Einstellung von horizontaler und vertikaler Monitorfrequenz

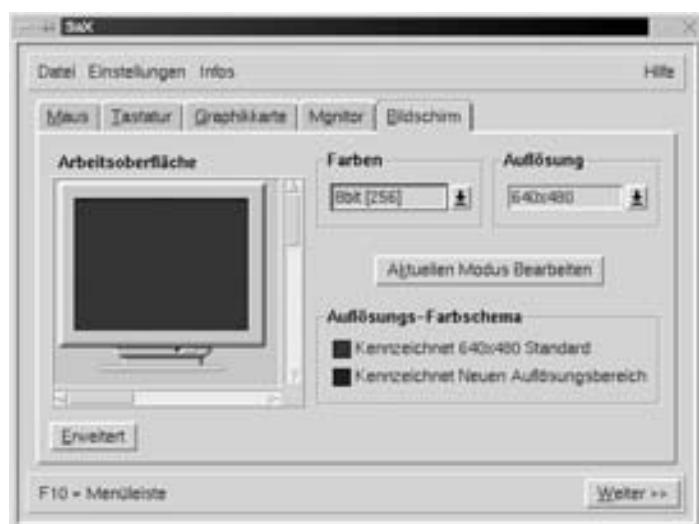


Hier müssen nun der horizontale und der vertikale Frequenzbereich nach den Angaben des Herstellers eingegeben werden, jeweils mit Unter- und Obergrenze. Diese Angaben genügen Sax – und damit dem X-Server – zur Ansteuerung des Monitors.

## Bildschirm

Abb. 12.8

Einstellung von Auflösung und Farbtiefe





Hier können nun – in Abhängigkeit von den Möglichkeiten der Grafikkarten und des vorhandenen Video-RAMs – die Farbtiefe und die Auflösung des Bildschirms eingestellt werden. Als Erstes sollte die gewünschte Farbtiefe eingestellt werden, danach die Auflösung, da diese von der Farbtiefe abhängt. Hier wird auf der Bildschirmdarstellung der Registerkarte in roter Farbe der minimale Standardbildschirm mit 640 mal 480 Bildpunkten dargestellt, in grüner Farbe die neue, derzeit gerade eingestellte Auflösung. Nachdem alle Einstellungen vorgenommen wurden, kann man nun auf die Schaltfläche *Weiter* rechts unten klicken, um erstens den Server zu testen und zweitens – wenn alles gut gegangen ist – weitere Feineinstellungen vorzunehmen. Sax fragt sicherheitshalber noch einmal nach:



Abb. 12.9

Die Sicherheitsabfrage vor dem ersten Start des X-Servers

Wie ersichtlich ist, gibt Sax noch einmal den Hinweis, was zu tun ist, wenn möglicherweise falsche Einstellungen gewählt wurden, die den Monitor beschädigen könnten. Wenn ungewohnte Effekte, konstantes Flimmern oder andere Arten unsauberer Darstellung auftreten, kann man den X-Server sofort durch die Tastenkombination **Strg+Alt+Entfernen** killen. In der Hoffnung, dass alles korrekt eingestellt ist, starten Sie den Server, und Sax startet seinen blauen Bildschirm zum Feintuning. Die Marken in den Ecken des Monitors sollten exakt sitzen, andernfalls kann man mittels der Buttons mit den entsprechenden Symbolen die sichtbare Fläche des Bildschirms verkleinern, vergrößern und verschieben, bis man ein zufriedenstellendes Ergebnis erzielt hat. Sax speichert nun die Daten nach entsprechende Abfrage. Die grafische Oberfläche ist einsatzbereit. Wie? Durch die Eingabe von

`startx`

an der Befehlszeile wird der X-Server und der voreingestellte Window-Manager gestartet. Dies ist bei einigen Distributionen *fvwm2*, bei SuSE ist es inzwischen KDE.





### Die Konfiguration mittels *xf86config*

In einigen wenigen Fällen kann es angehen, dass X nicht die meistens nagelneue Hardware erkennt, und es muss wieder zu älteren Bordmitteln gegriffen werden. Dies ist *xf86config*, welches ein textbasiertes Programm ist, das alle notwendigen Punkte nacheinander abfragt und so auch die genauen Daten für die grafische Umgebung in Erfahrung bringt. Hierbei sollte man sich aber mit allen notwendigen Handbüchern bewaffnen, da viele Details abgefragt werden.

Man wechselt also in das Verzeichnis */etc* und startet mit

*xf86config*

Nach einleitenden Worten (in Englisch) fragt das Programm folgende Punkte ab, wobei man dann die Nummer des angegebenen Menüpunkts eingeben muss: Wie oben werden Maustyp, Tastatur, Monitorfrequenzen (horizontal und vertikal), der Grafikkartentyp, der Videospeicher, eventuell der RAMDAC-Chip und der Clock-Chip der Grafikkarte, die Farbtiefe und Auflösung abgefragt. Hier ist auch einiges an Konzentration gefordert, da es keine Möglichkeit gibt, zurückzublättern. Im Falle der fehlerhaften Eingabe gilt: **[Strg]+[C]** drücken und von vorne beginnen.

*Abb. 12.10* ○

*xf86config* fragt die Mauseinstellungen ab

First specify a mouse protocol type. Choose one from the following list:

1. Microsoft compatible (2-button protocol)
2. Mouse Systems (3-button protocol)
3. Bus Mouse
4. PS/2 Mouse
5. Logitech Mouse (serial, old type, Logitech protocol)
6. Logitech MouseMan (Microsoft compatible)
7. MM Series
8. MM HitTablet
9. Microsoft IntelliMouse
10. Acacad tablet

If you have a two-button mouse, it is most likely of type 1, and if you have a three-button mouse, it can probably support both protocol 1 and 2. There are two main varieties of the latter type; mice with a switch to select the protocol, and mice that default to 1 and require a button to be held at boot-time to select protocol 2. Some mice can be convinced to do 2 by sending a special sequence to the serial port (see the ClearDIR/ClearRIS options).

Enter a protocol number:

### Grafikkarte zu neu?

In einigen, seltenen Fällen kann sein, dass die Grafikkarte derart neu ist, dass es keinen X-Server dafür gibt. Für eine Übergangszeit – der Erfahrungswert ist ca. drei Monate – bis es einen Server gibt, hilft es, den SVGA-Server zu installieren, so dass man zumindest mit 256 Farben arbeiten kann. Auch hilft etwas Suchen im Internet, ob sich schon jemand mit dem Problem beschäftigt und eventuell Zwischenlösungen anbietet. Zuerst sollte man aber natürlich bei der zuständigen Internetadresse nachsehen: [www.xfree.org](http://www.xfree.org).





## KDE

Nachdem viel Arbeit in den K-Desktop gesteckt wurde, Vieles sowohl in der grundlegenden Softwarearchitektur aber auch im Feinschliff geändert wurde, erblickte die Version 2.0 von KDE am 23.10.2000 offiziell das Licht der Welt. Vorausgegangen waren diverse Entwicklerversionen mit launigen Namen wie z.B. „Kaos“ oder „Kandidat“.

Abb. 12.11

Der neue Splashscreen  
von KDE 2.0



## K-Office 1.0

Der ehrgeizigste Teil des K-Projekts ist allerdings das K-Office, das zum Erscheinen von KDE 2.0 mit seiner eigenen Versionsnummer 1.0 aufwartete, also den Sprung in die „offizielle“ Welt gewagt hat.

K-Office besteht aus folgenden Komponenten:

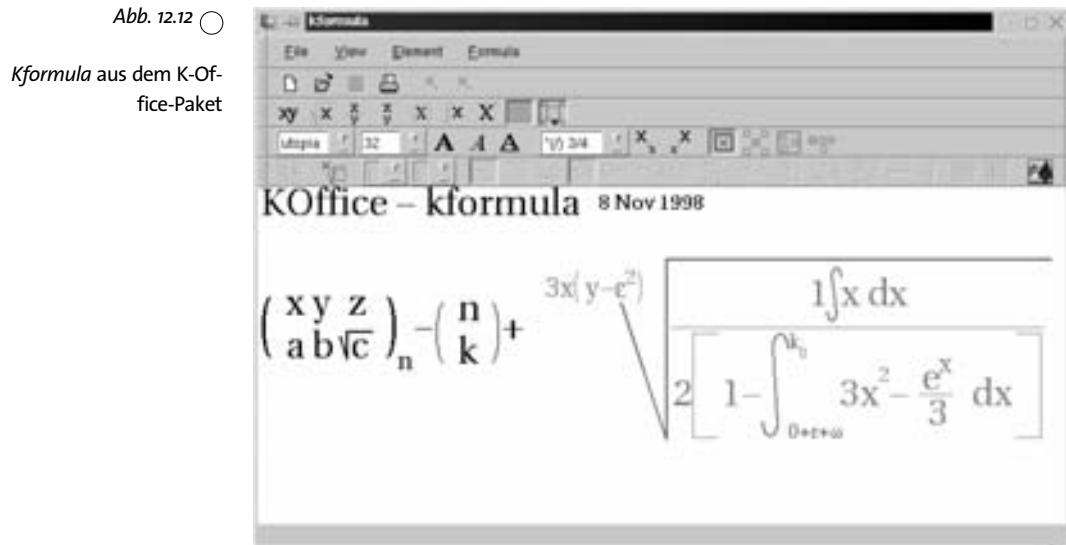
- *KWord* ist ein rahmenbasiertes Textverarbeitungsprogramm, das einige Leistungsmerkmale eines DTP-Programms aufweist. Vorbild ist der FrameMaker.
- *KSpread* ist eine Tabellenkalkulation, die u.a. auch Bilder und Diagramme in ihren Zellen beherbergen kann .
- *KPresenter* ist ein Präsentationsprogramm, das bereits weit gediehen ist. Grafiken aller Art können eingebunden werden, Rich Text kann importiert werden, die Struktur der Folien untereinander kann angezeigt werden, und die Präsentation kann als HTML-Code exportiert werden.



## X-Konfiguration und KDE



- *KIllustrator* ist ein Vektorprogramm, das sich in seinem derzeitigen Umfang durchaus mit frühen Versionen von CorelDraw messen kann.
- *Krayon* ist der Bildbearbeiter des K-Office-Packets.
- *Graphite* ist ein Grafikprogramm.
- *Katabase* ist die Datenbank des K-Projekts und noch in einem sehr frühen Entwicklungsstadium.
- *KFormula* ist ein Programm zum Editieren von komplexen Formeln.
- *Kivio* erzeugt aus Daten Diagramme aller Arten.



Das Bestechende ist, dass das K-Office ein Paket aus einem Guss wird, d.h. es ist nicht – wie so oft bei anderen Office-Paketen – ein Haufen von zusammengewürfelten Programmen, die sich in ihrer Bedienung reichlich unterscheiden. Im Gegenteil: Einzelne Komponenten eines beliebigen Programms des K-Office-Pakets können in einem anderen Programm aktiviert und benutzt werden.

## Die Bedienung des K-Desktops

Im Folgenden wird der K-Desktop in der Form beschrieben, wie er dem neuen Benutzer in seinen Standardeinstellungen entgegentritt. Wie Sie später sehen werden, können (fast) alle Bestandteile des Desktops konfiguriert werden, bis hin zu dem Punkt, dass auch ein geübter Betrachter ihn nicht notwendigerweise als K-Desktop erkennt – aber dazu später mehr. Frei nach dem Motto „Aus jeder Welt





das Beste“ wurde KDE programmiert, und so ist es dann kein Wunder, dass dem neuen Benutzer einige der grafischen Steuerungselemente bekannt vorkommen. Nachfolgend die Details.

## Das Panel

Abb. 12.13



Das Panel des K-Desktops

In der Standardeinstellung befindet sich das Panel am unteren Rand des Bildschirms und beherbergt mehrere Icons, die verschiedene Funktionalitäten beinhalten. Doch bevor darauf eingegangen wird, kommt der Neuling daher, der schon fleißig auf dem Bildschirm rumgeklickt hat, und beschwert sich bitterlich:

### Mein Panel ist nicht da!

Abb. 12.14



Der Panel-Button zum Ein- und Ausfahren des Panels

Es ist sehr wohl da; manchmal hilft auch: Erst lesen, dann klicken. Das Panel besitzt sowohl ganz rechts als auch ganz links einen Button inklusive Pfeil. Wenn auf einen dieser beiden geklickt wurde, verschwindet das Panel an der entsprechenden Seite und hinterlässt nur diesen Button, mit dessen Hilfe das Panel wieder „ausgefahren“ werden kann. Also rechts bzw. links unten den Button suchen, und mit einem Klick ist das gesamte Panel wieder sichtbar. Gedacht ist dieses Leistungsmerkmal für Situationen, in denen man möglichst viel Bildschirmfläche für sein Anwendungsprogramm zur Verfügung haben möchte.

Abb. 12.15



Der K-Button



## X-Konfiguration und KDE



Als erstes Icon neben diesem Schalter findet sich der K-Button. Ein Klick, und der Win/DOS-9x-Umsteiger fühlt sich zu Hause: Hier wird das so genannte „Start-Menü“ aktiviert, dass unter Windows derart heißt, weil dort u.a. der Menüpunkt *Beenden* verankert ist. Nachdem man also das K-Menü aktiviert hat, findet man hier alle bereits installierten KDE-Programme nebst Zugriffsmöglichkeiten auf den Desktop im direkten Zugang oder – der Übersichtlichkeit halber – in Untermenüs.

Abb. 12.16



Das K-Menü



Der unterste Menüpunkt mit der Beschriftung *Abmelden* bzw. *logout* soll hier besondere Erwähnung finden: Hier kann KDE verlassen werden und man kehrt zur Befehlszeile zurück.

Anmerkung: Wer KDE nach der Installation in englischer Sprache vorfindet und dieser nicht mächtig ist, der sollte jetzt kurz zum Abschnitt „Sprachen“ weiter unten im Text springen, um die Sprache seiner Wahl – wahrscheinlich Deutsch – ein-





zustellen. Weiterhin sind im Panel einige Programme, wie z.B. ein Taschenrechner, die Onlinehilfe oder die Terminal-Emulation des Desktops per Voreinstellung verankert.

## Der Konqueror

Der Konqueror ist *die* eierlegende Wollmilchsau für die Alltagsarbeit. Seinen Ursprüngen als simpler Datei-Manager ist er längst entwachsen. Er vereint zum einen die traditionellen Funktionen zur Dateimanipulation, wie man sie seit dem Erscheinen des Norton Commanders kennt: Kopieren, Umbenennen, Löschen und Verschieben von Dateien, das Erzeugen und Löschen von Verzeichnissen. Außerdem bietet er hier zusätzlich die für Linux notwendigen Zusatzaktionen, wie z.B. das Einbinden von CD-ROM- und Diskettenlaufwerken in den Dateibaum sowie die Vergabe und das Entziehen von Benutzerrechten.

Aber während es vor zehn Jahren noch genügte, die Dateien des eigenen Rechners zu manipulieren oder bestenfalls die im lokalen Netzwerk, so ist heute die gesuchte Information im Internet nur noch wenige Links entfernt – selbst wenn die Information Tausende von Kilometern entfernt liegt. Und so haben die KDE-Entwickler konsequenterweise einen Browser für das WWW sowie einen FTP-Client für Up- und Downloads integriert.

Hier fällt nun hoffentlich auch diversen Systemadministratoren ein Stein vom Herzen, denn diese Menschen sind in letzter Zeit immer auf der Suche nach einem kleinen, handlichen, pflegeleichten Browser, damit die Mitarbeiter im firmeneigenen Intranet arbeiten können. Hierzu ist Netscape inzwischen einfach zu monströs geworden und bietet auf schwachbrüstigen Rechnern schlicht unerträgliche Startzeiten. Mit dem Konqueror ist dieses Problem nun endlich gelöst.

Wie bekannt ist, kommen (fast) alle Linux-Distributionen mit einer großen Menge von Dokumentation. Die Info-Seiten und die Man(ual)-Pages sind die bekanntesten. Da man nun mit dem integrierten Browser ein entsprechendes Werkzeug besitzt, was liegt da näher, als die vorhandenen Dokumentationen in einem System unter der Oberfläche des Konqueror zu vereinen.

Der Konqueror kann auf die verschiedensten Arten gestartet werden – wie sollte es auch anders sein, bei dem sehr flexiblen KDE.

- Aus dem K-Menü muss der Menüpunkt *System* und dort eben *Konqueror* angeklickt werden.
- Im Panel ist ein sinnreiches Icon mit einem kleinen Häuschen sowie einem Aktenordner im Hintergrund eingelassen: Der Konqueror startet mit dem jeweiligen Heimatverzeichnis.
- Da die interaktive Hilfe sich auch des Konquerors bedient, kann bei fast alle K-Programmen auf **[F1]** gedrückt werden – ein Konqueror mit dem entsprechenden Handbuch erscheint.



## X-Konfiguration und KDE

- Befehlszeilenkünstler haben ja sowieso immer mindestens ein Terminal geöffnet, um das Hantieren mit der Maus zu reduzieren. Hier langt die Eingabe der Zeile *konqueror &*.
- Wer nicht extra ein Terminal öffnen möchte, aber lieber tippt als klickt, der tippt als erstes **Alt**+**C** ein, dann öffnet sich die Befehlszeileneingabe. Auch hier muss einfach *konqueror* eingegeben werden.
- Wer den Konqueror im Browermodus starten möchte, um zu surfen oder sich lokal HTML-Seiten anzusehen, der startet vom K-Menü aus über *Internet*, dort *Web-Browser* den Konqueror.
- Wird in einem anderen K-Programm auf eine Internetadresse oder auf ein Verzeichnis geklickt, so ist die Chance groß, dass der Konqueror startet. Aber nicht immer haben sich die Entwickler an diese Vorgabe gehalten. Schon „Kandalfs nützliche Tipps“, die beim ersten Start des KDE erscheinen, halten sich leider nicht an diese Vorgabe.

Wie Sie aber sehen werden, ist es ohne Probleme nahtlos möglich, zwischen den verschiedenen Modi des Konqueror umzuschalten. Nach dem Start durch einen Klick auf das Icon *Home* im Panel stellt sich der Konqueror wie folgt dar:



○ Abb. 12.17

Der Konqueror nach dem Start

Wie Sie es vom K-Desktop gewohnt sind, können die Einstellungen für das Look & Feel weitestgehend dem eigenen Geschmack angepasst werden. Hier werden lediglich die Voreinstellungen berücksichtigt.





## Die Modi des Konquerors

Zwischen den Modi des Konquerors kann wiederum auf verschiedensten Wegen umgeschaltet werden. Dies geschieht teilweise fast unmerklich getreu dem uralten UNIX-Motto „Alles ist eine Datei“. Im linken Fenster des Konqueror sehen wir schlicht die drei Einträge *Netzwerk*, *Persönliches Verzeichnis* und *Wurzelverzeichnis*. Während die letzten beiden Zugriff auf die entsprechenden Verzeichnisse geben und Dateimanipulationen ermöglichen, schlummern unter dem Eintrag *Netzwerk* die drei Menüpunkte *FTP-Archive*, *Websiten* und *Windows-Ressourcen*. Durch Klicken auf die ersten beiden Menüs öffnen sich jeweils Untermenüs mit der FTP-Seite und der Webseite der KDE-Organisationen. Wer an seinem Rechner einen Internetzugang besitzt, der kann sofort mit dem Surfen oder Downloaden loslegen. Die Interneteinstellungen des Konquerors werden übrigens im *Kapitel 14 Mit Linux ins Internet Internet* dieses Buches besprochen – eben weil der Konqueror auch ein praktischer und schneller Browser ist.

Unter dem Menüpunkt *Windows-Ressourcen* findet sich für den Linux-Benutzer in einem heterogenen Netzwerk eine Spezialität: Ohne sich mit der Konfiguration als SMB-Client auf Systemebene beschäftigen zu müssen, kann man mit dem Konqueror auch auf die Daten der berüchtigten Windows-Arbeitsgruppen zugreifen. Dazu muss nur vorher im KDE-Kontrollzentrum (s.u.) unter *Netzwerk > Windows-Ressourcen* ein für allemaal der SMB-Client eingerichtet sein. Dann wird der Datenaustausch mit den Kollegen auf ihren absturzgefährdeten Windows-Rechner zum Kinderspiel.

## Die URL-Leiste

Wie von anderen Browsern gewohnt, besitzt auch der Konqueror eine URL-Leiste. Hier wird normalerweise die gewünschte Internetadresse eingegeben, in der Regel ein FTP- oder WWW-Server. Die zugehörigen Protokolle, die von der eigentlichen Adresse durch zwei Slashes getrennt sind, heißen *ftp* und *http*. Von Netscape kennt man hier z.B. die Möglichkeit, auf eine lokale Datei zuzugreifen, indem als Protokollkennung *file* eingegeben wird. Dieses Prinzip haben die Entwickler massiv weiterentwickelt. Die einzelnen Präfixe für die verschiedenen Dienste lauten folgendermaßen:

- *http*: für eine Adresse im WWW.
- *ftp*: für einen File-Transfer-Server.
- *file*: für eine Datei auf dem lokalen Rechner oder Netzwerk. An bestimmten Dateitypen wird gleich die zugehörige Aktion ausgeführt.
- *man*: um eine bestimmte Manual-Page aufzurufen. Die Eingabe von *man*: gibt den Index der *man pages* aus.
- *smb*: um auf Windows-Netzwerke zugreifen zu können.
- *info*: um auf die GNU-Infoseiten zuzugreifen. *info:dir* gibt die Liste der verfügbaren Infoseiten aus.



## X-Konfiguration und KDE

---



Bei allen Eingaben versucht der Konqueror erstens die Eingaben auf ihre Schlüssigkeit zu überprüfen und etwaige Fehler – etwa das Verwechseln von `ftp://` und `http://` – selbstständig zu korrigieren. Zweitens sind einzelnen Dateitypen bestimmte Aktionen zugewiesen. So werden Bilder der Formate GIF, JPEG oder PNG nach einem Klick auf die entsprechende Datei angezeigt. Einfache unformatierte Texte, C-, C++ und Pascal-Quelltexte werden nach einem Klick zur Ansicht angezeigt. Für weitere Dateiformate, die der Konqueror nicht von Haus aus unterstützt, können jeweils Programme zugeordnet werden, die ein Betrachten und Bearbeiten dieser Datei ermöglichen. Dies geschieht unter dem Menüpunkt *Einstellungen > Einrichten > Dateizuordnungen*. Hier könnte man z.B. dem Dateityp TIFF den *gimp* zuordnen oder dem Dateityp RTF (Rich Text Format) das StarOffice.

### Ansicht und Einstellungen

Die Möglichkeiten, das Aussehen des Konquerors zu verändern, sind derartig vielfältig, dass wir dem verspielten Leser raten, sich damit einige Zeit zu beschäftigen. Die anderen beschränken sich auf das Notwendigste nach ihrem Geschmack. Der Inhalt des rechten Fensters lässt sich mittels Icons oder Text darstellen, mehrspaltig oder als Baum; die einzelnen Attribute können bei der Darstellung an- bzw. abgewählt werden; die Größe der Icons kann z.B. für Laptops erhöht werden, für große Bildschirme verringert; der Hintergrund kann eingefärbt werden oder mit einem Bild verziert werden – hier kann man wählen zwischen vorgefertigten Mustern, man kann aber auch das Bild seiner Herzallerliebsten dort anzeigen lassen. Und man kann Bilder der Formate GIF, PNG und JPEG gleich mit ihrem Inhalt auf Icongröße anzeigen lassen – Apple lässt grüßen.

Unter *Einstellungen* lassen sich – wie von anderen Programmen gewohnt – die verschiedenen Leisten zur Benutzerführung ein- bzw. ausblenden, im Einzelnen die Menüleiste, die Werkzeugleiste, die URL-Leiste und die Lesezeichenleiste. Wer sie alle ausblendet, sollte allerdings gut mit den Tastaturkürzeln des Konquerors vertraut sein. Unter *Einstellungen > Einrichten* lassen sich zum einen Schrifttyp und Verhalten des Datei-Managers verändern. Außerdem gibt es noch diverse Möglichkeiten, den Browser zu beeinflussen, wobei die wichtigste wohl die Eingabe des Proxy-Servers ist – falls man über ein lokales Netzwerk surft.

### Dateioperationen

Die Dateioperationen folgen den derzeitigen Standards, allerdings sollte man einige praktische Möglichkeiten erwähnen. Zum einen kann eine beliebige Anzahl von Instanzen des Konquerors gestartet werden. Das ist praktisch, wenn man z.B. in den Tiefen des Dateibaums Dateien verschieben oder kopieren muss. Man zieht die markierten Dateien von dem einen Konqueror in das Dateifenster des nächsten und muss dann nur noch die Frage des aufklappenden Menüs beantworten, ob man kopieren, verschieben oder verknüpfen möchte. Und in der Tat ist es praktisch, weitere Konqueror offen zu haben: einen zum Surfen, einen für die Onlinehilfe, einen für Dateioperationen und einen, um jede Menge Dateien von einem FTP-Server zu holen; z.B. die neuesten KDE-Programme!





Die zweite Möglichkeit ist das viel zu wenig bekannte Benutzen der Zwischenablage auf Dateioperationen: Das markierte Verzeichnis mittels **Strg+C** in die Zwischenablage befördern, das gewünschte Titelverzeichnis aktivieren und mittels **Strg+V** dorthin kopieren.

Sehr schön ist die Möglichkeit, *tar*-Dateien – auch in gezippter Form – schlicht als Verzeichnis zu behandeln. So kann zum einen der Inhalt eines Archivs durch einen Klick auf die Datei dargestellt werden, was bei größeren *tar*-Dateien, wie z.B. dem neuesten Linux-Kernel, natürlich etwas dauert. Zum zweiten können einzelne Dateien aus dem Archiv wie jede andere Datei auch an beliebige Stelle kopiert werden. Sehr praktisch.

## Hilfe!

An welcher Stelle des Konquerors man sich auch befindet: Ein beherzter Druck auf die Taste **F1** startet einen weiteren Konqueror mit dem interaktiven Hilfetext. Wenn man aber nur kurz wissen möchte, was z.B. die Icon in der Werkzeugleiste bedeuten, so langt die Eingabe von **↑+F1**: Der Cursor verwandelt sich in ein Fragezeichen, und nach dem Klicken auf das gewünschte Symbol erscheint ein kleiner, erklärender Text auf dem Schirm.

## K-Onsole – die Terminal-Emulation



Abb. 12.18

Das Symbol für die Konsole im Panel

Wer mit seiner grafischen Oberfläche arbeitet, muss nicht notgedrungen auf eine andere Textkonsole wechseln, wenn er damit arbeiten möchte: Es gibt die Terminal-Emulation des K-Desktops. Hier können alle Befehle wie gewohnt an der Befehlszeile eingegeben werden. Wem also die Befehlszeile vertraut ist und wer nicht lange nach dem grafischen Ersatz für *find* oder *tar* suchen will, der kann jederzeit ein Terminal starten und loslegen. Wem das zu unübersichtlich oder zu bunt ist, der kann allerdings wie gehabt mit **Alt+F1-F6** zu einer seiner sechs Textkonsolen wechseln. Von dort gelangt man zurück zum K-Desktop (wie zu jeder anderen X-Oberfläche auch) mit **Alt+F7**.

Wer nur schnell eine Befehlszeile eintippen möchte, muss dafür nicht gleich die Terminal-Emulation starten. Durch die Eingabe von **Alt+Shift** startet das kleine Fenster für die Befehlszeileeingabe, in die eben der Name des zu startenden Programms eingegeben werden soll. Warum die KDE-Entwickler bei ihrer sonst so schön verspielten Namensgebung nicht auf das Naheliegendste gekommen sind, ist unklar. Das Programm muss natürlich *KonGo* heißen.



## X-Konfiguration und KDE

Abb. 12.19

Der Autor bei der Arbeit an der Konsole

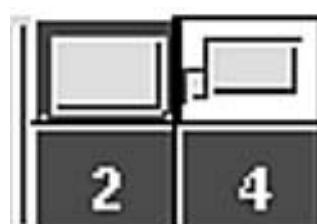


### Die virtuellen Bildschirme

Analog zu den sechs virtuellen Textkonsolen bietet KDE bis zu maximal acht virtuelle grafische Bildschirme – dies ist auch abhängig von dem vorhandenen Arbeitsspeicher. In der Grundeinstellung sind vier Bildschirme vorhanden, die durch die vier Buttons Eins bis Vier bzw. one bis four im Panel dargestellt werden. In älteren Versionen von KDE wurde diese virtuellen Bildschirme dem Anfänger sofort stolz präsentiert, da dort jeder einen anderen Hintergrund besaß ... Aber geplagte Besitzer von grafischen Oberflächen, die nur einen Bildschirm besitzen, werden den Vorteil sofort.

Abb. 12.20

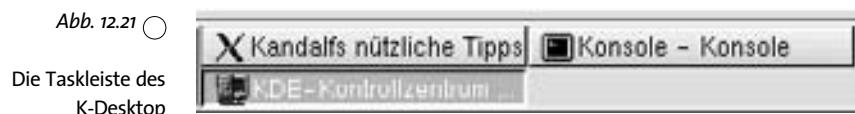
Die Buttons im Panel für die virtuellen Bildschirme





Ein Klick zurück auf den Button 1 gibt sofort das Pokerspiel bzw. das Terminal zurück. Wer unter MS-Windows gleichzeitig z.B. mit einem Desktop-Publisher, einem Bildbearbeiter, einer Datenbank sowie Werkzeugen wie ASCII-ANSII-ISO-Converter und ASCII-Editor arbeiten muss und irgendwann schlicht das gerade benötigte Programm nicht mehr findet, der weiß dieses Hilfsmittel sehr zu schätzen: Kein Klicken mehr durch endlose Fensterlisten, kein Klick mehr auf das nächste Fenster, welches natürlich nicht jenes ist, das man erwartete, sondern einfach eine saubere Verteilung auf die vier (oder mehr) Bildschirme, und die Arbeit geht deutlich schneller vonstatten. „Wenn ich jetzt aber mein Programm von Bildschirm zwei nach drei verschieben möchte“, fragt der mitdenkende Leser. Neben den – von anderen grafischen Oberflächen bekannten – drei Symbolen an der rechten Oberseite eines jeden Fensters, befindet sich unter KDE auf der linken Oberseite eine „Reißzwecke“, auch als „Sticky Button“ bezeichnet. Wenn darauf geklickt wird, wird das entsprechende Fenster von dem aktuellen virtuellen Desktop gelöst; dieses Fenster befindet sich jetzt automatisch auf allen Desktops. Ein Wechsel zu dem gewünschten, ein weiterer Klick auf den „Sticky Button“, und das Fenster ist auf dem aktuellen Desktop wiederum „festgenagelt“. Wenn allerdings ein Programm derart wichtig ist, dass es auf allen Desktops erscheinen soll, so muss die Reißzwecke „eingefahren“ bleiben. Das beste Beispiel dafür ist natürlich „Fernsehen unter Linux“ (Hallo Birger!), aber es soll noch andere Situationen geben, wo es sinnvoll sein kann.

## Die Taskleiste



In vorherigen Versionen war die Taskleiste wirklich eine eigenständige Leiste. Seit 2.0 ist die Standardeinstellung, dass die Tasks im Panel eingebettet sind, was die Bildschirmfläche, die von Anwendungsprogrammen nutzbar ist, natürlich erhöht. Wer das altgewohnte Verhalten aus früheren Versionen wiederherstellen möchte, kann dieses im Kontrollzentrum (s.u.) unter *Erscheinungsbild > Fensterleiste*.

Durch einen Klick auf den entsprechenden Button in der Taskleiste wird das Programm aktiviert bzw. minimiert. Nach seiner Minimierung ist das zugehörige Fenster dann nur noch über die Taskleiste zu erreichen. Wenn ein Programm bzw. ein Dokument über die Taskleiste aktiviert wird und dieses zuletzt auf einem anderen virtuellen Bildschirm aktiv war, wird zugleich auf den zugehörigen Bildschirm gewechselt. Wie Sie später sehen werden, kann auch die Taskleiste nach eigenen Wünschen gestaltet werden.

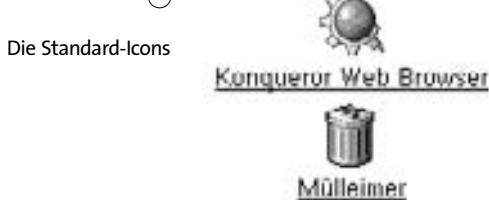
## X-Konfiguration und KDE



### Icons auf dem Desktop

Die nächste der vielfältigen Möglichkeiten, mit Programmen oder Dokumenten zu arbeiten, ist die Ablage eines zugehörigen Icons auf dem Desktop. Wie folgende Abbildung zeigt, werden von der Standardinstallation einige Icons auf dem Desktop abgelegt:

Abb. 12.22



Auch wenn Apple ja seinerzeit prozessiert hat, da sie der Ansicht waren, alle Papierkörbe der Welt gehören Ihnen, so sehen Sie hier unter anderem einen Apfel-freien Papierkorb, neuerdings mit *Mülleimer* beschriftet. Sie werden im Abschnitt „*Konfiguration des Desktops*“ sehen, wie man als Benutzer seinen Desktop mit jeder Menge Icons verzieren kann. Das Wichtige für den Umsteiger ist, dass alle Icons unter KDE nur einen Klick benötigen und der gewohnte Doppelklick die gewünschte Aktion zweimal ausführt. Um es deutlich zu machen: Wenn man ein Icon mit Netscape auf dem Desktop abgelegt hat und darauf einen Doppelklick ausführt, so werden zwei (!) Instanzen von Netscape gestartet, wobei die zweite Instanz – typisch für Netscape – sich damit meldet, dass bereits eine Instanz gestartet ist.

### Das Kontextmenü

Als Microsoft vor nicht allzu langer Zeit entdeckte, dass ihre selbstentwickelte Maus auch eine rechte Taste besitzt, da hatten die Entwickler der verschiedenen X-Oberflächen der mittleren Taste bereits diverse Funktionen zugewiesen. Bleiben wir bei der rechten Taste: Unter K entwickelte Programme aktivieren auf den Klick mit der rechten Maustaste das *Kontextmenü*. Wie das Wort bereits sagt: Es ist abhängig von dem entsprechenden Programm bzw. der Situation innerhalb des Programms. Wichtig ist zu wissen, dass der Klick mit der rechten Maustaste auf eine leere Stelle des Desktops folgendes Kontextmenü aktiviert:

Hier können diverse Einstellungen vorgenommen werden, ohne dass man sich durch Menübäume quälen muss. Das Ausloggen aus der K-Oberfläche kann also schnell mittels des letzten Menüpunkts *Abmelden* bzw. *logout* erfolgen, ohne dass man z.B. das Panel wieder ausrollen, danach eventuell darüber liegende Fenster verkleinern oder verschieben muss, um dann endlich den K-Button zu drücken, um sich danach dann endlich auszuloggen zu können: Es gibt viele Wege, um auf dem K-Desktop ans Ziel zu gelangen. Um aber noch einmal klar darzustellen, dass das





Kontextmenü eben im Kontext mit dem jeweiligen Programm steht, wird das Kontextmenü von *kworldwatch*, der Weltzeituhr für globale Projekte, gezeigt, wie es nun einmal ist.

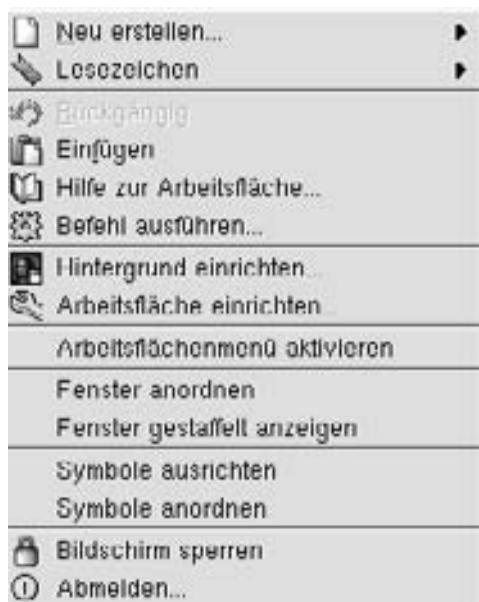
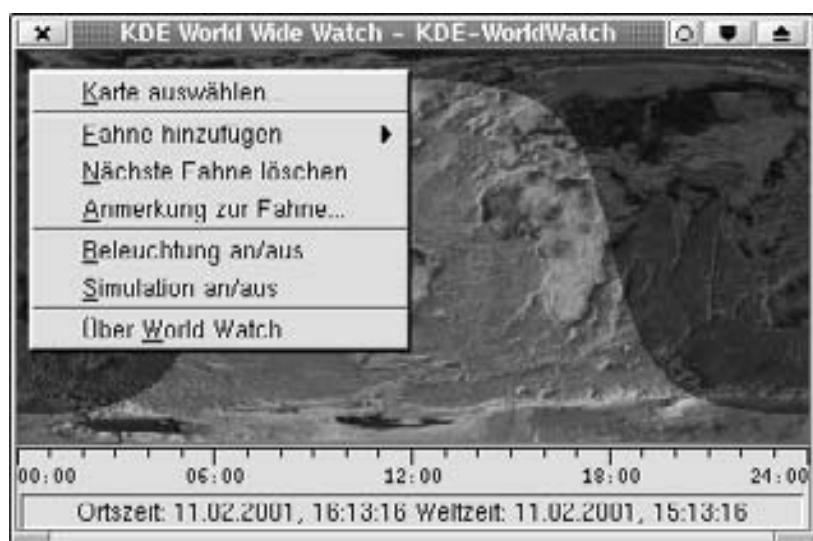


Abb. 12.23

Das Kontextmenü des Desktops

Abb. 12.24

Auch *kworldwatch* besitzt ein Kontextmenü



## X-Konfiguration und KDE



Wie das Popup-Menü zeigt, sind die Menüpunkte völlig abhängig von dem jeweiligen Programm und nur ihm zugehörig.

### Der Disk-Navigator

Falls man eine bestimmte Datei sucht, diese aber (noch) nicht auf dem Desktop abgelegt hat, so hilft der Disk-Navigator. Dieser bietet die Möglichkeit, mit einigen Mausbewegungen auch in die tiefsten Dateibäume einzudringen, um längst verloren geglaubte Schätze zu heben.

Abb. 12.25



Der Disk-Navigator wurde in der KDE-Version 2.0 aus unerfindlichen Gründen in „Schnell-Anzeiger“ umbenannt. Er wird im unteren Bereich der Programmleiste aktiviert. Mehrere Ausgangsverzeichnisse sind bereits definiert, wie *KDE*, das Wurzelverzeichnis *root* und das Heimatverzeichnis des Benutzers. Von hier aus kann man sich dann weiter durch die Unterverzeichnisse hangeln, bis man an der gewünschten Stelle ist. Dann kann man neuerdings über die obersten beiden Menüeinträge entweder ein Terminal oder den Datei-Manager öffnen – in beiden Fällen befindet man sich im ausgewählten Verzeichnis des Dateibaums.

Die zuletzt geöffneten Dateien, die vorher in den Disk-Navigator integriert waren, haben im unteren Teil des K-Menüs inzwischen einen eigenen Menüpunkt erhalten.





## Kandalf

Dass die Programmierer des KDE von frühkindlichen Schädigungen nicht frei sind, hat niemand behauptet. Zeugnis darüber legt *Kandalf* ab, in seiner Namensnennung einer der frühen Helden der Fantasy-Literatur nicht unähnlich. Nun, der „Herr der Ringe“ war vor 25 Jahren in gewissen Kreisen „Pflichtliteratur“, und weil die Eltern ihre Kinder damit langweilten, muss dieses Trauma noch heute von den Programmierern bewältigt werden.

Kandalf erscheint nach dem ersten Start von KDE sofort auf dem Bildschirm, und wenn man das Kontrollkästchen *Beim Start ausführen* nicht deaktiviert, auch bei jedem weiteren Start. Ansonsten gibt Kandalf hilfreiche Tipps zu der mittleren Maustaste, zu Tastatatkürzeln und zum Teekonsum seiner Programmierer, durchaus hilfreiche Informationen über all die kleinen Nettigkeiten, die den Alltag am Rechner erleichtern. Kandalf versteckt sich im K-Menü unter *Kleinigkeiten* neben anderen mehr oder minder sinnvollen Progrämmchen, wie dem Mausgeschwindigkeitsmesser oder dem Teekocher – eine spezialisierte Eieruhr.

Abb. 12.26

Kandalf gibt nützliche  
Tipps



## Konfiguration des K-Desktops

Die Benutzeroberfläche des K-Desktops lässt sich in fast allen Bestandteilen konfigurieren, und sie lässt sich wirklich derart verstellen, dass selbst der K-Profi seine Schwierigkeiten haben kann, sich zurechtzufinden. Ob dies sinnvoll ist, mag wiederum eine andere Frage sein; auf jeden Fall lässt sich K seinen Wünschen und auch krudesten Bedürfnissen entsprechend in weitesten Möglichkeiten konfigurieren. Wer nur bestimmte Bestandteile der Oberfläche verändern will, der kann eine detaillierte Auswahl im Untermenü *Einstellungen* des K-Menüs finden. Wer alle Möglichkeiten auf einmal gerne präsent hätte, der muss das KDE Kontrollzentrum aktivieren. Das ist wiederum auf verschiedenen Wegen möglich: Entweder wird im *Start*-Menü der Menüpunkt *KDE Kontrollzentrum* aktiviert, oder es wird auf den entsprechenden Button im Panel geklickt.



## X-Konfiguration und KDE



Abb. 12.27

Das Icon im Panel für das Kontrollzentrum



In beiden Fällen startet das Kontrollzentrum mit folgendem Erscheinungsbild:

Abb. 12.28

Das Kontrollzentrum



Das Kontrollzentrum ist in seiner neuesten Version deutlich aufgeräumter und strukturierter als bisher. Hier können nun wirklich (fast) alle Elemente der Benutzeroberfläche konfiguriert werden. Nachfolgend einige kurze Informationen zu den einzelnen Punkten.

### Angeschlossene Geräte

Hier können Tastatur und Maus konfiguriert werden, u.a. Wiederholrate der Tastatur oder das Vertauschen der Maustasten für Linkshänder.





## Dateien

Man kann zum einen das Verhalten und das Erscheinungsbild des Datei-Managers anpassen, zum anderen können unter *Dateizuordnungen* bestimmten Dateitypen jeweils Programme zugewiesen werden, die sich beim Klick auf die Datei automatisch aktivieren. Dies ist bereits weiter oben beim Konqueror beschrieben worden.

## Energiekontrolle

Für die immer beliebter werdenden Laptops können hier zum einen Informationen aktiviert werden, die über den Füllstand der Akkus informieren und beim Unterschreiten bestimmter Werte Warnhinweise auslösen. Außerdem können die Energiesparfunktionen eines Laptops aktiviert werden.

## Erscheinungsbild

Unter diesem Menüpunkt findet man endlich alle – früher etwas verstreut liegenden – Untermenüs für das Aussehen des Desktops versammelt.

## Arbeitsfläche

Der Benutzer kann nun kräftig in Farben und Formen zulangen und seinen Desktop individuell – und auch bis zur Unleserlichkeit – gestalten.

## Allgemein

Hier können die Aktionen definiert werden, die die drei Maustasten auf der Arbeitsfläche auslösen. Außerdem können Font und Farbe für die Texte auf der Arbeitsfläche eingestellt werden.

## Hintergrund

Für jeden virtuellen Bildschirm kann eine Hintergrundfarbe ausgewählt werden. Durch Klicken auf die voreingestellte Farbe wird der Farbdialog aktiviert, der einen Farbverlauf aller möglichen Farben anbietet. Wer keine Farbe, sondern ein Bild auf seinem Desktop wünscht, kann unter *Hintergrundbild* ein solches selektieren und unter *Modus* die Darstellung dieser Bitmap wählen, wie z.B. zentriert oder gekächelt.

## Ränder

Dieses Leistungsmerkmal lässt den Benutzer automatisch den virtuellen Bildschirm wechseln, sobald der Mauszeiger in die Nähe des Bildschirmrandes kommt. Die einzelnen Bildschirme werden hier also als Teil eines größeren virtuellen Bildschirms gedacht. Dieses Verhalten kann aktiviert werden, und die Verzögerung als auch die Breite der Randzone kann eingestellt werden.



## X-Konfiguration und KDE



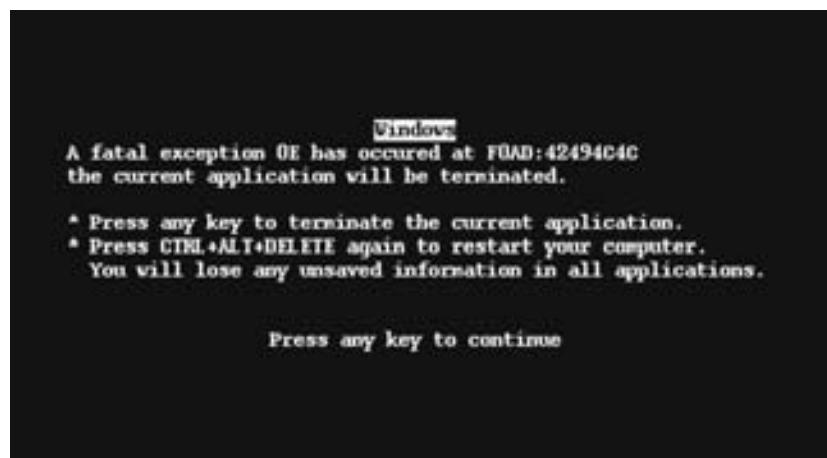
### Virtuelle Arbeitsfläche

Hier können maximal 16 virtuelle Arbeitsflächen eingerichtet werden, wobei die als Standardeinstellung vorhandenen vier in aller Regel völlig ausreichen.

### Bildschirmschoner

Hier kann ein Bildschirmschoner ausgewählt, mit einem Passwort versehen, die Zeit bis zur Aktivierung eingestellt werden und außerdem die Einstellung getestet werden. Wir empfehlen „Im Zeichen des Totenkopfs“, der Fehlermeldungen der Betriebssysteme von Amiga, Atari sowie Windows NT zeigt: Gerne genommen zur Verblüffung der Kollegen.

Abb. 12.29



### Design

Unter den beiden Unterpunkten *Stil* und *Symbole* kann das Aussehen an grundlegende Standards angepasst werden. Wer also sein KDE erscheinen lassen möchte, als ob es NeXT, SGI, Motif, CDE oder WingX wäre, kann dies mit wenigen Mausklicks einstellen. Außerdem können Mac-Fans hier die Systemmenüleiste am oberen Bildschirmrand aktivieren.

### Farben

Hier können vordefinierte Farbschemata für die einzelnen Fensterelemente ausgewählt werden. Einige sind anderen Betriebssystem entlehnt, wie z.B. BeOS, Solaris oder Windows. Was hier fehlt, ist die Einstellung „Hot Dog“, wie Sie sie aus den alten Tagen von Windows 3.1 kennen. Da taten die Augen richtig weh ...





## Fenstereigenschaften

### Aktionen

Es können wiederum zahlreiche Details kontrolliert werden. Wichtig für leistungsschwache CPUs ist z.B. die Deaktivierung der Kontrollkästchen *Fensterinhalt beim Verschieben anzeigen* und *Fensterinhalt bei Größenänderung anzeigen*. Auch kann umgeschaltet werden zwischen dem typischen X-Window-Manager-Verhalten, dass das Fenster unter dem Mauszeiger aktiv ist, während Umsteiger von anderen Betriebssystem wohl ihr vertrautes Verhalten bevorzugen, dass sich ein Fenster erst auf einen Klick aktiviert.

### Mausverhalten

Hier werden nicht die Verhaltensparameter der Teilnehmer vom Girlscamp eingestellt, sondern die Aktionen, die die drei Maustasten – auch in Kombination mit der Taste **Alt** – erzeugen. Die Voreinstellungen sind hinreichend praktisch und: Vorsicht, mit unüberlegten Einstellungen kann man an dieser Stelle seinen Desktop nahezu unbrauchbar machen!

### Fensterleiste

So heißt jetzt die Leiste, die Ihnen bisher neudeutsch als „Taskbar“ bekannt war. Die Position kann in den vier Himmelsrichtungen eingestellt werden.

### Kontrollleiste

So heißt jetzt das Panel, das in seiner Standardeinstellung am unteren Bildschirmrand seinen Platz hat. Die Position auf dem Bildschirm kann gewählt werden. Außerdem kann die Größe des Panels bestimmt werden, wobei die Beschriftung etwas irreführend ist: *sehr klein* ist mindestens für alle Bildschirmdarstellungen bis 800 mal 600 Bildpunkten genügend. Bei größeren Auflösungen empfehlen wir *normal*. Die Einstellung *groß* ist wahrscheinlich für Sehbehinderte gedacht. Unter *Einstellungen* kann hier ein relativ störendes Leistungsmerkmal unschädlich gemacht werden: Das Panel kann ausgeblendet werden; sobald der Mauszeiger jeweils in die Nähe der (versteckten) Position kommt, wird es aktiviert. Hierfür können auch noch die Verzögerung und Geschwindigkeit reguliert werden. Ansonsten birgt dieses Registerblatt noch Einstellungen für die Menütipps.

### Schrift

Hier kann für Datei-Manager, Werkzeugeiste, Menü und Fenstertitel jeweils die Schrifttype für die einzelnen Fensterelemente eingestellt werden.

### Symbole

Den Icons auf der Arbeitsfläche und in der Werkzeugeiste kann eine bestimmte Größe zugewiesen werden. Außerdem kann das Aussehen von aktiven, deaktivierten und Icons im Normalzustand eingestellt werden, so dass sich das Aussehen bei Mausbewegungen über dem Icon ändert.



## X-Konfiguration und KDE

---



### Systemnachrichten

Für jede Systemnachricht (!) von KDE bzw. des KDE-Fenstermanagers kann eine der folgenden vier Möglichkeiten ausgewählt werden:

- *In Log-Datei schreiben*
- *Klang abspielen*
- *Benachrichtigungsfeld anzeigen*
- *Standard-Fehlerausgabe*

Auch hier sollten die Einstellungen mit Bedacht gewählt werden: Wer zu jedem Schließen eines Fensters einen Klang aktiviert, wird über kurz oder lang von den lieben Kollegen erschlagen!

### Tastenzuordnung

Den verschiedensten Aktionen können Tastaturkürzel zugeordnet werden. Neben diversen Standardeinstellungen können sich Tastaturkünstler hier für die verschiedensten Fensteraktionen Hotkeys selbst definieren, was deutlich zur Schonung der Maus beiträgt.

### Hilfe

Unter dem Untermenüpunkt *Index* findet sich eine Online-Suchmaschine zur Volltextsuche, für die man allerdings einen Internetanschluss benötigt. In der SuSE-Distribution findet sich außerdem noch eine Liste mit allen möglichen Informationsquellen, die in einer Datenbank verwaltet werden. In beiden Fällen kann man die Informationsquellen verwalten, die mittels der Hilfefunktion des Konquerors abgefragt werden können.

### Information

An dieser Stelle findet man jede Menge Informationen über Hardware und zugehörige Treiber seines Rechners. Für den Benutzer älterer Versionen zeigt sich der schnelle Fortschritt der Entwicklung von KDE – es hat sich in letzter Zeit viel getan. Die Menüpunkte in Kürze:

*DMA-Kanäle.* Es werden die DMA-Kanäle angezeigt sowie die Geräte, von denen sie benutzt werden.

*Ein-/Ausgabe-Ports.* Hier findet man eine Liste der I/O-Adressen und den zugehörigen Geräten.





*Geräte.* Es werden Informationen aus dem virtuellen Dateisystem `/proc` entnommen und die im System vorhandenen Geräte mit ihren zugehörigen Gerätenummern angezeigt.

*Interrupts.* Alle belegten Interrupts werden mitsamt ihren Geräten angezeigt. Praktisch, wenn Soundkarte, Netzwerkkarte und ISDN-Controller sich um die raren Interrupts streiten.

*Klänge.* Die Informationen über alle vorhandenen Hard- und Softwaregeräte wie Soundkarte, Mixer, Midi und zugehörige Treiber werden angezeigt.

*PCI.* Hier werden alle PCI-Geräte des Rechners, wie z.B. Grafikkarte oder IDE-Controller, mit weiteren Details angezeigt.

*PCMCIA.* Falls der Rechner einen PCMCIA-Controller besitzt, werden Details über ausgegeben.

*Partitionen.* Die Partitionen der Festplatten, die Blockgröße, sowie Haupt- und Nebengerätenummern werden aufgeführt. Außerdem werden Informationen über die virtuellen Dateisysteme `/proc` und `/devpts` angezeigt.

*Prozessor.* Die Details eben des Prozessors werden angezeigt, wie der Hersteller lautet und ob diverse Hardwaredefekte vorhanden sind, wie z.B. der berühmte Fließkomma-Divisionsfehler sowie die Geschwindigkeit der CPU in den berühmten, von Linus Torvalds kreierten „Bogo-Mips“.

*SCSI.* Gibt detaillierte Informationen über SCSI-Geräte aus, falls welche angeschlossen sind.

*Samba-Status.* Samba ist das Protokoll des Windows-Netzwerks, was Linux übrigens hervorragend emuliert. Falls man Mitglied eines derartigen Netzes ist, erhält man detaillierte Informationen.

*Speicher.* Hiermit ist der Arbeitsspeicher sowohl in den RAM-Chips als auch auf der Swap-Partition gemeint. Die Anzeige wird dynamisch aktualisiert. Dem Anfänger empfohlen: Ein größeres Programm – z.B. StarOffice – starten und staunend die Anzeige beobachten.

*X-Server.* Auflistung der Versions- und Revisionsnummer des X-Servers, der Auflösung und diverser anderer Details der verfügbaren Bildschirme sowie der unterstützten Pixmap-Formate.

*Klänge.* Hier sind unter den Menüpunkten *Midi*, *Mixer*, *Signalton* und *Soundserver* diverse Einstellungen abzulesen, die an dieser Stelle auch interaktiv geregelt werden können.



## X-Konfiguration und KDE

---



### **Netzwerk:**

*Talk-Einrichtung.* Die Einrichtung des Talk-Programms, das zusätzliche Aktivieren eines Anrufbeantworters und der Weiterleitung ist möglich.

*Windows-Ressourcen.* Falls sich der Rechner in einem SMB-Netz befindet und die Windows-Ressourcen von Linux aus genutzt werden sollen, so müssen Angaben über das Netzwerk und den SMB-Server eingegeben werden. Danach können dann freigegebene Windows-Ressourcen, wie z.B. Festplatten oder Drucker, auch vom Linux-Client genutzt werden.

### **Persönliche Einstellungen:**

*E-Mail.* Falls man mit *KMail* seine elektrische Post erledigen möchte, so müssen hier die Standardeingaben erledigt werden, wie Absenderangaben, die Adressen, der Servertyp für Mail-Ein- und Ausgang und natürlich der Benutzername nebst Passwort.

### **Land und Sprache:**

Schon früh erkannte das KDE-Team, dass eine Internationalisierung des Desktops einen wichtigen Moment in der Entwicklung darstellt, vielleicht auch, weil die Entwicklergemeinde quer über diesen Erdball verteilt ist. So wurde bereits früh beim Design darauf geachtet, dass Dialoge und Beschriftungen an zentraler Stelle zusammengefasst sind und so eine Übersetzung in andere Sprachen ermöglichen. Dies wurde weltweit begrüßt, und so kann der K-Desktop derzeit in 34 Sprachen aktiviert werden.

Nun haben auch kleinste Sprachräume das erste Mal die Möglichkeit, einen Desktop in ihrer Heimatsprache zu bedienen, ja mehr noch: Ganze Gruppen von Anwenderprogrammen können in gewohntem Idiom bedient werden. Für die Übersetzungen in andere Sprachen haben die Macher des K-Desktops vorgesorgt: Während dies früher „zu Fuß“ mit einem ASCII-Editor bewerkstelligt werden musste, gibt es hierfür heutzutage ein Werkzeug namens *ktranslator*. Wer also die Sprache seine Wahl in obiger Liste nicht gefunden hat, kann sich aufmachen, das K-Projekt zu übersetzen.

In der Version 2.0 wurde das System der Ländereinstellungen gründlich überarbeitet. Da die Liste der unterstützten Sprachen immer länger und unübersichtlicher wurde, hat man sie in Großräume zerlegt, und dort die jeweiligen Länder zugeordnet. Außerdem können Land und Sprache getrennt voneinander gewählt werden. Für Sprachen, die einen anderen Zeichensatz als das lateinische Alphabet benutzen, kann außerdem der gewünschte Zeichensatz, z.B. kyrillisch oder griechisch, eingestellt werden.

Außerdem können die nationalen Einstellungen für das Dezimalzeichen, für die Währung sowie für die interne Anordnung von Datum und Zeit eingestellt bzw. angepasst werden.





*Passwörter.* Es kann angegeben werden, ob während der Eingabe von Passwörtern Sterne angezeigt werden sollen oder ob es auf dem Bildschirm kein Echo geben soll. Außerdem kann angegeben werden, ob Passwörter im System gespeichert werden sollen.

*Tastaturlayout.* Jedes Land hat sein eigenes Tastaturlayout, um nationale Sonderzeichen einfach erreichbar zu machen. Hier wird nun die alphabetisch sortierte Liste angeboten – von Armenien bis Vietnam.

*Verschlüsselung.* Für viele KDE-Programme ist die Möglichkeit der Verschlüsselung vorgesehen. Die verschiedenen Schlüsselarten können hier verwaltet werden.

*Zugangserleichterungen.* Hier können einige Standard-Systemeinstellungen umdefiniert werden. Statt des Signaltoms als Warnhinweis kann man den Bildschirm zum kurzen Aufleuchten bewegen. Außerdem kann man die Möglichkeit aktivieren, die Maus mittels des Ziffernblocks zu bewegen.

## System

*Anmeldungsmanager.* Früher hieß er Login-Manager und betrifft den Fall, dass ein grafisches Login bei der Installation aktiviert wurde. Dann können Sprache, Schriftart und Hintergrundfarbe gewählt werden. Auch kann das Benutzerlogo gewählt oder gar ein selbst erstelltes Bild des Benutzers eingesetzt werden.

*Datum und Zeit.* Sehr hilfreich für den Anfänger ist die interaktive Eingabe von Systemdatum und -zeit, denn die Syntax des Befehls *date* ist zugegebenermaßen etwas lästig. Auch die Zeitzone kann an dieser Stelle eingestellt werden.

## Webbrowser

*Cookies.* Hier können die so genannten „Cookies“ für den Konqueror an- bzw. abgeschaltet werden. Die Einstellung kann auch am Konqueror selbst vorgenommen werden.

*Erweitertes Webbrowsen.* In dieser Tabelle werden Abkürzungen für häufig benutzte Internetadressen gespeichert werden, die derart funktionieren, dass man die Abkürzung in der URL-Leiste des Konquerors eingibt und er die zugehörige Webadresse ansteuert. Wenn man z.B. schlicht *av* eingibt, erscheint die Suchmaschine *www.altavista.com*. Diese Liste kann vom Benutzer bearbeitet und erweitert werden.

*Konqueror-Browser.* Die Details des Erscheinungsbildes, wie z.B. die Schriftart des Konquerors, können beeinflusst werden. Außerdem können Java und JavaScript aktiviert bzw. deaktiviert werden. Diese Einstellungen können auch am Konqueror direkt vorgenommen werden.

*Netscape-Plugins.* Falls Netscape installiert wurde, kann hier die Liste der installierten Plugins verwaltet werden.



## X-Konfiguration und KDE

---



*Proxy-Server.* Falls das Surfen über einen hauseigenen Proxy-Server stattfindet, muss dessen Kennung hier angegeben werden.

*User-Agent.* Viele Webserver fragen die Kennung des Browsers ab, um danach Browser-spezifische Seiten zu aktivieren. Mit der Kennung *Konqueror* können viele Internetangebote noch nichts anfangen. Um derartig übereifrige Server zu überlisten, muss man sich bei bestimmten Servern als Netscape ausgeben, wenn man mit dem Konqueror surft. Genau das wird hier erledigt.

### YaST2

In der SuSE-Distribution verewigt sich im Kontrollzentrum außerdem noch YaST2, dessen Module auch von dieser Stelle aus geladen werden können. Die Details zu YaST2 entnehmen Sie dem *Kapitel 1 Erstinstallation*.

## Das Einrichten von Icons auf dem Desktop

Manche Programme zum täglichen Gebrauch möchte man nicht in tief verschachtelten Menüleisten versteckt halten, sondern mit einem Mausklick aktivieren können. Dazu bieten sich Icons auf dem Desktop an. Einige werden automatisch beim ersten Start eingerichtet, wie z.B. der Müllheimer und der Konqueror. Um ein weiteres Icon für ein beliebiges Programm (oder auch Dokument) auf dem Desktop abzulegen, geht man wie folgt vor: Man klickt mit der rechten Maustaste auf eine freie Stelle des Desktops – das Popup-Menü öffnet sich. Man wählt den ersten Menüeintrag *Neu erstellen*, in dem zugehörigen Untermenü *Anwendung*. Eine Dialogbox öffnet sich, die den Dateinamen des zugehörigen Links erwartet. Die voreingestellte Beschriftung *Anwendung* ändert man in eine entsprechend sinnreiche, wie z.B. *Netscape*. Danach aktiviert man das Registerblatt *Ausführen*. Hier muss in das erste Editierfeld der Name und – falls notwendig – der Pfad des Programms eingegeben werden. Falls man diesen nicht weiß oder zu faul zum Tippen ist, so kann man sich mittels *Durchsuchen* durch den Dateibaum hangeln, bis man das gewünschte Binary gefunden hat.

Danach kann man im Registerblatt *Allgemein* durch Klicken auf das Zahnrad ein Icon für den Desktop auswählen. Etwas Geduld ist hier nötig, da die Anzahl der mitgelieferten Icons inzwischen auf 500 zustrebt, und es etwas dauert, bis diese geladen sind. Ein Klick auf das gewünschte Icon sowie auf die Schaltfläche *OK*, danach ein Klick auf das *OK* des gesamten Dialogs, und das Icon liegt auf dem Desktop – ein (nur ein) Klick aktiviert das zugehörige Programm. Hier kann hemmungslos gespielt werden, denn ein Klick mit wiederum der rechten Maustaste bringt ein Popup-Menü hervor, und der Menüpunkt *Löschen* bringt das Icon unwiderruflich in den Daten-Orkus. Wer übrigens trotz der Vielfalt an Icons kein passendes findet, der kann sich relativ schnell selber eins erstellen: Das gewünschte Bild in den Bildbearbeiter der Wahl – z.B. *gimp* – laden, auf die Größe von 48 mal 48 Pixeln reduzieren und als Dateityp XPM speichern. Die Datei muss dann in das Verzeichnis */opt/kde/share/icons* kopiert werden, und schon steht das persönliche Icon zu Verfügung.



## Das Einrichten von Icons auf dem Desktop



### Nie mehr die Befehlszeile?

Der heutige Zeitgeist will es, dass der Benutzer (fast) all die Einstellungen, die man unter Linux vornehmen kann, inzwischen auch mit Mausgeklicke erledigen kann. Das KDE-Projekt trägt dem Rechnung, indem entweder für bestehende befehlszeilenorientierte Programme grafische Schnittstellen programmiert werden, oder indem die Funktionalität komplett neu erstellt wird, einschließlich grafischer Oberfläche. Hier sei eine Auswahl vorgestellt:

- *kam*, das grafische Front-End für *vbox*, das ISDN Voice Mail Packet, mit dem man u.a. seinen digitalen Anrufbeantworter betreiben kann.
- *karchive*, das grafische Gegenstück zu *tar*; zum Ein- und Auspacken von Archiven.
- *kcdwrite*, die grafische Oberfläche zum CD-Brennprogramm *cdwrite*.
- *kcmbind*, das Kontrollmodul für die Konfiguration von *bind*.
- *kcmclock*, Einstellen von Datum und Zeit.
- *kcmjoy*, Konfiguration des Joysticks.
- *kcmprint*, Einstellen der Druckerkonfiguration.
- *krontab*, ein Editor für *crontab*, der Datei, die häufig vorkommende Aufgaben, die automatisch ablaufen sollen, steuert.
- *kdesu*, das grafische Front-End zum Befehl *su*; zum Einloggen als Superuser.
- *kdf*, das Gegenstück zu *df*; Anzeige der Festplattenbelastung.
- *keirc*, jetzt wird grafisch gechattet.
- *kfirewall*, grafische Konfiguration einer Firewall.
- *kfstab*, ein Editor für die Datei *etc/fstab*, der die Mount-Points und das Dateisystem der Festplatten enthält.
- *kheise*, Suche im Register des Heise-Verlags, der u.a. die ct und die iX herausgibt.
- *kimon*, der grafische ISDN-Monitor.
- *klilo*, das Konfigurationsprogramm für LILO, den Linux-Loader.
- *kmodem* zeigt den Modemstatus an.
- *kmysql*, grafischer Client für die Datenbank *MySQL*.



## X-Konfiguration und KDE

---



- *kmysqlad*, Administration der Datenbank *MySQL*.
- *knc*, der 472te Clone des Norton Commanders.
- *knewmail*, ist neue Mail für mich da?
- *k pov*, ein grafisches Front-End für den Ray-Tracer *Povray*.
- *kpsqlman*, Zugriff auf Postgres-Datenbanken.
- *kpstree*, alle derzeitigen Prozesse als Baumdarstellung.
- *ksamba*, für die Konfiguration von Samba.
- *ksms*, SMS vom Rechner aus versenden.
- *ktelnet*, ein grafisches Telnet.

### KDE vs. Gnome

Parallel zur Entwicklung von KDE fand die Entwicklung von Gnome statt, ebenfalls eine grafische Oberfläche, die auch viele Leistungsmerkmale von KDE enthält. Die Gründe an dieser Stelle KDE und nicht Gnome zu besprechen, sind zahlreich und vielschichtig, einige seien hier genannt.

Der erste Grund ist trivial: KDE ist in Europa gestartet, während Gnome ein amerikanisches Kind ist. Dies ist natürlich in den Zeiten des Internets und der abnehmenden Einflüsse der Nationalstaaten kein Argument, hat aber dann doch aufgrund der regionalen Verbindungen ganz triviale Einflüsse auf die entsprechenden Distributionen: SuSE förderte schon früh das KDE-Projekt und hat inzwischen KDE zum Standard-Desktop befördert, während Red Hat ähnliche Aktivitäten bezüglich Gnome entwickelte. Der zweite Grund sind die technischen Details: Die grundlegende Technologie, auf die hier aus Platzgründen leider nicht eingegangen werden kann, unterscheiden sich deutlich, wobei die Grundlagen von KDE durchdachter sowie zukunftssicherer sind. Und zum Dritten besitzt KDE einen Entwicklungsvorsprung von ca. sechs bis neun Monaten, was natürlich dem Benutzer zugute kommt.

Der teilweise sehr unsachlich im WWW geführte Krieg zwischen den Anhängern (nicht: den Entwicklern) der beiden Desktops bringt allerdings niemanden weiter und ist verschwendete Energie. Und in der Tat bewegen sich die beiden Entwicklergruppen aufeinander zu, mit dem langfristigen Ziel, beide Desktops kompatibel zueinander zu machen. Wichtiger ist es da doch, dass Linux sich immer weitere Anteile des Marktes auch im Desktop-Bereich erobert. Und dabei ist Microsoft immer gerne behilflich: Das mexikanische Erziehungsministerium ist mithilfe der Unesco dabei, an den dortigen Schulen im Lauf der nächsten Jahre 140.000 Rechner zu installieren. Eine Bitte an Microsoft um Unterstützung mittels verbilligter Software wurde dort abgelehnt, und so geht man den billigeren und stabileren



## Das Einrichten von Icons auf dem Desktop



Weg: Die Schulrechner werden alle samt und sonders mit Linux als Betriebssystem und mit Gnome als grafischer Oberfläche ausgestattet. Genaueres findet man unter <http://luthien.nuclecu.unam.mx/~arturo/scholar/>.

### KDE – Heute und morgen

Nachdem KDE 2.0 für den Frühling 2000 angekündigt war, wurde es dann doch Herbst. Zum Trost sei gesagt, dass der Linux-Kernel 2.4 deutlich größere Verzögerung zeigte. Der lange Weg der Entwickler ging über so wohlklingende Versionsnamen wie *Kaos* oder *Krash* – damit auch jeder Normalsterbliche merkte, dass es eine mehr oder minder stabile Entwicklerversion und nicht für die Alltagsarbeit gedacht war.

In den nun vorliegenden Versionen 2.1.x ist – neben vielen „Polierarbeiten“ im Aussehen und in der Bedienung – auch in vielen Bereichen eine grundlegende Vereinfachung und Überarbeitung der Strukturen zu erkennen. Dies schlägt sich z.B. sehr positiv im Kontrollzentrum nieder. In vorangegangenen Versionen hatte man oft das Gefühl, dass die Einordnung der Menüpunkte doch arg aus Sicht der Programmierer geschah und nicht aus der Sicht des Anwenders.

Nebst unzähligen praktischen Verbesserungen für die Alltagsarbeit, ist mit dem Konqueror ein großer Wurf gelungen. Der Überlegung folgend, dass die Dateien auf dem persönlichen Rechner und im lokalen Netzwerk nur noch einen kleinen Teil der Welt darstellen, wurden hier eine zukunftsweisende Benutzerschnittstelle geschaffen. Der große andere Teil der Welt ist das Internet und muss von vielen Menschen im Alltag genauso oft benutzt werden wie ein Datei-Manager. Was lag da näher, die beiden Welten zu verbinden und Browser, FTP-Client und Datei-Manager in einem Programm zu vereinen? Vermutlich werden andere Hersteller diese Idee recht bald übernehmen.

Ein Kritikpunkt bleibt KDE 2.0 nicht erspart: Es ist fett geworden und etwas behäbig. Große Teile dieses Kapitels wurden auf einem Notebook mit „nur“ 32 MByte und einem „alten“ 233-MHz-Prozessor erstellt. Zum einen waren heftige Swap-Bewegungen zu verzeichnen – kein Wunder: Schon die Aktivierung von wenigen Programmen, wie KSnapshot, zwei Konsolen und dem Kontrollzentrum, verursachten einen Speicherbedarf von ca. 52 MByte – gemessen vom Kontrollzentrum selbst. Zum anderen war ein sehr langsames Starten und Schließen verschiedener Applikationen zu bemerken. Ich persönlich habe jedenfalls keine Lust, mir für die nächste KDE-Version einen neuen Laptop kaufen zu müssen. Solch ein Unfug sollte eigentlich Windows vorbehalten sein.

Für die Zukunft hingegen wünschen wir uns für den K-Desktop einen ausgewachsenen Desktop-Publisher mit mehr Möglichkeiten, als der FrameMaker bietet, intelligenter und kompatibler als Quark und nicht so eigenwillig wie der Ventura-Publisher. Unsere Unterstützung ist zugesichert, und man würde schließlich in die Welt der Zeitungen und Zeitschriften, also der Meinungsmacher einmarschieren.







# Linux im Netzwerk



|                                                 |            |
|-------------------------------------------------|------------|
| Offene Thin-Ethernet-Verkabelung                | <b>497</b> |
| Thin-Ethernet mit Dosen                         | <b>503</b> |
| Twisted-Pair-Verkabelung                        | <b>504</b> |
| IP-Adressen                                     | <b>506</b> |
| Linux im heterogenen Netzwerk                   | <b>514</b> |
| Besondere Netzwerkimplementierungen             | <b>516</b> |
| Die Konfiguration des SAMBA-Servers             | <b>531</b> |
| Einbindung eines SAMBA-Clients – a case study   | <b>546</b> |
| Der MARS-Server                                 | <b>551</b> |
| Up, up and away – die Anbindung an das Internet | <b>565</b> |



**13**

## Linux im Netzwerk

Bisher wurde das Linux-System so behandelt, als sei es ein Einzelplatzsystem, womit allerdings ein wesentliches Leistungsmerkmal ausgeklammert wurde: Als UNIX-Klon bietet Linux sämtliche erforderlichen Werkzeuge, um innerhalb eines Netzwerks agieren zu können.

Doch was ist ein Netzwerk überhaupt? Als Netzwerk ist „eine Gruppe von Computern und angeschlossenen Peripheriegeräten, die über einen Kommunikationskanal miteinander verbunden sind und so mehreren Benutzern den Zugriff auf Dateien und andere Ressourcen ermöglichen“ definiert.

Als Kommunikationskanal dienen hier unterschiedliche Hardwarekomponenten, deren Unterschiede unter einer einheitlichen Schnittstelle, den „abstrakten Netzwerkgeräten“, zusammengefasst werden, die angesprochen werden können, wenn die Initialisierungsfunktion die entsprechende Hardware gefunden hat.

Ähnlich wie beim Dateisystem kristallisiert sich bei der Analyse der Netzwerkspezifika eine Struktur heraus, deren Schichten (*Layers*) unterschiedlichen, ausgehend von der Hardware aufsteigenden Abstraktionsniveaus entsprechen.

Es handelt sich bei diesen Schichten um Protokolle, die meist vereinfacht unter dem Schlagwort *TCP/IP* zusammengefasst werden, die jedoch bei genauerer Betrachtung in folgende Schichten aufgeschlüsselt werden können.

Die in einer Netzwerkimplementierung verwendeten Schichten mit ihren Funktionen

| Schicht                           | Aufgabe                                                                                                                                                                                                                                                   |
|-----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>BSD-Socketschicht</i>          | verwaltet eine allgemeine Datenstruktur für Sockets; sie hat somit ähnliche Aufgaben wie das virtuelle Dateisystem                                                                                                                                        |
| <i>INET-Socketschicht</i>         | verwaltet die Kommunikationsendpunkte für die auf IP basierenden Protokolle TCP und UDP                                                                                                                                                                   |
| <i>UDP, TCP, IP</i>               | Typ des Sockets. <i>UDP</i> realisiert das <b>User Datagram Protocol</b> , <i>TCP</i> das <b>Transmission Control Protocol</b> für zuverlässige Kommunikationsverbindungen, und in der <i>IP</i> -Schicht ist das <b>Internet Protokoll</b> programmiert. |
| <i>PLIP, SLIP, Ethernet-Karte</i> | abstrakte Netzwerkgeräte, an die das IP die Pakete für den physikalischen Transport der Informationen über die serielle oder die parallele Schnittstelle oder eine Ethernet-Karte übergibt                                                                |

Tabelle 13.1

Grafisch können diese Schichten wie in Abbildung 13.1 dargestellt werden.



## Offene Thin-Ethernet-Verkabelung

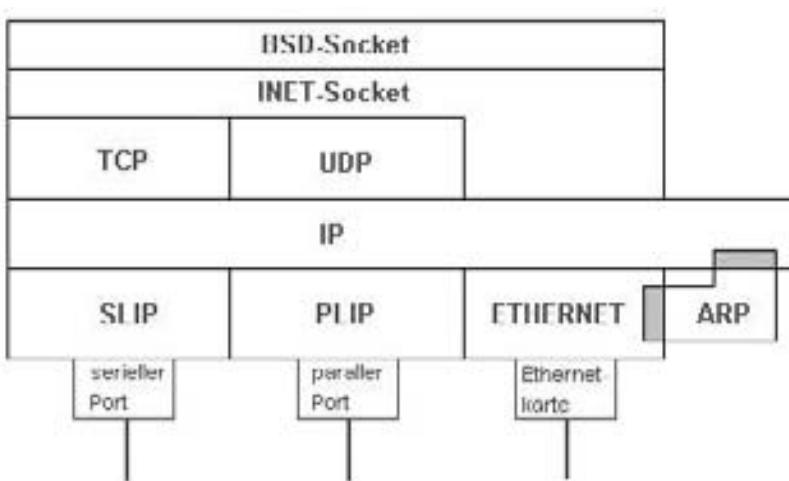


Abb. 13.1

Die Layer-Struktur des Netzwerks

Der in *Tabelle 13.1* beschriebene Vorgang findet beim Empfänger der Daten natürlich in umgekehrter Reihenfolge statt.

In der Kommunikation spricht man beim Sender von Daten auch von der „Datenquelle“ und beim Empfänger von der „Datensenke“.

Für die lokale Vernetzung hat sich das ursprünglich von Xerox entwickelte und nun durch die Normen IEEE 802.3 und ISO 8802.3 spezifizierte Ethernet durchgesetzt, da es bei geringem Kostenaufwand eine Reihe von Konfigurationsmöglichkeiten bietet.

Es ist, bezogen auf die Hardwareanforderungen relativ kostengünstig, einfach zu verlegen, genügt bezüglich der Übertragungsrate den meisten Anforderungen und lässt sich recht problemlos mit anderen Netzen koppeln.

Grundsätzlich wird bezüglich der Ethernet-Verkabelung zwischen so genannter „Thin-Ethernet“, die einfach und damit auch kostengünstig erfolgen kann, und „Twisted-Pair-Verkabelung“ unterschieden, wobei eine Thin-Ethernet-Verkabelung wiederum „offen“ oder mit Wanddosen erfolgen kann.



Hinweis

## Offene Thin-Ethernet-Verkabelung

Bei der offenen Thin-Ethernet-Verkabelung sind folgende Punkte zu beachten:

- Es werden Koaxialkabel vom Typ RG58 zwischen den einzelnen an das Netz angeschlossenen Rechnern verlegt, wobei der Anschluss der Kabel an die Netzwerkkarte mit **BNC-Kupplungen** (**B**aucus **N**auer **C**onnecto) an so



## Linux im Netzwerk

genannte „BNC-T-Stücke“ erfolgt; die Position des Servers innerhalb dieses Netzes ist nicht festgelegt (man spricht bei dieser Art der Verkabelung auch von einem „Bus“).

- Die Gesamtlänge eines solchen Thin-Ethernet-Busses sollte 128 Meter nicht überschreiten.
- Der Mindestabstand zwischen den einzelnen T-Stücken muss 50 cm betragen.
- Es ist möglich, bis zu 30 T-Stücke in einem solchen Netzsegment zu verwenden, d.h., dass insgesamt bis zu 30 Rechner (einschließlich des Servers) in einem solchen (Teil-)Netz vorhanden sein können.
- Das Thin-Ethernet muss an beiden Enden mit  $50\Omega$ -Widerständen (Abschlusswiderständen oder Terminatoren) abgeschlossen (terminiert) werden. Abbildung 13.2 zeigt die Struktur eines einfachen Thin-Ethernets.

Abb. 13.2 ○

Einfaches Thin-Ethernet mit einem Server und bis zu 29 Workstations

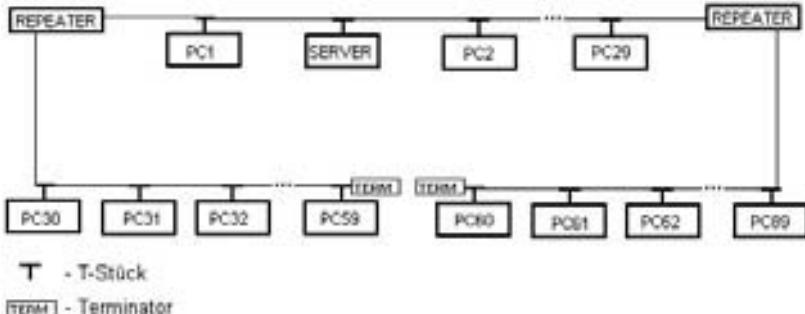


T - T-Stück

TERM - Terminator

Abb. 13.3 ○

Verbindung dreier Thin-Ethernet-Segmente mit jeweils bis zu 30 Computern pro Segment über Repeater



## Offene Thin-Ethernet-Verkabelung

- Es können bis zu drei solcher Netzsegmente über so genannte „Repeatere“ (Verstärker) miteinander verbunden werden, so dass die theoretisch maximal in einem Thin-Ethernet angeschlossene Zahl von Computern 90 beträgt, wobei die Länge der Verkabelung nicht mehr als 350 Meter betragen sollte.
- Mehrere solcher Netze können wiederum über so genannte „Router“ miteinander verbunden werden.

Beachten Sie, dass die Position des Servers auf dem Bus freigestellt ist

Diese Form des Netzwerks wurde als relativ kostengünstig und einfach zu verlegen charakterisiert. In Mark und Pfennig ausgedrückt entspricht diese Aussage der Aufstellung in *Tabelle 13.2* (beachten Sie hierbei, dass diese Preisangaben natürlich nur Näherungswerte sind, die anbieterspezifisch bis zu 20 oder mehr Prozent nach oben oder unten abweichen können).



Hinweis

### Kosten, die bei einer Thin-Ethernet-Verkabelung entstehen

| Komponente                                           | Preis          |
|------------------------------------------------------|----------------|
| NE2000-kompatible Netzwerkkarte mit BNC-Anschluss    | DM 30,00/Karte |
| RG58-Kabel (50W)                                     | DM 1,00/Meter  |
| BNC-Kupplungen (meist Teil des Netzwerkkartenpakets) | DM 2,00/Stück  |
| 50 W -Terminatoren                                   | DM 1,50/Stück  |

*Tabelle 13.2*



### No-Name vs. Markenprodukte

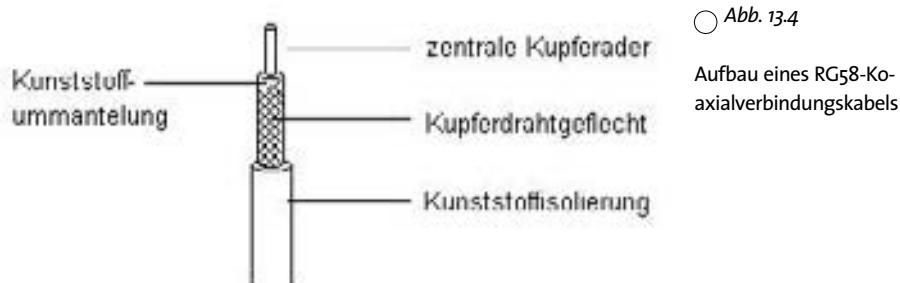
*Um Problemen bei der Konfiguration Ihres Netzwerks aus dem Wege zu gehen, sollten Sie in Erwägung ziehen, etwa DM 100,00 für eine gute Karte zu investieren. Die Karten von Markenherstellern bereiten nach den Erfahrungen der Autoren selten Probleme bei der Installation.*



### Konfektionsware vs. Eigenbau

Fertig konfektionierte Netzwerkkabel, bei denen Sie die Kupplungen nicht selbst mit dem Kabel verbinden müssen (es handelt sich hierbei um Quetschverbindungen, *crimp connections*) sind in jedem gut sortierten Elektronik-Fachgeschäft oder auch bei den meisten Computerhändlern erhältlich. Sie sind jedoch in der Regel teurer, als wenn Sie sie selber „crimpen“ würden, und nur in bestimmten Längen zu erhalten. Diese Tatsache führt dann meist zu zusätzlichem Kabelsalat auf oder unter dem Schreibtisch, wenn Kabel-Überlängen frei herumliegen. Aus diesem Grund finden Sie nachfolgend eine kurze „Bauanleitung“ für Netzwerkkabel vor. Wenn Sie technisch nicht gänzlich unversiert sind und darüber hinaus über das entsprechende Werkzeug verfügen, sollten Sie dazu in der Lage sein, Ihre Kabel selbst zu crimpeln.

Schauen Sie sich hierzu zunächst den Aufbau eines RG58-Koaxialkabels an, der in Abbildung 13.4 skizziert ist.



Mit diesem Kabel muss nun die BNC-Kupplung fest verbunden werden. Sie besteht (grob unterteilt) aus drei Komponenten:

- der eigentlichen Kupplung,
- einem Kupferdorn,
- einem kurzen Metallrohr.

Um die Kupplung anzubringen, verfahren Sie in folgenden Schritten:

- ❶ Schieben Sie das Röhrchen über die Kunststoffisolierung nach hinten.
- ❷ Isolieren Sie die Kunststoffisolierung ca. 1,5 cm ab.
- ❸ Biegen Sie das Kupfergeflecht nach hinten, wobei Sie es entflechten sollten.



## Offene Thin-Ethernet-Verkabelung

- (4) Isolieren Sie die zentrale Kupferader so weit ab, dass sie passgenau in den Dorn geschoben werden kann; achten Sie hier bitte wirklich darauf, dass die Ader nicht zu weit abisoliert ist – Thin-Ethernet-Verkabelungen haben die wunderbare Eigenschaft, mal auf Spannung zu sitzen und dann mal wieder nicht. In diesem zweiten Fall ist es (wie der Autor dieses Beitrags aus leidvoller eigener Erfahrung feststellen musste, nachdem er fast zwei Tage nach dem Fehler im Netz fahndete) schon vorgekommen, dass eine zu lang abisierte Kupferader einen Kurzschluss verursacht hat. (Was glauben Sie, was der Autor von der Firma hielt, die diese Verkabelung durchgeführt hat?)
- (5) Schieben Sie das abisierte Ende vorsichtig in die BNC-Kupplung, um zu überprüfen, ob es genau in die Kupplung passt. Notfalls kürzen Sie das abisierte Ende noch-mals um die überzählige Länge.
- (6) Schieben Sie den Dorn auf die zentrale Kupferader und quetschen Sie ihn. Zu diesem Zweck gibt es im Fachhandel spezielle Quetsch- (Crimp-)Zangen, notfalls tut es aber auch eine Kombizange.
- (7) Kürzen Sie das Kupfergeflecht, so dass es passgenau bis zur Verbreiterung der Kupplung abschließt. Schieben Sie das Röhrchen über das Ganze und crimpfen Sie es ebenfalls.
- (8) Schneiden Sie das Kabel in der gewünschten Länge ab, und verfahren Sie mit dem anderen Ende der Kabels ebenfalls wie gerade beschrieben.

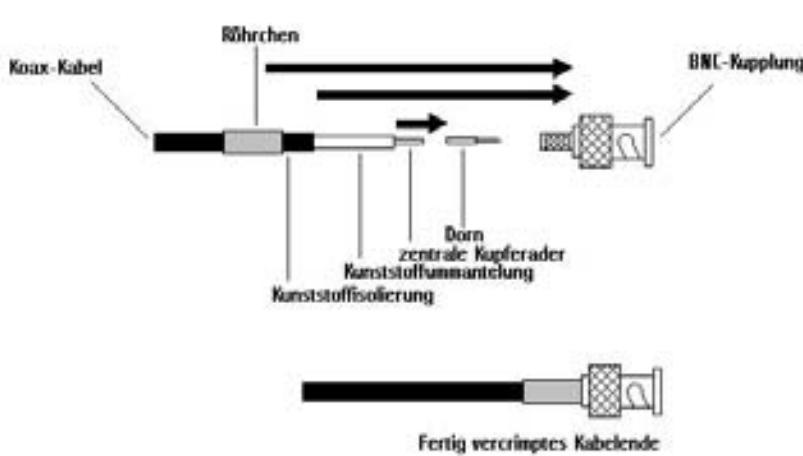


Abb. 13.5

Verkrampfen einer BNC-Kupplung mit einem RG58-Koaxialkabel

## Linux im Netzwerk

---



Eine Bemerkung noch zu den Netzwerkkarten: NE2000-kompatible Netzwerkkarten haben sich als Quasi-Standard in der Ethernet-Vernetzung durchgesetzt. Bezuglich der Kompatibilität von NE2000-Clones ist jedoch (jedenfalls für die Karte, die in den Linux-Rechner eingebaut werden soll) frei nach George Orwells „Animal Farm“ zu bemerken:

*Alle NE2000-Clones sind gleich kompatibel, aber manche sind kompatibler als andere.*

Dies heißt in der Praxis, dass Sie sich bei Ihrem Hardwarehändler für den Fall, dass die von Ihnen angeschaffte Karte unter Linux nicht funktioniert, ein Umtauschrech- ausbitten sollten.

Doch wie macht sich eine nicht 100 %ige Kompatibilität bemerkbar? Linux ver- sucht, die Netzwerkkarte beim Booten selbstständig zu erkennen und zu initialisie- ren – hängt sich der Rechner während dieses Vorgangs auf, wissen Sie, dass Sie unter Umständen eine Fehlinvestition getätigt haben.

Unter <http://cdb.suse.de/> finden Sie eine Hardware-Kompatibilitätsliste.

### Tipp



Doch zurück zu den Leistungsmerkmalen eines Thin-Ethernets: Bezuglich der Über- tragungsrate, die zuvor als „den meisten Anforderungen entsprechend“ charakteri- siert wurde, ist zu be-merken, dass darüber maximale Übertragungsraten von bis zu 10 Megabit pro Sekunde (Mbps) möglich sind, wobei sich allerdings alle Anwen- der im Netz diese Bandbreite teilen müssen. Diese Geschwindigkeit sollte im Regel- fall ausreichend sein, außer

- bei sehr starkem Netzbetrieb – dieser verlangsamt das Netzwerk spürbar,
- wenn sehr große Datenmengen, wie z.B. umfangreiche Bild- und Toninfor- mationen, über das Netz verteilt werden sollen. Solche Dateien können leicht Größenordnungen von mehreren Megabyte annehmen, so dass deren Über- tragung merklich länger dauert.

Der entscheidende Nachteil einer solchen Thin-Ethernet-Verkabelung liegt (außer in den zuvor beschriebenen Fällen, in denen diese Verkabelungsmethode hinsicht- lich ihrer Übertragungsge-schwindigkeit nicht mehr ausreichend ist) darin, dass, sollte z.B. ein Kabel zwischen den Endwi-derständen unterbrochen sein, das gesamte Netzwerk nicht länger funktionsfähig ist (siehe die weiter oben beschrie- bene leidvolle Erfahrung).

Die fehlerhafte Verbindung muss dann durch schrittweise Verkürzung der Netz- werkstrecke mittels Aufsetzen des zweiten Endwiderstands in Richtung des ersten lokalisiert werden. (Es sei denn, Sie verfügen über das geeignete Equipment, Ihr Netz durchzumessen.) Wurde das fehler-hafte Kabel gefunden, kann es entweder repariert oder ersetzt werden.



## Thin-Ethernet mit Dosen



### Thin-Ethernet mit Dosen

Eine (wenn auch etwas teurere, dafür bezüglich des anfallenden Kabelsalats und der damit einhergehenden Stolperfallen elegantere) Methode der Thin-Ethernet-Verkabelung besteht darin, die Kabel fest in Kabelschächten zu verlegen, an die der Anschluss über eine Kabelschlaufe von einer Wanddose zum Rechner erfolgt.

Hierbei fallen (bauartbedingt) zusätzlich folgende Kosten an:

**Zusätzliche Kosten, die bei einer Thin-Ethernet-Verkabelung mit Dosen anfallen**



| Komponente         | Preis    |
|--------------------|----------|
| EAD-Aufputzdose    | DM 25,00 |
| EAD-Anschlusskabel | DM 45,00 |

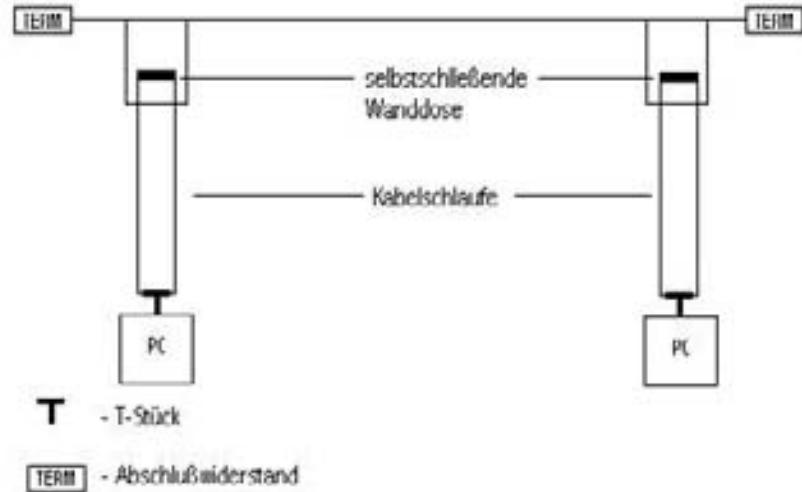
Tabelle 13.3



Folgende Abbildung zeigt die Verkabelung mit EADs.

Abb. 13.6

Thin-Ethernet-Verkabelung mit Dosen





### Twisted-Pair-Verkabelung

Bei der Twisted-Pair-Verkabelung handelt es sich um Kabel, die zwei oder mehr isolierte Draht-paare umfassen, welche sechs Verdrillungen (*Twists*) pro Zoll enthalten, wobei die Kabel abge-schirmt (*shielded*) oder ungeschirmt (*unshielded*) sein können. Eines der Kabel überträgt das Signal, das andere ist die Masseleitung.

Die sternförmige Verkabelung (*Stern-Topologie*) durch Verwendung so genannter „Hubs“, von denen aus zu jedem PC einzelne Leitungen gelegt werden müssen, bietet im Vergleich zum Thin-Ethernet eine erhöhte Betriebssicherheit, die durch den Einsatz von Dosen noch unterstützt wird, da durch sie Kabelsalat vermieden wird, der zur Stolperfalle werden kann.

Zwar ergibt sich durch dieses Prinzip ein höherer Verkabelungs- und Kostenaufwand, doch lässt sich für den Fall, dass ein PC im Netz ausfallen sollte, sofort das entsprechende Kabel isolieren und reparieren/ersetzen, ohne dass hiervon der Rest des Netzes beeinflusst wird.

Bei der Twisted-Pair-Verkabelung sind folgende Punkte zu beachten:

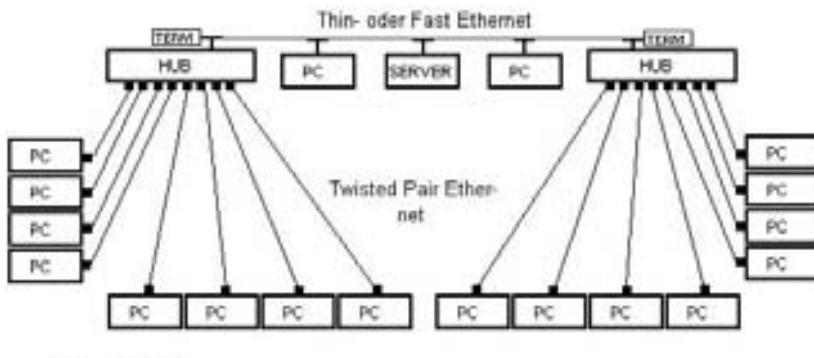
- Ein Hub wird mit einem bereits vorhandenen Thin-Ethernet-Segment über einen BNC-Anschluss verbunden.
- Die Verbindung zwischen Hub und PC erfolgt über Twisted-Pair-Patch-Kabel mit RJ45-Steckern.
- Die Länge der Patch-Kabel kann bei entsprechender Kabelqualität bis zu 100 Meter betragen.

Eine Twisted-Pair-Verkabelung könnte dann beispielsweise wie in Abbildung 13.7 erfolgen.

Wie weiter oben beschrieben, fallen bei einer Twisted-Pair-Verkabelung höhere Kosten an. Diese liegen zum einen im höheren Kabelaufwand begründet, zum anderen aber in der Anschaffung der Hubs und teurerer Netzwerkkarten, die neben einem BNC-Anschluss auch über eine RJ45-Buchse verfügen (man spricht hierbei auch von „Combo-Karten“, es gibt aber auch „reine RJ45-Karten“, die etwas günstiger ausfallen), wie folgendes Preisbeispiel zeigt.



## Twisted-Pair-Verkabelung



- - T-Stück
- - Terminator
- - RJ45-Buchse

Abb. 13.7

Verbindung eines Thin-Ethernet mit einem Twisted-Pair-Ethernet über Hubs

### Kosten, die bei einer Twisted-Pair-Verkabelung entstehen

| Komponente                                 | Preis            |
|--------------------------------------------|------------------|
| NE2000-kompatible Combo-Netzwerkarte       | DM 65,00/Karte   |
| Hub mit 8 RJ45-Buchsen und 1 BNC-Anschluss | DM 180,00/Hub    |
| Twisted-Pair-Kabel mit 2x RJ45-Steckern    | DM 25,00/5 Meter |

Tabelle 13.4

Viele Firmen bieten zu günstigen Preisen 100-Mbit-Karten, die Sie vorziehen sollten, damit im Netzbetrieb kein Datenstau zustandekommt. Für solche Karten ist dann natürlich ein entsprechend teurerer Hub anzuschaffen, wobei dieser, wenn er 10-Mbit- und 100-Mbit-Karten „bedienen“ soll, über ein so genanntes „Autosensing“ verfügen sollte, um den jeweils verwendeten Kartentyp selbstständig identifizieren zu können. Man spricht in diesem Fall dann auch besser von „autosensing switches“.

Neben diesen hardwareseitigen Vorarbeiten sind noch einige Grundlagen zum einführend beschriebenen TCP/IP-Protokoll zu klären.





### IP-Adressen

Dieser Abschnitt befasst sich mit der IP-Adressierung und mit Subnetzen. Hierbei sollen folgende Fragen geklärt werden:

- Was ist eine IP-Adresse?
- Was sind Adressklassen und Netzwerkadressen?
- Was sind Subnet-Masken und Subnet-Adressen?
- Wie werden Subnet-Masken definiert und verwendet?
- Wie wird all dies praktisch eingesetzt?

### IP-Adressierung

Eine IP- (**I**nternet **P**rotocol) Adresse ist ein eindeutiges Mittel zur Identifizierung eines Netzwerkgeräts. Sie ist eine 32 Bit lange Binärzahl, die im Regelfall durch vier dezimale Werte repräsentiert wird, von denen jeder 8 Bit darstellt. Diese Werte können zwischen 0 und 255 liegen und werden allgemein als „Oktette“ (*octets*) bezeichnet, z.B.:

134.245.010.033

Die binäre Darstellung dieser Adresse lautet:

10000110.11110101.00001010.00100001

Jede IP-Adresse besteht aus zwei Teilen:

der eine identifiziert das Netzwerk, der andere das Gerät (man spricht hier allgemein auch von einem „Node“ (Netzknoten). Dabei kann es sich zum Beispiel um einen PC oder auch einen Netzwerkdrucker mit einer eigenen Adresse handeln.

Die Adressklasse und die Subnet-Maske legen fest, welcher Teil zur Netzwerkadresse und welcher Teil zur Node-Adresse gehört.

### Adressklassen

Es gibt fünf verschiedene Adressklassen. Anhand der ersten vier Bits einer IP-Adresse kann festgestellt werden, welcher Klasse sie zuzurechnen ist:

- Adressen der Klasse A beginnen mit *0xxx* oder dezimal mit 1 bis 126.
- Adressen der Klasse B beginnen mit *10xx* oder dezimal mit 128 bis 191.
- Adressen der Klasse C beginnen mit *110x* oder dezimal mit 192 bis 223.



## IP-Adressen



- Adressen der Klasse D beginnen mit *1110* oder dezimal mit *224* bis *239*. Diese sind für das so genannte „Multitasking“ reserviert.
- Adressen der Klasse E beginnen mit *1111* oder dezimal mit *240* bis *254*. Diese sind für zukünftige Zwecke reserviert und sollten nicht verwendet werden.

Wenn Sie in vorstehender Aufstellung darüber gestolpert sind, dass ja wohl die *127* „fehlt“, können Sie dahingehend beruhigt sein, dass auch diese nicht vergessen wurden: Adressen, die mit *011111* (dezimal *127*) beginnen, sind für ein Loopback und für interne Tests der lokalen Maschine reserviert.

Mit diesen Informationen versorgt, können Sie nun erkennen, welcher Teil der IP-Adresse das Netzwerk (*N*) und welcher den Node (*n*) identifiziert:

- Klasse A NNNNNNNN.nnnnnnnn.nnnnnnnn.nnnnnnnn
- Klasse B NNNNNNNN.NNNNNNNN.nnnnnnnn.nnnnnnnn
- Klasse C NNNNNNNN.NNNNNNNN.NNNNNNNN.nnnnnnnn

Bei dem Beispiel *134.245.010.033* handelt es sich um eine Adresse der Klasse B; der Netzwerkteil (die Netzwerkadresse) wird durch die beiden ersten (*134.245.x.x*) und der Node-Teil durch die beiden letzten Oktette festgelegt (*x.x.010.033*).

### Subnet-Masken

Das Subnetting eines IP-Netzwerks kann aus einer Reihe von Gründen erfolgen. Hierzu gehören:

- übersichtlichere Organisationsstrukturen,
- die Verwendung unterschiedlicher physikalischer Medien (wie Ethernet, FDDI, WAN usw.),
- die Erhaltung von Adressräumen,
- eine höhere Sicherheit.

Der häufigste Grund, ein Subnetting vorzunehmen, liegt darin, den Netzwerkverkehr zu steuern.

In einem Ethernet-Netz „sehen“ alle Nodes eines Segments sämtliche Pakete, die von den anderen Nodes des Segments übertragen werden.

Der Datendurchsatz kann hierdurch negativ beeinflusst werden, da es durch das hohe Datenaufkommen häufiger zu Kollisionen (*packet collisions*) kommt, so dass eine große Zahl an Neuübertragungen dieser Pakete erforderlich ist.



## Linux im Netzwerk

---



Die Anwendung von Subnet-Masken auf eine IP-Adresse erlaubt es Ihnen, den Netzwerk- und den Node-Teil einer Adresse zu identifizieren, indem Sie durch eine bitweise logische UND-Verknüpfung zwischen der IP-Adresse und der Subnet-Maske die Netzwerkadresse ermitteln.

Betrachten Sie das Beispiel:

```
10000110.11110101.00001010.00100001
134.245.010.033 Klasse-B-IP
11111111.11111111.00000000.00000000
255.255.000.000 Standard Klasse-B-Subnet-Maske
```

Hieraus folgt:

```
10000110.11110101.00000000.00000000
134.245.000.000 Netzwerkadresse
```

Die Standard-Subnet-Masken sind für

- Klasse A 255.0.0.0 1111111.oooooooooooo.oooooooooooo.oooooooooooo
- Klasse B 255.255.0.0 1111111.1111111.oooooooooooo.oooooooooooo
- Klasse C 255.255.255.0 1111111.1111111.1111111.oooooooooooo

### Restriktivere Subnet-Masken

Der Standard-Subnet-Maske einer Klasse können weitere Bits hinzugefügt werden, um eine detailliertere Untergliederung zu erreichen. Wenn eine bitweise logische UND-Verknüpfung zwischen der Subnet-Maske und der IP-Adresse durchgeführt wird, resultiert hieraus die Subnet-Adresse.

Für sie gelten einige Einschränkungen. Node-Adressen, die sich ausschließlich aus Nullen und Einsen zusammensetzen, sind für die Spezifizierung des lokalen Netzes reserviert (wenn ein Host seine Netzwerkadresse nicht kennt) bzw. für die Gesamtzahl der Hosts eines Netzes (Broadcast-Adresse).

Dies gilt auch für Subnetze: Eine Subnet-Adresse kann nicht vollständig aus Nullen oder Einsen bestehen. Dies impliziert, dass eine „ein-bitige“ Subnet-Maske nicht erlaubt ist

Um die Gesamtzahl der Subnets oder Nodes zu berechnen, können Sie die Formel

$$2^n - 2$$

anwenden, wobei  $n$  gleich der Zahl der Bits in jedem Feld ist. Die Multiplikation der Zahl der Subnetze mit der Zahl der Nodes ergibt die Gesamtzahl der möglichen Nodes ihrer Klasse.



## IP-Adressen



Ein Subnetting reduziert grundsätzlich die Zahl der möglichen Nodes eines Netzwerks. Entnehmen Sie den folgenden Tabellen die für jede Netzwerkklasse möglichen Subnet-Masken mit den Zahlen der Netzwerke sowie Nodes und den Gesamtzahlen der Hosts pro Netz.

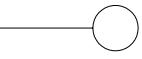
**Mögliche Subnet- und Host-IP-Adressen in einem Netz der Klasse A**

| Anz.Bits | Subnet-Maske    | Anzahl der Subnets | Anzahl der Hosts | Anzahl Netze x Anzahl Hosts |
|----------|-----------------|--------------------|------------------|-----------------------------|
| 2        | 255.192.0.0     | 2                  | 4.194.302        | 8.388.604                   |
| 3        | 255.222.0.0     | 6                  | 2.097.150        | 12.582.900                  |
| 4        | 255.240.0.0     | 14                 | 1.048.574        | 14.680.036                  |
| 5        | 255.248.0.0     | 30                 | 524.286          | 15.728.580                  |
| 6        | 255.252.0.0     | 62                 | 262.142          | 16.252.804                  |
| 7        | 255.254.0.0     | 126                | 131.070          | 16.514.820                  |
| 8        | 255.255.0.0     | 254                | 65.534           | 16.645.636                  |
| 9        | 255.255.128.0   | 510                | 32.766           | 16.710.660                  |
| 10       | 255.255.192.0   | 1.022              | 16.382           | 16.742.404                  |
| 11       | 255.255.224.0   | 2.046              | 8.190            | 16.756.740                  |
| 12       | 255.255.240.0   | 4.094              | 4.094            | 16.760.836                  |
| 13       | 255.255.248.0   | 8.190              | 2.046            | 16.756.740                  |
| 14       | 255.255.252.0   | 16.382             | 1.022            | 16.742.404                  |
| 15       | 255.255.254.0   | 32.766             | 510              | 16.710.660                  |
| 16       | 255.255.255.0   | 65.534             | 254              | 16.645.636                  |
| 17       | 255.255.255.128 | 131.070            | 126              | 16.514.820                  |
| 18       | 255.255.255.192 | 262.142            | 62               | 16.252.804                  |
| 19       | 255.255.255.224 | 524.286            | 30               | 15.728.580                  |
| 20       | 255.255.255.240 | 1.048.574          | 14               | 14.680.036                  |
| 21       | 255.255.255.248 | 2.097.150          | 6                | 12.582.900                  |
| 22       | 255.255.255.252 | 4.194.302          | 2                | 8.388.604                   |

*Tabelle 13.5*



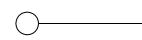
## Linux im Netzwerk



Mögliche Subnet- und Host-IP-Adressen in einem Netz der Klasse B

| Anz. Bits | Subnet-Maske    | Anzahl der Subnets | Anzahl der Hosts | Anzahl Netze x Anzahl Hosts |
|-----------|-----------------|--------------------|------------------|-----------------------------|
| 2         | 255.255.192.0   | 2                  | 16.382           | 32.764                      |
| 3         | 255.255.224.0   | 6                  | 8.190            | 49.140                      |
| 4         | 255.255.240.0   | 14                 | 4.094            | 57.316                      |
| 5         | 255.255.248.0   | 30                 | 2.046            | 61.380                      |
| 6         | 255.255.252.0   | 62                 | 1.022            | 63.364                      |
| 7         | 255.255.254.0   | 126                | 510              | 64.260                      |
| 8         | 255.255.255.0   | 254                | 254              | 64.516                      |
| 9         | 255.255.255.128 | 510                | 126              | 64.260                      |
| 10        | 255.255.255.192 | 1.022              | 62               | 63.364                      |
| 11        | 255.255.255.224 | 2.046              | 30               | 61.380                      |
| 12        | 255.255.255.240 | 4.094              | 14               | 57.316                      |
| 13        | 255.255.255.248 | 8.190              | 6                | 49.140                      |
| 14        | 255.255.255.252 | 16.382             | 2                | 32.764                      |

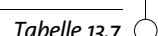
Tabelle 13.6



Mögliche Subnet- und Host-IP-Adressen in einem Netz der Klasse C

| Anz. Bits | Subnet-Maske    | Anzahl der Subnets | Anzahl der Hosts | Anzahl Netze x Anzahl Hosts |
|-----------|-----------------|--------------------|------------------|-----------------------------|
| 2         | 255.255.255.192 | 2                  | 62               | 124                         |
| 3         | 255.255.255.224 | 6                  | 30               | 180                         |
| 4         | 255.255.255.240 | 14                 | 14               | 196                         |
| 5         | 255.255.255.248 | 30                 | 6                | 180                         |
| 6         | 255.255.255.255 | 62                 | 2                | 124                         |

Tabelle 13.7



510



### Broadcast-Adresse

Neben IP-Adresse und Subnet-Maske muss noch die Broadcast-Adresse beachtet werden. Diese Adresse gibt Auskunft darüber, welche Rechner im Netz zu befragen sind, wenn ein Rundruf erfolgt.

Solche Rundrufe erfolgen beispielsweise bei manchen kommerziellen UNIX-Anwendungen (ARC-Info, Interleaf Publisher), um zu überprüfen, ob die Lizenzbestimmungen des Herstellers eingehalten werden (dass zum Beispiel nicht zwei Leute gleichzeitig mit dem Publisher arbeiten, obwohl doch nur eine Benutzerlizenz vorliegt ...).

Um nun zu verhindern, dass Rechner befragt werden, die gar nicht zum eigenen Subnetz gehören, muss man den Bereich entsprechend einschränken (siehe auch das folgende Beispiel).

Beispiel:

Gehen Sie davon aus, dass Ihnen die Netzwerksnummer 200.133.175.0 zugewiesen wird (sorry, falls es irgend jemanden dort draußen gibt, der diese Domain hat ...). Sie möchten nun dieses Netzwerk in verschiedene kleine Gruppen einteilen.

Sie können dies bewerkstelligen, indem Sie dem Netzwerk eine Subnet-Adresse zuweisen.

Sie werden dieses Netzwerk in 14 Subnetze mit jeweils 14 Nodes splitten, so dass Ihnen maximal 196 Nodes zur Verfügung stehen, statt der 254, die Sie sonst zur Verfügung hätten.

Dies hat aber den Vorteil, dass der Netzverkehr isoliert in Subnetzen ablaufen kann (was darüber hinaus natürlich die Sicherheit erhöht).

Sie verwenden hierzu eine vier Bit lange Subnet-Maske.

Rufen Sie sich noch einmal in Erinnerung, dass die Standard-Subnet-Maske eines Klasse C-Netzwerks

255.255.255.0 (11111111.11111111.11111111.00000000)

ist. Indem Sie diese um vier Bit ausdehnen, erhalten Sie folgende Maske:

255.255.255.240 (11111111.11111111.11111111.11110000)

Dies gibt Ihnen 16 mögliche Netzwerksnummern, von denen Sie allerdings zwei nicht verwenden können.



## Linux im Netzwerk

Mögliche Subnet- und Host-IP-Adressen in einem Netz der Klasse C mit einer vier Bit langen Subnet-Maske

| Subnet-Bits | Netzwerknummer  | Node-Adressen | Broadcast-Adresse |
|-------------|-----------------|---------------|-------------------|
| 0000        | 200.133.175.0   | RESERVIERT    | KEINE             |
| 0001        | 200.133.175.16  | 17 bis 30     | 200.133.175.31    |
| 0010        | 200.133.175.32  | 33 bis 46     | 200.133.175.47    |
| 0011        | 200.133.175.48  | 49 bis 62     | 200.133.175.63    |
| 0100        | 200.133.175.64  | 65 bis 78     | 200.133.175.79    |
| 0101        | 200.133.175.80  | 81 bis 94     | 200.133.175.95    |
| 0110        | 200.133.175.96  | 97 bis 110    | 200.133.175.111   |
| 0111        | 200.133.175.112 | 113 bis 126   | 200.133.175.127   |
| 1000        | 200.133.175.128 | 129 bis 142   | 200.133.175.143   |
| 1001        | 200.133.175.144 | 145 bis 158   | 200.133.175.159   |
| 1010        | 200.133.175.160 | 161 bis 174   | 200.133.175.175   |
| 1011        | 200.133.175.176 | 177 bis 190   | 200.133.175.191   |
| 1100        | 200.133.175.192 | 193 bis 206   | 200.133.175.207   |
| 1101        | 200.133.175.208 | 209 bis 222   | 200.133.175.223   |
| 1110        | 200.133.175.224 | 225 bis 228   | 200.133.175.229   |
| 1111        | 200.133.175.240 | RESERVIERT    | KEINE             |

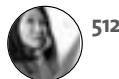
Tabelle 13.8

## Router

Um Daten von einem Teilnetz zum nächsten weiterzureichen, benötigt man so genannt „Router“. Unter einem Router versteht man eine Maschine, die an zwei oder mehreren Netzwerken angeschlossen ist und die Daten auf dem Netzwerk je nach Zieladresse an die verschiedenen Netzwerke übergibt.

Der Router hat dann mindestens so viele Internetadressen, wie er Netzwerkanschlüsse hat.

Will nun ein Rechner Daten an einen anderen Rechner schicken, der sich nicht im gleichen Subnetz befindet, kann er diese nicht direkt absenden, sondern muss sie diesem Router schicken, der „Mitglied“ beider Subnetze ist.



## IP-Adressen



Befindet sich der Zielrechner an einem an diesem Router angeschlossenen Subnetz, kann er die Daten des Routers dort abliefern. Ansonsten muss er die Daten zum nächsten Router weiterreichen.

Wohin die Daten weitergereicht werden, wird anhand von Routing-Tabellen bestimmt. Die Routing-Tabellen der meisten am Internet angeschlossenen Rechner beinhalten nur einen Eintrag, nämlich die so genannte „*default-Route*“, d.h. die Adresse des Routers, an den alle Daten zu senden sind, die nicht an den Rechner des lokalen Subnetzes adressiert sind.

Bei UNIX-Rechnern kann diese Tabelle mit dem Befehl *netstat -r* abgefragt werden.

Die Routing-Tabellen der wichtigeren Internet-Router beinhalten viele Tausend Einträge. Daher handelt es sich bei ihnen meist um sehr leistungsfähige Rechner, die ein schnelles Nachschlagen in diesen Tabellen ermöglichen.

Möchten Sie einmal feststellen, welchen Weg Datenpakete über das Netz nehmen, können Sie ihn mit dem Befehl *traceroute* verfolgen.

Ein Versuch, den Weg zu routen, der zum Host-Rechner führt, welcher die SYBEX-Homepage im WWW beherbergt, ergab folgendes Ergebnis:

```
gehrke@brunetti:~>traceroute www.sybex.de
traceroute to www.sybex.de (149.221.231.163), 30 hops max, 40
byte packets
 1 kiel-d.kielnet.net (62.104.199.10) 23 ms 23 ms 22 ms
 2 kiel-c.kielnet.net (62.104.199.9) 24 ms 23 ms 23 ms
 3 S2-0.hmb1-t.mcbone.net (62.104.199.1) 25 ms 25 ms 25 ms
 4 G3-0.hmb2-gsr.mcbone.net (62.104.214.5) 25 ms 25 ms 25 ms
 5 L0.bln2-gsr.mcbone.net (62.104.191.139) 29 ms 29 ms 29 ms
 6 L0.mdb2-gsr.mcbone.net (62.104.191.140) 31 ms 31 ms 31 ms
 7 L0.lpz2-gsr.mcbone.net (62.104.191.137) 32 ms 33 ms 33 ms
 8 L0.nbg2-gsr.mcbone.net (62.104.191.135) 36 ms 36 ms 37 ms
 9 L0.ffm4-gsr.mcbone.net (62.104.191.128) 40 ms 39 ms 39 ms
10 decix-gw.f.dpn.net (194.31.232.192) 48 ms 41 ms 40 ms
11 cisco2.d.dpn.net (194.77.111.33) 45 ms 47 ms 46 ms
12 rp2-gw.dpn.net (194.77.111.118) 48 ms 49 ms 47 ms
13 wwwserv1.rp-online.de (149.221.232.6) 243 ms 49 ms 49 ms
```

Ein abschließender Hinweis zur IP-Adressenvergabe: Zu der Zeit, als das IP-Protokoll entwickelt wurde, hatte noch niemand eine Vorstellung davon, welche Dynamik die Entwicklung des Internets einmal annehmen würde.

Aus diesem Grund schien den Entwicklern die Beschränkung auf ca. 4 Milliarden mögliche Adressen (bedingt durch die 32-Bit-Breite) mehr als ausreichend.



## Linux im Netzwerk



Da jedoch viele dieser Adressen ungenutzt bleiben, weil an bestimmte Internetteilnehmer, wie zum Beispiel Universitäten, ganze Nummernkontingente vergeben werden, die somit von anderen Teilnehmern nicht mehr genutzt werden können, obwohl sie vielleicht an den entsprechenden Hochschulen überhaupt keinen Einsatz finden, ist diese Zahl rein theoretischer Natur.

Als zweiter Nachteil ist zu bemerken, dass der Adressraum mit 32 Bit Breite einfach zu klein ist, um ein vernünftiges, hierarchisch organisiertes Routen zu ermöglichen.

Deshalb hat die Internet Engineering Task Force (*IETF*) eine neue IP-Version entwickelt, die „IPng“ genannt wird (**I**nternet **P**rotocol **n**ext **g**eneration).

---

Interessierte seien  
auf den Standard  
RFC 1752 verwiesen

IPng-Adressen sind 128 Bit lang, womit sich die schier unvorstellbare Zahl von 340.282.366.920.938.463.374.607.431.768.211.456 möglichen Adressen ergibt. Damit sollten es möglich sein, selbst auch noch Ihren Eierkocher aus dem Netz anzusteuern ...

Die Umstellung von IP auf IPng wird sicherlich im Laufe der nächsten Jahre vonstatten gehen – hier müssen Sie sich allerdings zunächst einmal mit IP begnügen.

## Linux im heterogenen Netzwerk

Nachdem Sie bisher erfahren haben, welches die physikalischen und die softwareseitigen Grundlagen für eine Vernetzung sind, soll Ihnen der nächste Abschnitt zeigen, wie Sie die Stärken eines Linux-Rechners in einem Rechnerverbund als Server für Computer nutzen, die mit einem anderen Betriebssystem arbeiten.

Die hier beschriebenen Techniken lassen sich natürlich auch für die Vernetzung mehrerer Linux-Rechner einsetzen.

Der Nutzen einer solchen Konstellation liegt auf der Hand: Es gibt kein aktuell verfügbares System, das gleichzeitig die Stabilität von Linux mit einem ähnlich „moderaten Preis“ verknüpft. Sprich: Die meisten kommerziellen UNIX-Betriebssysteme sind oftmals (zumindest für einen kommerziellen Einsatz) mit Anschaffungskosten verbunden, während gleichzeitig die Produkte eines bekannten Herstellers aus Redmont geradezu einen Hang zum „Reboot“ haben, so dass von einem geordneten Arbeiten nur selten die Rede sein kann.

Da man jedoch häufig noch darauf angewiesen ist, mit kommerziellen Programmen zu arbeiten, die auf DOS und dem darauf basierenden Fenstersystem laufen, weil die Geschäftspartner, mit denen man kommuniziert, diese noch verwenden und die Linux-basierten Alternativen wie StarOffice, WordPerfect Office oder die ApplixWare bezüglich der von ihnen gestellten Im- und Exportfilter nicht immer zufrieden stellend arbeiten, kann Linux dann jedoch als Schadensbegrenzer eingesetzt werden: Die anfallende Arbeit kann jederzeit auf dem Linux-Rechner zwischengelagert und somit auch sicher verwahrt werden.



## Linux im heterogenen Netzwerk



Ein weiterer Fall ist, dass für ein bestehendes UNIX-System neue Hardware erforderlich ist. Bevor hier für eine teure Workstation von Silicon Graphics, Sun Microsystems oder IBM z.B. eine neue Festplatte für den doppelten Preis einer normalen Platte beschafft wird, ist es möglich, einen Linux-Rechner als Platten- oder auch Fileserver einzusetzen.

Um weitere Kosten zu sparen, können Sie den Linux-Rechner als Printserver einsetzen. Dieser kann von den anderen Rechnerplattformen angesteuert werden, um Ihre Druckjobs, die Sie auf den unterschiedlichen Systemen mit unterschiedlicher Software erstellt haben, auf dem Server abarbeiten zu lassen.

Die vierte, wenn auch nicht die letzte Möglichkeit, Linux im heterogenen Netz zu nutzen, besteht darin, das System als Kommunikationsserver für alle angeschlossenen Netzwerkrechner zu benutzen. Hier erweist es sich ebenfalls als eine stabile Plattform, die alle standardisierten Protokolle und Programme standardmäßig mitliefert und somit die ideale Plattform für einen Übergang ins Internet oder in ein anderes entfernt stehendes Netz bietet.

Einige der Nutzungsmöglichkeiten werden im weiteren Verlauf des Kapitels vor gestellt.

### NFS

Ein wichtiger Hinweis zu Beginn dieses Abschnitts: Mit der Version 6.3 hat SuSE leider einige Änderungen bezüglich der Paketinhalte und -funktionalitäten im Vergleich zu vorhergehenden Versionen vorgenommen, die nicht unbedingt sauber dokumentiert sind. Um einige der hier und detaillierter im folgenden Abschnitt beschriebenen Leistungsmerkmale nutzen können, sollten Sie:

- bei Verwendung des Kernel-basierten NFS-Servers die Datei `/etc/rc.config` mithilfe von YaST dahingehend modifizieren, dass Sie unter *Administration des Systems > Konfigurationsdatei verändern* den Wert für die Variable `USE_KERNEL_NFSD` auf `yes` setzen.
- bei Verwendung des Userspace NFS Servers gegebenenfalls das Paket `nfserv` aus der Serie `n` nachinstallieren und die Datei `/etc/rc.config` mithilfe von YaST dahingehend modifizieren, dass Sie unter *Administration des Systems > Konfigurationsdatei verändern* den Wert für die Variable `NFS_SERVER` auf `yes` setzen.

Um den unter Linux zur Verfügung stehenden Plattenplatz auch anderen Rechnern im Netzwerk zugänglich zu machen, gibt es den *NFS-Server* (respektive den *NFS-Client*).

Hiermit besitzen Sie ein leistungsstarkes Werkzeug, das es, knapp formuliert, ermöglicht, den gesamten von NFS-Servern (in vorliegendem Fall handelt es sich hierbei „nur“ um den einzelnen Linux-Server) zur Verfügung gestellten Plattenplatz von den Rechnern, die Zugriff auf das Netzwerk haben, wie ein einziges großes physikalisches Laufwerk behandeln zu lassen.



## Linux im Netzwerk

---



Dateisysteme können von den einzelnen Rechnern sowohl exportiert als auch importiert werden. Das Netzwerk wird somit transparent für seine Benutzer.

Unter Linux sind nun einige Schritte notwendig, um das System so einzurichten, dass Benutzer von anderen Rechnern mittels NFS auf den Linux-Server zugreifen können.

Voraussetzung, um NFS zu benutzen, ist der Start des RPC-Portmappers (**Remote Procedure Calls**). Dies wird über das Skript `/sbin/init.d/rpc` (distributionsabhängig) erledigt.

Als Nächstes müssen die Dämonen `nfsd` und `mountd` gestartet werden.

Normalerweise werden sie bei der Einrichtung von NFS im Initialisierungsskript `rc.inet2` eingetragen. Sollte dies nicht automatisch geschehen, müssen Sie es noch „von Hand“ erledigen.

Des Weiteren ist zu regeln, welche Rechner im Netz überhaupt Plattenplatz vom NFS-Server in Anspruch nehmen dürfen.

NFS stellt Ihnen diesen Platz ab einem bestimmten Verzeichnis zur Verfügung. Welches Verzeichnis freigegeben wird, welchen Benutzern auf welchen Rechnern dort Platz eingeräumt wird, muss in der Datei `/etc/exports` vom Systemverwalter festgelegt werden. Diese Datei wird im folgenden Abschnitt im Zusammenhang mit dem Auto-Mounter genauer analysiert.

## Besondere Netzwerkimplementierungen

In der Regel wird man in einer Netzwerkumgebung den Wunsch haben, dass ein User – egal, an welche Workstation er sich setzt – stets Zugriff auf die Daten in seinem Heimatbereich hat. Man könnte einen solchen Zugriff dadurch realisieren, dass der Heimatbereich, der sich auf einem Server oder der eigenen Workstation befindet, in den Dateibaum des lokalen Rechners eingehängt wird, was allerdings mehrere Probleme aufwirft:

- Es ist dem „Normal-User“ standardmäßig nicht gestattet, Dateisysteme zu mounten.
- Sollte dem „Normal-User“ dieses Privileg zugestanden werden (Sie müssen hierzu entsprechende Einträge in der Konfigurationsdatei `/etc/fstab` vornehmen), müsste dieser dazu in der Lage sein, selbstständig den Mount- (und im Anschluss den Unmount-)Vorgang durchzuführen (Sie könnten ihm hierzu natürlich entsprechende Shell-Skripte zur Verfügung stellen). Vergisst dieser jedoch, das Dateisystem wieder aus dem lokalen System auszuhängen, wenn er seine Arbeit beendet, führt dies zu unnötigen Netzwerkverbindungen.



## Besondere Netzwerkimplementierungen



Diese statischen Mounts führen dazu, dass es zu „Hängern“ im System kommt, sollte der entsprechende NFS-Server aus irgendwelchen Gründen nicht ansprechbar sein (wie zum Beispiel in dem Fall, dass die entsprechende Maschine – beabsichtigt oder unbeabsichtigt – nicht mehr am Netzwerk angeschlossen ist).

Einen Ausweg aus diesem Dilemma bietet der so genannte „Auto-Mounter“.

### Der Auto-Mounter

Voraussetzung für die Nutzung seiner Leistungsmerkmale ist die Unterstützung durch den Kernel. Diese steht zwar bei Verwendung der SuSE-Standard-Kernel zur Verfügung, Sie müssen sie jedoch für den Fall, dass Sie einen eigenen Kernel kompilieren wollen oder müssen, explizit aktivieren.

Wechseln Sie hierzu als User *root* in das Verzeichnis */usr/src/linux* und rufen Sie dort

```
make xconfig
```

auf. Aktivieren Sie in der Sektion *Filesystems* den Eintrag *Kernel automounter support*, wie in Abbildung 13.8 dargestellt.

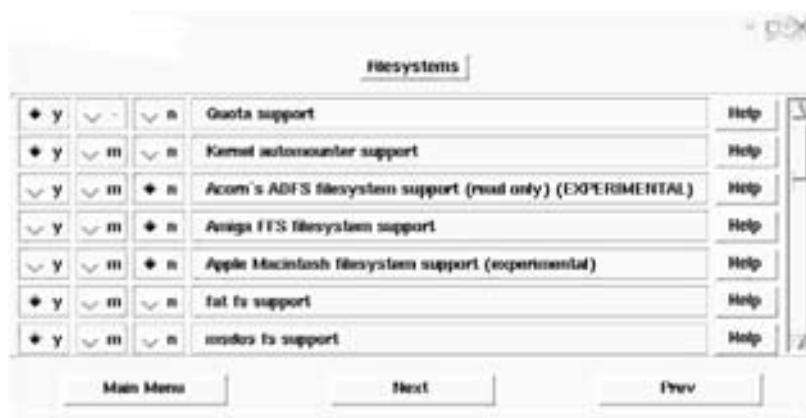


Abb. 13.8

Aktivieren der Kernel-Auto-Mounter-Funktion

Nehmen Sie gegebenenfalls weitere Kernel-Anpassungen vor, kompilieren Sie den Kernel (beachten Sie hierzu auch Kapitel 8 *Konfiguration und Kompilierung des Kernels*) im Anschluss, und starten Sie Ihr System neu – es ist nun für die korrekte Interpretation der folgenden Anpassungen vorbereitet.



## Linux im Netzwerk

---



Im Folgenden wird davon ausgegangen, dass die Heimatverzeichnisse derjenigen User, welche die Dienste des Auto-Mounters in Anspruch nehmen sollen, unterhalb von `/export/home` auf der „Heimatmaschine“ des jeweiligen Users liegen, z.B. unter `travelmate:/export/home/gehrke`, wenn der Rechner des Users *gehrke* den Namen *travelmate* hat.

### Hinweis



Warnung: Um die in diesem Kapitel dargestellten Leistungsmerkmale des Auto-Mounters nutzen zu können, dürfen die Heimatbereiche der User des Serversystems/der Serversysteme *nicht* unter `/home` liegen, sondern sollten in einem Verzeichnis wie etwa dem hier vorgeschlagenen `/export/home` untergebracht sein!

Wenn der Auto-Mounter Dateisysteme, wie weiter unten dargestellt, in das lokale Dateisystem einhängt, sich unterhalb des „genutzten“ Mount-Points allerdings noch Dateistrukturen befinden, sind diese für die Dauer der Aktivität des Auto-Mounters nicht länger sichtbar. Dies jagt einem zunächst einmal einen gehörigen Schrecken ein, da man vermeintlich seine sämtlichen User-Bereiche mit ihren Inhalten verloren hat, was allerdings nicht der Fall ist, da diese Bereiche – wie beschrieben – zeitweise nur nicht mehr sichtbar sind.

Um jedoch jeglichen Eventualitäten aus dem Weg zu gehen, sollten Sie Ihr Dateisystem lieber entsprechend den Vorgaben „umstricken“ oder Experimente mit dem Auto-Mounter nur auf neu installierten Systemen vornehmen, in deren Dateisystemen Sie die hierfür geforderten Strukturen von vornherein schaffen.

Dieser Zweig des Dateibaums muss für weitere Rechner innerhalb des Netzwerks freigegeben werden, damit Sie ihn in Ihr lokales Dateisystem einhängen können. Hierzu dient die bereits zuvor angesprochene Datei `/etc(exports`, die im Folgenden etwas detaillierter betrachtet werden soll.

### Die Datei `/etc(exports`

Die Datei `/etc(exports` legt fest, welche Verzeichnisse (samt Unterverzeichnissen) innerhalb des Dateibaums für NFS-Dienste zur Verfügung gestellt („exportiert“) werden sollen und welche Rechner im Netzwerk darauf in welcher Form zugreifen dürfen. Die in ihr enthaltenen Informationen werden sowohl vom NFS-Mount-Daemon `mountd` als auch vom NFS-Dateiserver-Daemon `nfsd` verwendet.

Das Dateiformat ähnelt dem der Exportdatei von SunOS, bietet jedoch eine Reihe zusätzlicher Konfigurationsmerkmale. Jede Zeile enthält:

- den Namen des für den Export freizugebenden Verzeichnisses,
- eine optionale Liste an Maschinen- oder Netzgruppennamen (NFS-Clients), denen es erlaubt sein soll, den entsprechenden Teil des Dateibaums in ihr Dateisystem zu mounten und
- eine optionale, geklammerte Liste mit Mount-Parametern, welche die Art des Zugriffs festlegen.



## Besondere Netzwerkimplementierungen



Wie bei anderen Konfigurationsdateien auch, werden Leerzeilen ignoriert, und ein einer Zeile vorangestelltes Doppelkreuz (#) kennzeichnet diese als Kommentarzeile. Wenn ein Eintrag mehrzeilig sein soll, muss jeder Zeilenumbruch durch einen Backslash (\) gekennzeichnet sein.

### Formate von Maschinennamen

NFS-Clients können auf unterschiedliche Weise spezifiziert werden:

- *Als einzelne Hosts:* Sie können diesen entweder mit seinem Hostnamen (*travelmate*), seinem vollen Domännamen (*travelmate.sybex.de*) oder seiner IP-Adresse (*192.168.0.175*) angeben.
- *Als Netzwerkgruppe:* NIS-Netzwerkgruppen können mit dem *Commercial at* (@) als solche gekennzeichnet werden (auf Einzelheiten zu NIS muss hier leider verzichtet werden).
- *Mit Hilfe von „Jokerzeichen“ (Wildcards):* Maschinennamen können die Jokerzeichen \* und ? in der Form \*.sybex.de enthalten, um den Inhalt der Exportdatei kompakter zu gestalten. In diesem Beispiel würden alle Hosts der Domain *sybex.de* Berücksichtigung finden.

Bei dieser Syntaxvariante ist zu beachten, dass etwaige in den Hostnamen enthaltene Punkte nicht durch das Jokerzeichen abgedeckt wären – ein Rechner mit dem Namen *pamsinck.sybex.de* würde bei der Ressourcenfreigabe also nicht bedient werden.



Hinweis

- *Durch IP-Netzwerke:* Verzeichnisse können ebenfalls an alle Hosts eines IP-(Sub-)Netzwerks gleichzeitig vergeben werden. Dies geschieht durch die Angabe eines IP-Adressen-/Netzwerkmasken-Paares, wie zum Beispiel: 134.245.35.0/255.255.255.0 (beachten Sie zu Subnetzmasken den entsprechenden Abschnitt weiter oben in diesem Kapitel).
- *Durch =public:* Diese Syntaxvariante, auf die hier nicht besonders eingegangen werden soll, da sie in erster Linie im Zusammenhang mit Web- und FTP-Servern von Bedeutung ist, deren Darstellung den Rahmen dieses Buches sprengen würde, dient der Festlegung des „Wurzelverzeichnisses“, von dem aus Clients Dienste nutzen können.

### Optionen

Die Daemons *mountd* und *nfsd*, welche die Datei */etc/exports* lesen, „verstehen“ die folgenden Optionen:

## Linux im Netzwerk

### Optionen der Datei `/etc(exports`

| Option                     | Bedeutung                                                                                                                                                                                                                                                                         |
|----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>secure</code>        | Diese (standardmäßig verwendete) Option bewirkt, dass nur Anfragen an einen Internet-Port gestellt werden können, dessen Portnummer kleiner ist als 1.024. Um sie zu deaktivieren, geben Sie <code>insecure</code> an.                                                            |
| <code>ro</code>            | erlaubt nur lesende ( <i>read only</i> ) Zugriffe auf den spezifizierten NFS-Bereich. Standardmäßig sind ebenfalls Schreibzugriffe erlaubt, was durch die Angabe von <code>rw</code> ( <i>read write</i> ) explizit geschehen kann.                                               |
| <code>noaccess</code>      | Diese Option sperrt für die angegebenen Clients den Zugriff auf Verzeichnisstrukturen unterhalb des angegebenen Verzeichnisses. Dies ist dann nützlich, wenn innerhalb einer freigegebenen Verzeichnishierarchie einzelne Verzeichnisse ausgeschlossen werden sollen.             |
| <code>link_relative</code> | konvertiert absolute symbolische Links (die Pfadangabe des Links beginnt mit einem Slash) in relative Links, durch „Bereitstellung“ einer entsprechenden Zahl an <code>..</code> , um von dem Verzeichnis, das den Link enthält, in das Wurzelverzeichnis des Servers zu gelangen |
| <code>link_absolute</code> | Standardeinstellung; alle symbolischen Links bleiben unberührt                                                                                                                                                                                                                    |

Tabelle 13.9

### User ID Mapping

Die Zugriffsrechte auf Dateien auf dem Server basieren auf der User ID (*UID*) und der Gruppen ID (*GID*), die mit jedem entfernten Zugriff (*RPC – remote procedere call*) via NFS übergeben werden. Als Normalanwender würde man wahrscheinlich erwarten, dass man den gleichen Zugriff auf Dateien auf dem Server hat wie in einem lokalem Dateisystem, was jedoch nicht in jedem Fall gegeben und auch nicht wünschenswert ist, auch wenn UID und GID auf dem Client- und auf dem Server-Host identisch sind.

So ist es in der Regel nicht erwünscht, dass ein lokaler Anwender *root* (mit der UID „0“) der Client-Maschine als eben dieser User *root* betrachtet wird, wenn er Dateizugriffe auf einen NFS-Server vornimmt. Aus diesem Grunde wird die UID 0 bei NFS-Zugriffen in ein anderes Benutzerkonto umgeleitet (man spricht hier von „Mapping“), nämlich dem so genannten „anonymen Konto“ des Users *nobody*. Diese standardmäßig verwendete Betriebsart wird auch als „*root squashing*“ bezeichnet; sie kann mit der Option `no_root_squash` deaktiviert werden.

## Besondere Netzwerkimplementierungen



Standardmäßig versucht der NFS-Daemon beim Starten des Dienstes, die anonyme UID und GID durch Durchsuchen der Datei `/etc/passwd` nach dem User `nobody` herauszufinden. Wenn dieser nicht gefunden werden kann, wird sowohl für UID als auch für GID -2 (das heißt der Wert 65534) verwendet. Diese Werte können durch die Optionen `anonuid` und `anongid` verändert werden.

Der NFS-Daemon erlaubt es Ihnen, neben dem User `root` weitere beliebige UIDs und GIDs zu spezifizieren, die ebenfalls auf den User `nobody` „gemapped“ werden sollen (vergleichen Sie dazu in Tabelle 13.10 „squash\_uids“ und „squash\_gids“).

Um bei allen Usern ein Mapping auf `nobody` vorzunehmen, verwenden Sie die Option `all_squash`.

In Netzwerksituationen, in denen die UIDs der gleichen User zwischen den verschiedenen Maschinen variieren, bietet der  `nfsd` verschiedene Mechanismen, um Server-UIDs an Client-UIDs oder anders herum Client-UIDs an Server-UIDs dynamisch zu mappen. Bei diese Mechanismen handelt es sich um *statisches Mapping*, *NIS-basiertes Mapping* und *ugidd-basiertes Mapping*, die im folgenden kurz vorgestellt werden sollen.

- Statisches Mapping: Dieses wird durch Verwendung der Option `map_static` aktiviert, die einen Dateinamen, der die entsprechenden Mappings enthält, als Argument verwendet.
- NIS-basiertes Mapping: Bei dieser Form des Mappings wird vom Client eine Anfrage an den NIS-Server gestellt, um eine Zuordnung von User- und Gruppennamen zwischen NFS-Server und -Client zu erhalten.
- *ugidd*-basiertes Mapping: Dieses wird durch Einsatz der Option `map_daemon` aktiviert, welche die Verwendung des UGID RPC-Protokolls bewirkt. Hierzu muss auf dem Client der Mapping -Daemon `ugidd` laufen. Da jedoch jeder beliebige Benutzer auf dem Client eine Liste gültiger User-Namen abfragen kann, ist diese Methode die am wenigsten sichere der hier dargestellten Methoden.

Tabelle 13.10 zeigt die für ein Mapping möglichen Optionen.

### Mapping-Optionen



| Option                      | Bedeutung                                                                                                                                                                                                                                                              |
|-----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>root_squash</code>    | Mapping-Anfrage eines Users mit der UID/GID 0 an die anonyme UID/GID. Beachten Sie hierbei, dass sich diese Form des Mappings tatsächlich nur auf den Superuser bezieht. Andere kritische Accounts, wie die des Users <code>bin</code> , sind hiervon nicht betroffen. |
| <code>no_root_squash</code> | Schaltet das <i>root squashing</i> aus.                                                                                                                                                                                                                                |



## Linux im Netzwerk

### Mapping-Optionen (Forts.)

| Option                                    | Bedeutung                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>squash_uids</i> und <i>squash_gids</i> | spezifiziert eine Liste mit UIDs oder GIDs, die ein anonymes Mapping erfahren sollen, z.B.<br><i>squash_uids=0-10,500,510-515</i>                                                                                                                                                                                                                                                                                                                                       |
| <i>all_squash</i>                         | Mapping aller UIDs und GIDs auf den anonymen User. Dies ist sinnvoll bei z.B. NFS-exportierten öffentlichen FTP-Verzeichnissen. Das Gegenteil hierzu ist die Option <i>no_all_squash</i> , welche die Standardeinstellung ist.                                                                                                                                                                                                                                          |
| <i>map_daemon</i>                         | schaltet dynamisches UID-/GID-Mapping ein. Jede UID in einer NFS-Anfrage wird in die entsprechende Server-UID übersetzt, und jede UID in einer NFS-Antwort erfährt ein entsprechendes gegenläufiges Mapping. Diese Option erfordert, dass auf dem Client der Prozess <i>rpc.ugidd</i> läuft. Standardmäßig ist die Option <i>map_identity</i> eingestellt, die alle UIDs unberührt lässt.                                                                               |
| <i>map_static</i>                         | Statisches Mapping. Dieser Option wird als Argument der Name der Datei, welche die Mapping-Informationen enthält, übergeben, z.B. <i>map_static=/etc/nfs/client1</i> . Das Dateiformat finden Sie unmittelbar nach der Tabelle in einem einfachen Beispiel erläutert.                                                                                                                                                                                                   |
| <i>map_nis</i>                            | NIS-basiertes UID-/GID-Mapping. Wenn der Server beispielsweise auf die UID 520 trifft, wird zunächst der mit dieser UID assoziierte Benutzername festgestellt und dann der NIS-Server des NFS-Clients kontaktiert, um festzustellen, welche UID der Client mit dem Namen verbindet. Um hierzu in der Lage zu sein, muss der NFS-Server die NIS-Domain des Clients kennen. Dieser wird der Option <i>map_nis</i> als Argument übergeben, z.B. <i>map_nis=sybex.com</i> . |



## Besondere Netzwerkimplementierungen

### Mapping-Optionen (Forts.)

| Option                            | Bedeutung                                                                                                                                                                                                                                                                                                                                                                                |
|-----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>anonuid</i> und <i>anongid</i> | setzt UID und GID explizit auf den anonymen Account. Diese Option ist speziell bei PC/NFS-Clients nützlich, bei denen alle Anfragen als die eines einzigen Users interpretiert werden sollen. Beachten Sie hierzu auch den Eintrag zu <i>/export/home/zasi</i> im Beispiel weiter unten, der ein Mapping aller Anfragen auf die UID 150, welche die UID dieses Users sein soll, bewirkt. |

Tabelle 13.10

Eine Datei für statisches Mapping könnte wie folgt aussehen:

```
Mapping fuer client1:
remote local
uid 0-99 -- # squash dieser User
uid 100-500 1000 # map User 100-500
 # auf die uids 1000-1500
gid 0-49 -- # squash dieser Gruppen
gid 50-100 700 # map Gruppen 50-100
 # auf die gids 700-750
```

Folgendes Beispiel soll einige typische Einträge in */etc(exports* zeigen:

```
Beispieleinträge fuer /etc(exports
/ dbserver(rw) mate(rw,no_root_squash)
/projects proj*.sybex.com(rw)
/usr *.sybex.com(ro) @trusted(rw)
/export/home/zasi
pc001(rw,all_squash,anonuid=150,anongid=100)
/pub (ro,insecure,all_squash)
/pub/private (noaccess)
```

Die zweite Zeile erlaubt das Exportieren des gesamten Dateisystems (*/*) für die Hosts *dbserver* und *mate*; über den Schreibzugriff hinaus wird das UID-Squashing für den Host *mate* ausgeschaltet.

Der zweite und der dritte Eintrag zeigt Beispiele für Jokerzeichen in Hostnamen und für eine Netzwerkgruppe (den Eintrag *@trusted*).

Die fünfte Zeile zeigt den Eintrag für den PC/NFS-Client, der in *Tabelle 13.10* ange- sprochen wurde.



## Linux im Netzwerk

---



Zeile sechs exportiert das öffentliche FTP-Verzeichnis `/pub` an jeden Host der Welt, wobei Anfragen grundsätzlich unter dem Account `nobody` bearbeitet werden. Die Option `insecure` ermöglicht es darüber hinaus Clients mit NFS-Implementierungen, die keinen der reservierten Ports für ihre Anfragen nutzen, dennoch auf das Verzeichnis zuzugreifen.

Die letzte Zeile verwehrt allen NFS-Clients Zugang zum Verzeichnis `/pub/private`.

Nach dieser theoretischen Einführung soll das ursprüngliches Ziel, die Heimatbereiche von Usern per Auto-Mounter in das jeweilige lokale Dateisystem einzubinden, weiterverfolgt werden. Hierzu stellen Sie sich bitte folgendes Szenario vor:

- Sie verfügen über einen Server namens *spati*.
- Auf diesem haben Ihre User ihre Heimatbereiche; so lautet der Heimatbereich des Users *zasi* gemäß den oben gemachten Ausführungen `/export/home/zasi`.
- Egal, ob *zasi* sich nun an einem der drei Arbeitsplätze *client1*, *client2* oder *client3* oder an dem Server *spati* direkt einloggt: *spati:/export/home/zasi* soll immer per Auto-Mounter in das jeweilige lokale Dateisystem unter `/home` eingehängt werden, so dass er stets Zugriff auf seiner benutzerspezifischen Daten hat.

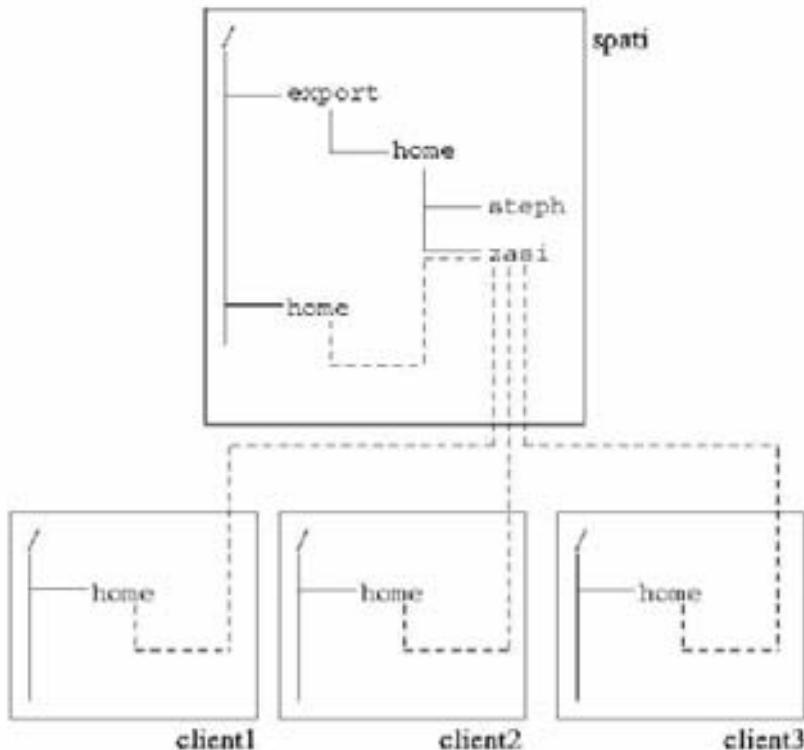
*Abbildung 13.9* soll dies illustrieren.

Sie gehen hierzu wie folgt vor:

- Ergänzen Sie als User *root* die Datei `/etc/exports` der „Heimatmaschine“ (des NFS-Servers also, der Teile seines Dateisystems exportieren soll) um mindestens den folgenden Eintrag (etwaige weitere notwendige Anpassungen sollten Sie aus dem zuvor Dargestellten ableiten können)  
`/export/home (rw)`
- Rufen Sie nun das Konfigurationstool *yast* auf und wählen dort den Menüeintrag *Administration des System > Konfigurationsdatei verändern*. „Blättern“ Sie mithilfe der Cursor-Tasten durch die Konfigurationsoptionen, bis Sie den Eintrag *START\_AUTOFS* finden, drücken Sie die Taste **F3**, und geben Sie als Wert für die Variable *yes* ein, wie die folgende Abbildung zeigt. Bestätigen Sie mit **Esc**, und verlassen Sie die Maske mit **F10**. Hierauf wird die Datei `/etc/rc.config`, welche die zentrale Konfigurationsdatei der SuSE-Linux-Distribution ist, aktualisiert, und einige Konfigurationsskripte werden abgearbeitet, was je nach Installationsumfang und Geschwindigkeit des Rechners zwischen einigen Sekunden und Minuten dauern kann. Verlassen Sie *yast* durch mehrmaliges Drücken der Taste **Esc**, bis Sie wieder am Eingabe-Prompt landen.



## Besondere Netzwerkimplementierungen



○ Abb. 13.9

Server *spati* und drei Clients



Legen Sie nun ein Verzeichnis */export/home/zasi* an:

```
mkdir -p /export/home/zasi
```



Legen Sie mithilfe des in *Kapitel 6 Userverwaltung* beschriebenen Befehls *useradd* einen User *zasi* an; verwenden Sie hierzu nicht *yast*, da dieses Problem mit der Zuweisung des „tatsächlichen“ physikalischen Home- und des automatisch zu mountenden Home-Verzeichnisses aufwerfen wird:

```
useradd -d /home/zasi -s /bin/bash zasi
```



Ändern Sie die User- und Gruppen-Rechteverhältnisse:

```
chown zasi /export/home/zasi
chgrp users /export/home/zasi
```



Vergeben Sie ein Passwort für den User *zasi*:

```
passwd zasi
```



## Linux im Netzwerk



Abb. 13.10

Festlegen der Umgebungsvariablen  
`START_AUTOFS` in YaST



Im nächsten Schritt wenden Sie sich der Konfiguration der Mapping-Dateien des Auto-Mounters zu, von denen als Erstes die Datei `/etc/auto.master` zu editieren ist (immer noch auf der „Heimatmaschine“).

### Die Datei `/etc/auto.master`

Die Datei `auto.master` wird auf die in ihr vorhandenen Einträge untersucht, wenn das Skript `/etc/rc.d/autofs` aufgerufen wird, um die für den Auto-Mounter notwendigen Mount-Points festzulegen. Jede Zeile dieser Datei beschreibt einen Mount-Point und gibt diejenige Datei an, die das Dateisystem beschreibt, auf das dieser Punkt gemountet werden soll.

#### Hinweis



Sie enthält (bis zu) drei Spalten, die durch eine beliebige Anzahl Leerzeichen oder Tabulatoren getrennt sein können; durch ein Doppelkreuz (#) eingeleitete Zeilen sind – wie gewohnt – Kommentarzeilen.

Bei der ersten Spalte handelt es sich um den Mount-Point, die zweite Spalte enthält den Namen der Datei (der so genannten „Map“), die bezüglich des Mount-Points Informationen darüber enthält, welches Dateisystem auf diesem eingehängt werden soll, und die dritte Spalte enthält optionale Ergänzungen, die sich auf alle Einträge dieser Map beziehen sollen.

Beachten Sie zum weiteren Format und zu den zur Verfügung stehenden Optionen auch den Eintrag zu `autofs` weiter unten in diesem Kapitel.

Beispiele:

|       |                |
|-------|----------------|
| /home | /etc/auto.home |
| /misc | /etc/auto.misc |



## Besondere Netzwerkimplementierungen



Diese Einträge definieren zwei Mountpoints `/home` und `/misc`; in jedem Fall, wenn auf diese zugegriffen wird, werden die entsprechenden Maps `/etc/auto.home` und `/etc/auto.misc` untersucht und das Mounting in Abhängigkeit des gefundenen Eintrags vorgenommen.

### Weitere Mapping-Dateien: `/etc/auto.home`

Im nächsten Schritt müssen Sie – um bei dem bisherigen Beispiel zu bleiben – eine Map `/etc/auto.home` anlegen, in der definiert ist, welcher User auf welchem Rechner wo seinen Heimatbereich hat. Diese soll in diesem Fall folgenden Inhalt haben:

```
zasi spati:/export/home/zasi
```

### Das Steuerungsskript `/etc/rc.d/autofs`

Bei diesem Programm handelt es sich um ein Shell-Skript für die Steuerung des Auto-Mounters in Abhängigkeit des Inhalts der Datei `/etc/auto.master`. Es muss nur dann aufgerufen werden, wenn Sie Änderungen an der „Master-Map“ vornehmen – Änderungen an weiteren Maps wie zum Beispiel `/etc/auto.home` werden automatisch vom Auto-Mounter registriert, so dass Sie das Skript in diesen Fällen nicht aufrufen müssen. Die Syntax des Aufrufs lautet:

```
/sbin/init.d/autofs start | stop | reload | restart |
status
```

Das Skript steuert die Operationen des Automount-Daemons Ihres Linux-Systems. Es wird normalerweise bereits während des Boot-Vorgangs mit dem Parameter `start` aufgerufen und beendet die Aktivitäten des Auto-Mounters beim Herunterfahren des Systems mithilfe des Parameters `stop`.

Manuell kann das Skript vom Systemadministrator aufgerufen werden, um den Auto-Mounter hiermit zu starten, ihn anzuhalten, ihn neu zu laden oder Statusinformationen zu ihm zu erhalten.

Wie zuvor beschrieben, wird beim Aufruf des Skripts die Datei `/etc/auto.master` daraufhin untersucht, auf welche Mount-Points Dateisysteme durch den Auto-Mounter in das lokale System eingehängt werden sollen. Für jeden dieser Mount-Points wird ein Auto-Mounter-Prozess mit seinen entsprechenden Parametern gestartet. Sie können diese für den Auto-Mounter aktiven Mount-Points durch Eingabe der Befehlszeile

```
/etc/rc.d/autofs status
abfragen, wie Abbildung 13.11 zeigt.
```



## Linux im Netzwerk

Abb. 13.11

Status des Auto-Mounters

```
brunetti:/etc/rc.d # ./autofs status
Checking for service autofs: OK
Configured Mount Points:

/usr/sbin/automount /misc: file /etc/auto.misc
/usr/sbin/automount /net program /etc/auto.net
/usr/sbin/automount /home file /etc/auto.home

Active Mount Points:

brunetti:/etc/rc.d #
```

Nach der Abarbeitung der Konfigurationsdatei *auto.master* wird nach einer NIS-Map gleichen Namens gesucht. Wenn eine solche existiert, wird sie ebenfalls abgearbeitet.

Die Eingabe der Befehlszeile

```
/sbin/init.d/autofs reload
```

stellt die aktuelle *auto.master*-Map laufenden Daemons gegenüber und killt diejenigen Daemons, deren ursprüngliche Einträge sich geändert haben, und startet dann Daemons für neue oder geänderte Einträge.

Mit Hilfe dieser Maßnahmen haben Sie nun Ihren Server *spati* so weit hergerichtet, dass beim Login des Users *zasi* auf diesem System der physikalische Bereich */export/home/zasi* auf den Mount-Point */home/zasi* in das lokale Dateisystem eingehängt wird.

Dies mag zunächst ein wenig witzlos erscheinen, könnte man argumentieren, denn der User könnte ja schließlich auch direkt seinen physikalischen Bereich nutzen, doch das Ziel soll ja sein, dass er – gleich auf welcher Maschine er arbeitet – immer die gleiche Umgebung vorfindet, was auch beispielsweise Zugriffspfade beinhaltet, so dass auch auf dem Serversystem ein Zugriff über */home/zasi* erfolgen können sollte.

Wenden Sie sich nun also den Client-Maschinen zu.

### Die Konfiguration der „Clients“

Bezüglich der Konfiguration der (Linux-) „Clients“ verfügen Sie mit dem weiter oben Dargestellten über sämtliche Werkzeuge/Informationen, die Sie benötigen, um das von geforderte Ziel, immer auf die benutzerspezifischen Informationen zugreifen zu können, erreichen zu können:

Die Relativierung des Begriffs „Client“ durch Anführungszeichen an dieser Stelle erfolgt aus dem Grunde, dass Sie in der Regel auf Ihren Linux-Maschinen lokale User „sitzen“ haben, deren lokaler Heimatbereich dann natürlich auch wieder per



## Besondere Netzwerkimplementierungen



Auto-Mounter eingehängt werden müsste, dieser Client also ein „normaler“ NFS-Server mit Automounting-Funktionen ist, so wie jede UNIX-Maschine den einen oder anderen Serverdienst zur Verfügung stellt.

In diesem Fall jedenfalls müssten Sie auf allen Linux-Maschinen innerhalb Ihres Netzwerks die zuvor beschriebenen Konfigurationsschritte ausführen. Wenn Ihre Linux-Clients tatsächlich als reine Klienten fungieren sollen, sind nur folgende Schritte auf ihnen auszuführen:

- Anlegen eines Mount-Points
- Anlegen der */etc/auto.master* und */etc/auto.home*
- Aktivieren des Auto-Mounters (mit *yast*)

Bezüglich des Zugriffs auf die die Home-Bereiche haben Sie somit Ihre Pflicht geleistet, doch sollen Sie sich auch an der Kür versuchen, indem Sie die bereits oben angedeutete Datei */etc/auto.misc* weiter ausbauen.

### Weitere Auto-Mounts mit */etc/auto.misc*

Weitere Konfigurationsmöglichkeiten des Auto-Mounters sollen anhand zweier Leistungsanforderungen wiederum praktisch dargestellt werden:

- Gegeben sei die Situation, dass innerhalb des Netzwerks von allen Workstations aus CD-ROMs genutzt werden sollen.

Statt nun einen eigenen zentralen CD-Server, bestehend aus 20 Laufwerken und eigener Netzwerkkarte, der über eine eigene IP-Adresse angesprochen werden kann, in das Netz zu integrieren (Kostenpunkt: ca. DM 15.000,00), nutzen Sie die Fähigkeit Ihres Servers, (Festplatten-) Ressourcen freizugeben, und die Fähigkeit Ihrer Clients, solche freigegebenen Ressourcen mithilfe des Auto-Mounters in ihre lokalen Dateisysteme einzuhängen.

Gehen Sie hierbei wie folgt vor:

- Schaffen Sie eine 20-GByte-Festplatte an (Kostenpunkt: ca. DM 300-400 (IDE) bzw. DM 800-900 (SCSI)), bauen Sie diese in Ihren Server *spati* ein und legen Sie auf dieser neuen Platte ein Linux-Dateisystem an. (*ext2fs* – beachten Sie hierzu auch die Ausführungen zum Befehl *mkfs* im Kapitel 3 *Linux-Dateisystem*. Wenn Ihnen dieser Vorgang zu kompliziert erscheint, kann das Anlegen ebenfalls über YaST im Menü *Einstellungen zur Installation > Festplatte partitionieren* erfolgen.) Diese Partition wird durch Editieren der Datei */etc/fstab* (beachten Sie hierzu auch im Kapitel 3 *Linux-Dateisystem* den Abschnitt „Erstellen eines Dateisystems“) oder mithilfe von YaST im Menü *Einstellungen zur Installation > Ziel-Partitionen > Dateisysteme festlegen* in das Dateisystem eingebunden, indem Sie es auf den Verzeichniseintrag */public-cd* mounten. Dieses Dateisystem kann die Inhalte von ca. 30 „randvollen“ CDs mit jeweils 650 MByte an Informationen aufnehmen.



## Linux im Netzwerk



Bezüglich der freizugebenden „CDs“ nutzen Sie nun „Techniken“, die Ihnen nach Studium der Kapitel über den Befehl *mount* und über das Brennen von CDs vom Prinzip her bereits geläufig sein sollten:

- Erstellen Sie mithilfe von *xdroast* Images Ihrer CDs, die Sie beispielsweise *cd1.raw* bis *cdX.raw* nennen, die Sie nach */public-cd* kopieren.

### Hinweis



Bei dem „X“ handelt es sich um einen variablen Wert, der von der Zahl der von Ihnen erstellten Images abhängig ist.

- Legen Sie nun die Verzeichnisse */export/home/cd1* bis */export/home/cdX* an.
- Ergänzen Sie die Datei */etc/fstab* um folgende Einträge. Durch das Festlegen des Dateisystemtyps auf den Wert *iso9660* und durch Ansprechen der entsprechenden Raw-Data-Dateien als so genanntes „loop-Device“ können Sie nun auf die Images zugreifen, als würde es sich dabei um CD-ROMs handeln:

```
/public-cd/cd1.raw /export/home/cd1 iso9660
rw,loop=/dev/loop0 0 0
/public-cd/cd2.raw /export/home/cd2 iso9660
rw,loop=/dev/loop1 0 0
...
/public-cd/cdX.raw /export/home/cdX iso9660
rw,loop=/dev/loopX 0 0
```

Geben Sie den Befehl *mount -a* ein, um einen (Re-)Mount aller Dateisysteme (auch der „CD-Dateisysteme“) zu bewirken.

Mithilfe dieser Maßnahmen haben Sie nun auf Ihrer Servermaschine CD-Images in das Dateisystem eingehängt, die für den weiteren Export bereitstehen.

Auf der Clientseite gehen Sie nun wie folgt vor:

- Ergänzen Sie die Datei */etc/auto.master* um den Eintrag (falls dieser noch nicht enthalten sein sollte).

```
/misc /etc/auto.misc
```

- Legen Sie eine neue Datei */etc/auto.misc* mit folgendem Inhalt an, oder ergänzen Sie die bereits existierende Datei um folgende Einträge:

```
cd1 spati:/export/home/cd1
cd2 spati:/export/home/cd2
...
cdX spati:/export/home/cdX
```

- Geben Sie folgenden Befehl ein, um die Änderungen der Map */etc/auto.master* wirksam werden zu lassen:

```
/etc/rc.d/autofs restart
```



## Die Konfiguration des SAMBA-Servers



Beim nächsten Zugriff auf eines der Verzeichnisse `/misc/cd1` bis `/misc/cdX` wird nun das entsprechende CD-Image vom Server *spati* in das lokale Dateisystem einge-hängt.

Abgesehen von der Kostensparnis bei Verwendung dieser Lösung gegenüber einem „echten“ CD-Server (es werden ca. 14.000,00 DM eingespart) hat sie einen weiteren Vorteil: Der Zugriff auf ein Loop-Device ist in jedem Fall schneller als selbst das schnellste SCSI-CD-ROM-Laufwerk – noch ein Grund also, sich für eine ähnliche wie die hier vorgestellte Form der Umsetzung zu entscheiden.

Diese Schritte müssten Sie dann jedoch auf jeder Client-Maschine durchführen, was bei einem Wachstum Ihres Netzwerks und der Zahl Ihrer User bezüglich der Pflege der Mapping-Dateien äußerst lästig wäre: Die manuelle Anpassung an die sich ändernden Begebenheiten wird mit der Zeit immer aufwändiger.

Solche zentralen Verwaltungsarbeiten zur Pflege von Rechner-, Benutzer- und Konfigurationsinformationen lassen sich jedoch durch Nutzung des *Network Information Service*, kurz *NIS*, erheblich vereinfachen. Für weitere Informationen zu diesem Netzwerk-Spezialthema konsultieren Sie die einschlägige Literatur.

## Die Konfiguration des SAMBA-Servers

Bei SAMBA handelt es sich um einen Datei- und Druckserver für UNIX-Systeme. Er besteht aus einer Reihe von Programmen, die das unter anderem in der Windows-Welt verwendete SMB-Protokoll implementiert, welches häufig auch als „LanManager“ oder „Netbios-Protokoll“ bezeichnet wird.

SAMBA besteht aus Komponenten, in deren Handhabung in diesem Abschnitt eingeführt werden soll. Wie in vielen anderen Bereichen ist auch hier lediglich eine einführende Darstellung möglich, die es Ihnen grundsätzlich erlaubt wird, Windows-Rechner oder weitere Linux-Rechner mithilfe des SMB-Protokolls als Klienten an Ihren Linux-Server anzubinden. Eine vollständige Darstellung würde den Rahmen dieses Buches mit Entschiedenheit sprengen (allein die Zahl der *man*-Zeilen zu den einzelnen SAMBA-Komponenten übersteigt 7.500 ...).

Das Buch *Linux Samba Administration* erscheint Mitte 2001 im SYBEX-Verlag.

Der Dämon *smbd*, den Sie mithilfe der weiter unten beschriebenen Datei `/etc/smb.conf` konfigurieren können, bietet den SMB-Klienten Datei- und Druckdienste an.

Das Hilfsprogramm *testparm* ermöglicht es Ihnen, diese Konfigurationsdatei `/etc/smb.conf` zu überprüfen.



Tipp

531



## Linux im Netzwerk

---



Das Programm *smbclient* implementiert einen einfachen, FTP-ähnlichen Client, der es Ihnen erlaubt, per SMB freigegebene Ressourcen anderer Rechner zu nutzen; hierbei kann es sich um einen weiteren Linux-Rechner handeln, aber auch um einen anderen Server, der SMB-Dienste zur Verfügung stellt, wie zum Beispiel einen Windows für Workgroups- oder einen Windows 95-Rechner.

Mithilfe des Hilfsprogramms *smbstatus* können Sie feststellen, wer die vom *smbd*-Server zur Verfügung gestellten Dienste gerade nutzt.

Der Dämon *nmbd* bietet eine NetBios-Nameserving-Unterstützung; er kann auch interaktiv dazu verwendet werden, weitere Namensdienste abzufragen.

### Die Datei */etc/smb.conf*

Die Datei */etc/smb.conf* dient der Konfiguration des Dämons *smbd*.

Sie enthält Konfigurationsinformationen, die zur Laufzeit eingelesen werden, Sie müssen den Dämon also nicht nach jeder Konfigurationsänderung neu starten.

#### Sektionen und Parameter

Die Konfigurationsinformationen liegen in Form von Sektionen („sections“) und Parametern („parameters“) zeilenbasiert vor, d.h., dass jede Zeile für sich einen Kommentar, einen Sektionsnamen oder einen Parameter enthält, wobei die Schreibweise dieser Einträge – anders als sonst von UNIX/Linux gewohnt – nicht nach Klein- und Großschreibung unterschieden wird.

Der Anfang einer Sektion wird durch ihren in eckige Klammern gestellten Namen kenntlich gemacht.

Eine Sektion enthält Konfigurationsparameter in der Form

`name = wert`

Verwendete *whitespace characters* (Leerzeichen, Tabulatoren etc.) am Zeilenanfang, innerhalb der Zeile oder am Zeilenende werden ignoriert, und nur das erste innerhalb der Parameterzeile verwendete Gleichheitszeichen ist signifikant. Wenn eine Zeile ausschließlich aus solchen Leerzeichen besteht, wird diese ebenfalls ignoriert.

Zeilen, die mit einem Semikolon beginnen, werden als Kommentarzeilen in der Abarbeitung übergangen.

Den UNIX-Konventionen entsprechend wird ein \ am Zeilenende als Zeilenumbruch interpretiert, der bewirkt, dass die Einträge der folgenden Zeile als der umbrochenen Zeile zugehörig interpretiert werden.

Bei den Gleichheitszeichen innerhalb der Parameter folgenden Werten handelt es sich entweder um Zeichenketten, um boolesche Werte, die als yes/no, 0/1 oder true/false angegeben werden dürfen, oder aber um nummerische Werte.



## Die Konfiguration des SAMBA-Servers



Zeichenketten müssen hierbei nicht in Anführungszeichen gesetzt werden. Ihre Schreibweise wird aber entgegen der Schreibweise der booleschen Werte nach Groß- und Kleinschreibung unterschieden.

### Dienste

Jede Sektion beschreibt einen Dienst. Der Sektionsname ist der Name des Dienstes, und die Parameter innerhalb der Sektion definieren die Eigenschaften des Dienstes.

Es gibt drei spezielle Sektionen, die weiter unten beschrieben werden: *[global]*, *[homes]* und *[printers]*; die folgenden Beschreibungen beziehen sich auf die allgemeinen Dienste.

Bei den Diensten kann es sich entweder um Speicherdienste handeln, bei denen es einem Client ermöglicht wird, die freigegebene Verzeichnisstruktur in seine lokale Verzeichnisstruktur „einzuhängen“ und auf den somit gewonnenen Massenspeicherplatz zuzugreifen, als handele es sich hierbei um ein natives Dateisystem, oder um Druckdienste, die vom Client dazu genutzt werden können, Drucker zu nutzen, die auf dem Server freigegeben wurden.

Dienste können als solche für „Gäste“ freigegeben werden, d.h., es wird kein Passwort benötigt, diese zu nutzen. Statt dessen wird ein spezielles „Gästekonto“ verwendet, um die Zugriffsrechte für diesen speziellen Fall zu regeln.

Andere als Gästedienste hingegen erfordern ein Passwort, um sie nutzen zu können, wobei der Username häufig vom Client direkt übergeben wird. In den Fällen, wo dies nicht geschieht und der Username nicht übergeben wird, können Sie eine Liste mit User-Namen anlegen, die mit dem angegebenen Passwort gegengeprüft werden soll, indem Sie die Option „user“ (siehe unten) in der Definition des Dienstes verwenden.

Beachten Sie hierbei, dass die Rechte, die ein User innerhalb des freigegebenen Dienstes besitzt, durch die Rechtesituation, die er lokal auf dem Host-Rechner hat, maskiert wird. Der SMB-Server kann diese allenfalls weiter einschränken, aber nicht erweitern!

Das folgende Beispiel zeigt einen Speicherdiensst:

```
[dokumentationen]
 path = /home/dokus
 writable = true
```

Der User besitzt einen Schreibzugriff auf den Pfad */home/dokus*, wobei der Zugriff über den Dienst *dokumentationen* erfolgt.

Folgendes Beispiel definiert einen Druckdienst. Der Parameter *path* legt fest, welche Druckerwarteschlange benutzt werden soll. Durch *read only = true* wird festgelegt, dass der einzige schreibende Zugriff auf die Warteschlange darin besteht, dass eine

## Linux im Netzwerk

---



Druck-Spool-Datei geöffnet, in diese hineingeschrieben und sie dann wieder geschlossen werden kann. Der Parameter `guest ok` signalisiert, dass ein Gästezugriff möglich ist.

```
[netzwerk-farbdrucker]
 path = /usr/spool/public
 read only = true
 printable = true
 guest ok
```

### Spezielle Sektionen

Folgende Sektionen besitzen besondere Bedeutung für die SAMBA-Dienste.

#### Die Sektion [*global*]

Die in dieser Sektion beschriebenen Parameter steuern das Verhalten des Servers als Ganzes oder sind Standardeinstellungen für Dienste, die nicht weiter parametrisiert sind. Beachten Sie hierzu auch den Abschnitt *Parameter* weiter unten.

#### Die Sektion [*homes*]

Wenn die Sektion [*homes*] in der Konfigurationsdatei enthalten ist, können Dienste, die Clients dynamisch an den Server binden, bereitgestellt werden. Hierbei wird wie folgt verfahren:

- Wenn eine Verbindung hergestellt wird, werden zunächst die vom Client angeforderten Dienste mit den in `/etc/smb.conf` definierten verglichen. Wird eine Übereinstimmung gefunden, wird dieser Dienst verwendet.
- Wird keine Übereinstimmung gefunden, wird der angefragte Dienstname als ein Username interpretiert, der in der lokalen Datei `/etc/passwd` gesucht wird.
- Wenn dieser Name gefunden und das Passwort korrekt eingegeben wurde, wird der Client dynamisch an den Server gebunden.

Die Sektion [*homes*] wird für den entsprechenden User „geklont“, wobei User-spezifische Modifikationen an den in dieser Sektion definierten Parametern erfolgen:

- Der Name des Dienstes wird von *homes* in den Namen des lokalisierten Users abgeändert.
- Wenn keine explizite Pfadangabe enthalten ist, wird der Pfad auf das Home-Verzeichnis des Users gesetzt.

Die Verwendung des *homes*-Dienstes ist ein einfaches Verfahren, einer großen Zahl Clients mit einem Minimum an Aufwand Zugriff zu ihren jeweiligen Home-Verzeichnissen zu geben.



## Die Konfiguration des SAMBA-Servers



In der Sektion `[homes]` können alle Parameter verwendet werden, die Sie auch in einer normalen Sektion spezifizieren können.

**Warnung:** Sie sollten es vermeiden, in der Sektion `[homes]` einen Gästezugang zuzulassen, da ansonsten alle Heimatverzeichnisse des Servers ohne Passworteingabe zugänglich wären. Sollte ein solcher Zugang dennoch aus irgendwelchen Gründen erwünscht sein, sollten Sie ihn zumindest als einen „Nur-Lesen“-Zugang beschränken, um unerwünschte Dateimanipulationen (seien diese nun böswillig, durch Unachtsamkeit oder durch Unbedarftheit bedingt) zu vermeiden.

### Die Sektion `[printers]`

Diese Sektion erlaubt einen allgemeinen Zugriff auf Drucker, die in der Datei `/etc/printcap` des Host-Rechners definiert sind.

Bei der Aufnahme einer Verbindung zum Host werden, wie oben für die Sektion `[homes]` beschrieben, die angeforderten Dienste geprüft und entweder verwendet oder, falls eine Sektion `[homes]` in `/etc/smb.conf` existiert, als Username interpretiert, der in der lokalen `/etc/passwd` gesucht wird.

Ist nun auch dieses nicht der Fall, wird der angegebene Dienst als Druckernname behandelt und ein entsprechender Eintrag in der Datei `/etc/printcap` des Host-Rechners gesucht.

Analog zum Vorgehen bei der Sektion `[homes]` wird ein neuer Dienst durch „Klonen“ der Sektion `[printers]` erzeugt, wenn ein Druckereintrag gefunden wurde.

Dieser Dienst wird noch wie folgt modifiziert:

- Der Name des gefundenen Druckers wird zum Namen des Dienstes.
- Falls kein Druckernname vorgegeben wurde, wird der Name des ermittelten Druckers zum Namen des Druckers.
- Wurde in `/etc/smb.conf` kein Gastzugang erlaubt und kein Benutzername angegeben, dann wird der Name des Druckers auch zum Benutzernamen.

Ein Beispieleintrag könnte wie folgt aussehen:

```
[printers]
 path = /usr/spool/public
 writable = no
 public = yes
 printable = yes
```

Achten Sie besonders auf den Parameter `printable`, er muss auf den Wert `yes` gesetzt werden, da ansonsten die Konfigurationsdatei vom Dämon `smbd` nicht gelesen wird!



## Linux im Netzwerk



Als Druckernamen können Sie im übrigen alle in der Datei `/etc/printcap` angegebenen Namen benutzen, einschließlich der Aliasnamen.

Falls Ihr Drucksystem nicht via `/etc/printcap` arbeitet, müssen Sie eine Pseudo-Printcap-Datei erstellen, die eine oder mehrere Zeilen der Form

`Aliasname| Aliasname| Aliasname ...`

enthält. Dabei sollte jeder Aliasname ein dem Drucksystem bekannter Druckername sein. Der Name der Pseudo-Printcap-Datei muss dann noch in der Sektion `[global]` als Name der Printcap-Datei eingetragen werden.

Mithilfe einer geeigneten Pseudo-Printcap-Datei lässt sich im übrigen der Zugriff auf einen Teil der verfügbaren Drucker einschränken.

## Variablensubstitution

Beim Setzen von Werten in der Datei `/etc/smb.conf` können eine Reihe von Variablensubstitutionen benutzt werden, sofern die Werte Zeichenketten sind. Soll zum Beispiel der Wert des Parameters `path` das Heimatverzeichnis des Benutzers sein, so kann man dies mit

`path = %H`

erreichen.

Neben der offensichtlich gesparten Tipparbeit hat diese Methode den wesentlichen Vorteil, dass der Pfad dynamisch an den Benutzer angepasst werden kann, für den der Dienst zur Verfügung stehen soll.

Typischerweise werden die Variablen zur Unterscheidung von gewöhnlichem Text durch ein Prozentzeichen eingeleitet.

Tabelle 13.11 listet die möglichen Variablen auf und erläutert kurz deren Bedeutung.

### Variablensubstitution in der Datei `/etc/smb.conf`



Variable wird ersetzt durch

|                 |                                                          |
|-----------------|----------------------------------------------------------|
| <code>%S</code> | Name des aktuellen Dienstes                              |
| <code>%P</code> | Wurzelverzeichnis des aktuellen Dienstes                 |
| <code>%u</code> | Benutzername des aktuellen Dienstes                      |
| <code>%g</code> | Name der Login-Gruppe des Benutzers ( <code>%u</code> )  |
| <code>%U</code> | Benutzername, der in der aktuellen Sitzung verwandt wird |
| <code>%G</code> | Name der Login-Gruppe des Benutzers ( <code>%U</code> )  |



## Die Konfiguration des SAMBA-Servers

### Variablenubstitution in der Datei `/etc/smb.conf` (Forts.)

Variable wird ersetzt durch

|                 |                                                                                                                                                       |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>%H</code> | Heimatverzeichnis des Benutzers ( <code>%u</code> )                                                                                                   |
| <code>%v</code> | die SAMBA-Version                                                                                                                                     |
| <code>%h</code> | Name des Host-Rechners, auf dem SAMBA läuft                                                                                                           |
| <code>%m</code> | Name des Clientrechner aus Sicht von NetBios                                                                                                          |
| <code>%L</code> | Name des Servers aus Sicht von NetBios. Erlaubt eine Änderung der Konfiguration in Abhängigkeit vom Aufruf des Client.                                |
| <code>%M</code> | Internetname des Clientrechners                                                                                                                       |
| <code>%N</code> | Name des Heimatverzeichnis des NIS-Servers (auf NIS wird in diesem Buch nicht weiter eingegangen)                                                     |
| <code>%p</code> | Pfad für das Heimatverzeichnis des Dienstes                                                                                                           |
| <code>%R</code> | verwendete Protokollebene; ab SAMBA Version 1.9.18 stehen zur Verfügung CORE, COREPLUS, LANMAN1, LANMAN2 oder NT1                                     |
| <code>%d</code> | Die <code>pid</code> (Prozess-id) des aktuellen Serverprozesses                                                                                       |
| <code>%a</code> | Typ des entfernten Rechners – erkannt werden zur Zeit in der Regel WfW, WinNT und Win95, andere Rechnerarchitekturen werden als UNKNOWN klassifiziert |
| <code>%l</code> | die IP Adresse des Clientrechners                                                                                                                     |
| <code>%T</code> | aktuelles Datum und aktuelle Zeit                                                                                                                     |

Tabelle 13.11

## Parameter

Über Parameter wird das Verhalten der Dienste gesteuert. In diesem Abschnitt soll mithilfe eines Beispiels eine Auswahl dieser Parameter vorgestellt werden, um Ihnen einen ersten Überblick über die Konfigurationsmöglichkeiten von SAMBA zu geben. Eine vollständige Liste der insgesamt fast 200 Parameter können Sie der *man page* zu `smb.conf` entnehmen.

Einige dieser Parameter sind ausschließlich für die Sektion `[global]` verfügbar (wie etwa `security`).

Parameter, die für die einzelne Dienste vorgesehen sind, lassen sich umgekehrt sehr wohl in der Sektion `[global]` verwenden. Man legt dadurch einen Standardwert für alle Dienste fest.



## Linux im Netzwerk

---



Für eine Reihe Parameter gibt es verschiedene Namen (Synonyme). Sie können in diesem Fall wahlweise den von Ihnen gewünschten Namen benutzen. So bedeuten zum Beispiel die Einträge

`create mode =0755`

und

`create mask = 0755`

in `/etc/smb.conf` dasselbe.

Folgendes Listing zeigt eine Beispielkonfiguration eines SAMBA-Servers; anhand dieses Beispiels werden die verwendeten Parameter vorgestellt.

Beispiel:

```
[global]
 security = user
 workgroup = arbeitsgruppe
 printcap name = /etc/printcap
 printing = bsd
 auto services = homes
 guest account = nobody
 wins support = false
 keep alive = 30
 load printers = yes
 mangled names = yes
 mangling char = ^

[homes]
 create mode = 0750
 writable = yes
 comment = home directory
 path = /export/home
 allow hosts = 192.6.1./255.255.255.0

[printers]
 comment = Alle Drucker
 printable = yes
 public = no
 read only = yes
 create mode = 0700
 directory = /tmp
```



## Die Konfiguration des SAMBA-Servers



### Die Sektion **[global]** mit Ihren Parametern

Wie zuvor beschrieben, definiert die Sektion **[global]** das generelle Verhalten des Servers.

Für dieses Beispiel bedeuten die im Einzelnen (Parameter, die ausschließlich in dieser Sektion verwendbar sind, sind mit (G) gekennzeichnet).

#### Parameter der Sektion **[global]**

| Parameter                              | Bedeutung                                                                                                                                                                                                                                                                                                                                                                                   |
|----------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>security (G)</i>                    | Mithilfe eines so genannten „security mode bit“ regelt dieser Parameter, ob und wie User- und Passwortinformationen zwischen Client und Server ausgetauscht werden. Die Standardeinstellung lautet <i>security = share</i> , hauptsächlich, weil es ursprünglich nur diese Einstellungsmöglichkeit gegeben hat. Alternativwerte sind <i>security = user</i> oder <i>security = server</i> . |
|                                        | Wenn auf den Windows-Clients die gleichen User-Namen verwendet werden, wie auf dem Linux-Server, sollten Sie die Einstellung <i>security = user</i> wählen.                                                                                                                                                                                                                                 |
|                                        | Wenn Sie auf den Client-Maschinen überwiegend User-Namen verwenden, die der Linux-Maschine nicht bekannt sind, sollten Sie die Einstellung <i>security = share</i> verwenden.                                                                                                                                                                                                               |
|                                        | Wenn Sie die Einstellung <i>security = server</i> wählen, versucht SAMBA, den User-Namen und das Passwort mithilfe eines anderen SMB-Servers zu validieren; dessen Name wird mittels des globalen Parameters <i>password server = servername</i> festgelegt. Wenn diese Validierung nicht möglich ist, verwendet SAMBA automatisch als Nächstes die Einstellung <i>security = user</i>      |
| Hier: <i>security = user</i>           |                                                                                                                                                                                                                                                                                                                                                                                             |
| <i>workgroup (G)</i>                   | Dieser Parameter steuert, in welcher Arbeitsgruppe/welchen Arbeitsgruppen der Server auftaucht, wenn er von Clients angeprochen wird.                                                                                                                                                                                                                                                       |
| Hier: <i>workgroup = arbeitsgruppe</i> |                                                                                                                                                                                                                                                                                                                                                                                             |

## Linux im Netzwerk



### Parameter der Sektion [global] (Forts.)

| Parameter                                  | Bedeutung                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|--------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>printcap name (G)</i>                   | der Name der zu verwendenden Printcap-Datei                                                                                                                                                                                                                                                                                                                                                                                                       |
| Hier: <i>printcap name = /etc/printcap</i> |                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <i>printing</i>                            | Dieser Parameter steuert, wie Statusinformationen des Druckers Ihres Systems interpretiert werden und beeinflusst die Standardeinstellungen des Druckbefehls ( <i>lpr</i> ) und des Befehls zum Löschen von Druckaufträgen ( <i>lprm</i> ). Zur Zeit werden sechs Druckmethoden unterschieden: <i>printing = bsd</i> , <i>printing = sysv</i> , <i>printing = hpx</i> , <i>printing = aix</i> , <i>printing = qnx</i> und <i>printing = plp</i> . |
| Hier: <i>printing = bsd</i>                |                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <i>auto services (G)</i>                   | eine Liste der automatisch dem Client anzugegenden Dienste                                                                                                                                                                                                                                                                                                                                                                                        |
| Hier: <i>auto services = homes</i>         |                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <i>guest account</i>                       | Name des Users, der „Gästedienste“ nutzen darf. Hierbei handelt es sich um Dienste, für deren Nutzung man keinen speziellen User-Account auf dem Server benötigt. Üblicherweise wird dieser Guest-User ebenfalls in der Datei <i>/etc/passwd</i> definiert.                                                                                                                                                                                       |
| Hier: <i>guest account = nobody</i>        |                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <i>wins support (G)</i>                    | Dieser boolesche Wert legt fest, ob der <i>nmbd</i> -Prozess als WINS-Server fungieren soll. Sie sollten diesen Parameter nur dann auf <i>true</i> setzen, wenn Sie mehrere Subnetze in Ihrem Netzwerk haben und einen speziellen <i>nmbd</i> -Server als WINS-Server einsetzen wollen. Sie dürfen diesen Wert niemals auf mehr als einer Maschine auf <i>true</i> setzen!                                                                        |
| Hier: <i>wins support = false</i>          |                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <i>keep alive (G)</i>                      | Dieser ganzzahlige Wert legt die Anzahl an Sekunden fest, die vergehen sollen, bis der Server feststellen soll, ob ein Client noch Dienste anfordert                                                                                                                                                                                                                                                                                              |
| Hier: <i>keep alive = 30</i>               |                                                                                                                                                                                                                                                                                                                                                                                                                                                   |



## Die Konfiguration des SAMBA-Servers



### Parameter der Sektion [global] (Forts.)

| Parameter                        | Bedeutung                                                                                        |
|----------------------------------|--------------------------------------------------------------------------------------------------|
| <i>load printers (G)</i>         | eine boolesche Variable, die steuert, ob alle Drucker in die Browser-Liste geladen werden sollen |
| Hier: <i>load printers = yes</i> |                                                                                                  |
| <i>mangled names</i>             | siehe letzter Abschnitt ( <i>Anpassung von Dateinamen</i> )                                      |
| <i>mangling char</i>             | siehe letzter Abschnitt ( <i>Anpassung von Dateinamen</i> )                                      |

Tabelle 13.12



### Die Sektion [homes] mit ihren Parametern

Wie bereits zuvor beschrieben, definiert die Sektion [homes] Heimatverzeichnisse.

### Die Sektion [homes] mit ihren Parametern



| Parameter          | Bedeutung                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>create mode</i> | Synonym: <i>create mask</i><br><br>Zur Festlegung der Rechtemaske einer zu erstellenden Datei. In einem ersten Schritt werden die Rechte aus den DOS-Attributen abgeleitet (etwa das Recht <i>w</i> aus dem Attribut <i>read/write</i> ). Die entstandene Rechtetabelle wird mit der Rechtemaske von <i>create mode</i> verknüpft. Ist ein Recht in beiden Fällen gesetzt, wird es übernommen, ansonsten wird es nicht erteilt. Durch <i>create mode</i> werden also die Rechte eingeschränkt. Die Maske ist analog zu <i>chmod</i> zu verstehen, nicht wie bei <i>umask</i> ! Angenommen, dass nach dem ersten Schritt die resultierende Rechtetabelle <i>rwxrwxrwx</i> ist, dann entzieht die Rechtemaske in diesem Beispiel (0750) der Gruppe das Recht <i>w</i> und allen anderen sämtliche Rechte.<br><br>Ergebnis: <i>rwxr-x--</i><br><br>Hier: <i>create mode = 0750</i> |
| <i>writable</i>    | Synonym: <i>write ok</i><br><br>umgekehrtes Synonym: <i>read only</i>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|                    | legt fest, ob Benutzer im Verzeichnis des Dienstes neue Dateien anlegen oder vorhandene Dateien ändern können. So legt vorliegendes Beispiel fest, dass im Verzeichnis neue Dateien angelegt und vorhandene Dateien verändert werden dürfen. Synonyme Einträge wären                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |

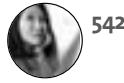


## Linux im Netzwerk

### Die Sektion [*homes*] mit ihren Parametern (Forts.)

| Parameter          | Bedeutung                                                                                                                                                                                                                                                                                                                                                                                            |
|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                    | <i>write ok = yes</i>                                                                                                                                                                                                                                                                                                                                                                                |
|                    | <i>read only = no</i>                                                                                                                                                                                                                                                                                                                                                                                |
|                    | Beachten Sie auch den Parameter <i>printable</i> , der im folgenden Abschnitt beschrieben wird                                                                                                                                                                                                                                                                                                       |
|                    | Hier: <i>writable = yes</i>                                                                                                                                                                                                                                                                                                                                                                          |
| <i>comment</i>     | Dies ist ein Text, der beim Client neben der freigegebenen Ressourcen angezeigt wird, wenn an diesem eine entsprechende Liste angezeigt werden soll                                                                                                                                                                                                                                                  |
|                    | Hier: <i>comment = home directory</i>                                                                                                                                                                                                                                                                                                                                                                |
| <i>path</i>        | Synonym: <i>directory</i>                                                                                                                                                                                                                                                                                                                                                                            |
|                    | legt den Pfad fest, auf den ein Benutzer des Dienstes Zugriff erhält. Hier ist eine Variablen-Substitution mit %u (Benutzername) und %m (Name des Client-Rechners) von besonderem Nutzen, da Sie so jedem Benutzer eine Pseudo-Heimatverzeichnis geben können. Besonderheiten bei Druckdiensten finden Sie im folgenden Abschnitt – ebenfalls unter <i>path</i> .                                    |
|                    | Hier: <i>path = /export/home</i>                                                                                                                                                                                                                                                                                                                                                                     |
| <i>allow hosts</i> | Synonym: <i>hosts allow</i>                                                                                                                                                                                                                                                                                                                                                                          |
|                    | umgekehrte Synonyme: <i>deny hosts, hosts deny</i>                                                                                                                                                                                                                                                                                                                                                   |
|                    | legt fest, von welchen Host-Rechnern aus dieser Dienst in Anspruch genommen werden kann. Mehrere Hostnamen werden durch Komma getrennt. Es können die Namen der Host-Rechner, die IP-Nummern oder Kombinationen von Netzwerknummer/Netzwerkmaske (wie in diesem Beispiel) benutzt werden. Mit Hilfe des Schlüsselwortes <i>EXCEPT</i> können einzelne Hosts ausgenommen werden. Zum Beispiel erlaubt |
|                    | <i>allow hosts = 194.195.155. \ except 194.195.155.105</i>                                                                                                                                                                                                                                                                                                                                           |
|                    | allen Rechnern des Subnetzes 194.195.155 den Zugriff, ausgenommen den Rechnern mit der IP-Nummer 194.195.155.105.                                                                                                                                                                                                                                                                                    |
|                    | Hier: <i>allow hosts = \192.6.1./255.255.255.0</i>                                                                                                                                                                                                                                                                                                                                                   |

Tabelle 13.13



## Die Konfiguration des SAMBA-Servers



### Die Sektion [*printers*] mit ihren Parametern

Die Sektion [*printers*] stellt SMB-Druckdienste bereit.

#### Parameter der Sektion [*printers*]

| Parameter          | Bedeutung                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>comment</i>     | siehe Sektion [ <i>homes</i> ]                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <i>printable</i>   | Synonym: <i>print ok</i> . Wenn dieser Parameter auf <i>yes</i> gesetzt ist, können Clients Spool-Dateien im spezifizierten Verzeichnis öffnen, schreiben und diese übergeben. Beachten Sie, dass ein „ <i>printable</i> -Dienst“ es grundsätzlich erlaubt, mithilfe von Spool-Diensten in den angegebenen Pfad zu schreiben. Insfern „überschreibt“ <i>printable = yes</i> den Parameter <i>writable = no</i> (siehe oben). Hier: <i>printable = yes</i> |
| <i>public</i>      | Synonym: <i>guest ok</i> . Wenn dieser Parameter auf <i>yes</i> gesetzt ist, ist kein Passwort für die Nutzung dieses Dienstes erforderlich. Die Rechte, die innerhalb dieses Dienstes zur Verfügung stehen, entsprechen denen des Gast-Accounts. Hier: <i>public = no</i>                                                                                                                                                                                |
| <i>read only</i>   | siehe <i>writable</i> in der Sektion [ <i>homes</i> ]. Hier: <i>read only = no</i>                                                                                                                                                                                                                                                                                                                                                                        |
| <i>create mode</i> | siehe Sektion [ <i>homes</i> ]. Hier: <i>create mode = 0700</i>                                                                                                                                                                                                                                                                                                                                                                                           |
| <i>directory</i>   | siehe <i>path</i> in der Sektion [ <i>homes</i> ]. Der hier definierte Pfad dient als Spool-Bereich für diesen SAMBA-Druckdienst; der Pfad sollte für alle User das Schreibrecht beinhalten, und das Sticky-Bit sollte gesetzt sein. Hier: <i>directory = /tmp</i>                                                                                                                                                                                        |

Tabelle 13.14



### Anpassung von Dateinamen

Unter SAMBA ist eine Anpassung von Dateinamen auf das 8.3-Format möglich, so dass Clients, die unter DOS oder Windows 3.x laufen, Dateien benutzen können, deren Namen dieser Konvention nicht genügt.

Die Anpassung wird generell dadurch ermöglicht, dass der Wert des Parameters *mangled names* auf *yes* gesetzt wird. Dies ist übrigens der voreingestellte Wert.

Die Anpassung erfolgt ähnlich der aus Windows 95 bekannten Methode. Die ersten (bis zu) fünf Zeichen des Dateinamens werden in Großbuchstaben umgewandelt und bilden so den Beginn des angepassten Namens. Dies betrifft Zeichen, die vor dem letzten Punkt im ursprünglichen Namen stehen, sofern sie gemäß den DOS-Konventionen gültige Zeichen sind.



## Linux im Netzwerk

---



An den so ermittelten Namen wird die Tilde (-) angehängt sowie ein zwei Zeichen langes Kürzel, das aus dem ursprünglichen Dateinamen mithilfe eines an dieser Stelle nicht weiter darzustelenden Algorithmus gebildet wird.

Gibt es eine Dateinamenserweiterung und ist diese länger als drei Zeichen, so wird sie auf drei Zeichen gekürzt. Die Erweiterung wird ebenfalls in Großbuchstaben umgewandelt.

Beginnt der originale Dateiname mit einem Punkt, so wird diese Datei dem DOS-Client als Datei mit dem Attribut *Hidden* dargestellt. Der angepasste Dateiname erhält als Endung drei Unterstriche.

Das folgende Beispiel stellt zwei lange Dateinamen und ihre Anpassung gegenüber:

| Original    | Anpassung  |
|-------------|------------|
| ei der daus | EIDER~VP   |
| smb.conf    | SMB~WX.CON |

Anstelle der Tilde lässt sich jedes andere Zeichen als Anpassungszeichen definieren. Dies ist nützlich, da einige Anwendungen die Tilde im Dateinamen nicht mögen! Die Verwendung des Zeichens ^ veranlassen Sie beispielsweise durch den Eintrag von

```
mangling char = ^
in die Datei /etc/smb.conf.
```

Wollen Sie gezielt einige Dateinamenerweiterungen anpassen, können Sie dies mit Hilfe von *mangling maps* tun. Die Zeile

```
mangling map = (*.html *.htm)
```

sorgt dafür, dass alle Dateinamen mit der Erweiterung *html* angepasst werden zu Dateien mit der Endung *htm*.

Durch eine Reihe weiterer Parameter kann die Anpassung noch verfeinert werden. Alle vorgestellten Parameter sind im übrigen für einzelne Dienste verfügbar.

Zu erwähnen bleibt noch, dass Dateinamen, die nicht dem 8.3-Format genügen, nicht angezeigt werden, wenn der Parameter *mangled names* auf *no* gesetzt wird!

### **testparm**

Mit Hilfe des Diagnoseprogramms */usr/bin/testparm* können Sie die von Ihnen erstellte Konfigurationsdatei */etc/smb.conf* auf syntaktische Korrektheit überprüfen lassen.

Dies ist auch netzwerkweit möglich, wie folgende Syntax zeigt:

## Die Konfiguration des SAMBA-Servers



### Syntax:

```
testparm [konfigurationsdatei [hostname hostIP]]
```

Wenn Hostname und dessen IP-Adresse angegeben werden, prüft *testparm* zusätzlich, ob der angegebene Host tatsächlich Zugriff zu den genannten Diensten hat.

Folgendes Listing zeigt eine Beispielausgabe für die Prüfung der */etc/smb.conf* mit einem Syntaxfehler:

```
Load smb config files from /etc/smb.conf
Processing section "[homes]"
Badly formed boolean in configuration file: "ys".
Processing section "[printers]"
Processing section "[pictures]"
Loaded services file OK.
Press enter to see a dump of your service definitions
```

Der letzten Zeile dieser Bildschirmausgabe können Sie entnehmen, dass Sie nach dem Drücken der -Taste Detailinformationen erhalten. Es handelt sich bei diesen um die Werte sämtlicher Parameter. Auf diesem Wege können Sie sich auch die Standardwerte der Parameter anzeigen lassen.

Für Sie ist an dieser Stelle nur Ausgabezeile 3 von Interesse, da hier ein Syntaxfehler gefunden wurde. Der Wert der gesetzten booleschen Variablen soll natürlich nicht „ys“, sondern „yes“ lauten...

### **smbclient**

Im Regelfall wird man SMB-Dienste nicht dafür verwenden, Linux-Rechner als Client an andere Linux-Rechner anzubinden, sondern andere Dienste wie NFS vorziehen.

Die Anbindung eines Linux-Rechners an einen Windows-Server sollte ebenfalls nicht Gegenstand eines Linux-Buches sein. ;-)

Aus diesem Grunde wird auf eine Darstellung des Dienstprogramms *smbclient* verzichtet.

Umgekehrt macht eine Anbindung sicherlich Sinn. Um nun die durch SAMBA zur Verfügung gestellten Ressourcen von einem Windows-Rechner aus nutzen zu können, müssen Sie diesen entsprechend als SMB-Client konfigurieren. Dies finden Sie im folgenden Abschnitt beschrieben.





### Einbindung eines SAMBA-Clients – a case study

Im folgenden Abschnitt soll eine einfache SAMBA-Anbindung dargestellt werden, wobei Sie sich eines Windows-95-Clients bedienen. Eine Übertragung der hier dargestellten Schritte auf Win98/NT-Systeme sollte problemlos möglich sein. :-)

Öffnen Sie die *Systemsteuerung* und wählen Sie dort den Eintrag *Netzwerk*.

Fügen Sie zunächst Ihre Netzwerkkarte hinzu, wenn dies nicht bereits schon geschehen ist, und wählen Sie dann mit *Hinzufügen > Protokoll > Hersteller: Microsoft* als Protokoll *NetBeui*.



Abb. 13.12

Auswahl des *NetBeui*-Protokolls

Im selben Dialogfenster müssen Sie das Netzwerkprotokoll *TCP/IP* wählen, wenn dieses nicht bereits zuvor geschehen ist.

Konfigurieren Sie das TCP/IP-Protokoll durch Doppelklick oder durch Anklicken der Schaltfläche *Eigenschaften*, und vergeben Sie dem Rechner eine IP-Adresse (beachten Sie hierzu auch den Abschnitt *IP-Adressen* am Anfang des Kapitels).



## Einbindung eines SAMBA-Clients – a case study



Abb. 13.13

Auswahl des TCP/IP-Protokolls



Abb. 13.14

Konfiguration des TCP/IP-Protokolls



## Linux im Netzwerk



Im Registerblatt *DNS-Konfiguration* des Dialogfensters *Eigenschaften fP* sollten Sie dem Rechner noch einen symbolischen Namen vergeben, falls dies nicht bereits zuvor geschehen ist.



Abb. 13.15

Das Registerblatt *DNS-Konfiguration* des Dialogfensters *Eigenschaften fP* für TCP/IP

Fügen Sie nun mit *Hinzufügen > Client > Hersteller: Microsoft* den *Client-Netzwerke* hinzu.

### Hinweis



Beachten Sie, dass der hier verwendete Arbeitsgruppenname natürlich auch dem SMB-Server bekannt sein muss (Parameter *workgroup*).

Ein sehr wichtiger Punkt ist die Modifikation der Eintragungen des Registerblatts *Identifikation* im Dialogfenster *Netzwerk*. Legen Sie hier den Computernamen und den Namen der Arbeitsgruppe, welcher der Computer angehört, fest.



## Einbindung eines SAMBA-Clients – a case study



Abb. 13.16

Auswahl des Clients für Microsoft-Netzwerke



Abb. 13.17

Festlegung des Computer- und Arbeitsgruppennamens

## Linux im Netzwerk



Nach Festlegung dieser Einstellungen werden Sie aufgefordert, Ihren PC neu zu booten.

Nach dem Neustart des Rechners sollten Sie (wenn Sie vorher keine Netzwerkfunktionen konfiguriert haben) ein neues Icon *Netzwerkumgebung* auf Ihrem Desktop liegen haben.



Abb. 13.18

Das Icon *Netzwerkumgebung*

Wenn Sie dieses doppelklicken, werden Ihnen alle im Windows-Netzwerk vorhandenen Rechner angezeigt, die ihre Ressourcen freigegeben haben. In diesem Beispiel ist dies nur der Server.

Abb. 13.19

Im Netzwerk verfügbare Rechner





### Der MARS-Server

Der Einzug der PCs in die Kleinbetriebe lief in den frühen 80er Jahren meistens eher chaotisch als gut geplant: Zuerst erhielt die Buchhaltung zur Datenerfassung einen PC, dann wurde die Schreibmaschine in die Ecke gestellt, und das Sekretariat erhielt den zweiten Personal Computer. Für die Herstellung oder die Produktion wurde dann der dritte Rechner angeschafft, um mit Multiplan o. Ä. kalkulieren zu können. Inzwischen hatten die Grafiker sowieso schon ihren Macintosh erhalten, der bereits vor 20 Jahren Standard im grafischen Gewerbe war.

Und irgendwann wurden die Probleme offensichtlich: Die Daten wurden auf den verschiedenen Rechnern benötigt. Aus einer Kalkulation entstand ein Angebot, aus einem Angebot wurde eine Rechnung und aus einer Rechnung ein Buchhaltungs-Datensatz. Die Tabellen aus Multiplan konnten auch gut im Prospekt verwendet werden ... So wanderten die Daten per Diskette von PC zu PC – man hatte das berüchtigte „Frisbee- oder Turnschuh-Netzwerk“. Und mit den Kollegen, die am Mac arbeiteten, konnte man wegen inkompatibler Diskettenformate überhaupt keine Daten tauschen. Die ersten Versuche, ein Billig-Netz über serielle Leitungen zu installieren, wurden recht schnell aufgegeben: man musste zwar nicht mehr von seinem Arbeitsplatz aufstehen, um Daten zu transportieren, aber der Datentransfer war kaum schneller als per Diskette: für 1 MByte über 115.000 Baud benötigte man mehr als eine Minute. Langsam wuchs die Erkenntnis: Ein Fileserver muss her.

Mitte der 80er Jahre gab es hauptsächlich zwei verschiedene Möglichkeiten für ein stabiles Serverbetriebssystem: zum einen UNIX, das aber als „kompliziert“ verschrien war, und deshalb mindestens einen Menschen in der jeweiligen Firma oder Abteilung benötigte, der sich damit auskannte. Die andere Alternative bot ab 1983 die Firma Novell: ein robustes und pflegeleichtes Serverbetriebssystem, das anfänglich sogar auf einem 286er lief. Der Server wurde in eine Ecke gestellt, die Büros wurden mit Ethernet verkabelt, der Server wurde eingerichtet und musste nur ausgeschaltet werden, wenn neue Hardware eingebaut wurde. Novell fand eine sehr große Verbreitung in kleinen und mittleren Betrieben und wurde so zum Quasi-Standard der Server-Welt. Die Version 3.12 war derartig stabil und ausgereift, dass kaum jemand ein Update wollte. So verkauft Novell inzwischen die Version 5.0, aber immer noch läuft zur vollen Zufriedenheit die Version 3.12 auf vielen Servern (für das Jahr-2000- oder Y2K-Problem wurden entsprechende Patches zur Verfügung gestellt). Und hier treten die Nachteile von Novell zutage. Erstens lässt Novell sich seine absturzfreien Dienste teuer bezahlen: Noch heute muss man für die Fünf-Benutzer-Lizenz der Version 3.12 knappe 1000,- DM bezahlen, für mehr User entsprechend mehr. Zum zweiten war Novell immer recht anspruchsvoll in Bezug auf die Hardware. Prinzipiell werden SCSI-Platten gefordert, was zwar der Geschwindigkeit zugute kommt, aber Löcher in die Kasse reisst. Weiterhin werden die Verzeichnisse der Festplatten im RAM gehalten: Um das Betriebssystem plus 4 Gigabyte an Festplatten zu verwalten, benötigt Novell mindestens 16 MByte RAM. In den Zeiten rapide fallender Hardwarepreise ist dieses zwar kein gravierendes Argument mehr, dieses mag aber auch für die Entwicklung des MARS-Servers



## Linux im Netzwerk

---



gesprochen haben. Außerdem war zusätzliche Hardware wie CD-ROM-Laufwerke oder ISDN-Karten nur mit Klimmzügen und zusätzlichen Programm-Modulen in das Novell-System zu integrieren.

Novell brachte sein eigenes Netz-Protokoll mit: IPX. Dieses läuft zwar auch auf einem Standard-Ethernet wie das TCP/IP der UNIX-Welt. Auch stören die beiden Protokolle einander nicht, aber wie so vieles in den 80er Jahren (und vorher): Firmen erzeugten Quasi-Normen, um die Märkte so gegenüber Konkurrenten abzuschotten.

In vielen Firmen stehen nun also zur Zufriedenheit aller Beteiligten ein oder mehrere Novell-Server, doch die LANs in den Betrieben haben ein eindeutiges Merkmal: Sie wachsen. Hier greift der MARS-Server ein. Der MARS-Server emuliert nun unter Linux all die Vorzüge von Novell 3.xx, ohne unbedingt alle seine historisch bedingten Nachteile zu übernehmen. Der MARS-Server ist kostenlos, er hat keine Beschränkung der Benutzerzahl, er läuft auch auf sparsam ausgestatteten Rechnern: Wie bekannt ist, läuft Linux klaglos mit 8 MByte RAM und IDE-Festplatten. CD-ROM-Laufwerke können problemlos eingebunden werden, und Festplatten fast jeder Größe können ohne zusätzliche Anforderungen an RAM angesprochen werden.

MARS richtet sich an alle Benutzer, die schon in einem funktionierenden Novell-Netz arbeiten und kostengünstig einen oder mehrere weitere Server ins lokale Netz integrieren wollen.

MARS wurde von **MARTin Stover** entwickelt und besitzt inzwischen eine große Gemeinde. MARS befindet sich auf der zweiten CD, welche diesem Buch beiliegt. Wer Interesse an der Weiterentwicklung hat, der findet die neuesten Versionen unter [http://www.compu-art.de/mars\\_nwe](http://www.compu-art.de/mars_nwe).

### Installation

MARS liegt auf der zweiten CD dieses Buches im Verzeichnis *MARS* als *mars\_nwe-0.99.pl13.tgz*, d.h. Version 0.99, Patchlevel 13 als komprimierte Tar-Datei. Als Erstes erzeuge man sich ein temporäres Verzeichnis zu Installation, etwa */mars*. Dann kopiere man die Tar-Datei von der CD-ROM in dieses Verzeichnis. Nachdem man in dieses temporäre Verzeichnis gewechselt hat, wird MARS ausgepackt:

```
tar -xvzf mars_nwe-0.99.pl13.tgz
```

In dem temporären Verzeichnis befindet sich jetzt das Unterverzeichnis *mars\_nwe*. Man wechsle dorthin und gebe ein:

```
make
```

MARS meldet, dass die Datei *config.h* erstellt wurde und man sie evtl. seinen Bedürfnissen anpassen möge. Dieses ist z.B. der Fall, wenn man mehr als 50 Benutzer – die Standardeinstellung – an seinem Server zulassen möchte. Bei der Erstinstallation ist dieses nicht der Fall. Ein weiteres Mal wird eingegeben:



## Der MARS-Server



`make`

Der Vorgang dauert etwas länger, nun werden die eigentlichen Binär-Dateien erzeugt. Mit folgendem Befehl werden jetzt die notwendigen Dateien installiert, wobei man als *root* eingeloggt sein muss:

`make install`

Nun muss „nur“ noch die Konfigurationsdatei bearbeitet werden. MARS wird von der Datei *nwserv.conf* gesteuert, die im Verzeichnis */etc* liegen muss. *nwserv.conf* ist eine Textdatei von ca. 32 KByte Größe, aber keine Panik: Der größte Teil ist Dokumentation mit vielen Hinweisen und Beispielen, die einem die Installation leicht machen.

Die Datei *nwserv.conf* kann also mit einem beliebigen Editor bearbeitet werden und ist in mehrere Sektionen unterteilt, die den jeweiligen Gegebenheiten angepasst werden müssen:

### Sektion 1: Die Novell-Volumes

Hier findet die Zuweisung der Linux-Verzeichnisse zu den MARS- bzw. Novell-Volumes statt, die dann auf der Clientseite einen Laufwerksbuchstaben zugewiesen bekommen. Die Namen sind frei wählbar, auf der MARS-Seite sollten nur Großbuchstaben verwendet werden; Umlaute und sonstige Sonderzeichen sind nicht erlaubt. Falls der MARS-Server der Server ist, in den die Clients sich einloggen, so muss folgendes beachtet werden: Das erste Volume muss *SYS* heißen. Außerdem müssen die Unterverzeichnisse *LOGIN*, *PUBLIC*, *SYSTEM* und *MAIL* vorhanden sein. Wenn die Clients den MARS-Server per Novell-Befehl *attach* als weiteren Server ansprechen, gelten diese Regeln nicht. Der MARS-Volume-Name darf maximal acht Buchstaben umfassen.

Die Syntax für Sektion 1 lautet:

```
Volume-Name Verzeichnis [Optionen] [Verzeichnis-Maske
Datei-Maske]
```

*Volume-Name* ist der exportierte Name des MARS-Volumes, *Verzeichnis* ist das Linux-Verzeichnis, das exportiert werden soll, die Optionen können folgender Tabelle entnommen werden. *Verzeichnis-Maske* und *Datei-Maske* sind oktale Werte, welche die Schreib- und Leserechte analog zu *chmod* beim Erzeugen vergeben.

Die wichtigsten Optionen von Sektion 1:

- *k* alle Dateinamen werden in Kleinbuchstaben benutzt, Standardeinstellung.
- *m* kennzeichnet ein wechselbares Laufwerk, wie z.B. ein CD-ROM- oder ZIP-Laufwerk, das während des Betriebes von MARS gemountet werden kann.

## Linux im Netzwerk

---

- *r* das Laufwerk ist read-only wie z.B. CD-ROM-Laufwerke.
- *O* der Novell-Namespace wird emuliert, um lange Dateiname zu ermöglichen für Clients unter OS/2 und Windows 95.

Beispiele:

```
1 SYS /mars/SYS k
1 CDROM /cdrom kmr
1 FUERALLE /mars/ALLE k 777 666
```

### Sektion 2: Der Servername

Hier wird der Servername des MARS-Servers vergeben. Er wird dann später von der Clientseite benötigt, um sich einzuloggen (s.u.). Nur Großbuchstaben sind erlaubt. Weitere Parameter sind hier nicht vorgesehen.

Beispiel:

```
2 JUPITER
```

Falls die optionale Sektion 2 nicht angegeben wird, so wird der Hostname des Rechners in Großbuchstaben gewandelt, evtl. auf acht Zeichen gekürzt und als Servername benutzt.

### Sektion 3: Interne Netzwerknummer

Auch bei einem IPX-Netz werden interne Netzwerknummern für jeden Rechner vergeben. Der Autor des MARS-Servers macht es Ihnen an dieser Stelle jedoch sehr einfach: man muss sich in keiner Weise um die internen Nummern kümmern, sie werden automatisch erkannt. Die Syntax lautet:

```
3 Interne_Netzwerknummer [Knoten]
```

Hier kommt jetzt der Komfort dieser Sektion zum Tragen: Standardmäßig sollte diese Zeile wie folgt aussehen:

```
3 auto 1
```

Hierdurch wird die eindeutige IP-Nummer des Linux-Rechners automatisch für das IPX-Netz benutzt. Wer aus Test- oder Debug-Gründen explizit die Werte angeben möchte, muss die Netzwerknummer in hexadezimalen Werten angeben. Es sei nochmals daran erinnert, dass es sich hier um Netzwerk- und Knotennummer des IPX-Netzes handelt, nicht um die Werte des TCP/IP-Netzes.





### Sektion 4: IPX-Devices

Hier werden die Parameter für die Geräte gesetzt, mit denen der Rechner die IPX-Pakete im Netz sendet und empfängt. Die Syntax lautet:

```
4 Netz_Nummer Gerät Frame [Ticks]
```

Es muss sichergestellt sein, dass die *Netz\_Nummer* nicht schon von einem andern IPX-Server im Netz benutzt wird. Man kann sich unter DOS oder Linux die Liste aller IPX-Server und ihrer Nummen mittels des Programms *slist* anzeigen lassen. Die *Netz\_Nummer* muss in hexadezimalen Werten mit einem vorangestellten ox angegeben werden. Der Wert oxo erzwingt die automatische Vergabe der *Netz\_Nummer*.

*Gerät* ist der Name der Linux-Geräte-Datei, über die der Netzwerkverkehr läuft. Dieses wird in der Regel *eth0* sein, aber auch andere Treiber sind möglich, wie z.B. *arco* für das Acnet, *isdn0* für IPX über ISDN oder auch *pppo* für PPP über ISDN. Der Stern \* an dieser Stelle erzwingt die automatische Erkennung der Geräte.

Falls weitere Novell- oder MARS-Server im Netz vorhanden sind, ist zu klären, welcher Frame-Typ von den anderen Servern benutzt wird. Dieser muss unter *Frame* eingetragen werden. Falls diese Information nicht vorhanden ist, kann die automatische Erkennung durch *auto* aktiviert werden. Mögliche Frame-Typen sind:

- ethernet\_ii für den Parallelbetrieb von IP- und IPX-Netzen via Ethernet
- 802.2 Novell-Standard seit Version 3.12
- 802.3 älterer Novell-Standard; nur noch selten im Gebrauch
- token für Token-Ring-Netzwerkarten
- auto automatische Erkennung des benutzten Frame-Typs

*Ticks* ist die Angabe der Zeit, die zum Versenden eines Pakets benötigt wird, gemessen in 1/18 Sekunden. Der Standardwert beträgt 1 und muss normalerweise nicht geändert werden. Falls der MARS-Server in großen, komplexen Netzen betrieben wird, muss dieser Wert evtl. erhöht werden.

Beispiele:

```
4 0x0 * auto 1
```

Automatische Vergabe der *Netznummer*, automatische Erkennung der *IPX-Geräte*, automatische Erkennung des *Frame-Typs*. Die Version für sehr bequeme Menschen, sollte aber in aller Regel funktionieren.

```
4 0x5 eth0 802.2 1
```

*Netz\_Nummer* 5 an der ersten Ethernet-Karte mit dem Standard-Frame-Typ 802.2.

## Linux im Netzwerk

---



4 0x5 ippp0 802.3 7

*Netz\_nummer* 5 für PPP über ISDN mit dem (veralteten) Frame-Typ 802.3 und erhöhter Tick-Anzahl.

### Sektion 5: Besondere Geräteoptionen

- 0x1 – logische IPX-Geräte und Routen werden nach dem Herunterfahren von Server oder Router nicht entfernt, was sonst normalerweise der Fall ist.
- 0x2 – da Windows 95 es manchmal nicht schafft, das IPX-Interface automatisch zu erzeugen, wird dieses unabhängig vom Client hierdurch erzwungen.

Die Standardeinstellung für Sektion 5 sollte normalerweise lauten:

5 0x0

### Sektion 6: Serverversion

Folgende Serverversionen können an den Client gemeldet werden:

- 0 Version 2.15
- 1 Version 3.11
- 2 Version 3.12

Falls lange Dateinamen unterstützt werden sollen, muss Einstellung 1 oder 2 gewählt werden. Die Standardeinstellung lautet:

6 0x0

### Sektion 7: Behandlung der Passwörter der DOS-Clients

Falls man von einem DOS-Client sein Passwort ändern will, kann das Passwort verschlüsselt werden, bevor es zum MARS-Server geschickt wird. Dieses erhöht die Sicherheit. Auf jeden Fall wird das Passwort auf dem MARS-Server nur verschlüsselt gespeichert.



## Der MARS-Server



Die möglichen Werte für Sektion 7 sind:

- 0 – das Passwort wird durch den DOS-Client verschlüsselt.
- 1 – wie zuvor; die nicht verschlüsselnden Versionen der Novell-Dienstprogramme werden aber unterstützt.
- 7 – alle nicht verschlüsselten Passwörter sind erlaubt, aber ein Passwort ist notwendig.
- 8 – alle nicht verschlüsselten Passwörter sind erlaubt, ein Passwort ist nicht notwendig.
- 9 – nur unverschlüsselte Passwörter sind erlaubt. Diese Option funktioniert nicht mit OS/2-Clients.

Die Standardeinstellung lautet:

7 0

## Sektion 8: Verhalten beim Login und Dateioperationen

Folgende Optionen sind möglich:

- ox1 – auch wenn man noch nicht eingeloggt ist, kann in ein anderes Verzeichnis als *SYS:LOGIN* gewechselt werden.
- ox2 – regelt das Verhalten einer Datei während des Öffnens. Falls die Datei zum Zweck des Schreibens von Daten geöffnet wird, dieses Datei aber die Eigenschaft *readonly* besitzt, so wird normalerweise eine Fehlermeldung zurückgegeben. Durch diese Option wird die Datei im Modus *readonly* geöffnet.
- ox4 – der Novell-Standard erlaubt aus Sicherheitsgründen das Umbenennen von Verzeichnissen nicht, sondern gibt eine Fehlermeldung zurück. Durch diese Option wird das Umbenennen von Verzeichnissen auf dem Server ermöglicht.
- ox8 – eventuelle Beschränkungen bezüglich der Arbeitsplatzrechner und der Arbeitszeit des Systemadministrators *SUPERVISOR* werden übergangen.
- ox10 – ermöglicht das Löschen einer Datei, auch wenn sie von einem anderen Benutzer geöffnet wurde.



## Linux im Netzwerk

---

- ox20 – der Dateieintrag wird sofort im Verzeichnis gespeichert, noch bevor die Datei geschrieben wird. Dieses ist notwendig, wenn *ncpfs* als Client benutzt wird.
- ox40 – beschränkt die Meldung des unbenutzten Plattenplatzes auf 2 GByte. Manche DOS-Programme können mit mehr freiem Platz nicht umgehen.

Die Standardeinstellung lautet:

8 0x0

### Sektion 9: Rechtevergabe beim Erzeugen von Verzeichnissen und Dateien

Die Syntax lautet

9 Verzeichnisrechte Dateirechte

*Verzeichnisrechte* und *Dateirechte* müssen oktale Werte sein, wie sie z.B. auch von *chmod* erwartet werden. Falls 0 angegeben wird, werden die Standardrechte benutzt. Wenn -1 angegeben wird, werden die Rechte des übergeordneten Verzeichnisses benutzt.

Die Standardeinstellung lautet:

9 755 664

### Sektionen 10 und 11 : User ID und Group ID vor dem Einloggen

Bevor der User sich in den MARS-Server einloggt, ist für ihn nur das Verzeichnis *SYS:LOGIN* sichtbar, das nur die beiden Programme. *LOGIN.EXE* und *SLIST.EXE* enthalten sollte. Von dem Rest des Servers ist vor dem Einloggen für den Client nichts sichtbar. Aus Sicherheitsgründen sollte in diesem Moment der Client nur minimale Rechte haben. Diese Rechte werden in diesen beiden Sektionen vergeben.

Die Syntax lautet:

10 GID  
11 UID

Die Standardeinstellungen sind:

10 65534  
11 65534





### Sektion 12: Supervisor-Login

In Novells Sprachgebrauch wird der Systemadministrator als „Supervisor“ bezeichnet. Er ist vergleichbar mit dem Benutzer *root* unter Linux. Als Supervisor kann man diverse Novell-/MARS-Konfigurationen ändern, die dem normalen Benutzer verwehrt sind.

Die Angabe des Supervisor-Passworts ist Pflicht, da es ohne diese Angabe beim ersten Start des MARS-Servers keinerlei Zugriffsmöglichkeit gibt! Auch hilft ein Eintrag in Sektion 13 als *Supervisor* nicht weiter, da dort nur Rechte für normale Benutzer vergeben werden.



Hinweis

Die Syntax lautet:

```
12 Mars_Login Linux_Login [Passwort]
```

- *Mars\_Login*: der Login-Name auf dem MARS-Server für den Supervisor, der standardmäßig *SUPERVISOR* lautet.
- *Linux\_Login*: der zugehörige Login-Name auf dem Linux-Rechner, traditionell *root*, falls man nicht aus Sicherheitsgründen dem Supervisor einen anderen Namen zugewiesen hat.
- *Passwort*: das Passwort für den *Mars\_Login*. Das Passwort muss hier im Klartext eingegeben werden. Nach dem ersten Login als Supervisor wird das Passwort verschlüsselt gespeichert, so dass diese Sektion gelöscht werden kann. Das Passwort kann in freundlichen Umgebungen auch weggelassen werden.

Beispiel:

```
12 SUPERVISOR root SUPERWEISER
```

### Sektion 13: User-Logins

Diese Sektion ist optional und als Vereinfachung für den Systemverwalter gedacht: jedem Benutzer, der Zugang zu dem Linux-Server besitzt, können automatisch Benutzerrechte für den MARS-Server zugewiesen werden.

Die Syntax ist entsprechend der von Sektion 12; zusätzlich gibt es noch einen optionalen Flag, der das Verändern des Passwortes durch den Benutzer verhindert. Dieser Flag hat den Wert *ox1*.

Beispiele:

```
13 PETER Peter
13 GAST gast gast 0x1
```

## Linux im Netzwerk

---



Vorsicht bei deutschen Umlauten und Sonderzeichen! Bei allen 8-Bit-Zeichen hat die Novell-/MARS-Welt andere Vorstellungen als Linux oder Windows: Hier gilt noch der gute alte 8-Bit-IBM-Zeichensatz. Wenn also der MARS-Benutzername *DÖRTE* lautet, so ist *D<Alt 153>RTE* einzugeben, damit es zu keinen Konflikten beim Einloggen kommt.

### Sektion 14 wird derzeit nicht genutzt

### Sektion 15: Die automatische Einordnung von Linux-Benutzern als MARS-Benutzer

MARS hat sich als vollwertiger Ersatz für Novell 3.xx-Server inzwischen in vielen Bereichen durchgesetzt. Auch in größeren Betrieben mit vielen Benutzern wird MARS genutzt. Dieses würde zu doppelter Arbeit zur Verwaltung der Benutzer-Daten führen: Wenn die jeweiligen Benutzer sowieso schon einen Linux-Zugang haben, so können diese Daten automatisch dem MARS-Server übergeben werden. Für all diese Benutzer muss ein Standard-Passwort vergeben werden.

Die Syntax lautet:

15 Flag Standard-Passwort

*Flag* kann folgende Werte besitzen:

- 0 – die Linux-Daten werden nicht von MARS übernommen, die Standardeinstellung.
- 1 – die Linux-Daten werden automatisch von MARS übernommen; als Passwort wird das *Standard-Passwort* übernommen.
- 99 – liest alle Login-Daten aus */etc/passwd*. Danach werden alle bereits bestehenden User-Daten überschrieben, und alle Passwörter werden auf *Standard-Passwort* gesetzt.

Beispiel:

15 1 geheim

Die Standardeinstellung ist:

15 0

### Sektion 16: Überprüfungen während des Hochfahrens

Falls diese Option angeschaltet werden soll (mit einer 1), so wird während des Hochfahrens des Servers zum einen überprüft, ob alle notwendigen Verzeichnisse auf SYS: vorhanden sind, wie z.B. *LOGIN*, *SYSTEM* oder *MAIL*. Außerdem werden

## Der MARS-Server



einige Interna überprüft. Beim ersten Start des Servers oder nach der Installation einer neuen Version sollte dieses immer geschehen. Da diese Überprüfung nur wenig Zeit benötigt, kann man diese Option getrost anstellen.

Standardeinstellung:

16 1

### Die Sektionen 17 bis 20 werden derzeit nicht benutzt

#### Sektion 21: Drucker-Queues

Novell verwaltet einen oder mehrere Queues zum Drucken, für jeden Drucker mindestens eine. Die Drucker müssen nicht direkt an dem Linux-Server angeschlossen sein, sondern können mittels Novell-Dienstprogrammen im Netz angesteuert werden, wo sie dann mit einem Arbeitsplatzrechner verbunden sein können. In dieser Sektion wird nun die MARS-Seite mit der Linux-Seite verbunden.

Die Syntax lautet:

21 Queue\_Name [Queue\_Verzeichnis] [Druck\_Befehl]

- *Queue\_Name*: Der Name der Print-Queue, wie der Benutzer sie sieht.
- *Queue\_Verzeichnis*: Das Verzeichnis, in dem die Druck-Jobs gespeichert werden. Der Name muss in DOS-Syntax angegeben werden. Das Verzeichnis muss auf dem ersten MARS-Volume liegen, traditionell also auf SYS:. Dieses Verzeichnis wird beim Hochfahren von MARS automatisch erzeugt. Um die Verwaltungsarbeit zu automatisieren, erzeugt der Bindestrich – das entsprechende Standardverzeichnis für jede Queue.
- *Druck\_Befehl*: Hier befinden Sie sich nun wieder auf der Linux-Seite: Mittels welcher Linux-Befehle soll der Druck-Job auf Papier gebracht werden? Standardmäßig sollte *lpr* genutzt werden. Etwaige Filter können hinzugefügt werden.

Beispiele:

```
21 HP550 -- lpr
21 LEXMARK -- lpr
```

#### Sektion 22: Printserver

Hier wird der Name des Printservers angegeben, normalerweise um den *ncpfs-pserver* zu aktivieren.



## Linux im Netzwerk

---



Die Syntax lautet:

```
22 Printserver_Name Queue_Name
```

Beispiel:

```
22 DRUCKER LEXMARK
```

Es folgen sehr spezielle Sektionen, die wichtigsten allgemeinen Angaben wurden hier aber abgehandelt.

## Starten des Servers

Der Server wird nun gestartet durch die Eingabe von:

```
nwserv
```

*nwserv* ist die Binärdatei, die normalerweise von der Installationsroutine in dem Verzeichnis */usr/sbin* abgelegt wird. *nwserv* selbst startet nun wiederum mehrere Prozesse, die mit *ps aux* angesehen werden können. Etwaige Fehlermeldungen gibt MARS unter */var/log/nw.log* aus. Man kann sie sich betrachten durch die Eingabe von:

```
tail -f /var/log/nw.log
```

Der Server läuft, nun können die einzelnen User sich einloggen.

## DOS-Clients für MARS

Alle folgenden Befehle können unter DOS interaktiv eingegeben werden; sie können aber genauso in die *autoexec.bat* geschrieben werden. Dieses ist natürlich zumindest mit Passwörtern problematisch. Es wird wiederum davon ausgegangen, dass der erste Server mit IPX-Protokoll ein „richtiger“ Novell-Server ist, sich also Novell-spezifische Dienstprogramme wie *LOGIN*, *ATTACH*, *MAP* oder *SLIST* auf diesem Server befinden. Zuerst muss sich auf dem Standard-Server eingeloggt werden mit:

```
LOGIN Name Passwort
```

Nun soll der zweite Server, in diesem Fall der MARS-Server, von dem Arbeitsplatzrechner angesprochen werden können. Dieses geschieht durch:

```
ATTACH SERVER/BENUTZER
```

Falls ein Passwort für den zweiten Server spezifiziert worden ist, wird nach diesem gefragt. Danach müssen die in Sektion 1 exportierten Laufwerke an einen Laufwerksbuchstaben gebunden werden. Dieses geschieht wie unter einem Standard-Novell-Server mit dem Befehl *MAP*:



## Der MARS-Server



```
MAP K:=MARS/SYS:
MAP L:=MARS/CDROM:
MAP M:=MARS/FUERALLE:
```

So werden den Beispielen aus Sektion 1 die DOS-Laufwerke *K*, *L* und *M* zugewiesen, die jetzt als weitere Laufwerke genutzt werden können.

## Linux-Client für MARS

Nicht nur auf der Serverseite wird Linux immer häufiger. Zusehends mehr Menschen benutzen auf der Clientseite inzwischen Linux und fahren Windows nur noch hoch, weil es (noch) keine adäquaten Linux-Programme gibt, wie z.B. für Home-Banking oder Desktop-Publishing. Auch von Linux aus kann auf ein bestehendes Novell-Netz und also auch den MARS-Server zugegriffen werden. Die Dienstprogramme wurden von Volker Lendecke geschrieben und sind Bestandteil vieler Distributionen. Bevor aber MARS-Volumes gemountet werden, muss das IPX-Interface konfiguriert werden, was glücklicherweise mit folgendem Befehl automatisch geschieht:

```
ipx_configure -auto_interface=on -auto_primary=on
```

Mit dem Befehl *ncpmount* können nun die MARS-Volumes in den Linux-Dateibaum eingebunden werden. Dazu sollte ein Verzeichnis erzeugt werden, welches praktischerweise auf der obersten Ebene des Linux-Dateibaums angesiedelt sein sollte; in den folgenden Beispielen wird von */MARS* ausgegangen. *ncpmount* hat eine sehr komplexe Syntax, so dass im Folgenden nur die wichtigsten Optionen besprochen werden:

```
ncpmount [-S Server] [-U User-Name] [-P Passwort |
-n] Mount-Point
```

- *-S Server* ist der Name des MARS-Servers, in diesem Fall also MARS
- *-U Username* ist der Benutzer-Name
- *-P* ist das Passwort des entsprechenden Benutzer
- *-n* wird benötigt, falls kein Passwort zugeordnet ist
- *Mount-Point* ist das Verzeichnis, in dem *ncpmount* die Volumes des MARS-Servers ablegt, in diesem Beispiel */MARS*

Nachdem *ncpmount* also erfolgreich die MARS-Volumes gemountet hat, sollte das Verzeichnis */MARS* wie folgt durch *ls* angezeigt werden, wobei wieder die Beispiele von Sektion 1 herangezogen werden:

```
rechner:/MARS # ls
sys cdrom fueralle
```

## Linux im Netzwerk



Wie auch bei DOS-Festplatten werden alle Verzeichnisse und Dateinamen in Kleinbuchstaben ausgegeben. Jetzt kann von jeder Applikation auf die Dateien der MARS-Volumes zugegriffen werden. Mittels des Linux-Dienstprogramms *slist* können außerdem alle erkannten Server angezeigt werden.

### Hinweis



Wir, die Autoren, haben schon mehrfach erlebt, dass das Netz sehr träge war, wenn ein Server frisch gebootet worden ist. Beharrlichkeit führt zum Ziel: *ncpmount* musste teilweise bis zu 20 mal ausgeführt werden, bis es willig war, die Volumes des MARS-Servers zu mounten!

## Drucken unter MARS

Im Gegensatz zu älteren Versionen kann MARS inzwischen auch das Verhalten eines Novell-Servers beim Drucken von einem Arbeitsplatzrechner aus emulieren. Die Vorarbeit wurde bereits in den Sektionen 21 und 22 erledigt: die Angabe der Druck-Warteschlange und des Namens des Print-Servers. Auch hier hilft ein Dienstprogramm aus dem Paket von Volker Lendecke namens *pserver*. *pserver* hat – verkürzt – folgende Syntax:

```
pserver [-S Server] [-U User-Name] [-P Passwort|-n] [-q Queue-Name] [-c Befehl]
```

- *-S Server* Server ist der Name des File-Servers, in diesem Fall also MARS
- *-U Username* ist der Name des Print-Servers auf dem MARS-Server, in diesem Fall also DRUCKER, *siehe Sektion 22*
- *P Passwort* – es wurde kein Passwort spezifiziert, also kommt die nächste Option zum Tragen
- *-n* kein Passwort
- *-q Queue-Name* ist der Name der MARS-Queue, wie in Sektion 21 angegeben, in diesem Beispiel also LEXMARK
- *-c Befehl* – der Befehl ist an dieser Stelle eigentlich überflüssig, da er schon in Sektion 21 spezifiziert wurde – die Standardeinstellung lautet lpr

Was passiert nun mit den Druckaufträgen, die der Benutzer an den Server loschickt? Mithilfe des Novell-Dienstprogramms *capture* werden z.B. alle lokalen Druckaufträge, die an LPT3 gedruckt werden, an den MARS-Server gesendet. Folgende Zeile sollte man in seine *autoexec.bat* schreiben, und zwar NACH dem ATTACH an den MARS-Server:

```
capture /L=3 /S=MARS /Q=LEXMARK
```

*capture* fängt alle Druckaufträge an LPT3 ab, auch wenn diese Schnittstelle am Arbeitsplatzrechner physisch nicht vorhanden ist. Der Druckauftrag wird der MARS-Queue auf dem Server MARS übergeben, in diesem Fall der Warteschlange



## Up, up and away – die Anbindung an das Internet



LEXMARK. Der MARS-Server entnimmt den Informationen aus Sektion 21 und 22, dass der Druckauftrag an *lpr* übergeben werden soll. Das Druckdienstprogramm *lpr* übergibt dann den Druck-Job an den Druck-Dämon *lpd*, der dann den Job an den entsprechenden Drucker weiterleitet, standardmäßig an die erste parallele Schnittstelle. Von Windows aus werden keine weiteren Filter benötigt, da die Windows-Druckertreiber die Daten auf das richtige Format des jeweiligen Druckers umwandeln, so dass MARS und Linux die Daten nur weiterreichen.

### Erfahrungen

Der Betrieb eines MARS-Servers in diesem Umfeld zeigt, dass MARS klaglos und stabil seinen Dienst versieht und keine großen Hardwareansprüche stellt. MARS läuft dort seit einem Jahr ohne Unterbrechung auf einem normalen Pentium-Rechner, der große Mengen grafische Daten trägt. Für die Benutzer ist nur ein weiterer Server sichtbar, was sich an den DOS-/Windows-Laufwerken I:, J: und K: bemerkbar macht. Von dem Standpunkt des Benutzers sind keine Unterschiede zum dem „echten“ Novell-Server sichtbar oder bemerkbar, auch nicht im Zeitverhalten. MARS ist stabil und zudem relativ schnell zu konfigurieren.

### Weitere Informationen

MARS hat sich im Laufe der Zeit etabliert, so dass es auch in vielen professionellen Umgebungen arbeitet. Wer also weitergehende Fragen hat, der kann sich auf dem zuständigen Mail-List-Server eintragen lassen: Schicken Sie eine E-Mail an *linware@sh.cvut.cz* mit dem Text *add linware*. Hier werden die unterschiedlichsten Fragen und Probleme diskutiert, wie z.B. MARS und DEC-Alpha, Arnet und MARS, PPP über IPX oder – natürlich – Windows 98 und MARS. Dass MARS den Kinderschuhen entwachsen ist, sieht man auch dort an einer öfter auftauchenden Frage: Immer dann, wenn bereits 248 Benutzer im MARS-Server eingeloggt sind, bekommen weitere Benutzer keinen Zugang ... Aber auch dafür gibt es eine Lösung. Damit man nicht zum wiederholten Male ein bereits gelöstes Problem diskutiert, gibt es die gesamte bisherige Mail unter der Adresse *ftp://klokan.sh.cvut.cz/pub/listserv/linware*.

## Up, up and away – die Anbindung an das Internet

„Meine Freundin wollte, dass wir unbedingt ins Internet gehen. Dabei habe ich doch keine Ahnung davon wie das funktioniert ...“

Nach einigen Schwierigkeiten, die ich mit meiner WurstDOSe hatte (irgendwie war das doch nicht so einfach ...), dachte ich mir, das doch mal anders zu versuchen ... Ich las hierzu folgenden Text.“





### Modem, ISDN-Karte oder ISDN-Modem?

Ob Sie Ihren Zugang zum Internet nun mit einem internen oder externen Modem, einem externen ISDN-„Modem“ oder einer internen ISDN-Karte realisieren wollen – die Unterschiede in der Konfiguration der ersten drei unterscheiden sich nur marginal, die Einrichtung der vierten Lösung zeigt sich als mindestens genauso einfach wie die der ersten.

#### Die Modemkonfiguration

Um ein Modem (intern, extern oder in Form eines an die serielle Schnittstelle angegeschlossenen ISDN-Adapters) zu konfigurieren, verfahren Sie wie folgt:

- Rufen Sie *yast* auf und wählen dort den Menüeintrag *Administration des Systems > Hardware in System integrieren > Modem konfigurieren*.
- Wie Sie der Abbildung 13.20 entnehmen können, müssen Sie hier angeben, an welche serielle Schnittstelle Ihr externes Modem angeschlossen ist. Diese Kommunikationsschnittstellen werden unter Linux als */dev/ttyS0* für die erste, */dev/ttyS1* für die zweite Schnittstelle und so weiter bezeichnet (unter DOS heißen diese etwas anschaulicher *com1*, *com2* usw.).



Abb. 13.20

Konfiguration des  
Modems

- Wählen Sie nun die Menüoption *Administration des Systems > Netzwerk konfigurieren > PPP Netzwerk konfigurieren*. Es wird das Konfigurationstool *wvdial* aufgerufen (siehe Abbildung 13.21).



## Up, up and away – die Anbindung an das Internet



Abb. 13.21

Das Konfigurationstool  
*wvdial*

Dieses gibt Ihnen folgende Einstellungsmöglichkeiten:

- Konfigurieren Sie Ihren Provider: Geben Sie hier die *Telefonnummer* Ihres Providers an. Legen Sie fest, ob eine *Amtsholung* erforderlich ist (dies ist bei Nebenstellenanlagen in der Regel eine vorweggewählte 0). Geben Sie die Ihnen von Ihrem Provider zugeteilte *Benutzerkennung* und das *Passwort* ein. Legen Sie mit der Leertaste fest, ob die *Nameserverkonfiguration* automatisch erfolgen soll oder nicht. Legen Sie das *Wählverfahren* fest (unterschieden wird hier zwischen [Im-] Puls- und Tonwählverfahren – beim Pulsverfahren hören Sie während des Wahlvorgangs mit Ihrem Telefon ratternde Geräusche, beim so genannte „Mehrfrequenz“ oder „Tonwählverfahren“ mehr oder weniger hübsche Melodien, gebildet aus unterschiedlich hohen Einzeltönen in Abhängigkeit der von Ihnen gewählten Rufnummer). Legen Sie fest, ob das von Ihnen verwendete Modem an eine *Telefonanlage* angeschlossen ist. Legen Sie den Einwahlmodus fest (dieser wird Ihnen ebenfalls von Ihrem Provider mitgeteilt) und verlassen Sie das Menü durch Bestätigung mit *Exit* (siehe Abbildung 13.22).
- Hinweis für Benutzer eines externen ISDN-Adapters („ISDN-Modems“): Um Ihr Gerät nutzen zu können, müssen Sie über die zuvor vorgenommenen Einstellungen hinaus im *Experten-Menü* Ihre besondere Hardware konfigurieren. Bestätigen Sie diesen und den darauf zusätzlich angebotenen Menüeintrag *Standard-Analog-Modem*, und wählen Sie den von Ihnen verwendeten ISDN-Adapter oder das Gerät aus, zu dem Ihres kompatibel ist.



## Linux im Netzwerk



Abb. 13.22

Konfiguration des Providers

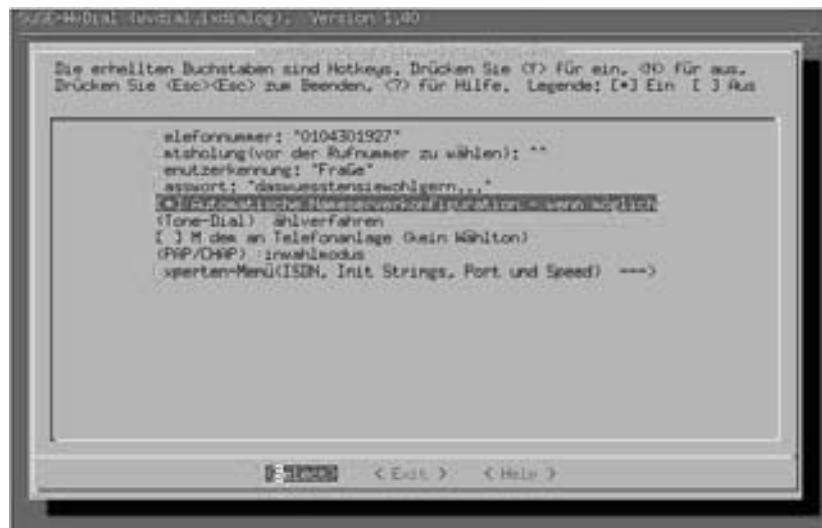


Abb. 13.23

Auswahl zwischen analogen Modem und externem ISDN-Adapter



Wählen Sie nun den Menüpunkt *WvDial starten und Provider anwählen*. Wenn alle Einträge korrekt von Ihnen vorgenommen wurden, sollten Sie mit einer Bildschirmausgabe ähnlich der in Abbildung 13.24 „belohnt“ werden.



## Up, up and away – die Anbindung an das Internet

```
Output of this wvdial run:
--> WxDial: Internet dialer version 1.40
--> Initializing modem.
--> Sending: ATZ
ATZ
OK
--> Sending: ATQ0 V1 E1 S0=0 &C1 &D2 S11=55 +FCLASS=0
ATQ0 V1 E1 S0=0 &C1 &D2 S11=55 +FCLASS=0
OK
--> Modem initialized.
--> Sending: AT&T,73066
--> Waiting for carrier.
AT&T,73066
CONNECT 14400/V92
--> Carrier detected. Starting PPP immediately.
--> Starting pppd at Sun Jan 23 17:48:16 2000
```

Abb. 13.24

Erfolgreicher Versuch der Herstellung einer Wählverbindung zum Provider

- Wenn Sie eine Fehlermeldung erhalten sollten, überprüfen Sie Ihre Eingaben und wiederholen Sie diesen Vorgang.
- Starten Sie nun beispielsweise *Netscape* und geben als WWW-Adresse ein: <http://www.sybex.de>. Sie sollten erfolgreich auf der Homepage des SYBEX-Verlags landen.
- Schließen Sie *Netscape* wieder und verlassen Sie *wvdial* mit *Exit*. Sie werden daraufhin abgefragt, ob Sie die von Ihnen vorgenommenen Einstellungen speichern möchten, was Sie im Erfolgsfall eines Verbindungsaufbaus wahrscheinlich bestätigend beantworten möchten.

### Konfiguration von ISDN

Wie bereits eingangs erwähnt, sollte auch die Konfiguration einer ISDN-Karte mit den aktuellen Linux-Distributionen keine große Hürde mehr darstellen, wobei sich aber speziell die SuSE als besonders anwenderfreundlich auszeichnet – dies sicherlich, weil ISDN nun einmal in der Bundesrepublik (anders etwa als in den USA) sehr weite Verbreitung gefunden hat.

Verfahren Sie für die Konfiguration Ihrer ISDN-Karte wie folgt:

- Wählen Sie in *yast* die Menüoption *Administration des Systems > Hardware in System integrieren > ISDN-Hardware*. Es erscheint die in Abbildung 13.25 dargestellte Eingabemaske.

## Linux im Netzwerk

Abb. 13.25

Konfiguration der ISDN-Hardware



- Aktivieren Sie hier mit der Leertaste das Optionsfeld *I4L Starten*, wählen Sie das gewünschte ISDN-Protokoll (dies sollte meist das heute übliche Euro-ISDN [EDSS1] statt des alten deutschen ISDN-Protokolls [7TR6] sein; eine häufige Ausnahme hierzu tritt dann auf, wenn Sie die ISDN-Karte an eine Telefonanlage [„TK-Anlage“] angeschlossen haben). Wählen Sie den Typ der von Ihnen verwendeten ISDN-Karte (zur Zeit werden bereits über 40 Modelle unterstützt – sollten Sie sich erst eine ISDN-Karte kaufen wollen, ist es empfehlenswert, einen Blick in die Kompatibilitätsliste zu werfen). Nehmen Sie gegebenenfalls erforderliche Einstellungen zum Interrupt (IRQ) und zur Ein-/Ausgabeadresse (I/O-Port) der Karte vor. Ob diese notwendig sind, hängt von dem von Ihnen verwendeten Kartentyp ab – die Autoren schwören auf „Nicht-Plug-and-Play-Karten“, die man idealerweise hardwareseitig einstellen kann. Speziell auf Maschinen, die mehrere Betriebssysteme starten können, kann es anderenfalls zu unliebsamen Überraschungen kommen, wenn ein anderes System meint, an der Konfiguration der Karte etwas ändern zu dürfen, und diese plötzlich unter Linux nicht mehr funktioniert. Die Standardeinstellungen Ihrer Karte entnehmen Sie bitte dem Handbuch. Rufen Sie nach Abschluss Ihrer Eingaben das I4L-Startskript auf, welches im Idealfall Folgendes ausgibt.
- Wenn Sie einen Fehler in Ihrer Konfiguration vorgenommen haben sollten, wird das entsprechend in diesem Dialog kommentiert. Verwenden Sie bitte diese Informationen, um den oder die Falscheingaben zu korrigieren, und wiederholen Sie den Aufruf des Startskripts. Wenn alle Eingaben korrekt sind, können Sie diese abspeichern.



## Up, up and away – die Anbindung an das Internet



Abb. 13.26



Ausgabe des I4L-Startskripts



Wählen Sie *Administration des Systems > Netzwerk konfigurieren > Netzwerk Grundkonfiguration*, und legen Sie mit der Taste **F5** ein neues Gerät an. Es erscheint folgende Auswahlliste, aus der Sie im Regelfall den Eintrag *ISDN SyncPP* wählen.



Mit der Taste **F6** rufen Sie den Dialog auf, um notwendige IP-Adressen zu vergeben. Bei einer Wahlverbindung zu einem Provider wird im Regelfall nach Verbindungsaufbau dynamisch eine IP-Adresse vergeben, so dass Sie an dieser Stelle für die IP-Adresse Ihres Rechners eine „Dummy-Adresse“ vergeben sollten, die ansonsten nicht in Ihrem Netzwerk verwendet wird. Die Adresse des Gateways ist im Regelfall die gleiche wie diese Dummy-Adresse und wird automatisch von *yast* vergeben. Die IP-Adresse Ihres *Point-To-Point-Partners* erhalten Sie von Ihrem Internet Service Provider (*ISP*).



## Linux im Netzwerk

Abb. 13.27

Netzwerktyp bestimmen



Abb. 13.28

Eingabe der Netzwerk-Adressen



Vergessen Sie nicht, nach Beendigung Ihrer Eingabe das neue Device mithilfe der Taste **F4** zu aktivieren.

Wählen Sie *Administration des Systems > Netzwerk konfigurieren > ISDN-Parameter konfigurieren*. Es erscheint folgende Eingabemaske.



## Up, up and away – die Anbindung an das Internet



Abb. 13.29

Konfiguration der ISDN-Parameter

Geben Sie hier Ihre eigene Telefonnummer, die anzurufenden Nummern (dies sind die Nummern Ihres ISPs) und Nummern an, die Sie gegebenenfalls anrufen dürfen. Aktivieren Sie auf jeden Fall das Kontrollfeld *Nur angegebene Nummern erlaubt*, da ansonsten jeder eine Verbindung zu Ihrem Rechner aufbauen dürfte. Mit Hilfe der Option *Wählmodus* legen Sie fest, ob – beispielsweise bei einem Aufruf von *Netscape* – der ISP automatisch angewählt werden oder ob dieses manuell geschehen soll. Mit der *Idle-Time* legen Sie fest, nach wie vielen Sekunden Inaktivität die Verbindung zum ISP abgebrochen (also wann „aufgelegt“) werden soll. Das Eingabefeld *Maximale Wählversuche* legt fest, wie häufig ein Verbindungsaufbau versucht werden soll, wenn der erste Anlauf nicht erfolgreich war. Wenn Sie weiter oben bei den anzurufenden Nummern zwei Rufnummern spezifiziert haben, ist die Anzahl der maximalen Wählversuche gleich der von Ihnen hier angegebenen Zahl mal 2. Bei *Name des PPP-Logins* handelt es sich um den Benutzernamen, unter dem Sie bei Ihrem ISP „bekannt“ sind, beim *Passwort des PPP-Logins* um Ihr Zugangspasswort. Nach Eingabe dieser Informationen können Sie einen Verbindungsaufbau zu Ihrem Provider mithilfe der Option *Starten* initiieren. Sollten im folgenden Dialog Fehler ausgegeben werden, korrigieren Sie diese entsprechend. Ansonsten speichern Sie Ihre Eingaben, und uns bleibt nur noch, Ihnen zu wünschen:

### Viel Spaß im Internet!

Nachdem Sie nun in diesem Kapitel die wesentlichen Grundlagen für eine lokale Vernetzung und eine Anbindung ans Internet kennengelernt haben, werden im folgenden *Kapitel 14 Mit Linux ins Internet* weitere nützliche Netzwerk-Tools wie FTP und TELNET vorgestellt.







## Mit Linux ins Internet



|                                        |            |
|----------------------------------------|------------|
| Voraussetzungen für den Internetzugang | <b>578</b> |
| Mailing                                | <b>579</b> |
| Newsgroups                             | <b>587</b> |
| World Wide Web                         | <b>588</b> |
| Datentransfer mit FTP                  | <b>602</b> |
| <i>telnet</i>                          | <b>615</b> |



**14**

## Mit Linux ins Internet

---



Nachdem in den Abschnitten zuvor die lokale Vernetzung einzelner Computersysteme erläutert wurde, erfahren Sie in diesem Kapitel zeigen, wie Sie Ihr LINUX-System bzw. Ihr heimisches „Intranet“ an das weltweite System des Internets anbinden.

Das Internet besteht aus mehreren Millionen Computern, die auf unterschiedliche Weise (sowohl physisch als auch softwareseitig) miteinander vernetzt sind, so dass man eigentlich nicht von „einem Computernetzwerk“ sprechen kann, sondern es als eine Verknüpfung unterschiedlicher heterogener Netzwerke oder Anbindungen von Einzelplatzsystemen an ein Teilnetz bezeichnen müsste. Das Internet wächst mit einer atemberaubenden Geschwindigkeit, die jegliche Schätzung seines aktuellen Umfangs unmöglich macht.

Es wurde ursprünglich eingerichtet, um den Bedürfnissen der US-Verteidigungsindustrie gerecht zu werden (ARPANET), und wuchs dann zu einem riesigen, globalen Netzwerk, an das Universitäten, akademische Forschungsstellen, kommerzielle Anbieter und Regierungsstellen in allen fünf Kontinenten angeschlossen sind. Somit können weltweit Personen aus unterschiedlichen Kulturkreisen und Gesellschaftsschichten miteinander in Verbindung treten oder sogar über Ländergrenzen oder Kontinente hinweg zusammenarbeiten.

Wie bereits in der Einführung erwähnt, ist LINUX ein solches Produkt der weltweiten Kooperation von Programmierern, das ohne das Internet nie hätte entstehen können.

Demzufolge ist es fast logisch, dass LINUX als „Kind des Internets“ alle wesentlichen Leistungsmerkmale bietet, die es ihm ermöglichen, sich sowohl als Client in dieses einzubinden als auch seine Ressourcen zur Verfügung zu stellen, indem ein mit ihm betriebener Rechner als Server fungiert. Als Kommunikationsprotokoll innerhalb des Internets dient das im *Kapitel 13 Linux im Netzwerk* vorgestellte TCP/IP-Protokoll, welches Sie bereits für die lokale Vernetzung verwendet haben.

Die zuvor erwähnte Kommunikation schlüsselt sich im Wesentlichen in folgende Anwendungsbereiche auf:



*E-Mail:* Der wohl am häufigsten von allen Usern verwendete Netzwerkdienst, der es ermöglicht, weltweit nicht nur Texte, sondern auch an diese angehängte Dateien (so genannte „Attachments“) – egal, ob es sich hierbei um Grafikdateien, Tabellenblätter oder sogar Programmdateien handelt – zu versenden. Über das Internet erreichbar sind mittlerweile auch Benutzer kommerzieller Netzdienste, wie America Online, CompuServe, lokale Provider (mehr zu diesen weiter unten) und (man lese und staune) der Telekom-Netzzugang T-Online.

### Hinweis



Im Internet-Jargon wird die Briefpost auch als „Snail-Mail“ (dt. „Schnecken-post“) oder einfach „Smail“ bezeichnet.



## Mit Linux ins Internet



- **IRC-Chat:** Abkürzung für **Internet Relay Chat**. Ein Dienst, der eine große Anzahl von Benutzern in Gruppendiskussionen in Echtzeit verbindet, so dass diese sich interaktiv (im Gegensatz zum Mailing, in dem sie, ähnlich der Briefpost, im Ping-Pong-Verfahren miteinander kommunizieren) über die Tastatur verständigen können.
- **Mailing-Listen:** Private Diskussionsgruppen, die durch E-Mail erreicht werden.
- **USENET Newsgroups:** Größere öffentliche Diskussionsgruppen, die sich auf ein bestimmtes Thema konzentrieren. Post und Informationen der USENET Newsgroups werden mithilfe eines so genannten „Newsreaders“ gelesen.
- **Gopher:** Ein modernes, menübasiertes System, das benutzt wird, um die Internetressourcen anzuzeigen, und das die darunter liegenden technischen Details des Internets vor dem Benutzer versteckt. Dieses Informationssystem verliert durch den Siegeszug des im Folgenden beschriebenen World Wide Web immer mehr an Bedeutung.
- **World Wide Web:** Kurz „WWW“, „W3“ oder auch einfach „Web“ genannt. Ein System, das auf Hyperlinks basiert und das genutzt wird, um Internetressourcen zu finden und auf sie zuzugreifen, indem man ein im Dokument hervorgehobenes Element anklickt, z.B. ein bestimmtes Textelement oder eine Grafik, und somit ein neues Dokument öffnet. Das World Wide Web ist einer der am schnellsten wachsenden Internetservices, dessen sich Hochschulen, Firmen oder auch Privatpersonen zur Eigendarstellung und Informationsvermittlung bedienen, wobei immer mehr multimediale Elemente wie Animationen, Filmsequenzen, akustisches Beiwerk usw. in diesen Selbstdarstellungen Verwendung finden.
- **FTP:** Eine Client/Server-Anwendung, die benutzt wird, um Dateien von und zu einem Internet-Hostcomputer zu übertragen. FTP ist auch der Name des Datenübertragungsprotokolls (**File Transfer Protocol**), das benutzt wird, um diese Aufgabe auszuführen.
- **TELNET:** Eine Client/Server-Anwendung, die benutzt wird, um sich in Internetcomputer einzuloggen und dann Anwendungen auf diesem auszuführen. Eine Version dieses Programms heißt „tn3270“ und wird benutzt, um auf IBM- und nicht auf UNIX-Computer zuzugreifen. Sie wird häufig auch als „3270-Emulator“ bezeichnet.
- Eine neuere Funktionalität stellt die Möglichkeit des *Telefonierens* über das Internet dar, bei der über den Internetzugang hinaus eine Soundkarte auf Ihrem und dem Rechner Ihres Kommunikationspartners installiert sein muss, welche die Sprachinformationen digitalisiert bzw. wieder in analoge Signale umwandelt. Hierbei ist zu bedenken, dass die Sprachübertragung ein enormes Datenaufkommen verursacht und somit das ohnehin schon notorisch überlastete Internet weiter bezüglich der erreichbaren Übertragungsraten beeinträchtigt. Tun Sie sich und uns einen Gefallen: Verzichten Sie auf die Nutzung dieses Services!



## Mit Linux ins Internet

---



- Eine Erweiterung dieses Telefonierens sind *Telekonferenzen*, bei denen zur reinen Sprachübertragung noch die Übermittlung visueller Informationen hinzukommt, wobei sich die Kommunikationspartner über so genannte „Webcams“ (digitale Kameras) gegenseitig sehen können.

Das Gesamtvolumen der Informationen des Internets ist schlicht erschlagend. Aufgrund der Tatsache, dass das Internet eine zufällige Gruppierung von vielen unterschiedlichen Netzwerken ist, gestaltet sich die Suche nach bestimmten Informationen oft als recht mühselig und zeitaufwändig, zumal sich die im Internet erreichten Datenübertragungsraten durch seine zunehmende Popularität und somit Überlastung ständig verringern.

Andere Problembereiche des Internets betreffen Netzwerksicherheit, Privatsphäre, Urheberrecht und Geheimnisschutz.

## Voraussetzungen für den Internetzugang

Voraussetzung für die Nutzung des Internets als Privatperson oder (kleineres) Unternehmen sind zumindest entweder ein Modem- oder aber ein ISDN-Anschluss. Diese ermöglichen eine Verbindung zu einem so genannten „Internet Service Provider“. Vielleicht gehören Sie aber auch zu den glücklichen Zeitgenossen, die über einen Zugang verfügen, wie Sie ihn innerhalb von Universitäten und anderen (Hoch-)Schuleinrichtungen in lokalen Hochschulnetzen über die Rechenzentren vorfinden.

Die zentralen Rechner der Service-Provider oder der Rechenzentren der einzelnen Hochschulen sind an weitere Rechner angebunden, meist über eine Standleitung.

Solche Standleitungen sind jedoch in der Regel für Privatpersonen zu teuer – eine günstigere Form der Anbindung, die theoretisch von ihrer Geschwindigkeit (beim Upload) das 12-fache eines normalen ISDN-Anschlusses und somit annähernd die Hälfte einer 2-MBit-Standleitung bietet, ist (A)DSL.

Weitere Möglichkeiten des Zugangs sind:

- *Internet per Satellit*. Bei dieser Zugangsart sind theoretisch bis zu 8 MBit/sec. möglich, wobei die anfallenden Kosten für Privatpersonen wohl (noch) zu hoch sind. In der Praxis sind die Übertragungsraten jedoch meist nicht viel höher als bei ISDN. Da eine normale Satellitenschüssel in Kombination mit einer DVB-Karte nur Daten empfangen kann, muss zusätzlich noch eine ISDN-Leitung oder ein analoger Anschluss zur Verfügung stehen, um auch Daten senden zu können – sendefähige Satellitenschüsseln kosten mehrere Zehntausend Mark.
- *Kabelmodems*. Bei Nutzung der bereits bestehenden Infrastruktur (TV-Kupferkabel; achten Sie jedoch darauf, ob das von Ihnen verwendete TV-Kabel rückkanalfähig ist) können mithilfe von Kabelmodems, die einfach über Ethernet-Netzwerkkarten angeschlossen werden, Übertragungsraten von theoretisch





bis zu 40 Mbit/sek. erreicht werden. Zurzeit gibt es jedoch kaum Provider, die einen Zugang über Kabelmodem zulassen – anders als in den USA, wo diese vermutlich aufgrund mangelnder ISDN- oder DSL-Versorgung schon früh ihren Durchbruch hatten.



*Richtfunk.* Übertragungsraten von 2 Mbit/sek. für den Endkunden (bei einer theoretisch maximalen Übertragungsrate von bis zu 155 Mbit/sek.) müssen hierbei teuer erkauft werden. Einrichtungsgebühren von im Schnitt DM 300,-, Grundgebühren zwischen DM 50,- und DM 150,- im Monat bei begrenzten Datentransfermengen (im Schnitt 2 GB), wozu häufig für jedes weitere Gigabyte weitere DM 50,- hinzukommen. Dies und die geringe Zahl an Anbietern machen diese ansonsten sehr attraktive Technologie für den Otto-Normalverbraucher (zurzeit noch) uninteressant.

Hiermit ist dann auch schon das Prinzip des Internets in seiner simpelsten Form beschrieben.

LINUX stellt in den meisten Distributionen Softwarelösungen zur Nutzung der zuvor beschriebenen Internetdienste zur Verfügung, von denen hier die wichtigsten für die am häufigsten genutzten Dienste (Mailing, Newsgroups, WWW, FTP und Telnet) vorgestellt werden.

## Mailing

Das Standard-Mailingprogramm, das auf jedem UNIX- (und damit auch auf jedem LINUX-)System zu finden ist, ist das Programm *mail*. Dieses eher einfache Programm wurde bereits im *Kapitel 4 Benutzerumgebung* besprochen.

Einen recht komfortablen Ersatz stellt das Public Domain-Programm *elm*, dar, das mit den meisten LINUX-Distributionen vertrieben wird.

### E-Mail mit *elm*

Das Mailingprogramm *elm* ist ein interaktives, ganzseitenorientiertes Mailing-system; Sie finden es in der Serie *n – Netzwerk-Support* der dem Buch beiliegenden SuSE-Distribution.

#### Syntax

```
elm[-achkKmrtwz][-f Verzeichnis][-d debug- level]
elm[-s subject] Liste von Aliasen oder Adressen
```

#### Beschreibung

*elm* ist ein interaktives, bildschirmorientiertes Mailingprogramm, das *mail* und seine Weiterentwicklung *mailx* ersetzt.



## Mit Linux ins Internet



Beim erstmaligen Aufruf von *elm* werden Sie mit dem Dialog aus *Abbildung 14.1* konfrontiert.

Abb. 14.1

Erster Installations-  
schritt beim Erstaufzug  
von *elm*

```
gehrke@travelmate2:/export/home/gehrke > elm
```

```
Notice:
This version of ELM requires the use of a .elm directory in your home
directory to store your elmrc and alias files. Shall I create the
directory .elm for you and set it up (y/n/q)?
```

Hinweis



*elm* benötigt ein Verzeichnis *.elm*, in dem die Dateien *elmrc* und Ihre Alias-Dateien gespeichert werden. Sie sollten diesen Dialog also mit **Y** beantworten.

Alias-Dateien enthalten Informationen zur Adressverwaltung.

Als Nächstes erfolgt die Abfrage, ob ein Verzeichnis angelegt werden soll, in dem Ihre Mail gespeichert wird.

Abb. 14.2

Zweiter Installations-  
schritt beim Erstaufzug  
von *elm*

```
gehrke@travelmate2:/export/home/gehrke > elm
```

```
Notice:
This version of ELM requires the use of a .elm directory in your home
directory to store your elmrc and alias files. Shall I create the
directory .elm for you and set it up (y/n/q)? Yes.
Great! I'll do it now.
```

```
Notice:
ELM requires the use of a Folders directory to store your mail Folders in.
Shall I create the directory /home/gehrke/Mail for you (y/n/q)?
```

Auch diesen Dialog sollten Sie mit **Y** beantworten.

### Verwendung von *elm*

Es gibt nun im Wesentlichen drei verschiedene Methoden, *elm* zu verwenden:



Man kann in der Befehlszeile beim Aufruf des Programms eine Liste von Mailingadressen angeben. Dies erlaubt, eine einzelne Nachricht mit allen normalerweise für das *elm*-System zur Verfügung stehenden Optionen an die angegebenen Empfänger zu versenden.

Die Eingabe der folgenden Befehlszeile

```
elm -s teest gehrke
```



## Mailing



würde das Senden einer Nachricht an den Alias-Namen *gehrke* mit der Betreffzeile *teest* initiieren und den von Ihnen gewählten Standardeditor für die Eingabe der Nachricht öffnen (an dieser Stelle wurde mit Absicht *teest* und nicht *test* gewählt; wie Sie im Kapitel 16 *Shell-Programmierung* noch sehen werden, ist der Begriff *test* ein Befehl, dessen Verwendung hier unter Umständen zu Problemen führen würde ...). Würde die Option *-s* mit ihrem Argument *teest* nicht angegeben werden, müssten Sie nun eine Betreffzeile eingeben.

Damit *elm* nicht *vi* als Standardeditor verwendet, müssen Sie den von Ihnen gewünschten Editor entweder in der Datei *.profile* durch Definition einer entsprechenden Umgebungsvariablen spezifizieren oder in der Datei *\$HOME/.elm/elmrc* folgenden Eintrag vornehmen (wenn Sie beispielsweise *joe* als Standardeditor verwenden wollen): *editor=/usr/bin/joe*.



Tipp

- Die zweite Methode, die meist verwendet wird, wenn zum Beispiel bereits zuvor geschriebene Dateien übertragen werden sollen, besteht darin, die Betreffzeile und den Empfänger in der Befehlszeile einzugeben und eine Eingabeumleitung durchzuführen.

Die Eingabe der folgenden Befehlszeile

```
elm -s teest gehrke < teest.c
```

würde eine Kopie der Datei *teest.c* mit der Betreffzeile *teest* an den User *gehrke* senden. Die Option *-s* und ihr Argument sind optional.

- Die dritte Methode besteht darin, *elm* ohne Betreffargument, Adresse oder Umleitung der Standardeingabe von der Befehlszeile aus aufzurufen. Sie können dann die eingehenden Mails lesen, auf Mails antworten, neue Nachrichten verfassen und so weiter.

Das Programm *elm* besitzt die in Tabelle 14.1 dargestellten Optionen.

### Befehlszeilenargumente für das Programm *elm*



| Option          | Bedeutung                                                                                                        |
|-----------------|------------------------------------------------------------------------------------------------------------------|
| <i>-a</i>       | ändert den Cursor in einen Pfeil, anstatt einen inversen Cursorbalken zu verwenden                               |
| <i>-c Alias</i> | gibt vollständige Angaben über das spezifizierte Alias aus                                                       |
| <i>-d level</i> | spezifiziert den Debug-Level. Die Ausgabe erfolgt nach <i>\$HOME/ELM/debug.info</i> .                            |
| <i>f Verz</i>   | durchsucht das durch <i>Verz</i> spezifizierte Verzeichnis statt des Standardverzeichnisses für eingehende Mails |
| <i>-h</i>       | zeigt eine Liste der Aufruptionen an                                                                             |



## Mit Linux ins Internet

### Befehlszeilenargumente für das Programm *elm* (Forts.)

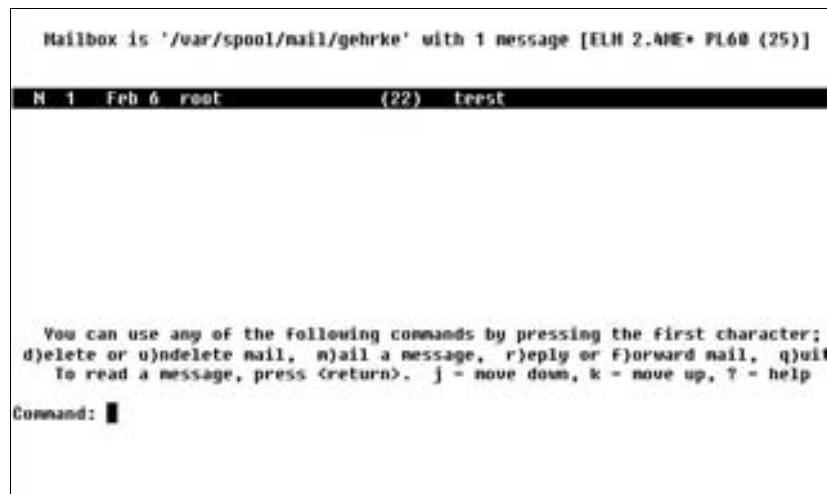
| Option                  | Bedeutung                                                                                |
|-------------------------|------------------------------------------------------------------------------------------|
| <code>-i Datei</code>   | lädt die durch <i>Datei</i> spezifizierte Datei mit in den Editier-Buffer                |
| <code>-?</code>         | wie <code>-h</code>                                                                      |
| <code>-k</code>         | erlaubt den Gebrauch einer HP-Terminal-Tastatur mit den auf ihr vorhandenen Sondertasten |
| <code>-K</code>         | erlaubt den Gebrauch von Softkeys auf einer HP-Terminal-Tastatur                         |
| <code>-m</code>         | blendet die Menüoptionen aus                                                             |
| <code>-s Betreff</code> | erlaubt die Angabe einer Betreffzeile                                                    |
| <code>-t</code>         | schaltet den Gebrauch von Termcap/Terminfo aus                                           |
| <code>-v</code>         | zeigt die Versionsnummer und Konfigurationsinformationen an                              |
| <code>-z</code>         | verhindert das Öffnen von <i>elm</i> , wenn keine Mail vorhanden ist                     |

Tabelle 14.1

Beim Aufruf von *elm* erscheint der in Abbildung 14.3 dargestellte Bildschirm.

Abb. 14.3

Der Startbildschirm von  
*elm*



The screenshot shows the terminal window of the elm mail client. At the top, it displays the message count: "Mailbox is '/var/spool/mail/gehrke' with 1 message [ELH 2.4HE+ PL68 (25)]". Below this, the message list header is shown: "M 1 Feb 6 root (22) test". A large empty area follows, representing the body of the message. At the bottom, there is a command prompt: "You can use any of the following commands by pressing the first character; d)elete or u)ndelete mail, m)ail a message, r)eply or F)orward mail, q)uit. To read a message, press <return>. j = move down, k = move up, ? = help" and "Command: █".

Wurde eine Mail noch nicht von Ihnen gelesen, wird dieser das Zeichen „N“ voran- gestellt.



## Mailing



Um sie nun zu lesen, müssen Sie sie nur mithilfe der Cursortasten anwählen, wenn Sie mehrere Meldungen erhalten haben sollten, und mit bestätigen.

Der von Ihnen gewählte Standardeditor wird geöffnet und zeigt die aktuelle Mail an.

In der Basis-Konfiguration (*elm* kennt die drei Benutzerebenen *Beginning User*, *Intermediate User* und *Expert User*, wobei im Rahmen dieser Einführung jedoch nur die Darstellung der Basisfunktionen für den Anfänger erfolgt) stehen Ihnen über das bloße Lesen von Mails hinaus noch weitere Menüoptionen zur Verfügung, die Sie in *Tabelle 14.2* dargestellt finden.

### Menüoptionen des Programms *elm*

| Menüoption             | Bedeutung                                                                |
|------------------------|--------------------------------------------------------------------------|
| <i>d)el</i> elete      | kennzeichnet eine Mail als beim Verlassen von <i>elm</i> zu löschen      |
| <i>u)n</i> delete mail | entfernt die Kennzeichnung wieder                                        |
| <i>m)ail a message</i> | erlaubt das Versenden einer neuen Mail                                   |
| <i>f)orward mail</i>   | erlaubt das „Weiterreichen“ der aktuellen Mail an einen neuen Adressaten |
| <i>q)uit</i>           | verlässt <i>elm</i>                                                      |

*Tabelle 14.2*



Im Folgenden werden die verschiedenen Schritte näher erläutert, die bei der Verwendung der Menüoptionen, die weitere Unteroptionen enthalten, durchgeführt werden müssen.

### ***m)ail a message***

Bei der Wahl dieses Menüpunkts erscheint ein neuer Eingabeprompt:

Send the message to:

Geben Sie hier die E-Mail-Adresse ein. Die Syntax hierfür lautet im Internet:

Username@Host-Rechner.Domain



## Mit Linux ins Internet



Möchten Sie beispielsweise eine E-Mail an die Autoren dieses Buches senden, können Sie hier eingeben:

Linux.team@kielnet.net

Als Nächstes werden Sie aufgefordert, eine Betreffzeile (*Subject*) einzugeben. Bestätigen Sie diesen Punkt einfach mit  (d.h., Sie möchten keine Betreffzeile angeben), werden Sie aufgefordert zu entscheiden, ob Sie dennoch fortfahren möchten.

Abb. 14.4

Das Verschicken einer Mail mit *elm*

```
Mailbox is '/var/spool/mail/gehrke' with 1 message [ELM 2.4HE+ PL6B (25)]
M 1 Feb 6 root (22) test

You can use any of the following commands by pressing the first character:
d)elete or u)ndelete mail, m)ail a message, r)eply or f)orward mail, q)uit
To read a message, press <return>. j = move down, k = move up, ? = help
Command: Mail To: linux-team@sue.uni-kiel.de
Subject of message: No subject - Continue with message? (y/n) █
```

Im folgenden Dialog können Sie festlegen, ob Kopien Ihrer Mail an weitere Adressaten erfolgen sollen:

Copies to:

Nachdem Sie dies (die Eingabe eines weiteren Empfängers einer Kopie ist natürlich optional) mit  bestätigt haben, wird Ihr Standardeditor geöffnet, mit dem Sie – wie gewohnt – Ihren Text verfassen können.

Die zuvor beschriebenen Eingabeinformationen werden auch als „Kopfzeilen“ (engl. *Header*) einer E-Mail bezeichnet. Folgende Tabelle gibt einen Überblick über die Bedeutung der einzelnen Kopfzeilen (die im Übrigen beim Empfänger im Kopf der Mail erscheinen werden).

### Die Header-Elemente einer Mail



| Kopfzeilenelement | Bedeutung                     |
|-------------------|-------------------------------|
| From:             | E-Mail-Adresse des Absenders  |
| To:               | E-Mail-Adresse des Empfängers |



## Mailing



### Die Header-Elemente einer Mail (Forts.)

| Kopfzeilenelement | Bedeutung                                                                                                                                                                                                                      |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Subject:</i>   | Betreffzeile der Nachricht                                                                                                                                                                                                     |
| <i>Reply-To:</i>  | Adresse, an die eine eventuelle Antwort geschickt werden soll; dies ist in der Regel die <i>From</i> -Adresse, kann aber auch eine andere sein                                                                                 |
| <i>Cc:</i>        | Adresse, an die eine Kopie (das <i>Cc</i> steht für <b>C</b> arbon <b>c</b> opy – dt. so etwas ähnliches wie ein Kohlepapier-durchschlag) der Mail geschickt werden soll                                                       |
| <i>Bcc:</i>       | Adresse, an die eine „blinde Kopie“ der Mail geschickt werden soll; der ursprüngliche Adressat erfährt jedoch nicht, dass die Nachricht in Kopie an einen Dritten geschickt wurde ( <b>B</b> lind <b>c</b> arbon <b>c</b> opy) |

Tabelle 14.3



Nach dem Verlassen des Editors stehen Ihnen folgende Optionen zur Verfügung, die Sie durch Drücken der durch eine Klammer hervorgehobenen Taste wählen können.

### Optionen der Funktion *mail a message* von *elm*

| Option               | Bedeutung                                                                                                                                                                                 |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>e)dit message</i> | kehrt zurück zum Editor                                                                                                                                                                   |
| <i>edit h)eaders</i> | ermöglicht eine nachträgliche Änderung der Kopfzeilen wie der Adresse, der Betreffzeile etc. Die hier weiterhin zur Verfügung stehenden Funktionen werden an dieser Stelle vernachlässigt |
| <i>s)end it</i>      | schickt die Mail ab                                                                                                                                                                       |
| <i>f)orget it</i>    | verwirft die zuvor von Ihnen verfasste Mail                                                                                                                                               |
| <i>r)eply</i>        | Erlaubt das direkte Beantworten der aktuellen Mail                                                                                                                                        |

Tabelle 14.4



Sie können eine erhaltene Mail direkt beantworten, wobei die Adresse des Absenders der zu beantwortenden Mail automatisch übernommen wird.



## Mit Linux ins Internet

---



Wenn Sie diese Funktion wählen, werden Sie zunächst aufgefordert festzulegen, ob die Originalnachricht kopiert werden soll, wobei die Verneinung voreingestellt ist:

Copy message (y/n) :

Der ursprüngliche Betreff wird in die Betreffzeile übernommen, wobei das *Re:* anzeigt, dass es sich um eine Referenzierung (einen Bezug) auf die ursprüngliche Mail handelt. Sie können diese Betreffzeile natürlich nach Belieben ändern oder ergänzen.

Nach der Bestätigung mit ist es auch hier wieder möglich, festzulegen, ob eine Kopie an einen weiteren Adressaten erfolgen soll.

Sollten Sie die Abfrage, ob die Originalnachricht kopiert werden soll, mit *y* beantwortet haben, wird der Originaltext nun in den Editor geladen, wobei die jeweiligen Originalzeilen durch ein vorangestelltes Größerzeichen (*>*) gekennzeichnet sind.

Eine Bemerkung zum Zitieren von Mails: Versuchen Sie dies auf ein absolutes Minimum zu reduzieren, denn bedenken Sie: Nicht nur Sie als Absender würden mit dieser Funktion unter Umständen ellenlange Texte verschicken, auch der Adressat müsste diese wieder von seinem Mailserver herunterladen, was auf beiden Seiten einen zusätzlichen (Online-)Zeitaufwand mit den damit verbundenen Kosten bedeutet und außerdem eine zusätzliche, vermeidbare Belastung des Internets darstellt.

Sie können diesen Text nun nach Belieben editieren.

Nach dem Speichern stehen Ihnen wiederum die in *Tabelle 14.3* dargestellten Optionen zur Verfügung, wonach wieder zum Hauptbildschirm zurückgekehrt wird.

### ***forward mail***

Diese Funktion erlaubt das Weiterleiten einer Mail, die Sie erhalten haben, an einen anderen Adressaten.

Sie werden zunächst aufgefordert, festzulegen, ob Sie die weiterzuleitende Mail editieren möchten:

Edit outgoing message (y/n) :

Es folgen wieder die Eingabe des Adressaten, einer Betreffzeile (der Standardtext, den Sie natürlich auch wieder abändern können, lautet *Forwarded mail...* – dt. „weitergeleitete Nachricht“) und eventueller weiterer Adressaten, die eine Kopie erhalten sollen.

Wenn Sie sich dafür entschieden haben, die zu sendende Mail noch weiter zu editieren, wird nun der Editor geöffnet und verfahren, wie im Abschnitt „*r)eply*“ dargestellt. Andernfalls stehen Ihnen sofort die in *Tabelle 14.3* dargestellten Optionen zur Verfügung.



## Newsgroups



### ***q)uit***

Wenn Sie die Menüoption *q)uit* wählen, werden Sie zunächst aufgefordert festzulegen, ob die Nachrichten, die Sie mit *d)elete* gekennzeichnet haben, tatsächlich gelöscht werden sollen.

`Delete messages (y/n) :`

Danach können Sie entscheiden, ob die Nachrichten, die Sie bereits gelesen haben, in die Datei für eingegangene Nachrichten verschoben werden sollen.

`Move read message to "received" folder? (y/n) :`

Abschließend quittiert *elm* mit einer entsprechenden Bildschirmmeldung, wie viele Nachrichten gelesen, gelöscht oder gespeichert wurden.

In engem Zusammenhang mit E-Mails stehen Mailinglisten und Newsgroups.

## Newsgroups

Die Usenet-News (auch „NetNews“ oder einfach „News“ genannt) sind ein verbreitetes Informations- und Diskussionssystem, bei dem jeder Teilnehmer Leser und Autor von Artikeln sein kann. Die einzelnen Artikel sind thematisch in Gruppen zusammengefasst, wobei man zwischen so genannten „moderierten“ und „unmoderierten Newsgroups“ unterscheidet.

Eine *unmoderierte Gruppe* ist ein unzensiertes Diskussionsforum, d.h., jeder, der Zugang zum so genannten „Posten“ (Senden von Artikeln) hat, kann Beiträge ergänzen oder auf bestehende Artikel antworten, ohne dass hierbei Rücksicht auf die Qualität des Beitrags genommen wird.

Bei den *moderierten Gruppen* werden die Beiträge an einen Moderator geschickt, der sie veröffentlicht, wenn ihm der Inhalt akzeptabel erscheint.

Die Newsgroups sind hierarchisch gegliedert und haben das Format *group.sub-group.subsub-group ...*

Diese Gliederung spiegelt sich auch in einer entsprechenden Bezeichnung der Gruppen wieder, zum Beispiel:

```
comp.security.unix
comp.databases
comp.databases.ingres
```

Drei wichtige Newsgroups (gerade für Anfänger) sind die Gruppen:

## Mit Linux ins Internet

---



- *news.newusers.questions* – Sie enthält Fragen und Antworten für neue News-Benutzer.
- *news.announce.newusers* – Diese Newsgroup enthält beispielhafte Postings, die man sich ansehen sollte, bevor man selbst zum ersten Mal einen Artikel sendet. Außerdem findet man hier Beiträge zur so genannten „Netiquette“, den „Benimm-Regeln“ im Internet (und speziell für die Teilnahme an Newsgroups).
- *news.answers* – In dieser Gruppe werden schließlich FAQs (Frequently Asked Questions – häufig gestellte Fragen) zu den verschiedensten Themenbereichen veröffentlicht.

Um an Newsgroups teilnehmen zu können, bedarf es eines so genannten „Newsreaders“.

### Newsreader

Der Newsreader ist die Schnittstelle zwischen dem Benutzer und dem News-System. Es handelt sich hierbei um ein Programm, mit dessen Hilfe Artikel gelesen, auf Artikel geantwortet sowie dem News-System neue Beiträge übergeben werden können.

Ein weit verbreiteter Newsreader, der hier jedoch nicht dargestellt werden soll, hat den schlichten Namen *nn*.

Beachten Sie bitte zum Thema Newsgroups den folgenden Abschnitt über das WWW und hier speziell die Ausführungen über *Netscape*.

## World Wide Web

Das World Wide Web ist der sich in den letzten Jahren explosionsartig ausbreitende multimediale Bereich des Internets, der auf Entwürfen des European Laboratory for Particle Physics (auch bekannt als CERN) basiert.

Es ist eine Ansammlung von Hyperlink-Dokumenten im Internet, die über so genannte „Links“ (Verbindungen, Verknüpfungen) Querbezüge zueinander herstellen und so den Zugriff auf verschiedene Informationstypen (Text, Grafik, Audio oder Video) ermöglichen. Diese Links verweisen hierbei nicht nur auf weitere Dokumente auf dem aktuellen Server, sondern können genauso die Verbindung zu einem anderen Server herstellen, der sich unter Umständen in einer völlig anderen Ecke der Welt befindet.

Bei der Informationssuche, bei der man sich von Seite zu Seite „hangelt“, spricht man auch vom „Surfen“.

Für den Zugriff auf das WWW benötigt man einen so genannten „Webbrowser“.

### Hinweis



## World Wide Web



Dabei handelt es sich um eine Client-Anwendung, die unter anderem für die Darstellung der in der Seitenbeschreibungssprache HTML (**Hypertext Markup Language**) kodierten Dokumente verantwortlich ist.

Bei den Browsern unterscheidet man zwischen reinen textbasierten Browsern, wie z.B. *Lynx*, oder den (in UNIX und seinen Derivaten) unter X Window laufenden grafikorientierten Browsern, wie z.B. *Arena*, *Mosaic*, *Netscape* oder dem KDE-Datei-Manager *Konqueror*, der ebenfalls als Internetbrowser eingesetzt werden kann.

### Lynx

*Lynx* hat zwar den Vorteil, dass es wirklich absolut frei erhältlich ist, d.h., es ist meist auch Bestandteil einer LINUX-Distribution, aber es ist – wie gesagt – rein textorientiert. Hierdurch können wesentliche Leistungsmerkmale des WWW (Textattribute, die Verwendung von Grafiken und Animationen) nicht genutzt werden, was natürlich ein wenig kontraproduktiv ist. Denn wenn man schon nach Informationen im multimedialen Teil des Internets sucht, dann möchte man natürlich auch all seine (vermeintlichen) Vorzüge nutzen können.

### Arena

Auch *Arena* ist als Bestandteil vieler LINUX-Distributionen zu finden, lässt jedoch bezüglich der gebotenen Funktionalität einiges zu wünschen übrig: Zwar unterstützt es alle offiziellen Leistungsmerkmale des Standards HTML 3, ist aber bezüglich der Bedienungsmöglichkeiten recht schlicht.

### Mosaic

*Mosaic* wurde ursprünglich vom NCSA (**National Center for Supercomputing Applications**) an der Universität von Illinois entwickelt und steht für die meisten UNIX-Derivate (so auch für LINUX) über anonymous FTP unter der Adresse

`ftp.ncsa.uiuc.edu`

zur Verfügung.

Es verwendet zwar die grafische Benutzeroberfläche (engl. **graphical user interface** – kurz *GUI*) X Window und ist somit bezüglich seiner Bedienbarkeit äußerst komfortabel, soll aber hier nicht (ebenso wenig wie *Lynx* und *Arena*) detailliert erklärt werden, da sowohl der im Folgenden dargestellte WWW-Browser Netscape als auch der KDE-Konqueror frei erhältlich sind.

### Netscape

Netscape ist sicherlich als eines der leistungsfähigsten Programme unter den Web-browsers zu bezeichnen, denn es entwickelt sich immer mehr zu einer Eier legenden Wollmilchsau: Über die reinen Browserfunktionen hinaus (Netscape unter-



## Mit Linux ins Internet

---



stützt nicht nur den HTML 3-Standard, sondern definiert darüber hinaus eigene Erweiterungen, die von vielen als große Bereicherung angesehen werden) bietet Netscape die Möglichkeit, Newsgroups zu lesen, E-Mails zu verschicken und zu empfangen, per FTP Dateien zu übertragen und so weiter. Sie sollten also über einen hinreichend leistungsfähigen Rechner verfügen, da dieses Mehrzweckprogramm natürlich eine Menge RAM benötigt.

In der Zwischenzeit wird Netscape 4.x in allen gängigen Distributionen mitgeliefert, da die Firma ihre bisherige Lizenzpolitik geändert hat und der Netscape Communicator für den Privatanwender kostenlos zu nutzen ist.

### Hinweis



Sie finden Netscape in seiner Version 4.76 in der Serie *xap – X Anwendungen* auf der dem Buch beiliegenden SuSE-Distribution.

Sie können sich die aktuellste Version des Programms aber auch selbst von einem der von der Firma Netscape betriebenen FTP-Server beschaffen.

Die Adresse hierfür lautet:

`ftpX.netscape.com`

wobei X eine Zahl zwischen 2 und 20 sein kann.

Um die aktuellste Version herunterzuladen, können Sie sich durch die Webseite von Netscape hindurchhangeln oder beispielsweise folgende Adresse anwählen:

`www.netscape.com/download/index.html`

Dabei sollten Sie die Adresse zunächst bis zum *Public*-Verzeichnis wählen, um dort weiter zu entscheiden, welche Version und in welchem Umfang Sie Netscapes Programme herunterladen möchten.

### Installation

Die Installation erweist sich zum Glück als recht problemlos:

### Hinweis



Bitte beachten Sie, dass die im Folgenden dargestellten Befehlszeilen alle sehr lang sind und aus satztechnischen Gründen nicht immer in eine Zeile passen, von Ihnen jedoch in eine einzige Zeile eingegeben werden müssen!

- Legen Sie zunächst ein Verzeichnis für Netscape an, z.B.:

`mkdir /usr/local/netscape`

- Entpacken Sie die gz-Datei:

`gzip -d  
Programm-vX-export.x86-unknown-Linux2.0.tar.gz`



## World Wide Web



- Installieren Sie das Archiv:

```
tar -xvf Programm-vX-export.x86-unknown-Linux2.0.tar
/usr/local/netscape
```

- Erstellen Sie einen Link (einen Verzeichniseintrag einer existierenden Datei unter anderem Namen):

```
ln -s /usr/local/netscape/netscape /usr/bin/X11R6
```

Jetzt ist nur noch eine Kleinigkeit zu beachten, bevor Sie unter X-Window mit Netscape arbeiten können.

- Sollte die Datei `/usr/X11R6/lib/X11/XKeysymDB` fehlen, müssen Sie die von Netscape mitgelieferte Datei hierhin kopieren:

```
cp /usr/local/netscape/XKeysymDB /usr/X11R6/lib/X11
```

Beim ersten Aufruf von Netscape (unter X wohlgemerkt) erscheinen zunächst einige Copyright-Meldungen und Warnungen, wonach Netscape selbstständig in Ihrem *HOME*-Bereich ein Verzeichnis `.netscape` anlegt. In diesem befindet sich neben lokalen Konfigurationsdateien ein weiteres Unterverzeichnis `HOME/.netscape/cache`, in dem Seiten, die Sie beim Surfen besucht haben, zwischengespeichert werden. So müssen diese, sollten Sie zu einem bereits besuchten Dokument zurückkehren wollen, nicht neu geladen werden.

Bedenken Sie, von Zeit zu Zeit (natürlich nicht während einer laufenden Sitzung) Ihren Cache (manuell) zu löschen, da dieser unter Umständen sehr viel Speicherplatz in Anspruch nehmen kann.

Die Oberfläche von Netscape zeigt sich hierbei wie in Abbildung 14.5.



Tipp



Abb. 14.5

Die Oberfläche von Netscape mit einer geladenen HTML-Seite

591



## Mit Linux ins Internet



### Konfiguration

Zur Konfiguration steht das Menü *Bearbeiten > Einstellungen* zur Verfügung, dessen wesentliche Einträge Ihnen nachfolgend kurz demonstriert werden.



Abb. 14.6

Das Menü *Bearbeiten > Einstellungen*

Wenn Sie im Menü *Bearbeiten > Einstellungen* den Punkt *E-Mail & Newsgruppen* wählen, steht Ihnen unter anderem ein Registerblatt *Identität* zur Bearbeitung zur Verfügung.

Tragen Sie hier Ihren richtigen Namen, Ihre Adresse, gegebenenfalls den Namen Ihrer Firma oder Dienststelle und den Pfad zu Ihrer Signaturdatei ein.



Eine Signatur-Datei ist eine kurze Textdatei, die automatisch an das Ende einer E-Mail gehängt wird. Sie enthält normalerweise Ihren Namen und Ihre (E-Mail-)Adresse; manche User fügen diesen Informationen noch einen schmierigen Spruch hinzu. Halten Sie diese Datei aber möglichst klein; bei einer Länge von mehr als fünf Zeilen erhalten Sie bei der „Erstverwendung“ eine Warnmeldung, die Sie jedoch ansonsten ignorieren können.

Weitere wichtige Einstellungen, die Sie vornehmen müssen, wenn Sie Netscape als Mail- und News-Programm verwenden wollen, erfolgen in der Kategorie *Mail-Server*, die Sie in Abbildung 14.8 dargestellt finden.



## World Wide Web



Abb. 14.7

Das Registerblatt *Identität*

Abb. 14.8

Die Kategorie *Mail-Server*



In diesem Registerblatt können Sie festlegen, welcher Host als *SMTP-Server* fungiert (dies wird der Server Ihres Internet Providers oder des zentralen Servers in Ihrem lokalen Ethernet sein). Dieser ist für die abgehenden Mails zuständig.



## Mit Linux ins Internet

### Hinweis

SMTP ist das TCP/IP-Protokoll zur E-Mail-Übertragung.

POP3 (das **Post Office Protocol**) unterstützt eine geordnete Übertragung von Nachrichten, die auf einem anderen Mailserver gespeichert sind, wobei es für den Empfang dieser Nachrichten verantwortlich ist.

Auch hier werden Sie wieder den Mailserver Ihres Internet Providers angeben, wobei die Wahrscheinlichkeit sehr groß ist, dass dieser POP3 (gegenüber dem etwas veralteten POP2) unterstützt.

Der *User Name* ist selbstverständlich der Name, den Sie zum Einloggen am Mailserver verwenden.

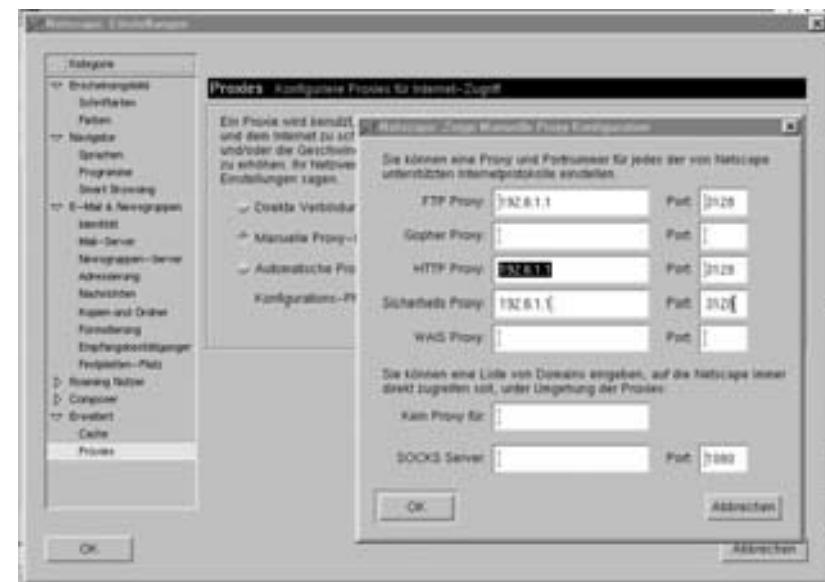
Ziemlich interessant ist noch die Angabe eines *Newsgroup-Servers*: Diesen müssen Sie konfigurieren, um mit Netscape auf USENET-News zugreifen zu können, wobei dieser Zugriff dann aber sehr komfortabel und damit der Verwendung von zum Beispiel *nn* (siehe den Abschnitt über Newsgroups) vorzuziehen ist. Informieren Sie sich bei Ihrem Internet Provider, welche Adresse Sie hier eintragen sollten.

### Proxies

In der Kategorie *Erweitert* finden Sie den Unterpunkt *Proxies*.

Abb. 14.9

Der Menüeintrag *Proxies*



Wenn Sie hier die Option *Manuelle Proxy-Konfiguration* wählen und dann die Schaltfläche *Anschauen* anklicken, wird ein neues Dialogfenster geöffnet, in dem Sie die Adressen verschiedener so genannter „Proxy-Server“ eintragen können.



## World Wide Web



Dazu eine kurze Erklärung: Stellen Sie sich vor, Ihre Arbeitskollegin Frau Vogel ruft in den Vereinigten Staaten eine bestimmte Webseite auf, deren Gesamtvolumen 400 KByte beträgt; abhängig von der Netzbela stung dauert die Übertragung dieser Seite unterschiedlich lange.

Wenn jetzt auch noch Sie diese gleiche Seite herunterladen wollen, tragen Sie natürlich zu einer zusätzlichen Belastung des Netzes bei, denn diese Datei muss den weiten Weg über den Ozean erneut antreten, was natürlich nicht sehr ökonomisch ist.

Hier greifen Proxies ein: Die von Ihrer Kollegin Vogel schon einmal abgerufene Seite wird über einen gewissen Zeitraum und in Abhängigkeit davon, wie häufig diese Information abgerufen wird, auf dem von ihr genutzten Proxy-Server zwischen gespeichert.

Für den Fall, dass Sie nun diese gleiche Seite abrufen und denselben Proxy-Server eingestellt haben, wird zunächst auf diesem nach der angefragten Webseite gesucht und, sollte sie noch dort zwischengespeichert sein, an Sie weitergeleitet.

Dieses Verfahren kann einerseits zur Beschleunigung der Übertragung von Daten beitragen und andererseits zu einer Gesamtentlastung des Internets führen.

Ein Hinweis noch zu dem Eingabefeld *Kein Proxy für:* Es macht sicherlich wenig Sinn, auf Proxy-Server zuzugreifen, wenn eine Seite aus der eigenen Domain übertragen werden soll. Tragen Sie also in dieses Feld den Namen Ihrer Domain ein.

Mit diesen Maßnahmen (auch bezüglich der Adresse der Proxy-Server sollten Sie sich an Ihren Provider oder Ihren Netzwerkadministrator wenden) haben Sie Netscape darauf vorbereitet, unterschiedliche Internetservices zu unterstützen.

### Handhabung

Die direkte „Anwahl“ eines WWW-Servers erfolgt durch Eingabe der so genannten „URL“ in die mit *Pfad:* beschriftete Eingabezeile.

Sie können einen Server natürlich auch über seine IP-Adresse anstatt über seinen symbolischen Namen ansprechen!



Hinweis

Bei der **URL** (**Universal Resource Locator**) handelt es sich um eine weltweit eindeutige, symbolische Adresse eines WWW-Servers.

Auf diesem finden Sie meist eine Startseite mit einem Index, der Ihnen Auskunft über die einzelnen Angebote dieses Servers, aber auch über Verweise zu weiteren WWW-Adressen gibt.



## Mit Linux ins Internet

Wie Sie sich sicherlich vorstellen können, gibt es beliebte WWW-Server, die sehr stark frequentiert werden und die somit zum einen bezüglich ihrer Übertragungsraten immer langsamer werden bzw. zum anderen den großen „Besucheransturm“ alleine überhaupt nicht bewältigen können, da sie nicht über eine ausreichende Anzahl an Zugangs-Ports verfügen.

Dem wird zum Teil dadurch Abhilfe geschaffen, indem die Verwalter anderer Server eine Kopie der Inhalte solcher „Renner des WWW“, die durch ihre extreme Belastung immer mehr zu Schnecken würden, auf Ihrem Server anbieten (man spricht hier auch von einer „Serverspiegelung“).

### Hinweis



Bei den Anbietern solcher „Serverspiegelungen“ handelt es sich häufig um Universitäten.

Wenn Sie ein absoluter WWW-Neuling sind, sei der Einstieg über folgende Adresse empfohlen: <http://www.entry.de>.

Geben Sie diese Adresse in die mit *Pfad* beschriftete Eingabezeile ein, und bestätigen Sie mit .

Sie erhalten eine Übersichtskarte der Bundesrepublik, auf der Sie zunächst die einzelnen Bundesländer anklicken können, in denen sich WWW-Server befinden. Es handelt sich bei dieser Webseite um einen Dienst des DFN-CIS (**D**eutsches **F**orschungsnetz – **C**enter for **I**nformation **S**ervices) an der Zentraleinrichtung für Datenverarbeitung der Freien Universität Berlin, der die offizielle Liste der deutschen WWW-Server führt.

Abb. 14.10

Eine Übersichtskarte der Bundesländer in Deutschland, über die Zugriff auf Webserver genommen werden kann



## World Wide Web



Die Angebote, welche die einzelnen Organisationen auf ihren Servern zur Verfügung stellen, können Sie dann ebenfalls wieder durch Anklicken in Anspruch nehmen.

Ein hervorragendes Beispiel einer Webseite ist folgende Adresse:

Abb. 14.11

Eine von der Universität Bonn gespiegelte Seite der Organisation TUCOWS



Auf dieser Seite sind deutlich verschiedene Elemente einer Webseite zu erkennen, die Ihnen nachfolgend kurz vorgestellt werden soll:

- Die jeweils hervorgehobenen und unterstrichenen Textelemente stellen Links zu weiteren Seiten dar.
- Die meisten der Grafiken (die Grafik mit der Beschriftung *Register your name...* ist übrigens eine animierte Grafik) können ebenfalls angeklickt werden; sie stellen die Verbindung zu weiteren Seiten her.
- Im Zentrum des oberen Viertels sehen Sie eine Eingabezeile, in die Sie einen Suchbegriff eingeben können. Bei diesem Suchbegriff handelt es sich (hier) um den Namen einer Software (TUCOWS ist ein Unternehmen, das kostenlos Public Domain und Shareware zur Verfügung stellt).
- Der Suchvorgang kann durch Anklicken der Schaltfläche *Go* aktiviert werden.



## Mit Linux ins Internet



—○ Diese Seite enthält darüber hinaus noch ein weiteres (ursprünglich von Netscape spezifiziertes) Leistungsmerkmal, nämlich so genannte „Frames“ (dt. Rahmen), welche die Unterteilung der Webseite in virtuelle Bereiche ermöglichen, durch die unabhängig voneinander gescrollt werden kann.

- 
- Die IPs für  
www.w3.org lauten:  
18.29.1.34, 18.29.1.35,  
18.176.0.26.
- Eine hervorragende Informationsquelle über das WWW finden Sie im Übrigen unter der Adresse <http://www.w3.org/WWW>.
- Dieses Angebot wird vom World Wide Web Consortium (W3C) betrieben.

Es wurde 1994 gegründet, um allgemein gültige Standards für die Weiterentwicklung des WWW zu entwickeln. Beim W3C handelt es sich um ein internationales Industriekonsortium, das gemeinsam vom **Massachusetts Institute of Technology Laboratory for Computer Science** (MIT/LCS) in den Vereinigten Staaten, dem **Institut National de Recherche en Informatique et en Automatique** (INRIA) in Europa und der Keio University Shonan Fujisawa Campus in Asien betreut wird. Ursprünglich wurde es in Zusammenarbeit mit CERN gegründet.

Einer weitergehenden Erklärung des Prinzips der Hyperlinks bedarf es hier wohl kaum, da man recht schnell die Handhabung solcher Web-Pages erfasst.

Zu detaillierteren Erklärungen der Funktionen und Leistungsmerkmale von Netscape konsultieren Sie bitte die einschlägige Fachliteratur, die sich ausschließlich mit diesem Programm befasst, und die Onlinedokumentationen.

### Konqueror

Eine zentrale Bedeutung innerhalb des *K Desktop Environment* (KDE – siehe auch *Kapitel 12 X-Konfiguration und KDE*) nimmt der Browser Konqueror ein, der sowohl als lokaler Datei-Manager als auch als Internetbrowser Verwendung finden kann.

Als Datei-Manager kann er beispielsweise zur Anzeige der Baumstruktur Ihres lokalen Dateisystems verwendet werden, wie folgende Abbildung zeigt:

Sie sehen hier in der Eingabezeile *URL* den Pfad des aktiven Verzeichnisses des lokalen Rechners.

Anstelle einer solchen lokalen Pfadangabe können Sie jedoch auch eine beliebige Internetadresse eingeben (wenn Sie Zugang zum Internet besitzen) und den Konqueror zum Surfen verwenden.

Wenn Sie mit einem Proxy arbeiten (siehe hierzu auch der Abschnitt „*Proxies*“ weiter oben in diesem Kapitel), wählen Sie gegebenenfalls im Menü *Einstellungen > Einrichten* die Option *Proxies* und nehmen Sie die entsprechenden Eintragungen vor.



## World Wide Web

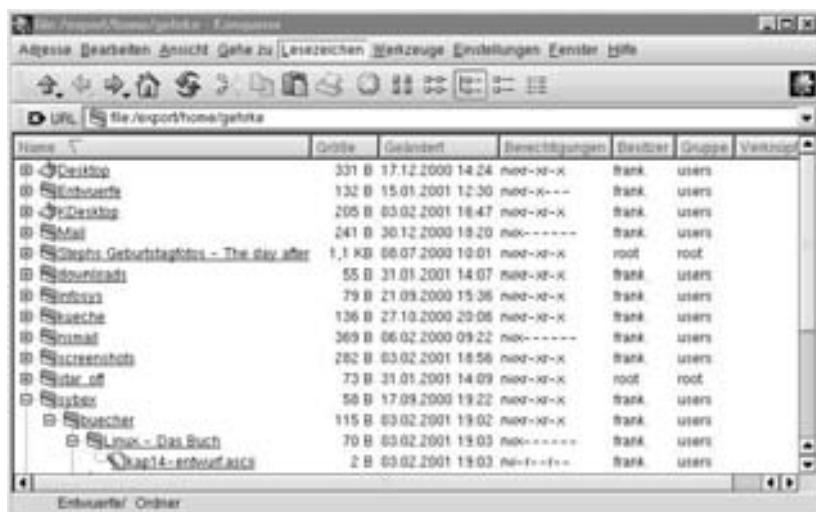


Abb. 14.12

Konqueror als lokaler Datei-Manager

Abb. 14.13

Festlegen von Proxies  
im Menü Einstellungen  
> Einrichten > Proxies



Nun können Sie wie beschrieben die von Ihnen gewünschte URL eintippen, um mit dem Surfen loszulegen.

Die in Abbildung 14.14 dargestellte Webseite bedient sich (wie in der Zwischenzeit eine Großzahl von Websites) der Programmiersprache JavaScript zur Generierung dynamischer Inhalte. In dem hier dargestellten Fall wird eine Grafik des Mondes bezüglich seiner „Vollheit“ in Abhängigkeit vom aktuellen Tagesdatum erzeugt.

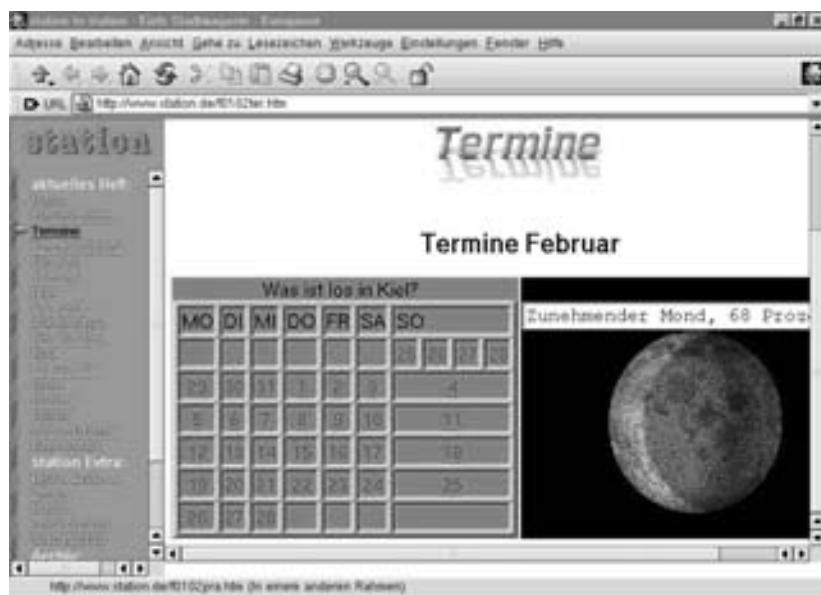


## Mit Linux ins Internet



Abb. 14.14

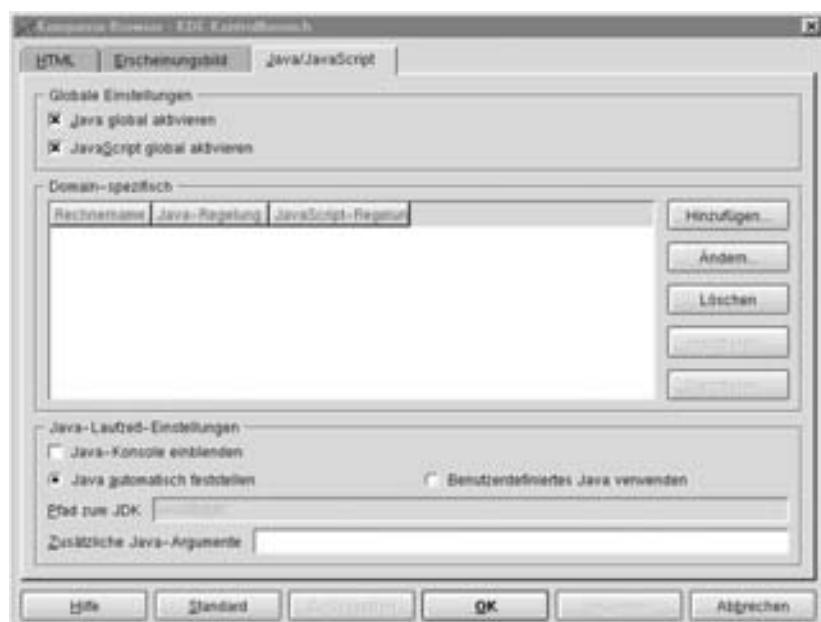
Der Konqueror als Web-  
browser



Um die durch Java und/oder JavaScript bereitgestellten Leistungsmerkmale nutzen zu können, müssen diese Dienste im Konqueror „freigeschaltet“ werden. Dies geschieht im Menü *Einstellungen > Einrichten > Browser* im Register *Java/JavaScript*.

Abb. 14.15

Die Freigabe von Java/  
JavaScript im Konqueror



## World Wide Web



Sinn oder Unsinn solcher aktiven Inhalte soll nicht Gegenstand dieser Betrachtungen sein. Ebenfalls sollen sicherheitsrelevante Aspekte in Bezug auf Java und JavaScript nicht an dieser Stelle diskutiert werden; wir weisen Sie jedoch ausdrücklich darauf hin, sich an anderer Stelle mit dieser Thematik auseinander zu setzen!

Bevor dieser Abschnitt über Browser schließt, sollen Sie jedoch noch erfahren, wie Sie gezielt an Informationen im WWW kommen können.

### Suchmaschinen

Wenn Sie Informationen zu einem bestimmten Thema suchen, aber die URL, wo Sie diese auffinden können, nicht kennen, sind so genannte „Suchmaschinen“ sehr hilfreich.

Es handelt sich dabei um (WWW-)Server, die riesige Datenbanken mit weltweiten Internetadressen verwalten, die Sie nach Schlagwörtern durchsuchen können.

In Abhängigkeit der von Ihnen definierten Abfrage (die logisch miteinander verknüpfte Suchbegriffe enthalten kann) werden neue Webseiten generiert, von denen aus Sie wiederum durch Anklicken der Hyperlinks direkt zu den entsprechenden Servern wechseln können, welche Seiten enthalten, die Ihre Suchkriterien entsprechen.

Beachten Sie zu den Möglichkeiten der logischen Verknüpfung von Suchbegriffen die Onlinehilfen der einzelnen Suchmaschinen.



### Hinweis

Das Suchergebnis kann, je weniger spezifisch die Abfrage formuliert wurde, leicht Verweise auf Hunderte von Dokumenten ergeben, die (suchmaschinenabhängig) in der Reihenfolge der prozentualen Wahrscheinlichkeit, in der sie Ihrer Spezifikation entsprechen, angezeigt werden.

Sie sollten sich also tatsächlich ein wenig mit der effektiven Handhabung solcher Suchmaschinen vertraut machen, um nicht eine lawinenartige Liste von WWW-Adressen zu erhalten, die bestenfalls am Rande Informationen anbieten, die mit dem von Ihnen gewünschten Thema etwas zu tun haben.

Um hierfür ein Beispiel zu geben: Die Suche nach dem Begriff *Linux* mit der Suchmaschine *AltaVista* ergab Anfang Februar 2001 17.713.697 Einträge gegenüber 4587.344, die im Februar 2000 gefunden wurden – eine Zahl, die, sollten Sie diese Suche wiederholen, in Zukunft noch weiter ansteigen wird.

Einen sehr guten Einstieg in das Thema Suchmaschinen bietet Ihnen die Seite [http://www.leo.org/www/search\\_de.html](http://www.leo.org/www/search_de.html). Sie zeigte sich beim Schreiben dieses Beitrags wie in Abbildung 14.16 dargestellt.



## Mit Linux ins Internet



Abb. 14.16

Links auf Suchmaschinen und so genannte „Meta-Suchmaschinen“



Sie finden hier ebenfalls Links auf so genannte „Meta-Suchmaschinen“, die Suchergebnisse unterschiedlicher Suchmaschinen als Quasi-Schnittmengen zusammenfassen.

## Datentransfer mit FTP

FTP ist die Abkürzung für das **File Transfer Protocol**. Es handelt sich hierbei um das TCP/IP-Internetprotokoll, das benutzt wird, wenn einzelne oder mehrere Dateien von einem Computersystem auf ein anderes übertragen werden.

Normalerweise werden Sie einen Account auf dem Rechner besitzen müssen, auf den Sie zugreifen möchten, wobei hierbei natürlich die gleichen Rechtebeschränkungen für Sie als User bestehen, als würden Sie direkt an diesem System arbeiten.

Für das zuvor Ausgeführte gibt es jedoch eine Ausnahme.

### anonymous FTP

Viele Firmen, Hochschulen und Vertreiber von Software stellen auf ihren Servern die Möglichkeit der Nutzung eines so genannten „anonymous FTP“ zur Verfügung. Hierbei benötigen Sie keinen eigenen Account auf dem Rechner, auf den Sie zugreifen möchten, sondern Sie können sich als Gast (als anonymer Anwender) dort einloggen.



## Datentransfer mit FTP



Der von Ihnen anzugebende Username beim Login ist dann *anonymous* (manche Systeme erlauben auch die Eingabe von *guest* oder *ftp*) und das Passwort, welches Sie angeben müssen, ist ebenfalls *guest* oder (und die Bereitschaft zu dieser Angabe wird in zunehmendem Maße auf vielen Servern von Ihnen verlangt) die Eingabe Ihrer E-Mail-Adresse.

Auch wenn die Eingabe Ihrer E-Mail-Adresse als Passwort auf dem anonymous Server oftmals nicht explizit verlangt wird, ist dies eine Maßnahme, die den Erfordernissen der „Netiquette“ – einem selbst auferlegten Verhaltenskodex innerhalb des Internets – entspricht!



Hinweis

Die Benutzung eines anonymous FTP-Servers ist naturgemäß bezüglich der Ihnen erlaubten Aktionen stark eingeschränkt: So werden Sie in der Regel nicht auf einen solchen schreiben dürfen, und das Leserecht beschränkt sich meist auf ein Verzeichnis */pub* (also ein „public“, ein öffentliches Verzeichnis) mit seinen ihm nachfolgenden Unterverzeichnissen.

### Das Prinzip von FTP

FTP benutzt für die Datenübertragung ein Client/Server-Modell, bei dem ein Clientprogramm auf dem lokalen Computer läuft, das auf einen FTP-Server zugreift, der auf einem Internet-Host läuft.

Es hält alle Werkzeuge bereit, die dazu benötigt werden, um sich den Inhalt von Dateien und Verzeichnissen anzusehen, um in andere Verzeichnisse zu wechseln und um binäre oder ASCII-Dateien von einem Computer zum anderen zu übertragen.

Auch für den FTP-Transfer stellen die LINUX-Distributionen wieder befehlszeilenorientierte Programme zur Verfügung, von denen hier zunächst eines detailliert betrachtet werden soll, da es entschieden zum Verständnis der bei FTP auftretenden Vorgänge beiträgt, bevor ein Programm für X Window vorgestellt wird.

### **ftp**

Ein Befehl, der die Übertragung und den Empfang zu oder von fernen Hosts über das **F**ile **T**ransfer **P**rotocol (FTP) ermöglicht. *ftp* kann dazu verwendet werden, sich in einen Internetcomputer einzuloggen und während der bestehenden Verbindung ASCII- oder Binärdateien zu übertragen.

#### Syntax

Die Syntax für den Befehl lautet:

```
ftp[Optionen] [Hostname]
```

603





Beim Aufruf von *ftp* wird ein Clientprogramm auf Ihrem Computer gestartet, das die Verbindung zum Serverprogramm des entfernten Hosts herstellt. Die mit *ftp* übergebenen Befehle werden in Instruktionen übersetzt, die der Server für Sie ausführt.



### Gefahren von FTP

*Bei der Verbindungsherstellung zu einem FTP-Server muss in der Regel ein Passwort angegeben werden. Die Übermittlung dieses Passworts geschieht in unverschlüsselter Form, so dass gegebenenfalls von Ihnen übertragene Passwörter ausgespäht werden können. Verwenden Sie aus diesem Grund niemals sicherheitsrelevante Passwörter in der Kommunikation mit Rechnern in Fremdnetzen.*

*Als sichere Alternative zu FTP gibt es Secure FTP (sftp), dessen Verwendung somit vorgezogen werden sollte. Es kann analog zu den im Folgenden vor- genommenen Darstellungen verwendet werden.*

### Optionen und Argumente

Beim Aufruf von *ftp* aus der Befehlszeile heraus können die in Tabelle 14.6 dargestellten Optionen verwendet werden. Für *Hostname* kann entweder ein Domainname oder eine IP-Adresse angegeben werden. Außer UNIX-Systemen erlauben viele weitere Computersysteme den Einsatz des **File Transfer Protocol (FTP)** zur Übertragung von Dateien.

### Der Gebrauch des *ftp*-Interpreters

Werden keine Befehlszeilenargumente übergeben, startet *ftp* den Befehlsinterpret er ohne weitere individuelle Anpassungen und wartet darauf, dass Befehle von Ihnen eingegeben werden. Das Standard-Bereitschaftszeichen ist:

*ftp>*

Folgende Optionen stehen zur Verfügung:



## Datentransfer mit FTP



### Optionen für den Gebrauch von *ftp*

| Option | Beschreibung                                                                                                                                                                                                                       |
|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -d     | aktiviert das Debuggen                                                                                                                                                                                                             |
| -g     | schaltet die Möglichkeit der Dateinamensubstitution aus, so dass sie literal gelesen wird                                                                                                                                          |
| -i     | schaltet den interaktiven Prompt während der Übertragung mehrerer Dateien ab                                                                                                                                                       |
| -n     | loggt nicht automatisch bei der ersten Verbindung in das entfernte System ein; wenn diese Option nicht gewählt wird, nimmt <i>ftp</i> an, dass der Login-Name auf dem entfernten System der gleiche wie auf dem lokalen System ist |
| -v     | zeigt alle Antworten sowie die Dateiübertragungsstatistik des entfernten Hosts an                                                                                                                                                  |

Tabelle 14.5



### Befehle in einer aktiven *ftp*-Sitzung

Nachdem eine Verbindung mit *ftp* hergestellt und das gewünschte Verzeichnis auf dem fernen System gefunden wurde, werden nur einige wenige Befehle benötigt, um Dateien über das Internet auf Ihren Computer zu übertragen.

Tabelle 14.6 gibt eine Übersicht über die während einer *ftp*-Sitzung allgemein gebräuchlichen Befehle (auch hier würde die Darstellung sämtlicher Befehle den Rahmen des Buches sprengen).

### Die Befehle von *ftp*

| Befehl               | Beschreibung                                                               |
|----------------------|----------------------------------------------------------------------------|
| Grundlegende Befehle |                                                                            |
| ?                    | zeigt eine Liste aller Befehle von <i>ftp</i>                              |
| ? <i>Befehl</i>      | zeigt Informationen über den spezifizierten <i>Befehl</i>                  |
| !                    | lässt <i>ftp</i> pausieren und startet eine Shell auf dem lokalen Computer |
| ! <i>Befehl</i>      | führt den spezifizierten <i>Befehl</i> auf dem lokalen Rechner aus         |
| bye                  | beendet <i>ftp</i>                                                         |



## Mit Linux ins Internet



### Die Befehle von *ftp* (Forts.)

| Befehl                               | Beschreibung                                                                                                                                                                                 |
|--------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>help</i>                          | zeigt eine Liste aller Befehle von <i>ftp</i>                                                                                                                                                |
| <i>quit</i>                          | beendet die Verbindung mit dem entfernten Computer und schließt <i>ftp</i>                                                                                                                   |
| Verbindungsbefehle                   |                                                                                                                                                                                              |
| <i>account [Passwort]</i>            | übermittelt ein <i>Passwort</i> , das von dem entfernten Computer gefordert wird                                                                                                             |
| <i>open [Hostname]</i>               | baut eine Verbindung zu dem spezifizierten entfernten Computer auf; hierzu muss zunächst eine eventuell bestehende Verbindung mit <i>close</i> geschlossen werden                            |
| <i>close [Hostname]</i>              | beendet die Verbindung mit dem spezifizierten entfernten Computer, aber führt weiter <i>ftp</i> aus                                                                                          |
| <i>disconnect</i>                    | wie <i>close</i>                                                                                                                                                                             |
| <i>user [Name [Passwort]]</i>        | identifiziert den Benutzer am entfernten System; dieser Befehl kann genutzt werden, um sich auf dem System, zu dem bereits eine Verbindung besteht, unter einem anderen Username einzuloggen |
| Verzeichnisbefehle                   |                                                                                                                                                                                              |
| <i>cd [Verzeichnis]</i>              | wechselt in das <i>Verzeichnis</i> des entfernten Computers                                                                                                                                  |
| <i>cdup</i>                          | wechselt aus dem aktuellen Verzeichnis des entfernten Computers in das Elternverzeichnis                                                                                                     |
| <i>delete [Entfernter_Dateiname]</i> | löscht die spezifizierte Datei auf dem entfernten Computer                                                                                                                                   |
| <i>dir [Verzeichnis]</i>             | zeigt den Verzeichnisinhalt des entfernten Computers an                                                                                                                                      |
| <i>lcd [Verzeichnis]</i>             | wechselt vom aktuellen Verzeichnis des lokalen Computers in das spezifizierte <i>Verzeichnis</i>                                                                                             |
| <i>ls</i>                            | zeigt den Inhalt des derzeitigen Verzeichnisses des entfernten Computers an                                                                                                                  |



## Datentransfer mit FTP



### Die Befehle von *ftp* (Forts.)

| Befehl                                                      | Beschreibung                                                                                                                                                                               |
|-------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>pwd</i>                                                  | zeigt den Namen des derzeitigen Verzeichnisses auf dem entfernten Computer an                                                                                                              |
| <i>rmdir [Verzeichnis]</i>                                  | löscht das spezifizierte <i>Verzeichnis</i> auf dem entfernten Computer                                                                                                                    |
| Dateübertragungsbefehle                                     |                                                                                                                                                                                            |
| <i>append [lokaler_Dateiname] [entfernter_Dateiname] cr</i> | schaltet das Entfernen eines Zeilenschaltungszeichens während der Übertragung einer ASCII-Datei an. Hängt eine lokale Datei an die Datei des entfernten Systems                            |
| <i>get [entfernter_Dateiname] [lokaler_Dateiname]</i>       | überträgt <i>entfernter_Dateiname</i> von dem entfernten Computer und benennt ihn um in <i>lokaler_Dateiname</i> auf dem lokalen Computer                                                  |
| <i>mget [entfernte_Dateinamen]</i>                          | überträgt die spezifizierten Dateien von dem entfernten Computer                                                                                                                           |
| <i>mput [lokale_Dateinamen]</i>                             | überträgt die spezifizierten Dateien an den entfernten Computer                                                                                                                            |
| <i>put [lokaler_Dateiname] [erntfernter_Dateiname]</i>      | überträgt die durch <i>lokaler_Dateiname</i> spezifizierte Datei an den entfernten Computer und benennt sie in <i>erntfernter_Dateiname</i> um                                             |
| Befehle zum Setzen der Optionen                             |                                                                                                                                                                                            |
| <i>ascii</i>                                                | setzt den Typ der Dateiübertragung auf ASCII                                                                                                                                               |
| <i>bell</i>                                                 | erzeugt am Terminal nach der Beendigung jedes Befehls einen Ton; eine sehr irritierende Option, die selten benutzt wird                                                                    |
| <i>binary</i>                                               | setzt den Typ der Dateiübertragung auf binär                                                                                                                                               |
| <i>hash</i>                                                 | druckt ein Doppelkreuz (#) für jeden übertragenen Datenblock. Dies ist nützlich, um zu erkennen, dass die Übertragung wie erwartet läuft, besonders, wenn große Dateien übertragen werden. |
| <i>prompt</i>                                               | schaltet auf den interaktiven Prompt um                                                                                                                                                    |
| <i>status</i>                                               | zeigt den derzeitigen Status von <i>ftp</i> an                                                                                                                                             |

Tabelle 14.6



## Mit Linux ins Internet

---



Mit nur fünf Befehlen besitzen Sie bereits ausreichendes Handwerkszeug für den Umgang mit *ftp: ascii, binary, get, put und quit*.

Abhängig von der Sicherheitsstufe im entfernten Computer kann es sein, dass viele der Befehle, die Dateien oder Verzeichnisse löschen oder erzeugen, nicht erlaubt sind. Auch kann die Möglichkeit, Dateien auf den entfernten Computer zu übertragen, eingeschränkt sein.

Ein Beispiel:

Im Folgenden soll eine typische *ftp*-Sitzung beschrieben werden. Zunächst loggt sich der User auf dem entfernten Rechner ein:

```
bash-2.03# ftp infosrv.rz.uni-kiel.de
Connected to infosrv.rz.uni-kiel.de.
220 infosrv.rz.uni-kiel.de FTP server (Version
wu-2.6.0(1) Fri Oct 22 17:30:31 MET DST 1999)
ready.
```

Möchten Sie sich auf dem entfernten Host unter dem gleichen Usernamen einloggen, unter dem Sie auf dem lokalen System angemeldet sind, brauchen Sie an dieser Stelle nur zu drücken, ansonsten müssen Sie den Usernamen eingeben. Da es sich jedoch hier um ein anonymous FTP handelt, muss als Username *anonymous* und im Anschluss die E-Mail-Adresse als Passwort eingegeben werden:

Hinweis



Das Passwort wird beim Eintippen nicht angezeigt.

```
Name (infosrv.rz.uni-kiel.de:root): anonymous
331 Guest login ok, send your complete e-mail address as
password.
Password:
230 Guest login ok, access restrictions apply.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

Nun soll ein Verzeichniswechsel in das Verzeichnis *pub/rz/pc/netscape* erfolgen:

```
ftp> cd pub/rz/pc/netscape
250-Please read the file README
it was last modified on Thu Dec 19 15:42:19 1996 -- 1142
days ago
250 CWD command successful.
ftp>
```



## Datentransfer mit FTP



Mit dem Befehl *ls* lässt sich der Inhalt des aktuellen entfernten Verzeichnisses anzeigen:

```
ftp> ls
200 PORT command successful.
150 Opening ASCII mode data connection for /bin/ls.
total 26452
-r--r--r-- 1 432 775 Dec 19 1996 README
dr-xr-xr-x 8 432 512 May 26 1999 communicator
-r--r--r-- 1 432 3115291 Dec 19 1996 g16e301.exe
-r--r--r-- 1 432 3674112 Dec 19 1996 g32e301.exe
-r--r--r-- 1 432 3070684 Dec 19 1996 n16e301.exe
-r--r--r-- 1 432 3632640 Dec 19 1996 n32e301.exe
226 Transfer complete.
ftp>
```

Zunächst geben Sie (da Sie gleich eine Binärdatei übertragen wollen und sich nicht mehr sicher sind, ob das System standardgemäß auf Binärübertragung eingestellt ist) den Befehl *bin* und, damit Sie den anschließenden Erfolg der Übertragung verfolgen können, den Befehl *hash* ein:

```
ftp> bin
200 Type set to I.
ftp> hash
Hash mark printing on (1024 bytes/hash mark).
ftp>
```

Nun beginnen Sie mit der Dateiübertragung.

*Abb. 14.17* ○ Die Datenübertragung mit *ftp*

```
ftp> hash
Hash mark printing on (1024 bytes/hash mark).
ftp> bin
200 Type set to I.
ftp> get g32e301.exe
local: g32e301.exe remote: g32e301.exe
200 PORT command successful.
150 Opening BINARY mode data connection for g32e301.exe (3674112 bytes).

```

Nach der erfolgreichen Dateiübertragung verlassen Sie das System mit *quit*.

Neben diesem traditionellen Ansatz, Dateien mit FTP zu übertragen, bieten Ihnen die meisten Browser (Lynx, Mosaic, Netscape, Konqueror), viele Editoren (z.B. Emacs) und Datei-Manager (Midnight Commander) ebenfalls die Möglichkeit einer solchen Übertragung.

Bezüglich seines Komforts und seiner Leistungsmerkmale sticht das von Neale G. Smith am Lawrence Livermore National Laboratory entwickelte Programm *xftp* hervor.



## Mit Linux ins Internet



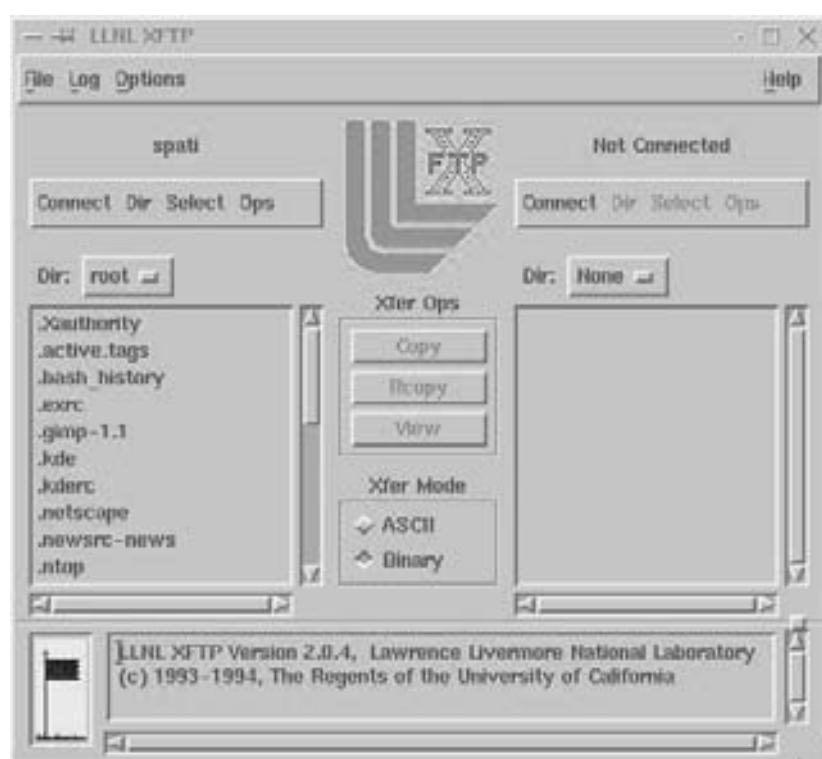
### xftp

Sie finden dieses Programmpaket in der Serie *xap – X Anwendungen* auf der dem Buch beiliegenden SuSE-Distribution.

Wie Sie folgender Abbildung entnehmen können, präsentiert sich *xftp* nach seinem Aufruf grundsätzlich zweigeteilt: Im linken Fenster finden Sie den Inhalt Ihres lokalen Verzeichnisses, das rechte Fenster ist für die Darstellung des Inhalts des aktiven Verzeichnisses des entfernten Systems reserviert (siehe *Abbildung 14.18*).

Abb. 14.18

Das Programm *xftp*



Möchten Sie nun eine Verbindung zu einem entfernten System herstellen, klicken Sie zunächst den Befehl *Connect* an; dieser bietet Ihnen (in Abhängigkeit von der aktuellen Verbindungssituation) folgende Möglichkeiten der Verbindungsherstellung.



## Datentransfer mit FTP

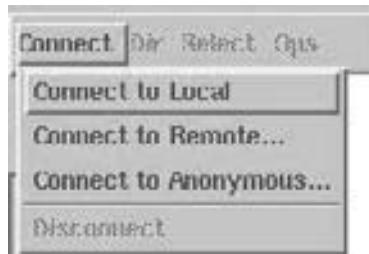


Abb. 14.19

Die Schaltfläche  
#Connect

Die einzelnen Menüpunkte haben hierbei folgende Bedeutungen:

### Unterpunkte des Menüs Connect

| Menüpunkt                   | Funktion                                                                                     |
|-----------------------------|----------------------------------------------------------------------------------------------|
| <i>Connect to Local</i>     | stellt eine FTP-Verbindung zum eigenen Server her                                            |
| <i>Connect to Remote</i>    | stellt eine FTP-Verbindung zu einem entfernten Server her, auf dem man einen Account besitzt |
| <i>Connect to Anonymous</i> | stellt eine Verbindung zu einem Server her, der anonymous FTP erlaubt                        |
| <i>Disconnect</i>           | unterbricht die laufende Verbindung                                                          |

Tabelle 14.7

Wenn Sie nun den Menüpunkt *Connect to Anonymous* wählen, wird folgendes Dialogfenster geöffnet, in dem Sie den *Host Name* entweder manuell eintippen oder, sollten Sie diesen FTP-Host schon einmal besucht haben, durch Anklicken aus einer Liste von Hosts auswählen.

Der *User Name anonymous* ist bereits voreingestellt, und als Passwort sollte an Stelle der Voreinstellung *User@Host* Ihre E-Mail-Adresse eingetragen werden.

Nachdem Sie diese Einstellungen vorgenommen haben, können Sie die Verbindung durch Anklicken der Schaltfläche *Connect* aufbauen.

Auf dem entfernten System können Sie sich nun nach erfolgreichem Verbindungsaubau durch Doppelklicken der Verzeichniseinträge durch den Verzeichnisbaum bis zum gewünschten Verzeichnis „durchhangeln“ oder durch Anklicken der Menüoption *Dir* ein Dialogfenster öffnen, in dem Sie entweder wieder durch Anklicken eines bereits zuvor in einer anderen Sitzung aktiven Verzeichnisses oder durch Eingabe des Verzeichnisnamens den gewünschten Eintrag wählen.



## Mit Linux ins Internet



Abb. 14.20

Das Dialogfenster zur Auswahl eines Hosts für anonymous FTP



Verzeichniswechsel auf dem lokalen System funktionieren nach dem gleichen Prinzip.

Auch in diesem Beispiel wechseln Sie bitte (wie in dem Beispiel zum befehlszeilenorientierten *ftp*) wieder in das Verzeichnis */pub/rz/pc/netscape*, um von dort einen Download durchzuführen.

Bestätigen Sie durch Anklicken der Schaltfläche *OK*, und wählen Sie im Anschluss (ebenfalls wieder durch Anklicken) die zu übertragende(n) Datei(en).



## Datentransfer mit FTP

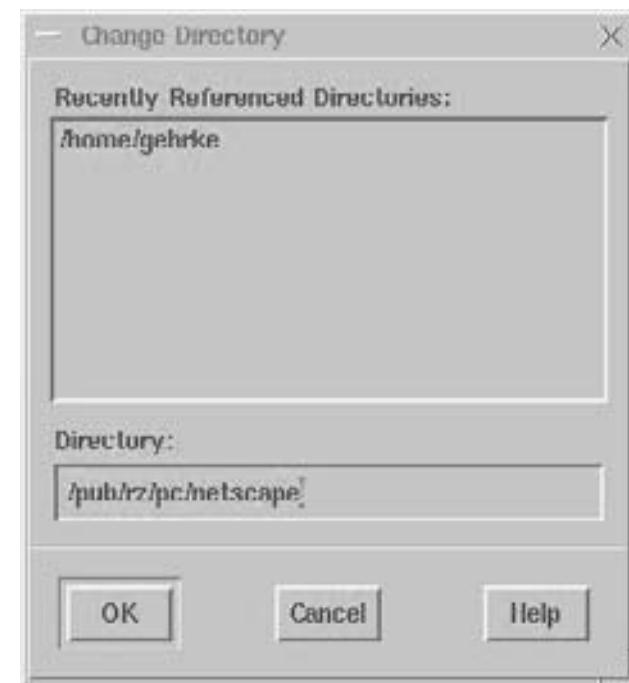


Abb. 14.21

Das Dialogfenster zur Festlegung eines Zielverzeichnisses

Bevor Sie durch Anklicken der Schaltfläche *Copy* die Übertragung einleiten, müssen Sie zwischen den Optionen *ASCII* und *Binary* die gewünschte Übertragungsart wählen.

Achten Sie darauf, den korrekten Übertragungsmodus einzustellen!!!

Sie werden noch einmal aufgefordert, die Liste der ausgewählten Dateien zu bestätigen, woraufhin der Übertragungsvorgang in einem weiteren Meldefenster angezeigt wird; den Fortgang können Sie anhand eines Statusbalkens verfolgen.

Mit diesen einfachen Schritten beherrschen Sie bereits die wichtigsten Funktionen, um mit *xftp* erfolgreich Daten übertragen zu können.

Für eine vollständige Beschreibung des Programms beachten Sie bitte den Handbucheintrag. Geben Sie dazu Folgendes ein:

```
man xftp
```



Hinweis



## Mit Linux ins Internet



### Konqueror

Wenn Sie mit KDE arbeiten, können Sie wie eingangs erwähnt auch den Konqueror als FTP-Client verwenden:

- Öffnen Sie zu diesem Zweck vorzugsweise zweimal den Konqueror, geben Sie in die URL-Eingabezeile des ersten die Adresse des von Ihnen gewünschten FTP-Servers an und wechseln Sie dort durch einfaches Anklicken in das gewünschte Downloadverzeichnis.
- Wechseln Sie im zweiten Konqueror in das gewünschte Zielverzeichnis.
- Ziehen Sie bei gedrückter linker Maustaste den oder die gewünschten Dateien aus dem Quellverzeichnis des FTP-Servers in das Zielverzeichnis Ihres lokalen Rechners.

Abbildung 14.22 zeigt, wie vom Server *infosrv.rz.uni-kiel.de* aus dem Verzeichnis */pub/rz/Linux/redhat/redhat-6.0* eine Datei in das lokale Verzeichnis */export/home/gehrke/downloads* kopiert wird.

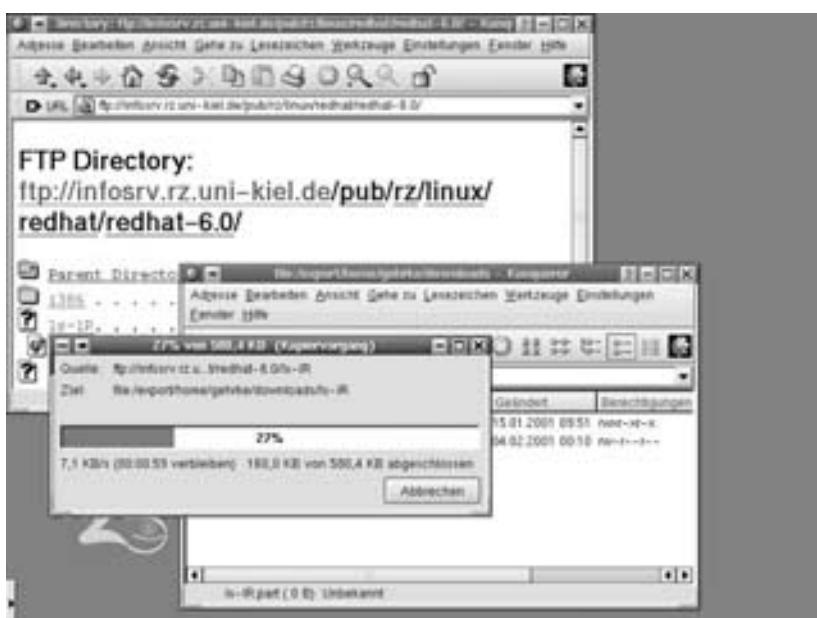


Abb. 14.22

FTP-Dateiübertragung  
mit dem Konqueror

Bei der letzten wichtigen Internetanwendung, die in diesem Kapitel vorgestellt werden soll, handelt es sich um die Möglichkeit, mit der TELNET-Funktion eine Verbindung zu einem UNIX-Rechner herzustellen und an diesem zu arbeiten, als wäre Ihr PC ein an diesen Rechner angegeschlossenes Terminal.

Verwenden Sie hierzu den gleichnamigen Befehl *telnet*.



## ***telnet***



### ***telnet***

Bei *telnet* handelt es sich also um einen Befehl zur Kommunikation mit entfernten Computern, dessen Grundlage das gleichnamige TELNET-Protokoll ist. Dieses ermöglicht das Arbeiten an dem entfernten Host, als wenn man an einem direkt an diesen angeschlossenen Standardterminal säße.

#### **Syntax**

Die Syntax für den Befehl *telnet* lautet:

```
telnet [Hostname[Port]]
```

*Hostname* kann eine symbolische Adresse (eine Domainadresse) oder eine IP-Adresse sein.

*Port* gibt die Nummer des Ports an, über den die Verbindung mit dem Host hergestellt werden soll.



#### **Gefahren von Telnet**

*Bei der Verbindungsherstellung zu einem Telnet-Server muss in der Regel ein Passwort angegeben werden. Die Übermittlung dieses Passworts geschieht in unverschlüsselter Form, so dass gegebenenfalls von Ihnen übertragene Passwörter ausgespäht werden können. Verwenden Sie aus diesem Grund niemals sicherheitsrelevante Passwörter in der Kommunikation mit Rechnern in Fremdnetzen.*

*Als sichere Alternative zu Telnet gibt es die Secure Shell (ssh), deren Verwendung somit vorgezogen werden sollte. Diese kann analog zu den im folgenden vorgenommenen Darstellungen verwendet werden.*

Nach erfolgreicher Verbindungsherstellung werden Ihre Tastatureingaben an den entfernten Host weitergegeben und auf diesem verarbeitet; sämtliche Bildschirmausgaben, die Resultate der Verarbeitung sind, und weitere Eingabeaufforderungen erscheinen auf Ihrem lokalen Bildschirm.

Sie können beim Aufruf von *telnet* entweder direkt den *Hostnamen* angeben, wie in der Syntaxbeschreibung dargestellt, oder Sie verzichten auf die Angabe dieser Adresse als Befehlszeilenargument, wodurch Sie *telnet* im so genannten „Befehlsmodus“ starten; *telnet* zeigt in diesem Fall seine Bereitschaft zur Entgegennahme weiterer Befehle durch seinen Eingabeprompt an:



## Mit Linux ins Internet



telnet>

### Hinweis



Wenn Sie *telnet* unter Angabe eines Hostnamens gestartet haben, können Sie mit der Tastenkombination **Strg+AltGr+9** in den Befehlsmodus umschalten.

Entnehmen Sie die wichtigsten *telnet*-Befehle der *Tabelle 14.8*.

### Die wichtigsten *telnet*-Befehle



| Befehl               | Beschreibung                                                                                                                                                             |
|----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>?</i>             | zeigt eine Liste mit <i>telnet</i> -Befehlen                                                                                                                             |
| <i>close</i>         | beendet die Verbindung zum fernen Host, ohne das <i>telnet</i> -Programm zu beenden                                                                                      |
| <i>display</i>       | zeigt aktuelle Betriebsparameter an                                                                                                                                      |
| <i>open Hostname</i> | stellt eine Verbindung zu dem Rechner <i>Hostname</i> her<br>(hierbei kann es sich sowohl um einen symbolischen Namen als auch um eine IP-Adresse handeln)               |
| <i>quit</i>          | beendet die Verbindung zu dem entfernten Host und schließt das Programm <i>telnet</i>                                                                                    |
| <i>z</i>             | unterbricht die Sitzung zu dem entfernten Host und kehrt zur Shell des lokalen Systems zurück; um die <i>telnet</i> -Sitzung fortzusetzen, können Sie <i>fg</i> eingeben |

*Tabelle 14.8*



Beachten Sie bitte, dass Sie mit *telnet* keine Dateien auf Ihr lokales Computersystem übertragen können – Sie benötigen hierzu ein Programm, welches das FTP-Protokoll verwendet (siehe weiter oben in diesem Kapitel).





# Programmierung



|                      |            |
|----------------------|------------|
| C                    | <b>618</b> |
| Shell-Programmierung | <b>620</b> |
| Perl                 | <b>621</b> |



**15**

## Programmierung

---



Dieser letzte Abschnitt des Buches beschäftigt sich mit dem Thema Programmierung.

Wie bereits erwähnt, handelt es sich bei UNIX ursprünglich um ein Betriebssystem von Programmierern für Programmierer. Diese Tatsache kann man sowohl negativ als auch positiv betrachten:

- Die für den „Normalanwender“ zum Teil mehr als krude Syntax der verschiedenen Befehle, die eine große Anzahl an Leistungsmerkmalen besitzen, so dass sie in manchen Fällen erschlagend wirken, mag vom Anfänger als negativ empfunden werden.
- Der Power-User hat hiermit jedoch eine Schatzkiste an Werkzeugen zur Hand, die, bei richtiger Handhabung, eine immense Arbeitserleichterung für viele Aufgabenbereiche darstellen.

Bezüglich der Handhabung vieler Tools erkennt man aber bereits die Herangehensweise des Programmierers an Problemlösungen: Viele dieser Hilfsmittel sind wiederum weiter programmierbar (es sie nur auf die im *Kapitel 4 Benutzerumgebung* vorgestellte bash und den Rechner *bc* verwiesen).

Daran erkennt man bereits, dass in einer UNIX-Umgebung anscheinend standardmäßig verschiedene Programmierumgebungen integriert sind.

Bevor Sie sich nun konkret einigen Beispielen zuwenden, sollen Sie zunächst einen Überblick über einige populäre Sprachen mit ihren jeweiligen Vorzügen und Nachteilen erhalten.

An (in jeder Hinsicht) erster Stelle sei hier C genannt.

### C

Die erste Version von UNIX wurde 1969 (ohne festen Auftrag hierfür) von Ken Thompson und einigen Kollegen an den Bell Laboratories entwickelt; sie wurde an einem ausgedienten DEC PDP-7-Rechner in Assembler geschrieben und sollte zu Beginn der 70er auf einen leistungsfähigeren Computer, den PDP-11, übertragen werden.

Da das Betriebssystem aber möglichst maschinenunabhängig sein sollte (Assembler-Programme sind prozessorspezifische Programme, die bei der Übertragung auf einen anderen Maschinentyp vollständig an diesen angepasst werden müssen), entschloss man sich, eine höhere Programmiersprache zu entwickeln.

Ken Thompson leitete aus der von Martin Richards entwickelten Sprache BCPL die Sprache B her, die wiederum von Dennis Ritchie überarbeitet und in C umbenannt wurde.





Mit Hilfe dieser Sprache gelang es der Gruppe um Thompson und Ritchie 1973, wesentliche Teile von UNIX in C neu zu programmieren, wobei nur ein (unumgänglicher) Teil in Assembler kodiert wurde.

Das eigentliche Ziel, ein möglichst portierbares Betriebssystem zu entwickeln, wurde durch die Entwicklung von C bei weitem übertroffen: Man besaß jetzt nicht nur eine Entwicklungsumgebung (UNIX), sondern mit dieser ein Entwicklungswerkzeug, das dieser Umgebung optimal angepasst war.

UNIX wäre also nicht ohne C, C nicht ohne UNIX denkbar – und diese Aussage trifft dann natürlich auch auf Linux zu.

Aus diesem Grund werden Sie keine Linux-Distribution finden, zu deren Umfang nicht auch ein C-Compiler gehört. Wie bereits mehrfach erwähnt, handelt es sich bei diesem meist um den GNU-C-Compiler *gcc*.

C ist eine Sprache, die sich sowohl durch einen hohen Grad an Strukturierung und Flexibilität auszeichnet als auch durch die Bereitstellung entsprechender Compiler auf anderen Systemen. Sie ermöglicht, relativ problemlos Programme von UNIX auf diese anderen Systeme zu portieren (zu übertragen).

Eine Standardisierung dieser Sprache trat durch das 1978 von Brian Kernighan und Dennis Ritchie im Verlag *Prentice Hall* veröffentlichte Buch *The C Programming Language* ein, welches auch heute noch als die C-Bibel angesehen wird, und außerdem durch die Normierung durch ein ANSI-Gremium „geadelt“ wurde. Dieses verabschiedete am 31. Oktober 1988 eine entsprechende Norm.

**ANSI:** Das **American National Standard Institute** ist eine in etwa dem deutschen Normenausschuss entsprechende Gesellschaft.



Hinweis

Die Entwicklung dieser Sprache hat dazu geführt, dass sie nunmehr auf (fast) allen Betriebssystemplattformen entweder direkt oder zumindest unter ihrer Beteiligung zur Entwicklung professioneller Software genutzt wird (Standardsoftware, Utilities, Spiele).

Da es sich bei ihr also um eine Sprache von Programmierern für Programmierer handelt, die strukturiert, flexibel und geeignet für die Systemprogrammierung ist, die darüber hinaus im hohen Grade standardisiert, portabel und weit verbreitet ist, wird sich der ernsthafte UNIXianer früher oder später mit ihr auseinander setzen wollen (müssen).

Es handelt sich dabei (aus nahe liegenden Gründen) jedoch nicht um eine Sprache, mit der man sich „mal so eben ein wenig“ befasst.

Aus diesem Grunde wird an dieser Stelle auf eine Einführung verzichtet und stattdessen auf die einschlägige Literatur zu diesem Thema verwiesen, die Ihnen eine ausführliche Einweisung bietet.

Eine Weiterentwicklung der Sprache C, die objektorientiert an die Lösung von Programmierproblemen herangeht, ist die Sprache C++.

## Programmierung

---



### C++

C++ kam in einer ersten Version ca. 1980 als Entwicklung der AT&T Bell Laboratories auf den Markt, hieß jedoch zu dieser Zeit noch „C with Classes“.

In erster Linie stellt es eine Weiterentwicklung von C dar, wobei es in den darauf folgenden Jahren immer mehr verbessert wurde und dadurch in der Zwischenzeit Leistungsmerkmale hinzugewonnen hat, die Sie im traditionellen C nicht finden.

Hierzu gehören in erster Linie das Konzept der objektorientierten Entwicklung und der Versuch, die bestehenden sprachlichen Mittel von C bezüglich ihrer Handhabung und Funktionalität zu verbessern.

C++-Compiler stehen in der Zwischenzeit für alle relevanten Betriebssystemplattformen zur Verfügung, so auch für Linux, unter dem der wiederum von GNU entwickelte *gpp*-Compiler läuft.

Noch weniger als für das traditionelle C empfiehlt es sich, überhaupt den Versuch einer Kurzeinführung für diese hochgradig komplexe Sprache zu wagen. Unsere Empfehlung lautet: Beschäftigen Sie sich zunächst mit dem traditionellen C, bevor Sie den Sprung ins kalte Wasser der Objektorientierung wagen, oder versuchen Sie sich zunächst einmal an den in diesem Buch vorgestellten „einfacheren“ Programmiersprachen.

Hier ist als Erstes das sehr leistungsfähige Instrument zur Programmierung, die schon mehrfach erwähnte Shell-Programmierung zu nennen.

## Shell-Programmierung

Bei Shell-Programmen („Skripten“ oder „Scripts“) handelt es sich um Programme, die unter Verwendung

- der Ihnen bereits bekannten Befehle,
- interner Befehle der Shell,
- von Kontrollanweisungen der Shell,
- von Shell-Variablen

erstellt werden.

Wenn Sie ein erfahrener DOS-Anwender sind, ist Ihnen das Prinzip der Shell-Programmierung bereits von den Stapeldateien (*Batches*) her bekannt. Auch bei diesen handelt es sich um Textdateien, die eine Zusammenfassung von Befehlen enthalten, die Sie auch direkt am Prompt eingeben könnten, deren Zusammenfassung jedoch die Abarbeitung sich häufig wiederholender gleicher Abläufe stark vereinfacht.





Sie bieten darüber hinaus durch die in ihnen verwendbaren Variablen und ihre verschiedenen Kontrollstrukturen (wie zum Beispiel Verzweigungen, Schleifen usw.) die Möglichkeit, so flexibel programmiert zu werden, dass sie auch variablen Ansprüchen und Situationen genügen.

Gerade für den Systemverwalter stellt die Shell-Programmierung ein unersetzliches Hilfsmittel dar, so dass das Kapitel 16 *Shell-Programmierung* ausschließlich dieser Programmiertechnik gewidmet ist.

Ein Nachteil der Shell-Skripte liegt darin, dass es sich bei ihnen um „interpretierte“ Programme handelt. Das heißt, dass der Programmcode bei jedem Programmaufruf neu interpretiert (übersetzt) werden muss, was durch die bash geschieht (oder den alternativ hierzu von Ihnen verwendeten Befehlszeilen-Interpreter). Dieser Übersetzungsprozess nimmt natürlich (Rechen-)Zeit in Anspruch, so dass die Ausführung sehr großer Shell-Skripte im Gegensatz zu kompilierten Programmen (wie zum Beispiel C- oder C++-Programmen) zeitintensiver ist.

Nichtsdestotrotz gilt das im vorherigen Absatz Geschriebene: Die Shell-Programmierung ist das wichtigste Hilfsmittel des Systemverwalters.

Eine Programmiersprache, die sich immer größerer Beliebtheit erfreut, ist *Perl*.

## Perl

*Perl* ist eine Programmiersprache, die hervorragend für die Manipulation von Texten, Dateien und Prozessen geeignet ist.

Speziell für die Zeichenkettenmanipulation (*String-Manipulation*) bietet sie Werkzeuge mit Funktionalitäten, die bisher (mit C oder der Shell-Programmierung) nur sehr schwer und umständlich realisiert werden konnten.

Gerade dieses Leistungsmerkmal macht Perl speziell im Bereich des World Wide Web immer beliebter: Webseiten können Formularelemente enthalten, die Eingaben in die HTML-Seite erlauben. Diese Eingaben müssen dann auf der Seite des Servers interpretiert werden, woraufhin häufig dynamisch eine neue Seite generiert und an den Client übergeben wird.

Diese Schnittstelle zwischen Webclient und Webserver wird auch als „**Common Gateway Interface**“ (*CGI*) bezeichnet.

Traditionell finden Sie häufig noch C-Programme als CGI. Immer häufiger wird jedoch auf Perl zurückgegriffen, da sich das Erstellen von Perl-Programmen wesentlich einfacher gestaltet als das Entwickeln von C-Routinen, so dass in vielen Fällen nur noch darauf zurückgegriffen wird, wenn die Grenzen Perls erreicht sind oder sich das Laufzeitverhalten als zu langsam erweist. Dies ist zum Beispiel beim Zugriff auf SQL-basierte Datenbanken der Fall, bei dem sich in C programmierte Abfrageroutinen durch die Integration von so genannten „Embedded SQL-Routinen“ (eingebettetes SQL) als sehr effektiv erweisen.



## Programmierung

---

### Hinweis



Structured Query Language (**SQL**). Relationale Datenbanken können mit dieser leistungsfähigen Abfragesprache, die stark an die natürliche Sprache angelehnt ist, manipuliert werden.

Wie Sie jedoch in den *Kapiteln 17 Programmierung in Perl* und *18 HTML und CGI* sehen werden, erlaubt die DBI-Schnittstelle ebenfalls einen sehr eleganten Zugriff auf relationale Datenbanken mithilfe von Perl.

Auch für Perl gilt, was zuvor über Shell-Skripte gesagt wurde: Es handelt sich ebenfalls um interpretierte Programme, die von ihrem Laufzeitverhalten natürlich nicht an die Geschwindigkeit von kompilierten oder assemblierten Programmen herankommen.

Ihr Sprachumfang (und ihre Verfügbarkeit auf nahezu allen Betriebssystem-Plattformen – so gibt es zum Beispiel in der Zwischenzeit auch Perl-Interpreter für Win/DOS und Win-98-Rechner, Apple usw.), ihre Flexibilität (Perl bietet oftmals mehrere Möglichkeiten zur Lösung ein und desselben Problems) und ihre relativ leichte Erlernbarkeit machen diese Sprache jedoch immer populärer.

Aus diesem Grund wird Ihnen im *Kapitel 17 Programmierung in Perl* eine Einführung in Perl geboten, die Ihnen einen „Vorgeschmack“ auf diese Sprache vermitteln soll. Anders als bei der Skriptprogrammierung, für die Sie nach Durcharbeitung der *Kapitel 3 Linux-Dateisystem*, *Kapitel 4 Benutzerumgebung* und *Kapitel 16 Shell-Programmierung* das wesentliche Rüstzeug besitzen, muss aber nochmals betont werden: Das *Kapitel 17 Programmierung in Perl* ist wirklich nur ein Vorgeschmack. Weiterführende (und auch wesentliche) Funktionalitäten, wie zum Beispiel die Dateiverwaltung, würden den Rahmen und den Ansatz dieses Buches entschieden sprengen, so dass Sie auch hier auf die entsprechende Fachliteratur zurückgreifen sollten.

Das Gleiche gilt für die (unter X) verwendete Programmiersprache *Tcl*.

### **Tcl/Tk**

Ein weiteres Beispiel für die große Zahl der unter UNIX (Linux) zur Verfügung stehenden Programmiersprachen ist *Tcl* mit seiner Erweiterung *Tk*.

Es handelt sich dabei ebenfalls um eine interpretierende Programmiersprache (der Interpreter heißt *Tcl-Shell*, kurz: *tclsh*), die sich gegenüber der Shell- und der Perl-Programmierung dahingehend abhebt, dass mit ihr

- (relativ) mühelos X-Programme erstellt werden können,
- ohne große Probleme C-Routinen verwendet werden können, die dann natürlich erheblich zur Geschwindigkeitssteigerung in der Verarbeitung beitragen.





Diese Eigenschaften haben *Tcl/Tk* zu der (neben Perl) wohl zurzeit populärsten Programmiersprache in der UNIX-Welt werden lassen. So bauen beispielsweise mittlerweile viele Installations- und Konfigurationsskripte von Linux-Distributionen auf dieser Sprache auf (wie zum Beispiel von Caldera oder Red Hat oder das Konfigurationsskript für den Kernel *make xconfig*).

Diese Popularität hat ebenfalls Auswirkungen auf die Nicht-UNIX-Welt: So sollen beispielsweise *Tcl/Tk*-Implementationen sowohl für die Windows- als auch für die Apple-Welt erscheinen.

## **Java**

Immer größerer Beliebtheit erfreut sich ebenfalls die von Sun Microsystems entworfene Programmiersprache Java.

Es handelt sich dabei um eine einfache, objektorientierte, netzwerktaugliche, interpretierte, robuste, sichere, architekturneutrale, portable Sprache, die sehr leistungsfähige Merkmale enthält.

Sie wurde entwickelt, um Anwendungen des World Wide Web interaktiv nutzen zu können, wobei sie sich bezüglich ihrer Syntax stark an C++ anlehnt, jedoch auf viele Leistungsmerkmale (die für die Lösung nur sehr spezieller Aufgabenstellungen benötigt werden und somit für einfachere Aufgaben nur eine Überfrachtung der Sprache darstellen) verzichtet.

Eine der wesentlichen Anforderungen an die Entwickler dieser Sprache war es, sie vom Sprachumfang möglichst klein zu halten, aber auch sehr kompakte Codes mit ihr erstellen zu können, da die in ihr entwickelten Programme (die so genannten „Applets“) ja schließlich über das Internet geladen werden müssen.

Java wird sicherlich in nächster Zeit zunehmend an Bedeutung gewinnen. Ob jedoch die Pläne von Sun fruchten werden, das Ende des PCs einzuläuten und diesen durch einen Java-basierten Netzwerk-PC zu ersetzen, der sich seine Applikationen grundsätzlich aus dem Netz holt, bleibt abzuwarten – wie man persönlich zu diesem Konzept steht, ist noch eine andere Frage.

## **Andere Sprachen**

Mit den meisten Linux-Distributionen bekommen Sie über diese hier vorgestellten Programmiersprachen hinaus eine Vielzahl an Compilern und Interpretern mitgeliefert, so dass sich eigentlich für jeden etwas finden sollte.

Exemplarisch soll hier auf die wichtigsten hingewiesen werden; so stehen Ihnen zur Verfügung:



## Programmierung

---



- ein 8086-Assembler (*bin86*)
- ein (man schnalle sich an, stelle die Sitzlehne senkrecht und das Rauchen ein ... ) BA-SIC-Interpreter (*bwbasic*)
- ein GNU *Fortran*-Compiler (*g77*)
- ein GNU *Ada*-Compiler (*gnat*)
- ein GNU *Objective C*-Compiler (*gobjc*)
- Linux kann auch als Wirtsumgebung von *PASCAL* dienen: Es gibt einen GNU *PASCAL*-Compiler (*gpc*) sowie das plattformübergreifende und Turbo-kompatible Free Pascal
- die Sprache, die in aller Munde ist (obwohl manche sie für kalten Kaffee halten ...): das *Java Development Kit* (*java*)
- ein *Modula-2* Compiler (*mocka*)
- ein Interpreter mit einer C-ähnlichen Syntax (*slang*)
- die Programmiersprache *SmallEiffel* (*smeiffel*)

Diese Programmiersprachen, bei denen Sie im Allgemeinen zunächst den Quellcode mit einem Editor Ihrer Wahl erstellen und den Objektcode dann befehlszeilenorientiert erstellen müssen, wobei Sie hierbei (wie sollte es in einem UNIX-System auch anders sein) auch oft mit haarsträubenden Syntaxvarianten konfrontiert werden, werden dann zum Teil unterstützt von Entwicklungsumgebungen, die Ihnen unter X zur Verfügung stehen. Hierzu gehören zum Beispiel:

- eine Entwicklungsumgebung à la *Borland C* (*xwpc*)
- ein grafisches User-Interface für den GNU-Debugger (*xxgdb*)
- das Programm *InterView*: eine Umgebung für die Entwicklung von plattformübergreifenden grafischen Oberflächen (*iv*)
- der Nachfolger von *InterView*: *Fresco*
- eine grafische Oberfläche im *OpenWin-Look*; diese unterstützt Sie beim Edieren, Kompilieren und Starten von C-Programmen (*xwb*)

Genug der vielen (theoretischen) Worte; lasst Daten sprechen ... Im folgenden *Kapitel 16 Shell-Programmierung* werden die Möglichkeiten der Skriptprogrammierung betrachtet.





# Shell-Programmierung



|                              |            |
|------------------------------|------------|
| Struktur eines Shell-Skripts | <b>627</b> |
| Starten von Shell-Skripten   | <b>628</b> |
| Ein-/Ausgabefunktionen       | <b>630</b> |
| Argumentwerte                | <b>632</b> |
| Kontrollstrukturen           | <b>635</b> |
| Verzweigungen                | <b>639</b> |
| Schleifen                    | <b>645</b> |
| Dialoge                      | <b>655</b> |



**16**

## Shell-Programmierung

---



Wie bereits im *Kapitel 4 Benutzerumgebung* angedeutet, ist ein effektiver Weg, UNIX- (und damit Linux-)Befehle einzusetzen, diese miteinander zu verknüpfen. Eine der Grundideen der UNIX-Entwickler war die Bereitstellung vieler kleiner, einfacher Werkzeuge, die sich miteinander verbinden lassen, mit der Wirkung, dass auch hier das Ganze mehr ist als die Summe seiner Einzelteile.

Diese „Verknüpfung“ haben Sie z.B. in Form von Pipelines oder Substitutionen bereits kennen gelernt. Wer jedoch komplexere Pipelines häufiger verwenden möchte, müsste diese immer wieder neu eintippen, was zum einen mühsam, zum anderen aber auch problematisch ist, da sich gerade beim „Vieltippen“ Fehler einschleichen.

Nehmen Sie eine Befehlszeile wie die folgende – sie durchsucht die *man pages* des Editors joe nach dem Suchbegriff *block*, wobei die Schreibweise (Groß-/Kleinschreibung) ignoriert wird und das Ergebnis sowohl auf dem Bildschirm erscheint als auch in eine Datei mit dem Namen *gefunden* umgeleitet wird.

```
man joe | grep -i block | tee gefunden
```

Sicherlich wäre es praktischer, einfach eingeben zu können:

```
suche block in joe
```

Noch eleganter wäre es natürlich, wenn diese Suche auf andere Begriffe in anderen *man pages* ausgedehnt werden könnte.

An dieser Stelle helfen Shell-Skripte.

Es handelt sich bei ihnen um (Text-)Dateien, die Abfolgen von Befehlen enthalten, die Sie normalerweise am Prompt eingeben können. Diese Möglichkeit, Befehlsfolgen zusammenzufassen, ist ein entscheidender Vorteil von Shell-Skripten. Dabei kommt als weiterer positiver Aspekt hinzu, dass solche Abfolgen nicht einfach von oben nach unten abgearbeitet werden müssen, sondern dass durch Kontrollstrukturen die Reihenfolge der Abarbeitung der Befehle im Shell-Skript in Abhängigkeit von Bedingungen geschieht, die *Sie* festlegen können.

Mit anderen Worten: Shell-Skripte stellen eine einfache Programmiermöglichkeit dar.

Gerade in der Systemverwaltung (und wahrscheinlich sind *Sie* die Person, die das Linux-System, mit dem *Sie* gerade arbeiten, verwalten) fallen Routinearbeiten an, die einerseits so komplex sind, dass sie eine einfache Abarbeitung mit zwei oder drei Befehlen nicht erlauben, andererseits aber keine großartigen neuen Hilfsprogramme erfordern, die *Sie* mithilfe einer höheren Programmiersprache, wie z.B. C, selbst erstellen müssen.

Das System bietet Ihnen ausreichend Werkzeuge, mit deren Hilfe *Sie* durch die Erstellung entsprechender Shell-Skripte mehr oder weniger komplexe Arbeiten elegant erledigen können.



## Struktur eines Shell-Skripts



Zu diesen Werkzeugen gehören:

- Ein-/Ausgabebefehle
- Shell-Variablen und Positionsparameter
- Kontrollstrukturen
- weitere interne Shell-Befehle

Darüber hinaus können Sie jeden beliebigen weiteren (externen) Linux-Befehl sowie jedes weitere Ihnen zur Verfügung stehende Linux-Programm im Skript verwenden.

## Struktur eines Shell-Skripts

Shell-Skripte sind, wie bereits erwähnt, Textdateien, d.h., Sie können sie mithilfe eines beliebigen Editors erstellen. Natürlich geht dies auch mit

```
cat > datei
```

wie im *Kapitel 3 Linux-Dateisystem* im Abschnitt „*cat*“ dargestellt.

Die Shell liest diese Datei bei der Ausführung, wobei die Zeilen wie Eingaben am Prompt interpretiert werden.

Sie liest die einzelnen Zeilen von oben nach unten, d.h., bei der Programmierung ist die Reihenfolge der Befehle dahingehend zu beachten, dass ein Befehl, der in einem Skript vor einem anderen steht, auch zuerst ausgeführt wird.

Bei der Abarbeitung eines Programms von oben nach unten spricht man auch von einer „sequenziellen Abarbeitung“.

Grafisch kann man diese „Struktur der Reihung“ (Sequenz) mithilfe eines so genannten „Struktogramms“ wie folgt darstellen.



Hinweis

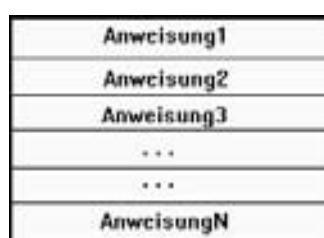


Abb. 16.1

Die Struktur der Reihung (Sequenz)

627



## Shell-Programmierung

---



Sie kann in ihrer Reihenfolge der Abarbeitung nur mithilfe von Kontrollstrukturen beeinflusst werden. Zur besseren Lesbarkeit eines Skripts können Sie beliebig viele Leerzeilen einfügen, wie Sie auch am Prompt einfach ein Ü eingeben können.

Sollte eine Befehlszeile länger als die Eingabezeile (des von Ihnen verwendeten Editors) sein, können Sie diese mit einem Backslash (\) am Ende der Zeile umbrechen, so dass sie korrekt von der bash als „durchgehende“ Befehlszeile interpretiert werden kann.

Wie in anderen Programmiersprachen erlaubt es Ihnen die Shell-Programmierung, Kommentare einzufügen. Ein solcher Kommentar wird durch ein Doppelkreuz (#) eingeleitet und endet mit dem Zeilenende.

Wenn Skripte in einem eigenen Verzeichnis stehen sollen, beachten Sie, dass Sie dieses Verzeichnis

- gegebenenfalls einrichten,
- im Suchpfad integrieren.

Sie sollten von der Möglichkeit des Kommentierens reichlich Gebrauch machen, um nicht nur sich selbst das Verständnis eines Skripts in der Zukunft zu vereinfachen, sondern auch anderen Personen, die das Skript benutzen oder eventuell bearbeiten sollen. Skripte werden häufig geschrieben, um mehreren Benutzern ein Werkzeug für ihre tägliche Arbeit zur Verfügung zu stellen. Ein schönes Beispiel hierfür ist das Installationsskript der SuSE-Linux-Distribution (YaST).

Wie Sie bereits im *Kapitel 3 Linux-Dateisystem* gesehen haben, zeichnet sich ein UNIX-System durch eine spezifische Verzeichnisstruktur aus. Dem sollten Sie beim Speichern Ihrer Shell-Skripte Rechnung tragen, indem Sie „persönliche“ Skripte, die nur für Sie selbst von Interesse sind, in einem speziell hierfür eingerichteten Verzeichnis Ihres Heimatverzeichnisses abspeichern. Gemäß Konvention ist dies das Verzeichnis \$HOME/bin (in der bash auch ansprechbar als ~/bin).

Skripte, die von allgemeinem Interesse sind, kann man (die entsprechenden Schreibrechte vorausgesetzt) im Verzeichnis /usr/local/bin speichern, das standardmäßig im Suchpfad integriert ist.

## Starten von Shell-Skripten

Grundsätzlich stehen drei verschiedene Möglichkeiten des Skriptaufrufs zur Verfügung:

- der implizite Aufruf,
- der explizite Aufruf,
- der Aufruf mithilfe des . (Punktes).



## Starten von Shell-Skripten



Der *implizite Aufruf* erfolgt, indem der Name des Skripts am Prompt eingegeben wird. Dies setzt voraus, dass der aufrufende Benutzer sowohl das Recht zum Lesen als auch das Recht zum Ausführen dieser Datei besitzt.

Verwenden Sie zur Rechtevergabe den Befehl *chmod*.

Die bash startet eine Sub-Shell, die das Skript liest und interpretiert. Nach Abarbeitung erfolgt die Rückkehr zur aufrufenden Shell.



Hinweis

Der *explizite Aufruf* erfolgt durch:

```
shell skript_name
```

wobei *shell* in der Regel in einem Linux-System die bash sein wird. Prinzipiell ist aber auch die Angabe einer beliebigen anderen Shell denkbar, wobei zu beachten ist, dass die internen Befehle der verschiedenen Shells voneinander abweichen können. In einem solchen Fall sollten nur solche Skripte Verwendung finden, die keine Shell-spezifischen Befehle enthalten, da deren Abarbeitung durch eine andere Shell nicht erfolgen kann.

Man beachte, dass für einen expliziten Aufruf kein Ausführungsrecht für die Skriptdatei notwendig ist.



Hinweis

Der Aufruf mithilfe des Punkts erfolgt durch Eingabe von:

```
. skript_name
```

Dabei wird keine Sub-Shell gestartet, d.h., das Skript wird in der aktuellen Shell abgearbeitet. Dies bedeutet unter anderem, dass Shell- (Umgebungs-)Variablen, die durch das Skript einen neuen Wert erhalten, in der aufrufenden Shell geändert werden.

Im Umkehrschluss heißt dies, dass eine Änderung von Variablen durch ein Shell-Skript, das nicht mit dieser Methode gestartet wird, eine Variablenänderung innerhalb einer Sub-Shell ist, die ihre eigene Systemumgebung hat.

Nach Beendigung des Skripts und damit der Sub-Shell sind deren Umgebungsvariablen nicht länger verfügbar. Um dies zu verdeutlichen, erstellen Sie ein einfaches Skript *neu\_pfad*, das die Shell-Variable *PATH* ändert, mit folgendem Inhalt:

```
echo Hier erfolgt die Pfadaenderung in der
echo Sub-Shell
PATH=/dev
echo Ausgabe des neues Suchpfades:
echo $PATH
echo Skript-Ende
```

Beim Aufruf dieses Skripts (ohne Punkt!) erhalten Sie folgende Bildschirmausgabe.

629



## Shell-Programmierung

### Hinweis



Um zu vermeiden, dass Ihre Arbeitsumgebung nachhaltig geändert wird, sollten Sie einen Aufruf dieses Skripts mit Punkt unterlassen!

The screenshot shows a terminal window titled "TNT - Lisserv, to host 134.245.35.7". The window contains the following text:  
Session Edit Commands Settings Help  
/g2pcc:/home/gehrke/scripts > neu\_pfad  
Hier erfolgt eine Pfadänderung in der Sub-Shell  
Ausgabe des neuen Suchpfads:  
/dev  
Skript-Ende  
F92pcc:/home/gehrke/scripts > █

Abb. 16.2

Bildschirmausgabe des Skripts *neu\_pfad*

Dabei stellt die dritte Zeile den neuen Inhalt der Variablen *PATH* der Sub-Shell dar.

Um zu überprüfen, dass diese Variablenzuweisung auf die aktuelle Shell keinen Einfluss genommen hat, geben Sie ein:

```
echo $PATH
```

Der für Sie aktuell gültige Zugriffspfad dürfte durch die Abarbeitung des zuvor vor gestellten Skripts keine Veränderung erfahren haben (was in diesem Fall ja auch so gewollt war).

## Ein-/Ausgabefunktionen

Viele Skripte werden erstellt, um langwierige, komplexe Prozesse im Hintergrund ablaufen zu lassen. In diesen Fällen ist eine Interaktion zwischen dem User und dem Skript nicht nur unerwünscht, sondern sogar störend.

In anderen Fällen jedoch ist eine Interaktion nicht nur erwünscht, sondern vielleicht sogar erforderlich (z.B. in der Form zusätzlicher Benutzereingaben, Bildschirrmeldungen usw.).

Hierzu stellt Linux verschiedene Instrumente zur Verfügung, von denen Sie den Befehl zur Bildschirmausgabe bereits in seiner Grundfunktion kennen gelernt haben.

### ***echo***

Der Befehl *echo* schreibt eine Zeichenkette auf die Standardausgabe.

Bei dieser Zeichenkette kann es sich sowohl um ein einzelnes Zeichen als auch tatsächlich um mehrere Zeichen handeln.



## Ein-/Ausgabefunktionen



### Syntax

Die Syntax für den Befehl *echo* lautet:

```
echo [-n | -e] Zeichenkette
```

### Beschreibung

Der Befehl *echo* schreibt eine Zeichenkette auf die Standardausgabe, wobei der Wert *o* zurückgegeben wird, wenn er erfolgreich abgeschlossen werden konnte. Ist dies nicht der Fall, wird ein Wert größer *o* zurückgegeben.

Folgende Besonderheiten sind bei der Verwendung dieses Befehls zu beachten:

- Enthält die Zeichenkette Sonderzeichen, muss sie in einfache oder doppelte Anführungszeichen gesetzt werden.
- Der Zeichenkette wird bei der Ausgabe ein Zeilenvorschub angefügt; dieser kann durch den Parameter *-n* ausgeschaltet werden.
- Die Verwendung des Befehls *echo* ohne Zeichenkettenargument bewirkt die Ausgabe einer Leerzeile.

Zur weiteren Formatierung von Zeichenketten können die in *Tabelle 16.1* beschriebenen so genannten „Escape-Sequenzen“ verwendet werden. Um diese nutzen zu können, muss der Parameter *-e* in die Befehlszeile eingegeben werden, z.B.:

```
echo -e "Hier folgt ein Warnton\b"
```

### Die Escape-Sequenzen



| Sequenz      | Bedeutung                                                       |
|--------------|-----------------------------------------------------------------|
| \a           | gibt einen Warnton („alarm“) aus                                |
| \b           | erzeugt ein Backspace                                           |
| \c           | unterdrückt den Zeilenvorschub                                  |
| \f           | erzeugt einen Seitenvorschub                                    |
| \n           | erzeugt einen Zeilenvorschub                                    |
| \r           | erzeugt einen Return                                            |
| \t           | erzeugt einen Tabulatorschritt                                  |
| \v           | erzeugt einen vertikalen Tabulatorschritt                       |
| \\\          | zeigt den Backslash (\) an                                      |
| \o Oktalzahl | gibt das durch <i>Oktalzahl</i> spezifizierte ASCII-Zeichen aus |

Tabelle 16.1

## Shell-Programmierung

---



Um Werte in ein Skript einlesen zu können, bietet Linux die Eingabefunktion *read*.

### ***read***

Der Befehl *read* liest eine Zeile aus der Standardeingabe.

#### **Syntax**

```
read [-r] Variable_1 ... Variable_n
```

#### **Beschreibung**

*read* liest eine Zeile aus der Standardeingabe und weist die Werte jedes Eingabeelements einer Variablen zu. *Variable\_1* bis *Variable\_n* spezifizieren die Namen der Shell-Variablen, denen die Werte der Eingabeelemente zugewiesen werden.

---

Beachten Sie hierzu auch die Ausführungen zu *expr* weiter unten in diesem Kapitel

Man beachte, dass es sich bei den eingegebenen Werten prinzipiell um Texteingaben handelt. Soll die Eingabe beispielsweise als nummerischer Wert verarbeitet werden, muss sie zunächst entsprechend „aufbereitet“ werden.

Enthält die Eingabezeile mehr Elemente als Variablennamen vorgegeben wurden, werden die noch verbleibenden Elemente der letzten Variablen zugewiesen.

Der Parameter *-r* bewirkt, dass bei Eingabe eines Backslashes dieser als Bestandteil der Eingabezeile und nicht als Steuerzeichen interpretiert wird.

Wurde der Befehl erfolgreich abgeschlossen, gibt er den Wert *o* zurück. Ein Wert größer *o* wird zurückgegeben, wenn das Zeichen EOF (End Of File – Dateiendemarke) für den Gebrauch mit dem Befehl *echo* entdeckt wurde oder ein Fehler aufgetreten ist.

Eine weitere Möglichkeit, Werte an Shell-Skripte zu übergeben, liegt in der Verwendung von Argumentwerten.

## Argumentwerte

Argumentwerte oder „Positionssparameter“ dienen dazu, bereits beim Aufruf des Skripts Werte an das Skript zu übergeben.

Zu Beginn dieses Kapitels wurde der Entwurf eines Skripts mit dem Namen *suche* gefordert. Dieses soll bei Eingabe der Befehlszeile

```
suche block in joe
den Befehl
man joe | grep -i block | tee gefunden
```



## Argumentwerte



ausführen.

Um dies realisieren zu können, bedienen Sie sich der Befehlszeilenargumente und verdeutlichen dies am folgenden Beispiel einer Befehlszeile:

```
cp /dev/null /home/gehrke/tmp
```

Mit dieser Befehlszeile wird der Inhalt der Datei */dev/null* in die Datei */home/gehrke/tmp* kopiert.

Die verschiedenen Elemente dieser Befehlszeile können bezüglich ihrer Position spezifiziert werden:

Wie bereits im *Kapitel 4 Benutzerumgebung* dargestellt, erfolgt eine Referenzierung von Variablen durch ein vorangestelltes Dollar-Zeichen. Es können bis zu neun Befehlszeilenargumente übergeben werden, die Sie mit \$1 bis \$9 referenzieren.



Hinweis

- *cp* – der eigentliche Befehl. Er nimmt die erste Position dieser Befehlszeile ein. Diese Position wird durch die Zahl 0 repräsentiert.
- */dev/null* – die Quelldatei. Sie stellt das erste Argument zum Befehl *cp* dar. Ihre Position wird durch die Zahl 1 repräsentiert.
- */home/gehrke/tmp* – die Zielfile. Sie stellt das zweite Argument zum Befehl *cp* dar. Ihre Position wird durch die Zahl 2 repräsentiert.

Um sich die Gesamtzahl der in einer Befehlszeile enthaltenen Argumente anzeigen zu lassen, können Sie diese mit \$# referenzieren.

\$0 ist immer der Befehl selbst!



Hinweis

Eine Liste der in einer Befehlszeile enthaltenen Argumente erhalten Sie mit \$\*.

Entwickeln Sie nun ein Skript mit dem Namen *kopiere*, das folgenden Inhalt hat:

```
cp $1 $2
```

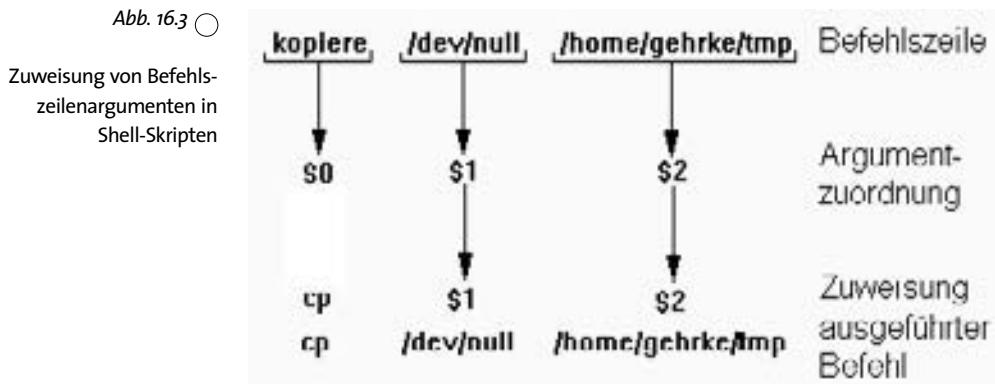
Beim Aufruf der Befehlszeile

```
kopiere /dev/null /home/gehrke/tmp
```

erfolgt die in *Abbildung 16.3* dargestellte Zuweisung der Befehlszeilenargumente.



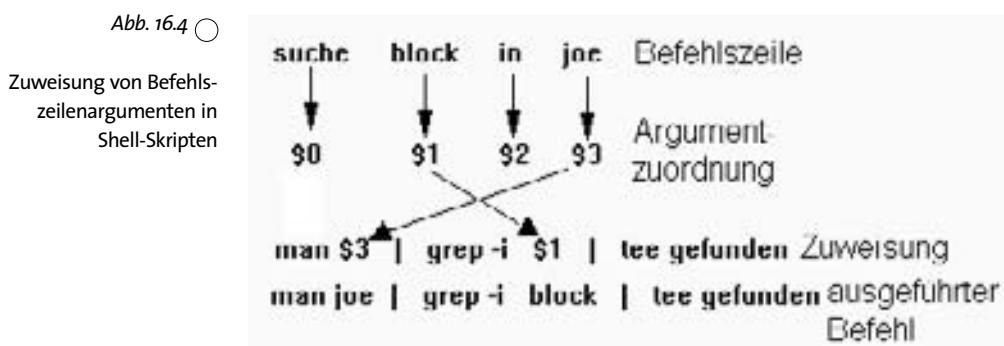
## Shell-Programmierung



Zurück zu dem Skript mit dem Namen *suche*. Mit der neu gewonnenen Programmiertechnik der Argumentübergabe können Sie dieses nun wie folgt formulieren:

```
man $3 | grep -i $1 | tee gefunden
```

Beim oben beschriebenen Aufruf finden nun folgende Zuweisungen statt.



Testen Sie einmal,  
was geschieht,  
wenn Sie das Skript  
ohne das Füllwort  
„in“ aufrufen

Beachten Sie bei diesem Beispiel, wie die den Konventionen der deutschen Grammatik entsprechende Syntax des Suchbefehls mithilfe der Positionsparameter den Konventionen der Linux-Syntax entsprechend „umgebogen“ werden. Das Befehlszeilenargument 2 (das Wort „in“) wird bei der Abarbeitung des Skripts vollständig ignoriert – es dient ausschließlich dazu, dem Anwender die Eingabe einer möglichst „natürlichen“ Befehlszeile zu ermöglichen.





## Kontrollstrukturen

Damit ein Programm variabel auf verschiedene Eingaben oder Parameter reagieren kann, bedient man sich so genannter „Steueranweisungen“ oder „Kontrollstrukturen“ (*control statements*).

Man unterscheidet hierbei zwischen Kontrollstrukturen, die ein mehrmaliges Abarbeiten einzelner Programmteile (Wiederholungsanweisungen oder Schleifen) oder die Durchführung bestimmter Aktionen in Abhängigkeit von bestimmten Bedingungen (Entscheidungsanweisungen, Verzweigungen oder Abfragen) erlauben.

In jedem Fall bedarf es eines Instrumentariums, so genannte „bedingte Ausdrücke“ auszuwerten. Linux (wie auch andere UNIX-Systeme) stellt hierzu den Befehl *test* zur Verfügung. *test* ist ein interner Befehl der bash. Da er nicht in jede Shell integriert ist, ist er häufig zusätzlich als externer Befehl in */usr/bin* zu finden.

### **test**

Der Befehl *test* bewertet bedingte Ausdrücke.

#### **Syntax**

`test Ausdruck`

oder

`[ Ausdruck ]`

#### **Beschreibung**

Der Befehl *test* bewertet den Parameter *Ausdruck* mit dem Rückgabewert *o*, wenn er logisch WAHR ist, und mit einem Rückgabewert ungleich *o*, wenn er logisch FALSCH ist oder keine Parameter angegeben wurden.

Der Rückgabewert des Befehls *test* kann mithilfe der Umgebungsvariablen `$?` abgefragt (referenziert) werden.

Beachten Sie die Rückgabewerte des Befehls *test*: Diese mögen, wenn Sie bereits mit einer anderen Programmiersprache Erfahrung haben, zunächst gewöhnungsbedürftig erscheinen.



**Hinweis**

Folgende Eigenarten dieses Befehls sind zu beachten:



- In der zweiten dargestellten Syntax ist darauf zu achten, dass der öffnenden Klammer ein Leerzeichen folgt und der schließenden ein Leerzeichen vorangestellt wird.



## Shell-Programmierung

- In der C-Shell muss eine Überprüfung auf explizite Dateinamen erfolgen. Eine Dateinamenssubstitution („Globbing“) führt zum Abbruch des Shell-Skripts.

Ausdruck kann entweder einen einstelligen (*unären*) oder einen zweistelligen (*binären*) Operator enthalten, wobei Ersterer ein einzelnes Argument benötigt (unäre Operatoren dienen oft zur Ermittlung des Zustands einer Datei) und Letzterer zwischen zwei Argumenten steht.

Mehrere Ausdrücke können durch spezielle Operatoren logisch miteinander verknüpft werden (siehe Tabelle 16.6).

Im Folgenden finden Sie tabellarische Darstellungen von Optionen und Operatoren des Befehls *test*.

### Prüfen des Zustands einer Datei

#### Optionen für den Gebrauch von *test*

| Option          | Beschreibung                                                                                                            |
|-----------------|-------------------------------------------------------------------------------------------------------------------------|
| <i>-b Datei</i> | WAHR, wenn <i>Datei</i> ein Block-Device ist                                                                            |
| <i>-c Datei</i> | WAHR, wenn <i>Datei</i> ein Zeichen-Device ist                                                                          |
| <i>-d Datei</i> | WAHR, wenn <i>Datei</i> ein Verzeichnis ist                                                                             |
| <i>-e Datei</i> | WAHR, wenn <i>Datei</i> vorhanden ist                                                                                   |
| <i>-f Datei</i> | WAHR, wenn <i>Datei</i> eine einfache Datei ( <i>Plain File</i> ) ist                                                   |
| <i>-g Datei</i> | WAHR, wenn für <i>Datei</i> das Bit SGID ( <i>set group ID</i> ) gesetzt ist                                            |
| <i>-G Datei</i> | WAHR, wenn <i>Datei</i> existiert und der Benutzergruppe des Users, der den Befehl <i>test</i> ausführt, zugeordnet ist |
| <i>-k Datei</i> | WAHR, wenn für <i>Datei</i> das „sticky bit“ gesetzt ist                                                                |
| <i>-L Datei</i> | WAHR, wenn <i>Datei</i> ein symbolischer Link ist                                                                       |
| <i>-O Datei</i> | WAHR, wenn <i>Datei</i> existiert und Eigentum des Benutzers ist, der den Befehl <i>test</i> ausführt                   |
| <i>-p Datei</i> | WAHR, wenn <i>Datei</i> eine Named Pipe ist                                                                             |
| <i>-r Datei</i> | WAHR, wenn <i>Datei</i> existiert und lesbar ist                                                                        |
| <i>-s Datei</i> | WAHR, wenn <i>Datei</i> existiert und größer als 0 Byte ist                                                             |



## Kontrollstrukturen



### Optionen für den Gebrauch von `test` (Forts.)

| Option                      | Beschreibung                                                                                                                                                          |
|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>-S Datei</code>       | WAHR, wenn <i>Datei</i> existiert und eine Socket ist                                                                                                                 |
| <code>-t Dateinummer</code> | WAHR, wenn die Datei mit der <i>Dateinummer</i> für ein Terminal geöffnet ist; ohne die Angabe von <i>Dateinummer</i> wird die Nummer 1 (Standardausgabe) angenommen. |
| <code>-u Datei</code>       | WAHR, wenn für <i>Datei</i> das Bit SUID ( <i>set user ID</i> ) gesetzt ist                                                                                           |
| <code>-w Datei</code>       | WAHR, wenn <i>Datei</i> existiert und in sie geschrieben werden kann                                                                                                  |
| <code>-x Datei</code>       | WAHR, wenn <i>Datei</i> existiert und ausführbar ist                                                                                                                  |

Tabelle 16.2



## Dateivergleiche

### Datei-Vergleichsoperatoren für den Gebrauch von `test`

| Operator                       | Beschreibung                                                                                                        |
|--------------------------------|---------------------------------------------------------------------------------------------------------------------|
| <code>Datei1 -ef Datei2</code> | WAHR, wenn <i>Datei1</i> und <i>Datei2</i> die gleiche I-Node auf dem gleichen Gerät belegen ( <i>equal files</i> ) |
| <code>Datei1 -nt Datei2</code> | WAHR, wenn <i>Datei1</i> neuer als ( <i>newer than</i> ) <i>Datei2</i> ist                                          |
| <code>Datei1 -ot Datei2</code> | WAHR, wenn <i>Datei1</i> älter als ( <i>older than</i> ) <i>Datei2</i> ist                                          |

Tabelle 16.3



## Zeichenkettenvergleiche

### Zeichenketten-Vergleichsoperatoren für den Gebrauch von `test`

| Operator                     | Beschreibung                                                |
|------------------------------|-------------------------------------------------------------|
| Unäre Operatoren             |                                                             |
| <code>-n Zeichenkette</code> | WAHR, wenn die Länge von <i>Zeichenkette</i> ungleich 0 ist |
| <code>-z Zeichenkette</code> | WAHR, wenn die Länge von <i>Zeichenkette</i> gleich 0 ist   |



## Shell-Programmierung

### Zeichenketten-Vergleichsoperatoren für den Gebrauch von `test` (Forts.)

| Operator                | Beschreibung                                                                        |
|-------------------------|-------------------------------------------------------------------------------------|
| Binäre Operatoren       |                                                                                     |
| <code>Zk1 = Zk2</code>  | WAHR, wenn Zeichenkette <code>Zk1</code> gleich Zeichenkette <code>Zk2</code> ist   |
| <code>Zk1 != Zk2</code> | WAHR, wenn Zeichenkette <code>Zk1</code> ungleich Zeichenkette <code>Zk2</code> ist |

Tabelle 16.4

### Arithmetische Vergleichsoperatoren

#### Arithmetische Vergleichsoperatoren für den Gebrauch von `test`

| Operator                    | Beschreibung                                                                                                     |
|-----------------------------|------------------------------------------------------------------------------------------------------------------|
| <code>Arg1 -eq -Arg2</code> | WAHR, wenn Argument <code>Arg1</code> gleich ( <i>equal</i> ) Argument <code>Arg2</code> ist                     |
| <code>Arg1 -ne -Arg2</code> | WAHR, wenn Argument <code>Arg1</code> ungleich ( <i>not equal</i> ) Argument <code>Arg2</code> ist               |
| <code>Arg1 -lt -Arg2</code> | WAHR, wenn Argument <code>Arg1</code> kleiner als ( <i>less than</i> ) Argument <code>Arg2</code> ist            |
| <code>Arg1 -gt -Arg2</code> | WAHR, wenn Argument <code>Arg1</code> größer als ( <i>greater than</i> ) Argument <code>Arg2</code> ist          |
| <code>Arg1 -le -Arg2</code> | WAHR, wenn Argument <code>Arg1</code> kleiner oder gleich ( <i>less equal</i> ) Argument <code>Arg2</code> ist   |
| <code>Arg1 -ge -Arg2</code> | WAHR, wenn Argument <code>Arg1</code> größer oder gleich ( <i>greater equal</i> ) Argument <code>Arg2</code> ist |

Tabelle 16.5



## Verzweigungen



### Logische Operatoren

#### Logische Operatoren für den Gebrauch von *test*

| Operator                | Beschreibung                                                                          |
|-------------------------|---------------------------------------------------------------------------------------|
| <i>! Ausdruck</i>       | WAHR, wenn Ausdruck falsch ist                                                        |
| <i>Ausdr1 -a Ausdr2</i> | WAHR, wenn Ausdruck <i>Ausdr1</i> UND ( <i>and</i> ) Ausdruck <i>Ausdr2</i> wahr sind |
| <i>Ausdr1 -o Ausdr2</i> | WAHR, wenn Ausdruck <i>Ausdr1</i> ODER ( <i>or</i> ) Ausdruck <i>Ausdr2</i> wahr ist  |

Tabelle 16.6



Um nun die Handhabung dieses Befehls genauer zu betrachten, sollte man sich zunächst mit den verschiedenen Kontrollstrukturen auseinander setzen. Diese sollen in den nächsten Abschnitten mit detaillierten Programmbeispielen vorgestellt werden.

## Verzweigungen

Verzweigungen werden dann verwendet, wenn innerhalb eines Programms aus zwei oder mehr Möglichkeiten des weiteren Programmverlaufs gewählt werden soll, d.h., ein bestimmter Programmteil entweder abgearbeitet oder übersprungen werden soll.

Verzweigungen werden auch „Entscheidungsanweisungen“ genannt.

Die einfachste Form der Verzweigung wird durch die *if*-Anweisung erreicht.



Hinweis

### **if**-Anweisung

Die *if*-Anweisung wird immer dann verwendet, wenn man aus zwei oder mehreren Möglichkeiten wählen möchte.

#### Syntax

```
if Ausdruck1 # Wenn Ausdruck1 WAHR ist,
then # dann
 Anweisung[sblock]1 # führe Anweisung(sblock)1 aus
[elif Ausdruck2 # Sonst, wenn Ausdruck2 WAHR ist,
then # dann
 Anweisung[sblock]2 # führe Anweisung(sblock)2 aus.
```



## Shell-Programmierung



```
[else # Sonst
 Anweisung[sblock]n] # führe Anweisung(sblock)n aus
fi
```

*fi* kennzeichnet das Ende der gesamten *if*-Anweisung. Die Shell muss schließlich unterscheiden können, welche Anweisungen noch bedingt abgearbeitet werden müssen, welche also zu *if* gehören und welche nicht. Diese Ende-Kennzeichnung einer Anweisung durch das rückwärts geschriebene Befehlswort

*if ... fi*

ist typisch für die bash.

Jeder Ausdrucksbewertung folgt eine Anweisung oder ein Block von Anweisungen, die jeweils in einer eigenen Zeile stehen müssen oder durch Semikola voneinander getrennt werden. Soll ein Anweisungsblock leer bleiben (leere Anweisung), müssen Sie an diese Stelle zumindest ein *echo* schreiben.

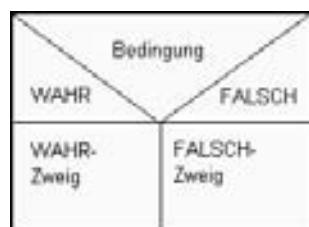
### Hinweis



Schreiben Sie die Einzelanweisungen eines Blocks möglichst jeweils in eine Zeile, um eine größere Übersicht zu wahren.

Die Struktur der Verzweigung kann mit folgendem Struktogramm beschrieben werden, wenn es sich um eine einfache Verzweigung handelt.

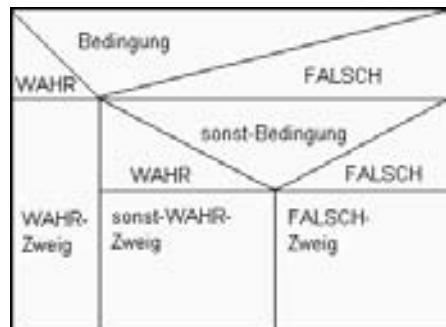
Abb. 16.5



Einfache Verzweigung mit *if*

Bei einer Mehrfachverzweigung sieht das Struktogramm wie folgt aus.

Abb. 16.6



Mehrfahe Verzweigung mit *if* und *elif*



## Verzweigungen



Im Folgenden werden Sie ein Beispielprogramm entwickeln, das den Gebrauch der `if`-Anweisung verdeutlichen soll. Die Problemstellung, um die es hier geht, ist folgende: An den Linux-Server ist ein VT100-Terminal angeschlossen, und ein PC ist mit ihm vernetzt. Der User soll nun beim Einloggen mitteilen, ob er sich direkt am Linux-Server (der „Konsole“), am PC, dessen Kommunikationssoftware eine VT220-Emulation ermöglicht, oder an dem VT100-Terminal befindet. Hierzu ergänzen Sie seine Startup-Datei `.profile` um folgende Einträge:

```
echo "An welchem Terminaltypen sitzen Sie?"
echo "[1] Konsole"
echo "[2] VT100"
echo "[3] VT220"
echo -n "Bitte waehlen Sie: "
read eingabe
if ["$eingabe" = "1"]
then
 export TERM=linux
elif ["$eingabe" = "2"]
then
 export TERM=vt100
else
 export TERM=vt220
fi
```

### Beschreibung

Die Bildschirmausgabe mit dem Befehl `echo` fordert den User auf, den entsprechenden Terminaltyp zu wählen. Die Eingabe wird mit dem Befehl `read` eingelesen und anschließend mit

```
if ["$eingabe" = "1"]
daraufhin überprüft, ob es sich bei der Eingabe um die Ziffer „1“ handelt.
```

Ist dies der Fall, wird der Terminaltyp auf `linux` gesetzt:

```
then
 export TERM=linux
```

Wurde keine „1“ eingegeben, erfolgt eine Überprüfung auf die Ziffer „2“:

```
elif ["$eingabe" = "2"]
```

Wurde eine „2“ eingegeben, wird der Terminaltyp auf `vt100` gesetzt:

```
then
 export TERM=vt100
```



Der Programmierer wird aus Gründen der Übersichtlichkeit, wenn möglich, immer diese Syntaxvariante vorziehen



## Shell-Programmierung

---



Wurde weder eine „1“ noch eine „2“ eingegeben, sondern ein anderes Zeichen oder eine andere Zeichenfolge, legt der *else*-Zweig fest, dass der Terminaltyp auf *vt220* eingestellt wird.

Der Wert der Variablen *TERM* wird in allen drei Fällen nun nicht einfach nur definiert (etwa *TERM=vt100*), die Variable wird außerdem noch exportiert, d.h. global allen Shells zur Verfügung gestellt.

Normalerweise ist eine Variable nur innerhalb der Shell verfügbar, in der sie definiert wird. Da ein Shell-Skript in der Regel in einer Sub-Shell ausgeführt wird, würde ein einfaches Definieren von *TERM* keinen Einfluss auf die Umgebung der aufrufen-den Shell haben. Gerade das soll aber vorliegendes Beispiel leisten: das Ändern der Systemumgebung der aktuellen Shell durch Aufruf eines Shell-Skripts.



### Hinweis

Zum *export*-Befehl siehe auch *Kapitel 4 Benutzerumgebung*.

Für dieses einfache Beispiel mag die *if*-Verzweigung noch genügen, doch zeigen sich bereits einige Schwachpunkte:

- Bei einer größeren Anzahl von Auswahlmöglichkeiten (stellen Sie sich vor, Sie wollten zwischen acht verschiedenen Terminal-Emulationen wählen) wird eine mehrfache *if*-Verzweigung sehr lang und damit unübersichtlich.
- Was geschieht bei einer unbeabsichtigten Falscheingabe?

Es wird versucht werden, diese und andere Probleme im weiteren Verlauf der Ausführungen noch auszuräumen.

Der erste Schritt ist, eine Schachtelung mehrerer *if*-Anweisungen zu vermeiden. Dies geschieht mit der so genannten „Fallunterscheidung“.



### Hinweis

Man spricht bei einer solchen Schachtelung auch von einer „Kaskadierung“.

## **case-Verzweigung**

Bei der Fallunterscheidung mit *case* können zwei oder mehr Ergebnismöglichkeiten eines zu bewertenden Ausdrucks übersichtlicher dargestellt werden; dieses Verfahren (auch „Struktur der Mehrfachauswahl“ genannt) ist einer geschachtelten *if*-Anweisung in der Regel vorzuziehen.

### Syntax

```
case wert in
 Ausdruck[sliste]1) Anweisung[sblock]1;;
 Ausdruck[sliste]2) Anweisung[sblock]2;;
 Ausdruck[sliste]_n) Anweisung[sblock]_n;;
esac
```



## Verzweigungen



Das Ende jedes Ausdrucks bzw. jeder Ausdrucksliste muss mit einer Klammer gekennzeichnet und jede Anweisung bzw. jede Anweisungsliste (mit Ausnahme der letzten, bei der die Semikola optional sind) muss mit zwei „;“ beendet werden.

Soll ein Standardwert (*default*) bestimmt werden, geschieht dies mit dem Asterisk (\*) als letztgenanntem Ausdruck der Fallunterscheidung.

Die Struktur der Mehrfachauswahl kann mit folgendem Struktogramm beschrieben werden (siehe Abbildung 16.7).



Abb. 16.7

Mehrfachauswahl mit `case`

Folgendes Beispiel zeigt die Lösung des im vorigen Abschnitt beschriebenen Programmierproblems mithilfe der Mehrfachauswahl:

```
echo "An welchem Terminaltypen sitzen Sie?"
echo "[1] Konsole"
echo "[2] VT100"
echo "[3] VT220"
echo -n "Bitte waehlen Sie: "
read eingabe
case $eingabe in
 "1") export TERM=linux;;
 "2") export TERM=vt100;;
 "3") export TERM=vt220;;
 *) echo "Falscheingabe! Der Terminaltyp wird"
 echo "auf VT100 gesetzt"
 export TERM=vt100;;
esac
```

### Beschreibung

Die Bildschirmausgabe mit dem Befehl `echo` fordert den User auf, den entsprechenden Terminaltyp zu wählen. Die Eingabe wird mit dem Befehl `read` eingelesen und anschließend mit `case` ausgewertet: Wurde eine „1“ eingegeben, wird der Terminaltyp auf `linux` gesetzt, wurde eine „2“ eingegeben, auf `vt100`, und bei der Eingabe einer „3“ wird `vt220` als zu verwendende Terminal-Emulation festgelegt. Bei der Ein-

## Shell-Programmierung

---



gabe eines beliebigen anderen Zeichens oder einer beliebigen anderen Zeichenkette erfolgt eine entsprechende Fehlermeldung und der Terminaltyp wird auf `vt100` gesetzt.

Was ist jedoch zu tun, wenn mehrere Ausdrücke das gleiche Ergebnis bringen sollen wie im folgenden Beispiel, bei dem sowohl die Eingabe eines Klein- wie auch eines Großbuchstabens die gleiche Bildschirmausgabe bewirken soll:

```
echo -n "Moechten Sie fortfahren [J/N]: "
read eingabe
case $eingabe in
 "j") echo "Sie haben JA gewaehlt";;
 "J") echo "Sie haben JA gewaehlt";;
 "n") echo "Sie haben NEIN gewaehlt";;
 "N") echo "Sie haben NEIN gewaehlt";;
 *) echo "Ungueltige Eingabe!!!"
esac
```

Sie können die verschiedenen Ausdrücke in einer so genannten „Ausdrucksliste“ zusammenfassen, wie folgendes Beispiel zeigt:

```
echo -n "Moechten Sie fortfahren [J/N]: "
read eingabe
case $eingabe in
 [jJ]) echo "Sie haben JA gewaehlt";;
 [nN]) echo "Sie haben NEIN gewaehlt";;
 *) echo "Ungueltige Eingabe!!!"
esac
```

Dieses Beispiel zeigt, dass Sie verschiedene Eingabemöglichkeiten in Ausdruckslisten zusammenfassen können, um die Fallunterscheidung noch übersichtlicher zu gestalten.

Komplexere Ausdruckslisten können mithilfe von Sonderzeichen definiert werden, die ähnlich wie Shell-Metazeichen Zeichen(ketten)mengen definieren. Entnehmen Sie diese der *Tabelle 16.7*.

### Zeichenketten zur Bildung von Ausdruckslisten

---



| Zeichenkette         | Bedeutung                                                                                                                                                                                                              |
|----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>Zk1*Zk2</code> | <code>Zk1hier_steh_t_eine_beliebige_ZeichenfolgeZk2</code> , z.B. <code>ab*m</code> . Die Zeichenketten <code>abraham</code> und <code>abdulrahim</code> wären hierdurch beispielsweise abgedeckt.                     |
| <code>Zk1?Zk2</code> | <code>Zk1hier_steh_t_ein_beliebiges_ZeichenZk2</code> , z.B. <code>P?ter</code> . Diese Ausdrucksliste würde beispielsweise die Zeichenketten <code>Peter</code> , <code>Pater</code> und <code>Puter</code> abdecken. |



## Schleifen



### Zeichenketten zur Bildung von Ausdruckslisten (Forts.)

Zeichenkette      Bedeutung

$Zk1[ \quad ]Zk2$        $Zk1$  beliebiges Zeichen aus der Liste in  $[ \quad ]Zk2$ , wobei Bereichsangaben möglich sind, z.B. [a-nt-y], [A-G], [2-9] usw.

$Zk1 | Zk2$       Zeichenkette  $Zk1$  ODER Zeichenkette  $Zk2$ . In dem Beispiel oben hätte man also auch formulieren können: „j“ | „J“.

\      hebt die Sonderbedeutung des nachfolgenden Zeichens auf; z.B. steht \\$ für „\\$“

Tabelle 16.7



## Schleifen

Standardmäßig werden Shell-Skripte (wie auch andere Programme) zeilenweise von oben nach unten abgearbeitet. Bei der Verwendung von Schleifen kehrt das Programm bei Erfüllung einer gestellten Bedingung wieder zum Schleifenanfang zurück, um den Schleifenkörper erneut zu durchlaufen.

Schleifen werden in der Programmierung auch „Wiederholstrukturen“ oder „Iterationen“ genannt.



Hinweis

### Die **while**-Schleife

Die *while*-Schleife wird so lange durchlaufen, bis der Rückgabewert einer gestellten Bedingung ungleich 0 ist.

#### Syntax

```
while Ausdruck # Solange der Rückgabewert 0 ist
do
 Befehl[sliste] # führe aus
done # Befehl(sliste)
 # Schleifenende
```

Folgendes Struktogramm zeigt die Struktur dieser Iteration.

645



## Shell-Programmierung



Abb. 16.8 ○

Struktogramm der *while*-Schleife



Das zuvor beschriebene Beispielprogramm zur Festlegung des Terminaltyps soll dahingehend ergänzt werden, dass im Falle einer Falscheingabe der User erneut aufgefordert wird, eine Terminal-Emulation zu wählen. Dieser Vorgang soll wiederholt werden, bis einer der gültigen Werte „1“, „2“ oder „3“ eingegeben wird:

```
echo "An welchem Terminaltypen sitzen Sie?"
echo "[1] Konsole"
echo "[2] VT100"
echo "[3] VT220"
echo -n "Bitte wählen Sie: "
read eingabe

while ["$eingabe" != "1" -a "$eingabe" != "2" \
 -a "$eingabe" != "3"]
do
 echo "Falscheingabe! Bitte nur 1, 2 oder 3"
 echo "eingegeben"
 echo -n "Bitte wählen Sie: "
 read eingabe
done
case "$eingabe" in
 "1") export TERM=linux;;
 "2") export TERM=vt100;;
 "3") export TERM=vt220;;
esac
```

### Beschreibung

Zunächst wird der User – wie gehabt – aufgefordert, einen Terminaltyp zu wählen. Erfolgt eine Falscheingabe (d.h., es wurde keine der drei Ziffern „1“, „2“ oder „3“ eingegeben), erfolgt eine entsprechende Fehlermeldung, die wiederholt wird, bis eine gültige Eingabe vorgenommen wurde.

Ist dies der Fall, wird die gewünschte Terminal-Emulation aktiviert.





### Auswertung von Argumenten als Ausdrücke

Wie Sie in der Beschreibung des Befehls `test` gesehen haben, verfügt er über arithmetische Vergleichsoperatoren. Um diese praktisch im Zusammenhang mit der `while`-Schleife nutzen zu können, ist es nützlich, sich mit der arithmetischen Funktionalität der bash auseinander zu setzen.

Hierbei wird der Inhalt einer referenzierten Variablen zunächst arithmetisch manipuliert, wobei der gesamte Ausdruck in eckigen Klammern stehen muss. Dieser Ausdruck muss erneut durch \$ referenziert werden und kann damit einer Variablen zugewiesen werden.

Diese Funktionalität der bash ausnutzend, könnten Sie ein Programm wie das folgende formulieren:

Wenn Sie nicht mit der bash, sondern einer anderen Shell arbeiten, müssen Sie sich gegebenenfalls des Befehls `expr` für die Durchführung arithmetischer Operationen bedienen.



Hinweis

```
a=1
while [$a -le 5]
do
 echo "$a. Schleifendurchlauf"
 a=$((a+1))
done
echo "Ende des Schleifenkoerpers"
```

Doch wenden wir uns weiteren Möglichkeiten der `while`-Schleife zu: Mithilfe des Befehls `true` kann eine `while`-Schleife als Endlosschleife konzipiert werden.

Die bash kann nur mit Ganzzahlen rechnen.



Hinweis

#### **true**

Der Befehl `true` liefert immer den Wert WAHR (o) zurück.

#### **Syntax**

```
true
```

Folgendes Beispiel zeigt eine Endlosschleife, die mit `while true` gebildet wurde:

```
while true
do
 echo "Hallo Welt"
done
```

Ein solches Konstrukt ist in der Regel jedoch nicht gewünscht, da diese Schleife nur mit Brachialmethoden wie dem Drücken der Taste `[Entf]`, der Tastenkombination `[Strg]+[C]` oder Ähnlichem abgebrochen werden kann.

## Shell-Programmierung

---



Was Sie also hier benötigen, ist eine Methode, die Endlosschleife geregelt verlassen zu können. Hierzu bietet die bash den Befehl *break*.

### Die *break*-Anweisung

Zum geordneten Verlassen einer *while*-Schleife (wie auch der weiter unten beschriebenen *until*- und *for*-Schleifen) kann die *break*-Anweisung verwendet werden.

#### Syntax

```
break [n]
```

*break* bewirkt für sich allein genommen das Verlassen der aktuellen Schleife. Sollen mehrere verschachtelte Schleifen verlassen werden, kann über den optionalen Parameter *n* die Verschachtelungstiefe angegeben werden.

Im nächsten Beispiel wird mithilfe einer Endlosschleife die Möglichkeit gegeben, einen Wert gefolgt von einem Operator und einem zweiten Wert einzugeben. Die beiden Werte werden dann mithilfe des von Ihnen eingegebenen Operators arithmetisch miteinander verknüpft, das Ergebnis wird ausgegeben, und es erfolgt eine Abfrage, ob eine weitere Berechnung stattfinden soll.

Aus diesem Grunde musste die Verzweigung

```
if ["$op" = "*"]
```

in den Programmcode eingefügt werden.

#### Hinweis



Da der Asterisk (\*) nicht nur als arithmetischer Operator benutzt werden kann, sondern auch ein Metazeichen der Shell ist, muss er durch einen vorangestellten Backslash (\) zitiert werden.

Dies ist nur eines der Probleme, die bei der Lösung dieses Programmierbeispiels zu beachten sind. Versuchen Sie doch einfach einmal, eine Division durch 0 durchzuführen – das Ergebnis wird Sie überraschen. Dieses Beispiel zeigt wieder einmal, dass sich ein scheinbar sehr einfach zu lösendes Programmierproblem sehr schnell in der Praxis komplexer als ursprünglich angenommen erweist.

```
while true
do
 clear
 echo -n "Zahl1: "
 read z1
 echo -n "Operator: "
 read op
 echo -n "Zahl2: "
 read z2
 if ["$op" = "*"]
```



## Schleifen



```
then
 z3=$[$z1*$z2]
else
 z3=$[$z1opz2]
fi
echo "$z1 $op $z2 = $z3"
echo -n "Neue Berechnung [J/N]: "
read eingabe
if ["$eingabe" = "n" -o "$eingabe" = "N"]
then
 break
fi
done
```

Um den Gebrauch des optionalen Parameters *n* zum Verlassen mehrfach geschachtelter Schleifen zu demonstrieren, wird das vorige Beispiel nochmals verkompliziert. Hierzu konstruieren Sie für die Abfrage, ob eine erneute Berechnung erfolgen soll, eine innere *while*-Schleife konstruiert.

```
while true
do
 echo -n "Zahl1: "
 read z1
 echo -n "Operator: "
 read op
 echo -n "Zahl2: "
 read z2
 if ["$op" = "*"]
 then
 z3=$[$z1*$z2]
 else
 z3=$[$z1 $op $z2]
 fi
 echo "$z1 $op $z2 = $z3"
 while true
 do
 echo -n "Neue Berechnung [J/N]: "
 read eingabe
 if ["$eingabe" = "j" -o "$eingabe" = "J"]
 then
 break
 elif ["$eingabe" = "n" -o "$eingabe" = "N"]
 then
 break 2
 fi
 done
done
```

## Shell-Programmierung

---



In diesem Programmbeispiel wird, anders als im vorigen Beispiel, bei Eingabe eines „j“ oder eines „J“ nur der innere Schleifenkörper verlassen. Die äußere Endlosschleife wird also fortgesetzt, und es sind somit weitere Berechnungen möglich. Bei der Eingabe eines „n“ oder eines „N“ wird jedoch durch die Anweisung *break 2* nicht nur der innere Schleifenkörper verlassen, sondern es wird durch den Parameter *2* die Schachtelungstiefe definiert, die verlassen werden soll.

Für vorliegendes Beispiel bedeutet dies, dass auch die äußere *while*-Schleife beendet und somit das Programm verlassen wird.

Eine ähnliche Funktion wie die *break*-Anweisung besitzt die *continue*-Anweisung.

### Die *continue*-Anweisung

Die *continue*-Anweisung bewirkt, dass der aktuelle Schleifendurchlauf abgebrochen wird, d.h., dass alle dem Befehl *continue* folgenden Anweisungen des Schleifenkörpers übersprungen werden. Die Schleife wird jedoch nicht verlassen, sondern es wird mit dem folgenden Iterationsschritt fortgefahren – zurück zum Schleifenanfang.

#### Syntax

```
continue [n]
```

Auch hier gibt der optionale Parameter *n* die Schachtelungstiefe an.

Folgendes Beispielprogramm gibt alle Werte von 1 bis 50 mit Ausnahme der Werte 10, 20, 30 und 40 aus:

```
a=0
while [$a -lt 50]
do
 a=$((a+1))
 if [$a -eq 10 -o $a -eq 20 -o $a -eq 30 \
 -o $a -eq 40]
 then
 continue
 fi
 echo $a
done
```

### Die *until*-Schleife

Die *until*-Schleife wird durchlaufen, bis der Rückgabewert einer gestellten Bedingung gleich 0 ist. Sie stellt somit eine Variante der *while*-Schleife dar, bei der die Bedingung umgekehrt ist.



## Schleifen



### Syntax

```
until Ausdruck # Solange der Rückgabewert 1 ist
do # führe aus
 Befehl[sliste] # Befehl(sliste)
done # Schleifenende
```

Folgendes Struktogramm zeigt die Struktur dieser Iteration.



Abb. 16.9

Struktogramm der *until*-Schleife

Folgendes Beispielprogramm gibt den Wert einer Variablen so lange aus und zählt ihn im Anschluss um den Wert 1 hoch, bis die Variable den Wert 10 überschritten hat. Ist dies der Fall, wird das Programm mit dem Rückgabewert 0 beendet:

```
a=1
until [$a -gt 10]
do
 echo "Wert von a: $a; Rueckgabewert: $?"
 a=$[$a+1]
done
```

Um die Verwandtschaft mit der *while*-Schleife und die dabei auftretenden Unterschiede noch einmal zu verdeutlichen, hier die entsprechende Kodierung mit *while*:

```
a=1
while [$a -le 10]
do
 echo "Wert von a: $a; Rueckgabewert: $?"
 a=$[$a+1]
done
```

### Die Zählschleife mit *for*

Wenn bereits im Voraus bekannt ist, wie häufig eine Schleife wiederholt werden soll oder wenn eine Liste von Befehlszeilenargumenten verarbeitet werden soll, empfiehlt sich der Einsatz einer *for*-Schleife.

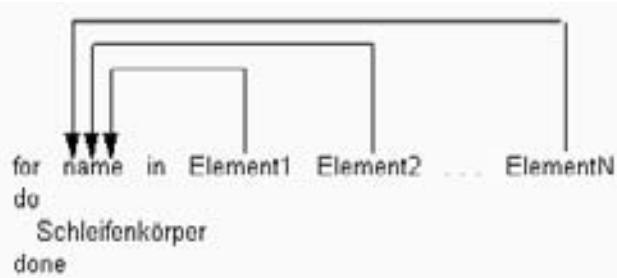


## Shell-Programmierung



Abb. 16.10

Die *for*-Schleife



### Syntax

```
for Variable [in Liste]
do
 Befehl[sliste]
done
```

### Beschreibung

Wenn *for* eine Liste von Argumenten übergeben wird, wird die Schleife so häufig durchlaufen, wie Listenelemente vorhanden sind:

```
for i in hund katze maus
do
 echo "Hallo Welt"
done
```

Da die Anzahl der Listenelemente drei beträgt, wird die Schleife dreimal durchlaufen, der Text „Hallo Welt“ wird also dreimal hintereinander ausgegeben, wobei bei jedem Schleifendurchlauf das aktuelle Listenelement der Variablen (hier: *i*) zugewiesen wird, wie folgende Programmänderung zeigt:

```
for i in hund katze maus
do
 echo "Ein(e) $i"
done
```

Sollen nun Befehlszeilenargumente übergeben werden, ist dies mit der Konstruktion

```
for i in $*
möglich, wobei eine vereinfachte Schreibweise
```

```
for i
```

wäre. Hier wird die Schleife so oft durchlaufen, wie Befehlszeilenargumente übergeben wurden.



## Schleifen



Beachten Sie weiter oben den Abschnitt „*Argumentwerte*“: Die Argumentliste wird mit `$*`, die Gesamtzahl der Argumente mit `$#` referenziert.



Hinweis

Folgendes Beispiel ermöglicht, dem Skript eine beliebige Anzahl an Dateinamen zu übergeben, deren Inhalte der Reihe nach auf dem Bildschirm angezeigt werden:

```
a=0
for i in $*
do
 a=$[$a+1]
 echo "$a. Schleifendurchlauf, basierend auf "
 echo "$# Befehlszeilenargumenten"
 echo "Angezeigt wird der Inhalt der Datei $i:"
 cat $i
done
```

Der Aufruf dieses Programms könnte wie folgt aussehen.

Abb. 16.11

Auswertung der Befehlszeilenargumentübergabe mit `for`

```
TNVT- Linserv, to host 134.245.35.7
Session Edit Commands Settings Help
fg2pcc:/home/gehrke/scripts > zeige datei1 datei2 datei3
1. Schleifendurchlauf, basierend auf 3 Befehlszeilenargumenten
Angezeigt wird der Inhalt der Datei datei1:
Dies ist der Inhalt der Datei1
2. Schleifendurchlauf, basierend auf 3 Befehlszeilenargumenten
Angezeigt wird der Inhalt der Datei datei2:
Dies ist der Inhalt der Datei2
3. Schleifendurchlauf, basierend auf 3 Befehlszeilenargumenten
Angezeigt wird der Inhalt der Datei datei3:
Dies ist der Inhalt der Datei3
fg2pcc:/home/gehrke/scripts > ■
```

Die Abarbeitung der Befehlszeilenargumente kann alternativ zur *for i-Schleife* mit dem Befehl `shift` in Kombination mit `while` oder `until` erfolgen.

## Shell-Programmierung

---



### Veränderung von Befehlszeilenargumenten mit *shift*

Wie in dem Abschnitt „Argumentwerte“ dargestellt, können Befehlszeilenargumente bezüglich ihrer Position referenziert werden, wobei die Positionsparameter \$1 bis \$9 zur Übergabe an das Skript (nämlich \$0) zur Verfügung stehen.

Was macht man aber, wenn die genaue Anzahl der übergebenen Befehlszeilenargumente entweder im Voraus nicht bekannt ist oder die Zahl 9 überschreitet?

Neben der *for*-Schleife steht die Funktionalität des *Parameter-Shiftings* zur Verfügung.

Hierbei wird zunächst der erste übergebene Parameter abgearbeitet; dann kann der zweite Parameter in die Position „1“ verschoben („geshiftet“) werden und entsprechend wieder als „neuer“ Parameter 1 abgearbeitet werden. Dabei verändern auch alle folgenden Befehlszeilenargumente (Parameter) ihre Position um eine Einheit nach links.

Dies geschieht mithilfe des Befehls *shift*.

Folgendes Beispielskript *shifttest* mag dies verdeutlichen:

```
i=0
while [$# -gt 0]
do
 i=$[$i+1]
 echo "Befehlszeilenargument $i hat den Inhalt: \"$1\""
 shift
done
```

Wird das Skript mit der Befehlszeile

```
shifttest Ich bin muede
```

aufgerufen, erhalten Sie folgende Ausgabe:

```
Befehlszeilenargument 1 hat den Inhalt: "Ich"
Befehlszeilenargument 2 hat den Inhalt: "bin"
Befehlszeilenargument 3 hat den Inhalt: "muede"
```

Zunächst wird die Anzahl der Befehlszeilenargumente überprüft. Da diese größer 0 ist, wird zunächst der Inhalt des ersten Arguments ausgegeben (*Ich*).

Alle weiteren Argumente werden durch *shift* um eine Position nach links verschoben, womit nun

- die Anzahl der Positionsparameter um 1 kleiner,
  - der bisherige Parameter 2 (*bin*) zu Parameter 1
  - und Parameter 3 (*muede*) zu Parameter 2
- wird.





Dieser Vorgang wird zwei weitere Male wiederholt, dann ist die Anzahl der Positionsparameter 0, und die Schleife wird somit beendet.

## Dialoge

Ein wesentliches Leistungsmerkmal der Shell-Programmierung unter Linux, das mit den meisten Distributionen erhältlich ist, ist der Befehl *dialog*. Aus diesem Grunde soll an dieser Stelle mit einigen detaillierteren Beispielen auf diesen Befehl eingegangen werden.

### ***dialog***

Der Befehl *dialog* ermöglicht es Ihnen, in Shell-Skripten Dialogboxen zu verwenden, die eine Interaktion zwischen Skript und Anwender erlauben.

#### Syntax

```
dialog [--clear] [--title Text] [Dialogoptionen]
```

Bei der mit diesem Buch ausgelieferten Version handelt es sich um die Version 0.62, die sowohl über die Tastatur als auch mit der Maus bedient werden kann. Sollten Sie mit einer anderen Linux-Distribution arbeiten, kann die Handhabung von der hier vorgestellten gegebenenfalls abweichen.

Sie finden das Paket in der Serie *ap – Anwendungen, die kein X11 benötigen*.

Die wichtigsten für den Befehl *dialog* zur Verfügung stehenden Optionen entnehmen Sie *Tabelle 16.9*.

#### Optionen für den Gebrauch des Befehls *dialog*

| Option                        | Beschreibung                                                                                                                                     |
|-------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>-clear</code>           | kann mit jeder anderen Dialogoption kombiniert werden. Bewirkt, dass der Dialog nach seiner Beendigung vom Bildschirm gelöscht wird.             |
| <code>-title text</code>      | kann mit jeder anderen Dialogoption kombiniert werden. Ermöglicht die Darstellung eines Dialogtextes.                                            |
| <code>-msgbox text y x</code> | zeigt einen Meldungstext in einem Dialogfenster von y Zeilen Höhe und x Zeichen Breite an. Dieser Dialog muss mit <code>→</code> beendet werden. |

## Shell-Programmierung

### Optionen für den Gebrauch des Befehls *dialog* (Forts.)

| Option                                                                   | Beschreibung                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|--------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>-infobox text y x</code>                                           | zeigt einen Meldungstext in einem Dialogfenster von <i>y</i> Zeilen Höhe und <i>x</i> Zeichen Breite an. Dieser muss jedoch nicht mit  beendet werden, sondern die Abarbeitung des Programms wird sofort weitergeführt. Der Inhalt dieses Dialogfensters wird so lange angezeigt, bis der Bildschirm gelöscht wird                                                                                       |
| <code>-yesno text y x</code>                                             | zeigt ein Dialogfenster von <i>y</i> Zeilen Höhe und <i>x</i> Zeichen Breite an, dessen Meldungstext eine Ja-/Nein-Entscheidung beinhaltet                                                                                                                                                                                                                                                                                                                                                  |
| <code>-inputbox text y x</code>                                          | ermöglicht die Eingabe einer Textzeile in einem Dialogfenster von <i>y</i> Zeilen Höhe und <i>x</i> Zeichen Breite                                                                                                                                                                                                                                                                                                                                                                          |
| <code>-textbox datei y x</code>                                          | zeigt den Inhalt der durch <i>datei</i> spezifizierten Textdatei in einem Fenster von <i>y</i> Zeilen Höhe und <i>x</i> Zeichen Breite an. Ist die Textdatei so lang, dass sie nicht vollständig in dem durch die <i>y/x</i> -Koordinaten definierten Fenster angezeigt werden kann, kann mithilfe der Cursor-tasten durch den vollständigen Text gescrollt werden.                                                                                                                         |
| <code>-menu text y x mp mp_1 mt_1 ... mp_n mt_n</code>                   | ermöglicht die Darstellung einer Menüliste in einem Dialogfenster von <i>y</i> Zeilen Höhe und <i>x</i> Zeichen Breite, aus der eine bestimmte Option gewählt werden kann. <i>mp</i> spezifiziert hierbei die Anzahl der Menüpunkte, mit Menüpunkt <i>mp_1</i> und dem diesem zugeordneten Menütext <i>mt_1</i> bis zu Menüpunkt <i>mp_x</i> mit dem ihm zugeordneten Menütext <i>mt_x</i> .                                                                                                |
| <code>-checkbox text y x op op_1 ot_1 stat_1 ... op_n ot_n stat_n</code> | ermöglicht die Darstellung einer Optionsliste in einem Dialogfenster von <i>y</i> Zeilen Höhe und <i>x</i> Zeichen Breite, aus der eine gleichzeitige Auswahl mehrerer Optionen möglich ist. <i>op</i> spezifiziert hierbei die Anzahl der Optionspunkte, mit Optionspunkt <i>op_1</i> , dem diesem zugeordneten Optionstext <i>ot_1</i> und seinem Status <i>stat_1</i> bis zu Optionspunkt <i>op_n</i> mit dem ihm zugeordneten Optionstext <i>ot_n</i> und seinem Status <i>stat_x</i> . |



## Dialoge



### Optionen für den Gebrauch des Befehls *dialog* (Forts.)



| Option                                                                    | Beschreibung                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|---------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>-radiolist text y x op op_1 ot_1 stat_1 ... op_n ot_n stat_n</code> | ermöglicht die Darstellung einer Optionsliste in einem Dialogfenster von <i>y</i> Zeilen Höhe und <i>x</i> Zeichen Breite, aus der eine ausschließliche Auswahl einer Option möglich ist. <i>op</i> spezifiziert hierbei die Anzahl der Optionspunkte, mit Optionspunkt <i>op_1</i> , dem diesem zugeordneten Optionstext <i>ot_1</i> und seinem Status <i>stat_1</i> bis zu Optionspunkt <i>op_n</i> mit dem ihm zugeordneten Optionstext <i>ot_n</i> und seinem Status <i>stat_x</i> . |
| <code>-gauge text y x -prozent</code>                                     | ermöglicht die Anzeige eines Statusbalkens in Prozent in einem Fenster von <i>y</i> Zeilen Höhe und <i>x</i> Zeichen Breite                                                                                                                                                                                                                                                                                                                                                              |
| <code>-backtitle text</code>                                              | stellt eine Variation der oben beschriebenen Option <code>-title</code> dar. Zeigt den Titel jedoch nicht in der Dialogbox, sondern am oberen Bildschirmrand an.                                                                                                                                                                                                                                                                                                                         |

Tabelle 16.8



*dialog* liefert die in Tabelle 16.9 aufgeführten Rückgabewerte, die mithilfe der Shell-Variablen \$? ausgelesen und somit für eine weitere Verarbeitung durch das Skript verwendet werden können.

Folgende Besonderheiten sind bei Verwendung des Befehls *dialog* zu beachten:

- Befehlszeilen, die länger als die Eingabezeile sind (ausgenommen solche, die durch Apostrophe eingeschlossen sind), müssen für die bash mit einem Backslash (\) als durchgängige Befehlszeilen gekennzeichnet werden.
- Lange Bildschirmausgaben werden entsprechend der Größe des Dialogfens-ters umbrochen. Sollen manuelle Umbrüche erfolgen, geschieht dies mithilfe von \n.
- Texte in *dialog*-Optionen, die Trennzeichen enthalten, müssen in einfache oder doppelte Anführungszeichen gestellt werden.



## Shell-Programmierung

### Rückgabewerte des Befehls *dialog*

| Rückgabewert | Beschreibung                                                  |
|--------------|---------------------------------------------------------------|
| 0            | Der Dialog wurde mit <i>OK</i> oder mit <i>Yes</i> beendet    |
| 1            | Der Dialog wurde mit <i>Cancel</i> oder mit <i>No</i> beendet |
| 255          | Der Dialog wurde mit E beendet                                |

Tabelle 16.9

Das Ergebnis von Dialogtypen, die einen Text, eine Option oder eine Liste von Optionen zurückgeben, kann in eine Fehlerdatei umgeleitet werden, die dann für weitere Zwecke ausgewertet werden kann.

## Beispiele

Um den Gebrauch der oben beschriebenen Optionen zu verdeutlichen, sollen an dieser Stelle einige Beispielprogramme entwickelt werden, die in zunehmendem Maße an Komplexität gewinnen werden.

### Anzeige von Informationen

Der Befehl *dialog* bietet drei Optionen zur Informationsausgabe:

- Die Option *-msgbox* zeigt den von Ihnen gewünschten Text an und hält den weiteren Programmablauf so lange an, bis Sie [Esc] drücken.
- Die Option *-infobox* zeigt den von Ihnen gewünschten Text auf dem Bildschirm an, wobei die weitere Programmabarbeitung ohne eine von Ihnen vorzunehmende Bestätigung erfolgt.

Da eine Infobox nur für die Dauer bis zum nächsten Löschen des Bildschirms angezeigt wird, sollte diese Funktion nicht mit der Option *-clear* zusammen verwendet werden, da sonst der Infotext sofort gelöscht würde.

#### Tipp



Möchten Sie den Bildschirm nach dem Anzeigen der Infobox dennoch löschen, lassen Sie diese mithilfe des Befehls *sleep* für ein paar Sekunden anzeigen.

- Die Option *-textbox* ermöglicht, den Inhalt einer als Parameter zu übergebenden Textdatei anzeigen zu lassen. Durch diese kann, wenn Sie größer als das von Ihnen definierte Dialogfenster ist, mithilfe der Cursortasten nach links/rechts und oben/unten geblättert werden.

Folgendes Beispielprogramm verwendet diese Dialogelemente:





```
dialog --clear --title "Messagebox 1" \
--msgbox "Hier steht Ihre Meldung" 5 27
dialog --title "Infobox" --infobox \
"Programm wird in 5 Sekunden\nfortgesetzt.
\nBitte warten ..." 6 31
sleep 5
dialog --clear --title "Textbox: inf_anzeige" \
--textbox inf_anzeige 20 78
dialog --clear --title "Ja/Nein-Entscheidung" \
--yesno \
"Moechten Sie fortfahren?" 7 30
janein=$?
case $janein in
 "0") dialog --clear --title "Messagebox 2" \
 --msgbox "Das Ergebnis war JA" 5 27;;
 "1") dialog --clear --title "Messagebox 3" \
 --msgbox "Das Ergebnis war NEIN" 5 27;;
 "255") dialog --clear --title "Messagebox 4" \
 --msgbox "Abbruch mit <ESC>" 5 27;;
esac
```

### Ja/Nein-Abfragen

Das folgende Skript *delbak* soll zum Löschen aller Sicherungsdateien im aktuellen Verzeichnis dienen, deren Namen mit einer Tilde (~) enden, dies sind z.B. Sicherungsdateien des Texteditors joe.

Hierzu wird mit der Option *--yesno* eine Entscheidungsabfrage realisiert, deren Ergebnis der Variablen *jn* zugewiesen wird (*jn=\$?*).

In Abhängigkeit von dem Wert, den *jn* besitzt, werden entweder die Sicherungsdateien gelöscht (*\$jn = 0*), die Dateien werden nicht gelöscht (*\$jn = 1*), oder es wird, sollte der Dialog mit E abgebrochen worden sein, eine entsprechende Meldung auf dem Bildschirm ausgegeben:

```
#Skript zur Darstellung einer Ja/Nein-Entscheidung
dialog --clear --yesno "Sollen die \
Sicherungsdateien geloescht werden?" 7 35
jn=$?
if [$jn = 0]
then
 rm *~
elif [$jn = 1]
then
 echo "Sicherungsdateien werden nicht geloescht"
else
 echo "Abbruch mit <ESC>"
fi
```



## Shell-Programmierung

---



### Texteingaben

Im nächsten Beispiel wird ein Skript *intdel* entwickelt, das zur Eingabe eines Dateinamens aufruft und diese Datei dann löscht.

Hierzu wird die Option *-inputbox* verwendet, deren Eingabeergebnis standardmäßig an die Standardfehlerausgabe weitergegeben wird. Um jedoch mit diesem Ergebnis etwas anfangen zu können, wird es in eine Fehlerdatei umgeleitet. Der Dateiname wird hierbei dynamisch vergeben, indem er aus seiner Prozessnummer generiert wird (*> \$\$tmp*).

#### Hinweis



Ein dynamischer Dateiname kann in einem Skript mit der Zeichenfolge *\$\$* generiert werden. *\$\$* liefert die Prozessnummer der aktuellen Shell.

```
#Skript zur Darstellung einer Zeichenketteneingabe
dialog --clear --title "Loeschbefehl" \
--inputbox "Dateiname:" \
10 30 2> $$tmp # Einlesen eines
 # Dateinamens
 # und Umlenkung der Eingabe
 # in eine temporaere Datei
ergebnis=$?
if [$ergebnis = 0] # Ueberpruefung darauf, ob
 # der Dialog
then # auch tatsaechlich mit OK
 # abgeschlossen wurde
 eingabe=$(cat $$tmp) # Einlesen des Inhalts
 # der temporaeren
 # Datei in die Variable
 # "eingabe"
 rm $eingabe # Loeschen der angegebenen
 # Datei
else
 echo "Programm wurde abgebrochen"
fi
rm $$tmp # Loeschen der dynamisch
 # generierten temporaeren
 # Datei
```

Die im Shell-Skript benutzte Syntax

```
eingabe=$(cat $$tmp)
```

ist eine Variante der Befehlssubstitution.





### Menüprogrammierung

Folgendes Beispiel zeigt die Realisierung eines Auswahlmenüs mithilfe der Option `-menu`. Es stellt die drei Menüoptionen 1 `joe` (für den Aufruf des Editors), 2 `X Window` (für den Aufruf des X-Window-Systems) und 0 `Menue verlassen` zur Verfügung (siehe Abbildung 16.12).

Vergleichen Sie hierzu auch den entsprechenden Abschnitt „*Befehlssubstitution*“ im Kapitel 4 *Benutzerumgebung*.



### Hinweis



Abb. 16.12

Beispiel für ein mit dem Befehl `dialog` programmiertes Auswahlmenü

Auch hier wird das Ergebnis wieder in eine Fehlerdatei umgeleitet, die dynamisch unter Zuhilfenahme der Prozessnummer gebildet wird und zur weiteren Auswertung ausgelesen werden muss (`auswahl=$(cat $$tmp)`).

Für den Fall, dass der Editor `joe` gewählt wurde, erscheint eine Eingabebox, in die der Name der zu editierenden Datei eingegeben werden muss. Diese Datei wird auf ihr Vorhandensein hin überprüft (`if ! test -e $datname`).

Existiert sie bereits, wird sie zum Editieren geöffnet; kann sie jedoch nicht gefunden werden, wird der Anwender in einem weiteren Dialogfenster aufgefordert, festzulegen, ob sie neu erstellt werden soll oder nicht.

Entscheidet sich der Anwender positiv, wird der Editor geöffnet, andernfalls wird zur äußeren Schleife zurückgekehrt (`continue`).

Unabhängig davon, ob der Editor oder das X-Window-System über das Menü gewählt wurde, wird nach Verlassen des entsprechenden Programms immer wieder in das Menü zurückgekehrt, bis der Menüpunkt 0 `Menue verlassen` gewählt wird (`until [ $auswahl = 0 ]`).

Beachten Sie, dass bei dieser Form des Dialogfensters nicht nur die Höhe und die Breite des Fensters angegeben werden müssen, sondern auch die Anzahl der zur Verfügung stehenden Menüpunkte.

Hier der komplette Code dieses (schon etwas gehaltvoller) Programms:



## Shell-Programmierung

---



```
#Skript zur Darstellung der Menueprogrammierung
until [$auswahl=0]
do
 dialog --clear --menu "Auswahlmenue" 15 25 3 \
 "1" "joe" "2" "X Window" "0" "Menue verlassen" \
 2> $$tmp
 auswahl=$(cat $$tmp)
 setterm -clear # Bildschirm löschen
 case $auswahl in
 "1") dialog --clear --title "J O E" \
 --inputbox "Welche Datei soll editiert \
 werden:" 10 25 2> $$tmp
 datname=$(cat $$tmp)
 if ! test -e $datname
 then
 dialog --clear --title "J O E" \
 --yesno "Datei nicht gefunden. Soll sie \
 neu erstellt werden?" 8 27
 jn=$?
 if [$jn = 0]
 then
 joe $datname
 else
 continue
 fi
 else
 joe $datname
 fi;;
 "2") startx;;
 "0") echo "Menue-Ende";;
 esac
 rm $$tmp
done
```

### Optionsauswahl

Optionslisten ermöglichen die Auswahl einer oder mehrerer Programmoptionen, die innerhalb des Dialogfensters mit der Leertaste K aktiviert bzw. deaktiviert werden können. Eine aktivierte Option wird durch ein in den vorangestellten Klammern stehendes X als solche kenntlich gemacht.

Wie bereits in *Tabelle 16.9* aufgeführt, müssen dem Dialogelement *-check-list* pro Option drei Informationen übergeben werden:

- Das Argument *Optionspunkt* (*op*) definiert die Gesamtzahl der möglichen Optionen.
- Der *Optionstext* (*op\_1* bis *op\_n*) ermöglicht es, Optionsbeschreibungen anzugeben.



## Dialoge



- Der Status der Option (*stat\_1* bis *stat\_n* mit den möglichen Werten *on* bzw. *off*).

Das Ergebnis lässt sich in eine Datei umleiten, aus der es zur weiteren Bearbeitung ausgelesen werden kann, wobei sich aber einige Probleme ergeben, wie Sie gleich sehen werden.

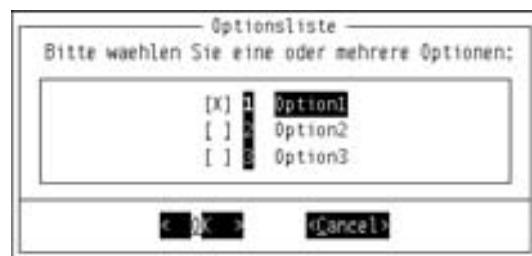
Zunächst ein einfaches Beispiel zur prinzipiellen Handhabung dieses Dialogelements, das nichts weiter tut, als die von Ihnen gewählten Optionen in einer Textbox auf dem Bildschirm auszugeben:

```
Skript \6opts\9 zur Darstellung der Optionsauswahl
dialog --clear --title "Optionsliste" \
--checkbox "Bitte wählen Sie eine oder \
mehrere Optionen:" 10 50 3 \
"1" "Option1" on \
Diese Option wird durch den
Status on als standardgemäß
voreingestellte Option
kenntlich gemacht
"2" "Option2" off \
Diese und die folgende Option
werden durch den Status \6off\9
standardgemäß ausgeschaltet
"3" "Option3" off 2> $$tmp
dialog --clear --title "Ergebnis der Optionsauswahl" \
--textbox $$tmp 10 34
rm $$tmp
```

Wie Sie in Abbildung 16.13 erkennen können, ist beim Aufruf dieses Skripts die *Option1* aktiviert (Status = *on*).

Abb. 16.13

Dialogbox mit Options-  
liste



Wenn Sie dieses Skript ablaufen lassen und alle drei Optionen wählen, erhalten Sie das Ausgabeergebnis aus Abbildung 16.14.



## Shell-Programmierung

---



Abb. 16.14 ○

Bildschirmausgabe des Ergebnisses des Skripts *opts* nach Auswahl aller drei Optionen



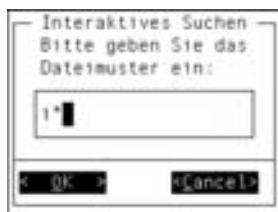
Sie sehen also, dass die Ergebnisse dieses Dialogs in Anführungszeichen stehen und jeweils durch Leerzeichen voneinander getrennt sind.

Eine Möglichkeit, dieses Problem für weitere Auswertungen in den Griff zu bekommen, soll das nachfolgende Beispiel zeigen.

Es handelt sich hierbei um ein Skript *suchdialog*, das eine interaktive Suche von Dateien, ausgehend vom aktuellen Verzeichnis, ermöglichen soll.

Abb. 16.15 ○

Interaktiver Suchdialog



Nach Eingabe des Dateimusters kann in einer Optionsliste festgelegt werden, welche weiteren Suchkriterien berücksichtigt werden sollen.

Abb. 16.16 ○

Optionsliste zur Auswahl der Suchkriterien



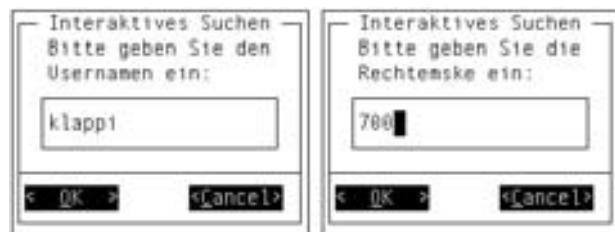
## Dialoge



In Abhängigkeit davon, ob eine Berücksichtigung des Usernamens und/oder der Rechtemaske erfolgen soll, werden weitere Inputboxen angezeigt.

Um das oben beschriebene Problem der jeweils durch Leerzeichen getrennten Rückgabewerte in Anführungsstrichen zu lösen, wurde hier der Stream-Editor *sed* benutzt, der diese Zeichen in zwei getrennten Durchgängen ausfiltert (siehe Programmzeilen 13 und 15):

Abb. 16.17 ○  
Inputboxen zur Eingabe  
des Usernamens und  
der Rechtemaske



Hier wurde der Editor *sed* verwendet, da interaktives Editieren natürlich in diesem Zusammenhang nicht erwünscht ist.



Hinweis

```
Dialog zum interaktiven Suchen von Dateien
until [$again=1]
do
 dialog --clear --title "Interaktives Suchen" \
 --inputbox "Bitte geben Sie das Dateimuster ein:" 9 25
2> $$tmp
 datname=$(cat $$tmp)
 dialog --clear --title "Interaktives Suchen" \
 --checkbox "Bitte legen Sie das Suchmuster fest:" 15 50
3 \
 "1" "Nach Namen" on \
 "2" "Nach User" off \
 "3" "Rechtemaske berücksichtigen" off 2> $$tmp
Ausfiltern der Anführungszeichen
sed s/"""/"/g $$tmp > ausgabe1.tmp
Ausfiltern der Leerzeichen
sed s/"/"/"/g ausgabe1.tmp > ausgabe2.tmp
auswahl=$(cat ausgabe2.tmp)

if ["$auswahl" = "12" -o "$auswahl" = "123" -o
"$auswahl" = "2" \
-o "$auswahl" = "23"]
then
 dialog --clear --title "Interaktives Suchen" \
 --inputbox "Bitte geben Sie den Usernamen ein:" 9 25
2> $$tmp
 username=$(cat $$tmp)
```

665



## Shell-Programmierung

---



```
fi

if ["$auswahl" = "123" -o "$auswahl" = "13" -o
"$auswahl" = "23" \
-o "$auswahl" = "3"]
then
 dialog --clear --title "Interaktives Suchen" \
 --inputbox "Bitte geben Sie die Rechtemaske ein:" 9 25
2> $$tmp
 rechte=$(cat $$tmp)
fi
case $auswahl in
 "1") find . -name "$datname" -print > suchliste
 dialog --clear --title "Suchergebnis"
 --textbox suchliste 10 40;;
 "12") find . -name "$datname" -user "$username"
 -print > suchliste
 dialog --clear --title "Suchergebnis"
 --textbox suchliste 10 40;;
 "123") find . -name "$datname" -user "$username"
 -perm "$rechte" \
 -print > suchliste
 dialog --clear --title "Suchergebnis"
 --textbox suchliste 10 40;;
 "13") find . -name "$datname" -perm "$rechte" -print >
 suchliste
 dialog --clear --title "Suchergebnis"
 --textbox suchliste 10 40;;
 "2") find . -user "$username" -print > suchliste
 dialog --clear --title "Suchergebnis"
 --textbox suchliste 10 40;;
 "23") find . -user "$username" -perm "$rechte" -print >
 suchliste
 dialog --clear --title "Suchergebnis"
 --textbox suchliste 10 40;;
 "3") find . -perm "$rechte" -print > suchliste
 dialog --clear --title "Suchergebnis"
 --textbox suchliste 10 40;;
esac
dialog --clear --title "Interaktives Suchen" \
--yesno "Neuer Suchvorgang?" 5 40
again=$?
done
rm $.tmp
rm ausgabe1.tmp prm ausgabe2.tmp
clear
```

Die Einschränkung der Auswahl auf nur eine Option aus einer Optionsliste geschieht mit dem Dialogelement `-radiolist`, welches bezüglich seiner Handhabung der soeben vorgestellten `-checkbox` entspricht.



## Dialoge



Das folgende kurze Programmbeispiel *radio* dient zur Festlegung zu löschernder temporärer Dateien. Es kann alternativ festgelegt werden, ob alle Dateien, die mit *.tmp* oder mit der Tilde (~) oder mit *.tst* enden, gelöscht werden sollen:

```
Dialog zum interaktiven Suchen von Dateien
until [$again = 1]
do
 dialog --clear --title "Loeschbefehl" \
 --radiolist "Zu loeschende Dateien:" 15 50 3 \
 "1" "*.tmp" on \
 "2" "*~" off \
 "3" "*.tst" off 2> $$tmp
 ergebnis=$(cat $$tmp)
 case $ergebnis in
 "1") rm *.tmp;;
 "2") rm *~;;
 "3") rm *.tst;;
 esac
 dialog --clear --title "Loeschbefehl" \
 --yesno "Neuer Loeschvorgang?" 5 40
 again=$?
done
rm $$tmp
clear
```

Beim Aufruf dieses Programms erhalten Sie die Bildschirmausgabe aus *Abbildung 16.18*.

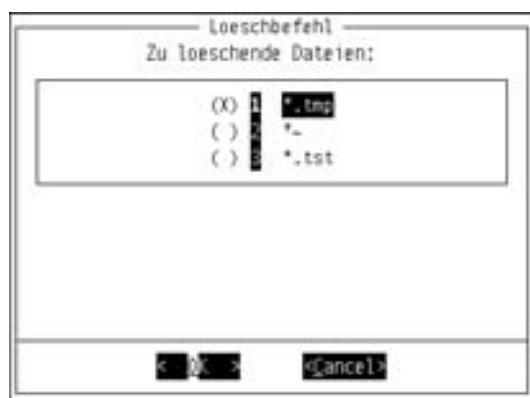


Abb. 16.18

Ausschließliche Auswahlaus einer Radiolist

Zum Abschluss dieses Kapitels sollen hier noch die beiden Dialogelemente *-backtitle* und *-gauge* vorgestellt werden.

Das Dialogelement *-backtitle* stellt eine Variation des zuvor wiederholt eingesetzten Elements *-title* dar, muss hier also nicht weiter beschrieben werden.

## Shell-Programmierung

---



Das Dialogelement `-gauge` dient zur Programmierung eines Statusbalkens, der eine prozentuale Anzeige darstellt, wie sie hinlänglich z.B. aus diversen Installationsprogrammen bekannt ist. Dort wird sie benutzt, um den Fortschritt der Installation zu visualisieren.

Hierbei wird der jeweilige anzugebende Prozentsatz von der Standardeingabe gelesen, was in einem Shell-Skript etwa durch eine Pipeline mit dem Befehl `echo` bewerkstelligt werden kann.

Im folgenden (nicht ganz so ernst zu nehmenden ...) Beispiel wird eine rückwärts laufende Statusanzeige erstellt, indem eine Variable `i` von `101` auf `0` herabgezählt und jeweils durch den Befehl `echo` an den Dialog übergeben wird:

```
i=101
until [$i -eq 0]
do
 i=$[$i-1]
 echo "$i" \
 | dialog --backtitle "Windoof '97 De-Installation \
 (C)opyleft 1997, by Mickeysoft" \
 --gauge "Fortschritt" 10 30 $i
done
clear
```

Während des Programmablaufs wird sich der Bildschirm bei jedem neu eingelesenen Wert der Variablen `i` neu aufbauen (was leider ein leichtes Flackern zur Folge hat). Der Wert wird dann sowohl als Prozentzahl sowie als (farbiger) Balken angezeigt.



Abb. 16.19

Mit dem Dialogelement  
`-gauge` erzeugter Sta-  
tusbalken





Als Variante zu den zuvor vorgestellten Programmen wurde hier das Dialogelement `-backtitle` statt `-title` verwendet, wodurch der Titel am oberen Bildschirmrand angezeigt wird.



### Ein Hinweis für die KDE-Fans unter Ihnen

*Das KDE-Projekt bemüht sich, einen Ersatz für wirklich alle Tools, die es für Linux gibt, zur Verfügung zu stellen.*

*Entsprechend gibt es mittlerweile das Tool kdialog, das Sie analog zu den hier dargestellten Ausführungen verwenden können (es gibt geringfügige Syntax-Abweichungen – so müssen Sie beispielsweise nicht länger die Größe der Dialogfenster angeben), wobei dann die hiermit von Ihnen erstellten Dialoge GUI-basiert sind.*

*Schauen Sie einmal auf die Webseiten des KDE-Projekts (<http://www.kde.org>), um sich über den Fortschritt auch dieses Projekts zu informieren.*





# Programmierung in Perl



|                                                                   |            |
|-------------------------------------------------------------------|------------|
| Aufbau von Perl-Programmen                                        | <b>679</b> |
| Ein- und Ausgabefunktionen                                        | <b>680</b> |
| Variablen                                                         | <b>683</b> |
| Arrays                                                            | <b>690</b> |
| Kontrollstrukturen                                                | <b>691</b> |
| Funktionen                                                        | <b>701</b> |
| Perl als Hilfswerkzeug für die Manipulation von Datenbankinhalten | <b>704</b> |
| Zusammenfassung                                                   | <b>716</b> |



**17**

## Programmierung in Perl

---



In diesem Kapitel erhalten Sie eine kurze Einführung in die Programmiersprache Perl, die sich speziell im Zuge des allgemeinen Internet-Hypes zunehmender Popularität erfreut.

Die Gründe hierfür wurden bereits eingangs dieses Buchabschnitts genannt: Die große Flexibilität bezüglich der Lösungsansätze zur Bewältigung von Programmierproblemen, die relativ einfache Erlernbarkeit und die universelle Verfügbarkeit haben diese Sprache in den letzten Jahren zu einer der beliebtesten und einflussreichsten Programmiersprachen gemacht.

Die Sprache Perl (**P**ractical **E**xtraction and **R**eport **L**

Tatsächlich gibt es mittlerweile „Übersetzungsprogramme“, die andere Programmiersprachen-Codes in Perl übertragen (*a2p* für die Übersetzung von awk- in Perl-Programme, *s2p* für die Übersetzung von sed- in Perl-Programme ...).

Die Bereitstellung im USENET führte sehr schnell zu einer weiten Verbreitung dieser Sprache und bewirkte, dass sich viele User mit der Bitte an den Autor wandten, der Sprache weitere Leistungsmerkmale hinzuzufügen bzw. die bereits vorhandenen bezüglich ihrer Implementierung anzupassen. Dies geschieht auch heute noch: Perl wird mittlerweile zwar von einer Entwicklergruppe, die sich *Perl5Porters* (oder auch *p5p*) nennt, gepflegt und ständig durch neue Funktionalitäten erweitert, Larry Wall hat jedoch immer noch „das letzte Wort“.

### Hinweis



Beachten Sie hierzu auch das *Kapitel 14 Mit Linux ins Internet*.

Diese Berücksichtigung vieler unterschiedlicher Interessen führte dazu, dass die Sprache recht umfangreich erscheint (Puristen würden vielleicht sagen: aufgebläht), doch erlaubt dieser Sprachumfang eine Flexibilität in der Programmierung, die bei anderen Sprachen in dieser Form nicht vorhanden ist: So, wie viele Wege nach Rom führen, bietet Perl eine große Zahl an Ansätzen zur Lösung eines und desselben Problems, wie sie woanders wohl kaum zu finden ist.

Die Vorteile können noch einmal wie folgt zusammengefasst werden:

- Perl vereint die besten Leistungsmerkmale von Programmiersprachen wie C, awk, sed, Basic, der Shell-Programmierung und weiterer Sprachen in sich.
- Perl arbeitet mithilfe der abstrakten Datenbankschnittstelle *DBI* mit Datenbanken wie Oracle, Sybase, Postgres und weiteren Datenbanksystemen zusammen.
- Perl kann mit HTML, XML und weiteren Mark-up-Sprachen zusammenarbeiten.



## Programmierung in Perl

- Perl bietet die Unterstützung von Unicode (also die Unterstützung verschiedener, auch nicht-lateinischer Zeichensätze).
- Perl unterstützt sowohl die prozedurale als auch objektorientierte Programmierung.
- Perl besitzt Schnittstellen zu externen C/C++-Bibliotheken durch XS oder SWIG.
- Das **Comprehensive Perl Archive Network (CPAN)** (siehe <http://www.cpan.org>) bietet über 400 Module. Hierbei handelt es sich um Sammlungen von Funktionen, die zur Erledigung häufig wiederkehrender Aufgaben dienen, ohne dass hierfür jedes Mal „das Rad wieder neu erfunden“ werden muss. Eine recht beachtliche Zahl dieser Module finden Sie bereits in der Serie *Perl*, wenn Sie Besitzer einer SuSE-Linux-Vollversion sind. Diese beinhaltet folgende Module (von denen Sie sich bitte an dieser Stelle nicht erschlagen fühlen sollen):

### Perl-Module als Bestandteil der SuSE-Vollversionen

| Modul                          | Inhalt                                                                                                                                                                                          |
|--------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>cstools</i>                 | Tools für den Umgang mit tschechischen Texten in Perl                                                                                                                                           |
| <i>dbdXbase</i>                | Das Modul XBase ermöglicht Zugriff auf XBase (dBBase, Fox*)-Datenbankdateien wie <i>.dbf</i> , <i>.abt</i> , <i>fpt</i> , <i>.ndx</i> , <i>.ntx</i> , <i>.mdx</i> , <i>.idx</i> und <i>.cdx</i> |
| <i>Perl-Apache-ASP</i>         | Dieses Modul stellt die Active-Server-Pages-Funktionalität für den Apache-Webserver mittels Perl als Skriptsprache zur Verfügung                                                                |
| <i>Perl-Apache-AuthNetLDAP</i> | Mit diesem Perl-Modul zu dem Webserver Apache ist es möglich, eine User-Authentifizierung über das LDAP Protokoll durchzuführen                                                                 |
| <i>Perl-Apache-AutoIndex</i>   | Dieses Modell ist ein kompletter Ersatz für <i>mod_autoindex</i> und <i>mod_dir</i> , welche Bestandteil von Apache sind                                                                        |
| <i>Perl-Apache-Filter</i>      | ändert den Output vorheriger Handler                                                                                                                                                            |
| <i>Perl-Apache-Icon</i>        | Dieses Modul extrahiert Teile aus <i>mod_autoindex</i> , um eine Perl-Schnittstelle zum Auffinden von Icons zu adaptieren                                                                       |
| <i>Perl-Apache-Language</i>    | Ziel dieses Moduls ist es, für das Apache-Modul <i>mod_perl</i> Mehrsprachenunterstützung bereitzustellen                                                                                       |

## Programmierung in Perl

### Perl-Module als Bestandteil der SuSE-Vollversionen (Forts.)

| Modul                               | Inhalt                                                                                                                                                                                                                                                                            |
|-------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Perl-Apache-SSI</i>              | Dieses Modul implementiert die Funktionalität des Apache-Moduls <i>mod_include</i> , um serverseitige HTML-Dokumente behandeln zu können. Es läuft mit dem Apache-Modul <i>mod_perl</i> .                                                                                         |
| <i>Perl-Apache-Sandwich</i>         | Mit diesem Perl-Modul zu Apache können sehr leicht Header und Footer zu bestehenden HTML-Dokumenten hinzugefügt werden                                                                                                                                                            |
| <i>Perl-Apache-Session</i>          | Dieses Modul stellt die persistente Speicherung von Daten in beliebigen Data-Management-Systemen bereit. Die Details der Interaktion mit den Data-Management-Systemen werden abstrahiert, damit verhalten sich alle Systeme gleich. Das Programm interagiert mit einem Tied Hash. |
| <i>Perl-ApacheDBI</i>               | Dieses Perl-Modul stellt für den Apache-Webserver die Möglichkeit zur Verfügung, sich über das Perl-DBI-Modul zu authentifizieren                                                                                                                                                 |
| <i>Perl-Compress-Zlib</i>           | Perl-Schnittstelle zu einem Teil der <i>info-zip zlib compression library</i>                                                                                                                                                                                                     |
| <i>Perl-Config-IniFiles</i>         | Modul zum Lesen von <i>.ini</i> -Style-Konfigurationsdateien                                                                                                                                                                                                                      |
| <i>Perl-Convert-ASN1</i>            | Zwischen Perl-Datenstrukturen und ASN.1-kodierten Paketen konvertieren                                                                                                                                                                                                            |
| <i>Perl-Convert-BER</i>             | Perl-Objektklasse zum Kodieren und Dekodieren von Objekten wie beschrieben durch den ITU-T-Standard X.209 (ASN.1)                                                                                                                                                                 |
| <i>Perl-Convert-TNEF</i>            | Perl-Modul zum Lesen von TNEF-Dateien                                                                                                                                                                                                                                             |
| <i>Perl-Convert-Uulib</i>           | Perl-Schnittstelle zur UULib-Library                                                                                                                                                                                                                                              |
| <i>Perl-Curses</i>                  | Dies ist ein dynamisch ladbares Perl-Modul für die Curses-Funktionen. Es ist in jedem CPAN-Archiv erhältlich.                                                                                                                                                                     |
| <i>Perl-Cyrus-IMAP</i>              | IMAP-Perl-Modul von Cyrus für den IMAPD von Cyrus                                                                                                                                                                                                                                 |
| <i>Perl-Cyrus-SIEVE-acap</i>        | Perl-Modul für Cyrus' SIEVE ACAP                                                                                                                                                                                                                                                  |
| <i>Perl-Cyrus-SIEVE-managesieve</i> | Perl-Modul für Cyrus' SIEVE                                                                                                                                                                                                                                                       |



## Programmierung in Perl



### Perl-Module als Bestandteil der SuSE-Vollversionen (Forts.)

| Modul                        | Inhalt                                                                                                                                                                                                                                                                                     |
|------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Perl-DBD-CSV</i>          | DBI-Treiber für CSV-Dateien                                                                                                                                                                                                                                                                |
| <i>Perl-DBI</i>              | Das Perl Database Interface von Tim Bunce                                                                                                                                                                                                                                                  |
| <i>Perl-Data-ShowTable</i>   | <i>ShowTable.pm</i> ist ein Perl5-Modul, das Perl-Funktionen zur formatierten Ausgabe von Listen und Tabellen in den Formaten einfache Tabelle, gerahmte Tabelle, Listenformat und HTML bereitstellt                                                                                       |
| <i>Perl-Date-Calc</i>        | Dieses Paket enthält eine C-Library und ein Perl-Modul für diverse Datumsberechnungen basierend auf dem gregorianischen Kalender, konform zu allen relevanten Normen und Standards, ISO/R 2015-1971, DIN 1355 und teilweise ISO 8601                                                       |
| <i>Perl-DateManip</i>        | Dieses Perl-Modul stellt eine Sammlung von Funktionen bereit, um viele gängige Zeit- und Datumsbearbeitungen zu vereinfachen. Operationen wie das Vergleichen zweier Zeitangaben, Berechnung von Zeitschnitten und die Verarbeitung von Zeitzonenangaben können leicht vorgenommen werden. |
| <i>Perl-Devel-Symdump</i>    | ermöglicht, die Symboltabelle Perls sowie die Klassenhierarchie aus einem laufenden Programm heraus zu untersuchen                                                                                                                                                                         |
| <i>Perl-Digest-MD5</i>       | Perl-Schnittstelle für Message Digests                                                                                                                                                                                                                                                     |
| <i>Perl-File-Magic</i>       | Perl-Modul zum Herausfinden des Typs einer Datei                                                                                                                                                                                                                                           |
| <i>Perl-GD</i>               | Das Paket enthält ein Schnittstellenmodul für die <i>libperl-GD</i> . Eine beliebte Perl-Bibliothek zur Erzeugung von GIF-Dateien.                                                                                                                                                         |
| <i>Perl-HTML-Clean</i>       | Dieses Perl-Modul dient als Filter für HTML-Code. Der erzeugt HTML-Code ist „bereinigt“: Er enthält keine White Spaces und kann damit schneller übertragen und im Browser dargestellt werden.                                                                                              |
| <i>Perl-HTML-Parser</i>      | Eine Sammlung von Modulen, die Informationen aus HTML-Dokumenten verarbeiten                                                                                                                                                                                                               |
| <i>Perl-HTML-SimpleParse</i> | Ein sehr einfacher HTML-Parser, ähnlich dem zuvor genannten Modul <i>HTML::Parser</i>                                                                                                                                                                                                      |

## Programmierung in Perl



### Perl-Module als Bestandteil der SuSE-Vollversionen (Forts.)



| Modul                           | Inhalt                                                                                                                                                                                                                              |
|---------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Perl-h8N-LangTags</i>        | Sprach-Tags sind ein Formalismus (beschrieben in RFC 1766), der definiert, in welcher Sprache eine Informationseinheit formuliert ist. Diese Bibliothek beinhaltet Funktionen für übliche Aufgaben, die Sprach-Tags involvieren.    |
| <i>Perl-IO-Socket-SSL</i>       | Dieses Modul ist eine Klasse, die eine objektorientierte Schnittstelle in SSL-Sockets implementiert. Die Klasse stammt von <i>IO::Socket::INET</i> und stellt eine Reihe von Schnittstellenmethoden der „Base Class“ zur Verfügung. |
| <i>Perl-IO-stringy</i>          | Ein-/Ausgaben auf <i>in-Core</i> -Objekte wie Strings und Arrays                                                                                                                                                                    |
| <i>Perl-MIME-Base64</i>         | Dieses Paket beinhaltet einen Encoder/Decoder für <i>base64</i> und Quoted-Printable. Diese Kodiermethoden sind im RFC 2045-MIME ( <b>M</b> ultipurpose <b>I</b> nternet <b>Mail <b>E</b>xtensions) spezifiziert.</b>               |
| <i>Perl-MIME-tools</i>          | Modul zum Parsen und Erzeugen von MIME-Entitäten                                                                                                                                                                                    |
| <i>Perl-MLDBM</i>               | Dieses Perl-Modul ist in der Lage, multidimensionale Hash-Strukturen in Tied Hashes umzuwandeln                                                                                                                                     |
| <i>Perl-Mailtools</i>           | Ein Set von Perl-Modulen für Mailapplikationen                                                                                                                                                                                      |
| <i>Perl-Mysql-MySQL-modules</i> | Dieses Paket enthält die entsprechenden Module, um eine MySQL-Datenbank mit Perl anzusprechen                                                                                                                                       |
| <i>Perl-Net-DNS</i>             | Perl-Interface zum DNS Resolver                                                                                                                                                                                                     |
| <i>Perl-Net-Daemon</i>          | Perl-Erweiterung für portable Daemons                                                                                                                                                                                               |
| <i>Perl-Net-Netmask</i>         | Parsen, Manipulieren und Überprüfen von IP-Netzwerkblöcken                                                                                                                                                                          |
| <i>Perl-Net-SNMP</i>            | Implementiert eine objektorientierte Schnittstelle in das <b>S</b> imple <b>N</b> etwork <b>M</b> anagement <b>P</b> rotocol ( <i>SNMP</i> )                                                                                        |
| <i>Perl-Net-SSLeay</i>          | Perl-Modul für den Einsatz von OpenSSL                                                                                                                                                                                              |
| <i>Perl-NetxAP</i>              | Dieses Modul bietet eine Schnittstelle zu den Protokollfamilien IMAP, IMSP, ACAP und ICAP                                                                                                                                           |



## Programmierung in Perl



### Perl-Module als Bestandteil der SuSE-Vollversionen (Forts.)

| Modul                        | Inhalt                                                                                                                                                                                                                                                                                                       |
|------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Perl-PDA-Pilot</i>        | Dieses Paket enthält Perl-Module, um mit einem PalmPilot zu kommunizieren                                                                                                                                                                                                                                    |
| <i>Perl-PDL</i>              | implementiert die Möglichkeit für numerische Berechnungen. Das PDL-Modul ermöglicht es, große N-dimensionale Felder kompakt zu speichern und schnell zu verarbeiten.                                                                                                                                         |
| <i>Perl-Parallel-MPI</i>     | Perl-Modul für MPI Message Passing ( <i>MPICH</i> )                                                                                                                                                                                                                                                          |
| <i>Perl-Parallel-Pvm</i>     | Perl-Modul für das Message Passing System ( <i>PVM</i> )                                                                                                                                                                                                                                                     |
| <i>Perl-Parse-RecDescent</i> | generiert rekursive Text-Parser aus yacc-ähnlichen Spezifikationen                                                                                                                                                                                                                                           |
| <i>Perl-Paw</i>              | stellt einige Elemente zur Erstellung grafischer Benutzeroberflächen im Textmodus bereit                                                                                                                                                                                                                     |
| <i>Perl-PerlMagick</i>       | PerlMagick ist ein objektorientiertes Perl-Interface für ImageMagick. Mit diesem Modul kann man Bilder oder Bildsequenzen aus einem Perl-Skript lesen, schreiben oder verändern. Dadurch lässt es sich z.B. für CGI-Skripte für das Web einsetzen. Es benötigt Image-Magick-4.0.8 und Perl 5.002 oder neuer. |
| <i>Perl-SNMP</i>             | Perl „SNMP Extension Module“ für den <i>ucd-snmpd</i>                                                                                                                                                                                                                                                        |
| <i>Perl-SQL-Statement</i>    | SQL stellt dem Benutzer einen Sprachschatz zur Verfügung, mit dessen Vokabeln er alle erdenklichen Abfragen und Manipulationen an Datenbanken vornehmen kann                                                                                                                                                 |
| <i>Perl-Storable</i>         | Implementierung persistenter Objekte                                                                                                                                                                                                                                                                         |
| <i>Perl-Syslog</i>           | Perl Interface zum Syslog                                                                                                                                                                                                                                                                                    |
| <i>Perl-Text-CSV_XS</i>      | Dieses Modul bietet die Möglichkeit, CSV-Werte ( <i>comma separated values</i> – ursprünglich also „komma-separierte Werte“, nunmehr aber Datenstrukturen, die durch ein klares Trennzeichen getrennt sind) zu bilden oder zu entschlüsseln                                                                  |
| <i>Perl-Tie-TxHash</i>       | Modul für ordnungserhaltende assoziative Arrays in Perl                                                                                                                                                                                                                                                      |
| <i>Perl-TimeDate</i>         | Diese Perl-Module stellen Funktionen zur Zeit- und Datumsberechnung zur Verfügung                                                                                                                                                                                                                            |
| <i>Perl-URI</i>              | Perl-Schnittstelle für URI-Objekte                                                                                                                                                                                                                                                                           |

## Programmierung in Perl

### Perl-Module als Bestandteil der SuSE-Vollversionen (Forts.)

| Modul                   | Inhalt                                                                                                                                                                                                                                                                          |
|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Perl-XML-Dom</i>     | Eine Erweiterung für das Perl-Modul <i>XML::Parser</i> , einer Perl-Schnittstelle zu James Clarks XML-Parser <i>expat</i> . <i>XML::DOM</i> ermöglicht es <i>XML::Parser</i> , eine objektorientierte Datenstruktur mit einer DOM-Ebene-1-konformen Schnittstelle zu erstellen. |
| <i>Perl-XML-Parser</i>  | XML-Parser als Perl-Modul                                                                                                                                                                                                                                                       |
| <i>Perl-gettext</i>     | <i>gettext</i> für Perl                                                                                                                                                                                                                                                         |
| <i>Perl-ldap</i>        | Ein Client-Interface zum LDAP-Server                                                                                                                                                                                                                                            |
| <i>Perl-ldap-ssl</i>    | SSL-Erweiterung zu <i>perl-ldap</i>                                                                                                                                                                                                                                             |
| <i>Perl-libnet</i>      | Einige Client-Interfaces für Netzwerkdienste wie SMTP, POP usw.                                                                                                                                                                                                                 |
| <i>Perl-libwww-perl</i> | <i>Libwww-perl</i> ist eine Sammlung von Perl-Modulen, die ein einfaches und konsistentes Application Programming Interface (API) für das World Wide Web (WWW) zur Verfügung stellt                                                                                             |
| <i>Perl_bf</i>          | Perl-Blowfish-Algorithmus                                                                                                                                                                                                                                                       |
| <i>Perl_cpx</i>         | Kryptografische Erweiterungen für Perl (RSA, Blowfish, IDEA ...)                                                                                                                                                                                                                |
| <i>Perl_qt</i>          | Perl QT ist ein Perl-Modul mit dessen Hilfe Qt Widgets aus Perl verwendet werden können                                                                                                                                                                                         |
| <i>Perl_tar</i>         | Perl-Modul zur Erzeugung und Manipulation von tar-Dateien                                                                                                                                                                                                                       |
| <i>Perl_tk</i>          | <i>Perl Tk</i> ist eine Erweiterung für Perl. Mit <i>Perl Tk</i> lassen sich die Vorteile von <i>Tk</i> und die mächtige Skriptsprache Perl auf angenehme Weise kombinieren.                                                                                                    |

Tabelle 17.1

An dieser Stelle kann nur eine kurze Einführung in diese leistungsfähige Sprache gegeben werden, die allenfalls dazu dienen kann, Ihnen einen Einstieg in diese Thematik zu bieten, der vielleicht ein tiefer gehendes Interesse bei Ihnen weckt, sich intensiver mit ihr auseinander zu setzen.

Zum Ende des Kapitels soll jedoch ein einigermaßen plausibles Beispiel zustande kommen, das einige der Möglichkeiten von Perl bei der Manipulation von Datenbanken aufzeigen soll.



## Aufbau von Perl-Programmen



### Aufbau von Perl-Programmen

Bei Perl handelt es sich um eine so genannte „formatfreie Sprache“. Dies bedeutet, dass Trennzeichen (*whitespace characters*) wie Zeilenumbrüche, Tabulatoren, Leerzeichen und Leerzeilen in erster Linie dazu dienen, den Programmcode übersichtlicher zu gestalten. Sie können also durchaus den gesamten Code eines Programms in eine einzelne Zeile schreiben oder auf Einrückungen logisch zusammenhängender Anweisungsblöcke verzichten, was dem Programm sicherlich einen Hauch von Genialität verleiht, doch seine Lesbarkeit nicht nur für andere, sondern auch für Sie selbst unnötig erschwert, wenn nicht sogar gänzlich unmöglich macht, falls Sie das Programm später einmal ergänzen oder ändern wollen.

Zur besseren Lesbarkeit (und damit zum besseren Verständnis) eines Programms trägt ebenfalls die Verwendung von Kommentaren bei, woraufg im *Kapitel 16 Shell-Programmierung* bereits mehrfach hingewiesen wurde. Diese werden (wie in der Shell-Skriptprogrammierung) durch ein Doppelkreuz (#) eingeleitet, welches bewirkt, dass sämtliche Zeichen bis zum Zeilenende als Kommentar interpretiert werden.

Perl-Programme bestehen aus Anweisungen, die aus einzelnen oder Kombinationen von so genannten „Ausdrücken“ gebildet werden, welche durch ein Semikolon abgeschlossen werden.

Die letzte Anweisung eines Anweisungsblocks benötigt eigentlich kein Semikolon, um sie als solche kenntlich zu machen, doch sollten Sie es aus Gründen der Einheitlichkeit des Erscheinungsbildes setzen.



Hinweis

Ausdrücke können Operationen mit Variablen, Funktionsaufrufe oder andere Konstruktionen sein, die einen so genannten „Ausdruckswert“ zurückgeben (dazu später mehr).

Mehrere logisch zusammenhängende Anweisungen werden durch Klammerung mit geschweiften Klammern {...} zu Anweisungsblöcken zusammengefasst, was zum Beispiel dann geschehen muss, wenn nach einer so genannten „Kontrollstruktur“, wie zum Beispiel einer Iteration, nicht nur eine einzelne Anweisung abgearbeitet werden soll, sondern wiederum eine Reihe (logisch) zusammenhängender Anweisungen.

Ein Perl-Programm könnte also allgemein wie im folgenden Beispiel aufgebaut sein:

```
Anweisung1;
Anweisung2;
Kontrollstruktur
{
 Anweisung3;
 Anweisung4;
 Kontrollstruktur
 {
```

679



## Programmierung in Perl

---



```
Anweisung5;
Kontrollstruktur
 Anweisung6;
 Anweisung7;
}
}
Anweisung8;
```

Grundsätzlich können Perl-Programme direkt am Prompt eingegeben und somit ausgeführt werden, doch werden Sie sie in der Regel (speziell, wenn es sich um längere Programme handelt) in eine Datei schreiben wollen, die selbst ausführbar ist.

Dies ist möglich, indem Sie in der ersten Zeile der Datei angeben, dass zur Interpretation der folgenden Zeilen der Perl-Interpreter verwendet werden soll. Geben Sie hierzu an, wo sich der Interpreter befindet, z.B.

```
#!/usr/bin/perl
```

Zwischen den einzelnen Teilen dieser Zeile darf *kein* Leerzeichen stehen!

Tipp



Sollen auch andere User Ihres Linux-Systems dieses Perl-Skript ausführen oder verändern dürfen, müssen Sie die Rechtemaske natürlich entsprechend ändern. Verwenden Sie hierzu den Befehl *chmod*.

Nach dem Abspeichern der Datei müssen Sie diese noch mit dem Ausführungsrecht versehen.

Geben Sie hierzu z.B. ein:

```
chmod 700 Dateiname
```

Um nun gleich mit einigen praktischen Beispielen beginnen zu können, wenden Sie sich zunächst den Ein- und Ausgabefunktionen von Perl zu.

## Ein- und Ausgabefunktionen

Um die Ausgabefunktionen einigermaßen aussagefähig darstellen zu können, muss an dieser Stelle kurz der Begriff der Variablen definiert werden: Diese werden (in ihrer einfachsten Form) durch ein vorangestelltes Dollarzeichen (\$) als solche kenntlich gemacht und können sowohl numerische Werte als auch Zeichenketten enthalten – näheres hierzu weiter unten in diesem Kapitel.

### Ausgabefunktionen

Zur Ausgabe von Text und Variableninhalten können Sie die beiden Funktionen *print()* und *printf()* verwenden, wobei Letztere eine Formatierung der Zeichenketten erlaubt.



## Ein- und Ausgabefunktionen



### Syntax:

```
print (Argument) bzw.
printf (Argument)
```

Bei *Argument* kann es sich sowohl um Konstanten als auch um Ausdrücke handeln. Ausdrücke werden zunächst ausgewertet, wodurch nicht sie selbst, sondern ihre Ausdruckswerte ausgegeben werden.

Die Verwendung der Klammern ist bei beiden Funktionen optional, doch sollten sie zum einen aus Gründen der besseren Lesbarkeit und zum anderen zur Vermeidung von Problemen bei der Ausgabe komplexerer Argumente verwendet werden.



Hinweis

Mit diesem Rüstzeug können Sie bereits ein erstes (zugegebenermaßen nicht sehr gehaltvolles) Programm erstellen, das bei seinem Aufruf nichts weiter tut, als die Zeichenkette „Hallo Welt“ auf dem Bildschirm auszugeben.

Geben Sie hierzu ein:

```
#!/usr/bin/perl
print ("Hallo Welt");
```

Beachten Sie in diesem Zusammenhang auf die *Tabelle 17.4* (Formatelemente in durch doppelte Anführungszeichen geklammerten Zeichenketten) weiter unten.



Tipp

Bei der Ausführung dieses Programms werden Sie feststellen, dass die Funktion *print()* zwar wie erwartet die Zeichenkette „Hallo Welt“ auf dem Bildschirm ausgegeben hat, jedoch steht der Systemprompt direkt hinter der Bildschirmausgabe, es erfolgt also kein Zeilenvorschub. Dieser wird durch die Zeichenkonstante \n bewirkt. Ergänzen Sie Ihr Programm also wie folgt und lassen Sie es erneut ablaufen:

```
#!/usr/bin/perl
print ("Hallo Welt\n");
```

Möchten Sie mit *printf()* eine formatierte Ausgabe von Variablen vornehmen, so können Sie dies mit den in *Tabelle 17.1* aufgeführten Formatelementen bewerkstelligen (beachten Sie hierzu auch den folgenden Abschnitt *Variablen*).

### Formatelemente für die Ausgabe unterschiedlicher Datentypen



| Element | Datentyp     |
|---------|--------------|
| %s      | Zeichenkette |
| %d      | Ganzzahl     |

## Programmierung in Perl

### Formatelemente für die Ausgabe unterschiedlicher Datentypen (Forts.)

| Element | Datentyp       |
|---------|----------------|
| %f      | Fließkommazahl |

Tabelle 17.2

## Eingaben von der Standardeingabe lesen

Die Eingabe von Benutzerdaten erfolgt in der Perl-Programmierung durch einfaches Einlesen der Standardeingabe und deren Zuweisung an eine Variable.

### Syntax

```
$variable=<STDIN>
```

Die Eingabebestätigung der Standardeingabe erfolgt bei einer Tastatureingabe immer über , welches innerhalb der Variablen mit abgespeichert wird.

Dies hat bei einer weiteren arithmetischen Verwendung der Variablen keinen Einfluss, führt jedoch beispielsweise bei Zeichenkettenvergleichen zu Problemen.

### Hinweis



Der Abschnitt *Funktionen* beschreibt die Funktionen *chop()* und *chomp()* für das Abschneiden der Standardeingabe.

## Eingaben als Befehlszeilenparameter lesen

Wenn Werte als Befehlszeilenparameter an das Perl-Skript weitergeleitet werden sollen, können diese aus dem übergebenen so genannten „Array“ (mehr zu Arrays weiter unten) ausgelesen und einzelnen Variablen übergeben werden. Dies geschieht mit folgender

### Syntax

```
$variable=$ARGV[position_des_parameters]
```

Wenn einem Perl-Skript also beispielsweise zwei Werte als Befehlszeilenparameter übergeben wurden, könnten diese wie folgt auslesen:

### Hinweis



Zu beachten ist an dieser Stelle, dass der erste übergebene Befehlszeilenparameter die Position 0, der zweite die Position 1 usw. hat.

```
#!/usr/bin/perl
$var1=$ARGV[0];
$var2=$ARGV[1];
```





```
print "Uebergebene Variablen:\n";
print "Variable 1: $var1\n";
print "Variable 2: $var2\n";
```

Hieße das Skript `bfz_para-test.pl` und Sie gäben ein:

```
bfz_para-test.pl Hallo Welt
```

so wäre das Ergebnis dieser Ausgabe:

```
Uebergebene Variablen:
Variable 1: Hallo
Variable 2: Welt
```

## Variablen

Die Flexibilität von Programmen wird dadurch erreicht, dass in ihnen variable Werte verarbeitet werden können, deren Werte sich von Programmdurchlauf zu Programmdurchlauf ändern können.

Perl verlangt (anders als andere Programmiersprachen, wie zum Beispiel C) nicht, dass Variablen vor ihrem ersten Gebrauch deklariert werden müssen; die Identifizierung des Variablentyps geschieht an beliebiger Stelle des Programms durch ein vorangestelltes Symbol.

Zahlen- oder Zeichenkettenvariablen werden durch ein vorangestelltes Dollarzeichen (\$), dem der Name der Variablen folgt, kenntlich gemacht. Für den Variablennamen kann eine beliebige Folge von Buchstaben (beachten Sie hierbei, dass Perl zwischen Groß- und Kleinschreibung unterscheidet), Ziffern und dem Unterstrich verwendet werden.

Arrays (beachten Sie hierzu den Abschnitt weiter unten in diesem Kapitel) werden durch ein vorangestelltes *commercial at* (@) kenntlich gemacht.

Perl unterscheidet zwischen verschiedenen Datentypen, die im Folgenden betrachtet werden sollen.

## Zahlentypen

Grundsätzlich scheint Perl zwischen so genannten „Ganzzahlen“ und „Fließkomazahlen“ zu unterscheiden, wobei Ganzzahlen entweder dezimal, oktal oder hexadezimal dargestellt werden können.

Statt des Begriffs „hexadezimal“ hört man auch manchmal den Begriff „sedezimal“.



Hinweis



## Programmierung in Perl

Dezimalzahlen werden mit den zehn gewohnten Ziffernsymbolen 0, 1, 2, 3, 4, 5, 6, 7, 8 und 9 als Ganzzahlen mit der Basis 10, Oktalzahlen durch die Voranstellung einer 0 mit den Ziffernsymbolen 0, 1, 2, 3, 4, 5, 6 und 7 als Ganzzahlen mit der Basis 8 und Hexadezimalzahlen durch ein vorangestelltes ox mit den Ziffernsymbolen 0 bis 9 und A, B, C, D, E und F (bzw. ihrer Kleinschreibung a bis f) als Ganzzahlen mit der Basis 16 dargestellt. Folgende Tabelle zeigt drei Dezimalzahlen mit ihren oktalen und hexadezimalen Entsprechungen:

**Dezimale, oktale und hexadezimale Darstellung von Ganzzahlen**

| Dezimal | Oktal | hexadezimal |
|---------|-------|-------------|
| 1       | 01    | 0x1         |
| 123     | 0173  | 0x7B        |
| 217     | 0331  | 0xD9        |

*Tabelle 17.3*

Perl bietet (anders, als dies in anderen Programmiersprachen üblich ist) die Möglichkeit, Oktal- und Hexadezimalzahlen ein negatives Vorzeichen voranzustellen, was in der Praxis allerdings kaum von Nutzen ist, da mit diesen Zahlen im allgemeinen Bitmuster beschrieben werden.

Fließkommazahlen werden durch die Verwendung eines Dezimalpunkts (achten Sie auf die Verwendung eines Punkts statt eines Kommas, da die Verwendung eines solchen zu unerwarteten Ergebnissen führen würde) als solche kenntlich gemacht.

Einleitend wurde erwähnt, dass Perl grundsätzlich zwischen Ganzzahlen und Fließkommazahlen zu unterscheiden „scheint“. Diese Unterscheidung erfolgt intern jedoch nicht: Ganzzahlen werden wie Fließkommazahlen mit acht Byte Länge gespeichert und verarbeitet.

Diese Beschränkung auf acht Byte Länge führt übrigens bei der Verwendung von wiederholten Fließkommaberechnungen im Laufe der Zeit zu Rundungsfehlern, die Sie bei Ihrer Programmgestaltung berücksichtigen sollten!

Sehr große Zahlen werden mit dem Zeichen *E* oder *e* zur Symbolisierung eines Zehnerexponenten dargestellt: 12.34e56 – 77E-8.

**Tipp**



684



### Zeichenkettentypen

Um Zeichenketten zu formulieren, steht Ihnen die gesamte Bandbreite der 256 ASCII-Zeichen zur Verfügung, wobei die Länge der Zeichenketten (noch) einzig durch die Größe des Arbeitsspeichers begrenzt ist. Sie können also somit rein theoretisch eine gesamte Text- oder sogar Binärdatei in einer einzelnen Zeichenkettenvariablen abspeichern, was jedoch in der Praxis selten sinnvoll ist.

Perl kennt zwei verschiedene Darstellungsformen für Zeichenketten:

- In einfache Anführungsstriche gesetzte Zeichen werden „wörtlich“ als Bestandteile einer Zeichenkette interpretiert. Ausgenommen hiervon sind das einfache Anführungszeichen selbst ("), das doppelte Anführungszeichen (") und der Backslash (\). Diese müssen mit einem vorangestellten Backslash als Bestandteil der Zeichenkette kenntlich gemacht werden.

#### In einfache Anführungszeichen gestellte Zeichenketten und ihr Ausgabeergebnis

| Kodierung        | Bildschirmausgabe                                               |
|------------------|-----------------------------------------------------------------|
| "Hallo Welt"     | Hallo Welt                                                      |
| "Hallo Welt\n"   | Hallo Welten ( <i>Keine Interpretation als Zeilenvorschub</i> ) |
| "Hallo \"Welt\"" | Hallo "Welt" (Zweimal ", nicht einmal ")                        |
| "Backslash: \\"  | Backslash: \                                                    |

Tabelle 17.4

- Doppelte Anführungszeichen (") erlauben die Verwendung von Ersatzdarstellungen für (im Allgemeinen) nicht druckbare Zeichen innerhalb der Zeichenkette. Entnehmen Sie diese der Tabelle 17.4.

#### Ersatzdarstellungen in durch doppelte Anführungszeichen geklammerten Zeichenketten

| Ersatzdarstellung | Bedeutung                           |
|-------------------|-------------------------------------|
| \"                | das doppelte Anführungszeichen ("") |
| \\"               | der Backslash (\)                   |
| \a                | Alarnton                            |
| \b                | die Taste                           |



## Programmierung in Perl

### Ersatzdarstellungen in durch doppelte Anführungszeichen geklammerten Zeichenketten (Forts.)

| Ersatzdarstellung | Bedeutung                                                                                |
|-------------------|------------------------------------------------------------------------------------------|
| \cZeichen         | die Tastenkombination [Strg] + Zeichen, z.B. \cd für die Tasten-kombination [Strg] + [D] |
| \e                | die Taste [Esc]                                                                          |
| \f                | Seitenvorschub                                                                           |
| \l                | setzt das folgende Zeichen in Kleinschreibung                                            |
| \n                | neue Zeile                                                                               |
| \r                | die Taste [Enter]                                                                        |
| \t                | ein Tabulatorschritt                                                                     |
| \u                | setzt die folgende Zeile in Kleinbuchstaben                                              |
| \v                | bewirkt einen Zeilenvorschub                                                             |
| \E                | schaltet \L und \U wieder aus                                                            |
| \L                | setzt alle Zeichen bis \E in Kleinbuchstaben                                             |
| \U                | setzt alle Zeichen bis \E in Großbuchstaben                                              |
| \XWert            | hexadezimale Ausgabe von Wert                                                            |
| \oWert            | oktale Ausgabe von Wert                                                                  |

Tabelle 17.5

## Typkonvertierung

Wenn mit einem Datentyp Operationen durchgeführt werden sollen, die eigentlich einem anderem Datentypen vorbehalten sind, dann versucht Perl diesen entsprechend umzuwandeln (zu konvertieren). Soll beispielsweise mit den Zeichenketten „3.25 DM“ und „7.75 DM“ eine Addition vorgenommen werden, so wandelt Perl diese in die numerischen Werte „3.25“ und „7.75“ um und liefert das korrekte Ergebnis „11“, wie das folgende Beispiel *konvert* zeigt:

```
#!/usr/bin/perl
Perl-Skript "konvert" zur Demonstration der korrekten
Typenkonvertierung
$a="3.25 DM"; # 1. Zeichenkettenvariable
$b="7.75 DM"; # 2. Zeichenkettenvariable
$c=$a+$b; # Addition der beiden
 # Zeichenkettenvariablen
```



## Variablen



```
Folgende Zeile gibt das korrekte Ergebnis der Addition
aus:
printf("Das Ergebnis von $a + $b = $c\n");
```

In den Fällen, in denen Perl diese Konvertierung nicht durchführen kann, wird der Wert 0 eingesetzt.

Die Konvertierung eines nummerischen Wertes in eine Zeichenkette hingegen ist immer erfolgreich.

## Operatoren

Um Variablen zu manipulieren, steht eine Reihe von so genannten „Operatoren“ zur Verfügung, wobei bei diesen zwischen Operatoren für Zeichenkettenoperationen und Operatoren für nummerische Operationen unterschieden wird.

Eine vollständige Aufzählung sämtlicher in Perl zur Verfügung stehender Operatoren würde den Rahmen dieser Einführung sprengen, darum seien hier nur die wichtigsten genannt.

### Zahlenoperatoren

Perl kennt arithmetische Operatoren für die Addition (wie im vorherigen Beispiel bereits gezeigt), Subtraktion, Multiplikation, Division, Modulo (ganzzahlige Division mit Ausgabe des Teilungsrests) und für die exponentielle Darstellung. Darüber hinaus gibt es die logischen Operatoren UND, ODER und NICHT, die einen Wert ungleich 0 für WAHR und den Wert 0 für FAALSCH zurückgeben, und Vergleichsoperatoren, die den Wert 0 für FAALSCH und den Wert 1 für WAHR zurückgeben.

### Logische Operatoren



| Operator | Beschreibung |
|----------|--------------|
| &&       | UND          |
|          | ODER         |
| !        | NICHT        |

Tabelle 17.6



### Vergleichsoperatoren



| Operator | Beschreibung   |
|----------|----------------|
| <        | kleiner        |
| <=       | kleiner-gleich |



## Programmierung in Perl



### Vergleichsoperatoren (Forts.)

| Operator | Beschreibung                                     |
|----------|--------------------------------------------------|
| >        | größer                                           |
| >=       | größer-gleich                                    |
| ==       | gleich                                           |
| !=       | ungleich                                         |
| <=>      | compare-Operator (liefert den Wert -1, 0 oder 1) |

Tabelle 17.7



### Arithmetische Operatoren

| Operator | Beschreibung   |
|----------|----------------|
| +        | Addition       |
| -        | Subtraktion    |
| *        | Multiplikation |
| /        | Division       |
| %        | Modulo         |
| **       | Exponent       |

Tabelle 17.8



Darüber hinaus kennt Perl Operatoren für die In- und Dekrementierung (das „automatische“ Hochzählen oder Herunterzählen) von Variablenwerten.

### Inkrement und Dekrement

| Operator | Beschreibung                                               |
|----------|------------------------------------------------------------|
| ++       | inkrementiert (zählt hoch) eine Variable um den Wert 1     |
| -        | decrementiert (zählt herunter) eine Variable um den Wert 1 |

Tabelle 17.9



## Variablen



### Zeichenkettenoperatoren

Für die Zeichenkettenmanipulation bietet Perl Operatoren zur Verknüpfung, zur Wiederholung, zum Vergleich und zur Suche nach Zeichenmustern an. Dies macht Perl für CGI-Programmierung so interessant.

#### Verknüpfung und Wiederholung von Zeichenketten

| Operator      | Beschreibung                                                                              |
|---------------|-------------------------------------------------------------------------------------------|
| $Zk \sim RA$  | Die Zeichenkette $Zk$ wird auf Vorhandensein des regulären Ausdrucks $RA$ überprüft       |
| $Zk !\sim RA$ | Die Zeichenkette $Zk$ wird auf Nicht-Vorhandensein des regulären Ausdrucks $RA$ überprüft |

Tabelle 17.10

#### Zeichenketten-Vergleichsoperatoren

| Operator      | Beschreibung                                                                                        |
|---------------|-----------------------------------------------------------------------------------------------------|
| $Zk1 cmp Zk2$ | überprüft, ob Zeichenkette $Zk1$ einen von Zeichenkette $Zk2$ verschiedenen Inhalt hat              |
| $Zk1 eq Zk2$  | überprüft, ob Zeichenkette $Zk1$ den gleichen Inhalt ( <i>equal</i> ) hat wie Zeichenkette $Zk2$    |
| $Zk1 ge Zk2$  | überprüft, ob Zeichenkette $Zk1$ größer oder gleich ( <i>greater equal</i> ) Zeichenkette $Zk2$ ist |
| $Zk1 gt Zk2$  | überprüft, ob Zeichenkette $Zk1$ größer als ( <i>greater than</i> ) Zeichenkette $Zk2$ ist          |
| $Zk1 le Zk2$  | überprüft, ob Zeichenkette $Zk1$ kleiner-gleich ( <i>less equal</i> ) Zeichenkette $Zk2$ ist        |
| $Zk1 lt Zk2$  | überprüft, ob Zeichenkette $Zk1$ kleiner als ( <i>less than</i> ) Zeichenkette $Zk2$ ist            |
| $Zk1 ne Zk2$  | überprüft, ob Zeichenkette $Zk1$ ungleich ( <i>not equal</i> ) Zeichenkette $Zk2$ ist               |

Tabelle 17.11



## Programmierung in Perl

### Musteroperatoren

| Operator  | Beschreibung                                                           |
|-----------|------------------------------------------------------------------------|
| .         | verknüpft zwei Zeichenketten miteinander, z.B. „Hallo“ . „Welt“        |
| Zk x Wert | zeigt die Zeichenkette Zk mit der durch Wert definierten Häufigkeit an |

Tabelle 17.12

Dazu gibt es noch Operatoren zur

- Bitmanipulation
- Dateimanipulation

und so weiter; der Sprachumfang von Perl ist, wie bereits mehrfach erwähnt, gewaltig.

## Arrays

Unter Arrays versteht man eine Datenstruktur, mit der gleichartige Daten unter einem Namen angesprochen werden können.

Wie schon zuvor erwähnt, beginnt der Name einer Array-Variablen mit dem vorangestellten Zeichen @.

Dieser werden in runden Klammern durch Kommata getrennte Array-Konstanten zugewiesen, wobei diese durchaus unterschiedlichen Datentypen zugehören können.

Auch ist die Verwendung von Variablen innerhalb der Array-Definition möglich.

Bilden Sie zunächst einen Array mit fünf Namen:

```
@i = ("Jan", "Hein", "Klaas", "Pitt", "Horst");
```

Auf die einzelnen Elemente des Arrays kann zugegriffen werden, indem der Name des Arrays mit dem Zeichen \$ als Variable referenziert und der Indexwert des gewünschten Elements angegeben wird.

Auch hierbei ist zu beachten, dass das erste Element den Index 0 besitzt.

Hinweis



690

## Kontrollstrukturen



Die Programmzeile

```
printf("$i[2]");
```

würde also den Namen „Klaas“ ausgeben.

Dies soll noch etwas deutlicher an dem (leicht) ergänzten Refrain eines bekannten Liedes demonstriert werden:

Folgendes Programm

```
#!/usr/bin/perl
@i = ("Jan", "Hein", "Klaas", "Pitt", "Horst");
@j = ("Baerte", "Triefnase");

printf("$i[0] und $i[1] und $i[2] und $i[3]\n");
printf("Die haben $j[0], die haben $j[0]\n");
printf("$i[0] und $i[1] und $i[2] und $i[3]\n");
printf("Die haben $j[0], die fahren mit\n");
printf("Nur $i[4] nicht, der hat naemlich "ne $j[1]\n");
```

würde bei seinem Aufruf folgende Bildschirmausgabe bewirken:

```
Jan und Hein und Klaas und Pitt
Die haben Baerte, die haben Baerte
Jan und Hein und Klaas und Pitt
Die haben Baerte, die fahren mit
Nur Horst nicht, der hat naemlich "ne Triefnase
```

Möchten Sie auf mehrere Elemente gleichzeitig zugreifen, geschieht dies in der Form:

```
@Array[Element1, Element2, ...]
```

Soll auf einen Bereich innerhalb eines Arrays zugegriffen werden, kann dies mit der Syntax

```
@Array= ("Element1" ... "Element_N")
```

geschehen.

## Kontrollstrukturen

Wie bereits im *Kapitel 15 Programmierung* beschrieben, dienen Kontrollstrukturen der Steuerung eines Programms, um einen nicht ausschließlich sequentiellen Programmablauf zu ermöglichen.



## Programmierung in Perl

---



Sollten Sie das Kapitel über die bash-Programmierung bereits gelesen haben, werden Sie feststellen, dass die von Perl zur Verfügung gestellten Kontrollstrukturen denen der Skriptprogrammierung gar nicht einmal unähnlich sind, doch werden Sie hier mit einigen Erweiterungen konfrontiert werden, die Sie so wohl in keiner anderen Programmiersprache vorfinden.

### Verzweigungen

#### Hinweis



Man nennt Verzweigungen auch „Entscheidungsanweisungen“.

Verzweigungen werden dann verwendet, wenn innerhalb eines Programms zwischen zwei oder mehr Möglichkeiten des Programmverlaufs gewählt werden soll. Sie legen somit fest, ob ein bestimmter Programmteil entweder abgearbeitet oder übersprungen wird. Die Entscheidung hierfür basiert auf der Auswertung einer Bedingung: Ist die Bedingung WAHR, wird der erste Teil der Verzweigung abgearbeitet, ist sie FALSCH, erfolgt eine Abarbeitung des SONST (*else*) -Zweigs. Die Auswertung geschieht durch den Befehl *if*.

#### *if*

Die Syntax für die einfache Verzweigung lautet:

```
if (Bedingung)
{
 Ja-Anweisung1;
 Ja-Anweisung2;
 ...
}
else
{
 Nein-Anweisung1;
 Nein-Anweisung2;
 ...
}
```

#### Hinweis



Die geschweiften Klammern zur Kennzeichnung der einzelnen Zweige müssen immer gesetzt werden, auch wenn es sich hierbei nicht um Anweisungsblöcke, sondern um Einzelanweisungen handelt!

Formulieren Sie zur Verdeutlichung dieser Kontrollstruktur ein Beispiel, wobei Sie sich ebenfalls der zuvor vorgestellten Vergleichsoperatoren bedienen. Es sollen zwei Zeichen überprüft und festgestellt werden, welches von beiden „größer“ ist, d.h. den höheren dezimalen ASCII-Wert besitzt:

```
#!/usr/bin/perl
printf("Bitte geben Sie ein Zeichen ein: ");
$a=<STDIN>;
printf("Und jetzt noch ein Zeichen : ");
$b=<STDIN>;
```





```
if($a gt $b)
{
 printf("Das Zeichen\n--->$a\nist \"groesser\" als das
Zeichen\n--->$b");
}
else
{
 printf("Das Zeichen\n--->$b\nist \"groesser\" als das
Zeichen\n--->$a");
}
```

Beachten Sie, dass im Nein-Fall (Wert der Variablen *\$a* ist *nicht* größer als der Wert von *\$b*) die Aussage konsequenterweise *groesser oder gleich* lauten müsste. Sie haben hier also ein wenig vereinfacht und werden deshalb weiter unten eine Variante vorstellen, mit der die drei Möglichkeiten *größer, kleiner und gleich* behandelt werden.

Eine Perl-spezifische Variante der Verzweigung ist die Konstruktion mit *unless*.

### Varianten zu *if*

Im Folgenden werden einige Variationen der *if*-Bedingung vorgestellt.

```
unless(Bedingung)
{
 Anweisung1;
 Anweisung2;
}
else
{
 Anweisung3;
 Anweisung4;
 ...
}
```

Diese Konstruktion stellt eine Vereinfachung der Überprüfung von *Bedingung* auf den Ausdruckswert FALSCH hin dar, d.h., sie „dreht“ die gestellte Bedingung um.

### Verschachtelung mit *elsif*

Eine Verschachtelung mehrerer Verzweigungen geschieht mit dem Schlüsselwort *elsif*.

Syntax

```
if(Bedingung1)
{
 Anweisung1;
 Anweisung2;
```



## Programmierung in Perl

---



```
 ...
}
elsif(Bedingung2)
{
 Anweisung_a;
 Anweisung_b;
 ...
}
else
{
 Anweisung_x;
 Anweisung_y;
 ...
}
```

Ergänzen Sie das Beispielprogramm von eben um dieses Konstrukt:

Zunächst soll überprüft werden, ob die beiden eingegebenen Zeichen gleich sind. Ist dies nicht der Fall, soll überprüft werden, ob das erste Zeichen größer ist, und eine entsprechende Bildschirmmeldung ausgegeben werden. Ansonsten soll eine Bildschirmmeldung erfolgen, dass das zweite eingegebene Zeichen größer ist:

```
#!/usr/bin/perl
printf("Bitte geben Sie ein Zeichen ein: ");
$a=<STDIN>;
printf("Und jetzt noch ein Zeichen : ");
$b=<STDIN>

if($a eq $b)
{
 printf("Die beiden eingegebenen Zeichen sind
gleich\n");
}
elsif($a gt $b)
{
 printf("Das Zeichen\n--->$a\nist \"groesser\" als das
Zeichen\n--->$b");
}
else
{
 printf("Das Zeichen\n--->$b\nist \"groesser\" als das
Zeichen\n--->$a");
}
```

Perl wartet auch bei Verzweigungen wieder mit einigen Überraschungen auf. Es gibt für die zuvor beschriebenen Syntaxbeschreibungen Kurzformen, die so wohl in keiner anderen Programmiersprache zu finden sind.





### Varianten

#### Syntaxvariante 1

Anweisung if Bedingung

Für das Beispiel von oben heißt das, Sie könnten auch formulieren:

```
#!/usr/bin/perl
printf("Bitte geben Sie ein Zeichen ein: ");
$a=<STDIN>;
printf("Und jetzt noch ein Zeichen : ");
$b=<STDIN>

printf("Das Zeichen\n--->$a\nist \"groesser\" als das
Zeichen\n--->$b")
if($a gt $b);
printf("Das Zeichen\n--->$b\nist \"groesser\" als das
Zeichen\n--->$a")
if($b gt $a);
```

Noch kürzer sind folgende Syntaxvarianten:

#### Syntaxvariante 2

Anweisung && Bedingung

Anweisung || Bedingung

Auch hier wieder die entsprechende Kodierung des Beispielprogramms in dieser Syntaxform:

```
#!/usr/bin/perl
printf("Bitte geben Sie ein Zeichen ein: ");
$a=<STDIN>;
printf("Und jetzt noch ein Zeichen : ");
$b=<STDIN>

printf("Das Zeichen\n--->$a\nist \"groesser\" als das
Zeichen\n--->$b")
&& ($a gt $b);
printf("Das Zeichen\n--->$b\nist \"groesser\" als das
Zeichen\n--->$a")
&& ($b gt $a);
```

### Schleifen

Wie in der Shell-Programmierung stehen auch in Perl sowohl eine *while*- als auch eine *until*-Schleife zur Verfügung, wobei letztere hier ebenfalls wieder nur die Laufbedingung anders auswertet, nämlich statt



## Programmierung in Perl

---



```
while (WAHRE Laufbedingung)
die Auswertung
until (FALSCHE Laufbedingung)
vornimmt.
```

### **while**

Allgemein formuliert lautet die Syntax der *while*-Schleife:

```
while (Ausdruck)
{
 Anweisung1;
 Anweisung2;
 ...
}
```

bzw. die Syntax der *until*-Schleife:

### **until**

```
until (Ausdruck)
{
 Anweisung1;
 Anweisung2;
 ...
}
```

Folgende zwei Beispiele zeigen jeweils den Text „Hello world“ zehnmal an:

```
#!/usr/bin/perl
$a=0;
while($a < 10)
{
 $a=$a+1;
 printf("Hello world\n");
}
```

Die Lösung derselben Aufgabe mit *until*:

```
#!/usr/bin/perl
$a=0;
until($a > 9)
{
 $a=$a+1;
 printf("Hello world\n");
}
```



## Kontrollstrukturen



Bei den vorherigen zwei Beispielen handelt es sich um Zählschleifen; diese würde man jedoch (eleganter?) mit der *for*-Schleife lösen:

### **for**

Bei der *for*-Schleife wird der Startwert gemeinsam mit der Laufbedingung (dies ist die Bedingung, die festlegt, wie lange die Schleife durchlaufen werden soll) und der Schrittweite im Schleifenkopf initialisiert.

Die Syntax der *for*-Schleife lautet:

```
for(Startwert; Laufbedingung; Schrittweite)
{
 Anweisung1;
 Anweisung2;
 ...
}
```

Sie könnten das Beispielprogramm von vorhin, das zehnmal den Gruß „Hello world“ auf dem Bildschirm ausgibt, also auch wie folgt formulieren:

```
#!/usr/bin/perl
for($a = 0; $a < 10; $a++)
{
 printf("Hello world\n");
}
```

Um den Begriff der Schrittweite noch ein wenig zu verdeutlichen, betrachten Sie folgende Programmmodifikation:

```
for($a = 0; $a < 10; $a = $a + 2)
```

Die „Schrittweite“  $\$a = \$a + 2$  legt fest, dass  $a$  bei jedem Schleifendurchlauf nicht um den Wert 1, sondern um 2 hochgezählt wird – die Schleife wird also in diesem Fall nur fünfmal durchlaufen.

Eine weitere Schleifen-Variante, die in vielen anderen Programmiersprachen nicht üblich ist, ist die *foreach*-Schleife.

### **foreach**

Die *foreach*-Schleife dient zur Abarbeitung von Listen, wobei die Schleife jeweils einmal pro Listenelement durchlaufen und der jeweils aktuelle Listenwert einer Variablen zugewiesen wird.

Die Syntax für diesen Schleifentyp lautet:

```
foreach $var (@liste)
{
 Anweisung1;
```

## Programmierung in Perl

---



```
Anweisung2;
...
}
```

Um dies an einem Beispiel zu demonstrieren, initialisieren Sie einen Array @a mit den Zeichen a bis k und lassen diese anschließend ausgeben:

```
#!/usr/bin/perl
@a = ("a" .. "k");
foreach $a (@a)
{
 printf("$a\n");
}
```

Die Variabilität von Perl wurde angesprochen und soll nun auch einmal an diesem Beispiel demonstriert werden. Bezüglich der Initialisierung des Arrays hätten Sie das Programm auch wie folgt formulieren können:

```
#!/usr/bin/perl
foreach $a (@a = ("a" .. "k"))
{
 printf("$a\n");
}
```

In diesem Beispiel geschieht die Initialisierung des Arrays also im Schleifenkopf und nicht zu Beginn des Programms. Dies ist eine interessante Syntaxvariante, die allerdings, wie Sie vielleicht mit uns übereinstimmen werden, nicht unbedingt zur Lesbarkeit des Programms beiträgt ...

## Sprunganweisungen

Sprunganweisungen dienen dazu, Schleifen in Abhängigkeit von bestimmten Bedingungen frühzeitig zu verlassen, einen erneuten Schleifendurchlauf zu erzwingen, auch wenn eine dieser widersprüchliche Bedingung eintritt, etc.

So bietet Perl beispielsweise einen Sprungbefehl, der BASIC-Programmierern mehr als vertraut ist, bei dem sich jedoch jemandem, der eine strukturierte Programmierung gewohnt ist, die Nackenhaare sträuben...

Es handelt sich hierbei um den Befehl *goto*.

### **goto**

Der Befehl *goto* springt innerhalb eines Programms in Abhängigkeit einer bestimmten Bedingung zu einer so genannten „Sprungmarke“, wodurch der sequentielle Ablauf des Programms unterbrochen wird.

Die Syntax für diesen Befehl lautet

```
goto Sprungmarke
```



## Kontrollstrukturen



wobei diese innerhalb des Codes durch einen nachgestellten Doppelpunkt als solche gekennzeichnet werden muss:

**Sprungmarke:**

Folgendes Beispielprogramm, welches ebenfalls wieder zehnmal den Text „Hello world“ anzeigt, soll dieses verdeutlichen:

```
#!/usr/bin/perl
$a = 0;

Marke1:
if($a < 10)
{
 printf("Hello world\n");
 $a = $a + 1;
 goto Marke1;
}
else
{
 goto Marke2;
}
Marke2:
```

In diesem Programm wird zunächst die Variable *a* mit dem Wert *0* initialisiert. Nun wird überprüft, ob sie kleiner als *10* ist. Ist dies der Fall, wird der Text angezeigt und anschließend die Variable um den Wert *1* hochgezählt; danach wird zur *Marke1* gesprungen. Dieser Vorgang wiederholt sich so lange, bis der Wert *10* erreicht wird; in diesem Fall wird zur *Marke2* gesprungen.

Da Sie bereits Schleifen kennen gelernt haben, wird Ihnen vielleicht einleuchten, warum diese Art von Programmen auch „Spaghetti-Code“ genannt wird ...

Sollten Sie bereits das *Kapitel 16 Shell-Programmierung* gelesen haben, kennen Sie den Shell-Befehl *continue*. Seine Entsprechung in Perl heißt *next*.

### **next**

Der Befehl *next* erzwingt, dass der aktuelle Schleifendurchlauf unterbrochen und zum Anfang der Schleife zurückgekehrt wird.

Die Syntax dieses Befehls lautet einfach:

```
next
```

Folgendes Beispielprogramm berechnet die Ergebnisse der Divisionen von *1* durch *5* bis *1* durch *-5*. Da eine Division durch *0* verboten ist, muss ein erneuter Schleifendurchlauf mit *next* erzwungen werden, da sonst das Programm mit einer Fehlermeldung abbrechen würde:



## Programmierung in Perl

---



```
#!/usr/bin/perl
for($b = 5;$b >= -5; $b--)
{
 if($b == 0)
 {
 printf("1/$b -- Diese Rechenoperation ist nicht");
 printf("zulaessig ...\\n");
 next;
 }
 else
 {
 $a = 1/$b;
 printf("1/$b = $a\\n");
 }
}
Dieses Programm erzeugt folgende Bildschirmausgabe:
```

1/5 = 0.2  
1/4 = 0.25  
1/3 = 0.33333333333333  
1/2 = 0.5  
1/1 = 1  
1/0 -- Diese Rechenoperation ist nicht zulaessig ...  
1/-1 = -1  
1/-2 = -0.5  
1/-3 = -0.33333333333333  
1/-4 = -0.25  
1/-5 = -0.2

Analog zum Befehl *next* gibt es den Befehl *last*.

### **last**

Der Befehl *last* erzwingt den Abbruch eines Schleifendurchlaufs.

Die Syntax lautet auch hier einfach

```
last
```

Verwenden Sie das gleiche Beispielprogramm wie eben und ersetzen Sie den Befehl *next* durch den Befehl *last*, wie im folgenden Programmfragment beschrieben:

```
...
if($b == 0)
{
 printf("1/$b -- Diese Rechenoperation ist nicht");
 printf(" (zulaessig ...\\n");
 last;# In dieser Zeile erfolgt die Programmaenderung
}
...
```



## Funktionen



Der Aufruf dieses Programms würde bei der Fehlermeldung abbrechen, ohne die negativen Divisionen durchzuführen:

```
1/5 = 0.2
1/4 = 0.25
1/3 = 0.3333333333333333
1/2 = 0.5
1/1 = 1
1/0 -- Diese Rechenoperation ist nicht zulaessig ...
```

## Funktionen

Die Programmiersprache Perl besitzt eine schier unendlich große Zahl an Funktionen.

Zu diesen gehören:

- arithmetische Funktionen
- Konvertierungsfunktionen
- Zeichenkettenfunktionen
- Array- und Listenfunktionen
- Funktionen zur Dateimanipulation
- Such- und Ersetzungsfunktionen

und so weiter und so weiter. Diese können beliebig durch die Definition eigener Funktionen ergänzt werden.

Exemplarisch für die Vielzahl der standardmäßig vorhandenen Funktionen werden Ihnen hier

- *sqrt()*
- *int()*
- *chop()*
- *chomp()*

vorgestellt.



### Die Berechnung der Quadratwurzel mit *sqrt()* und Ausgabe des ganzzahligen Anteils mit *int()*

Entwickeln Sie ein Programm, das die Eingabe einer Zahl erwartet, die im Bereich zwischen 1 und 999 liegt. Erfolgt eine Fehleingabe, wird der Benutzer des Programms entsprechend darauf hingewiesen und zu einer erneuten Eingabe aufgefordert. Liegt die Eingabe im geforderten Wertebereich, wird zunächst die Quadratwurzel aus dieser gezogen und dann der ganzzahlige Anteil dieses Rechnergebnisses ausgegeben:

```
#!/usr/bin/perl
$i=0;
until($i > 0)
{
 printf("Bitte geben Sie eine Zahl zwischen ");
 printf("1 und 999 ein: ");
 $i=<STDIN>;
 if($i <= 0 || $i > 999)
 {
 printf("Bitte nur eine Zahl zwischen ");
 printf("1 und 999 eingeben!\n");
 }
 else
 {
 $j=sqrt($i);
 printf("Die Wurzel von $i ist $j\n");
 $i=int($j);

 printf("Der ganzzahlige Anteil von $j lautet: $i\n");
 }
}
```

Beim Aufruf dieses Programms ergibt sich bei einer ersten Fehleingabe und der anschließenden Eingabe der Zahl 2 folgende Bildschirmausgabe:

```
Bitte geben Sie eine Zahl zwischen 1 und 999 ein: -50
Bitte nur eine Zahl zwischen 1 und 999 eingeben!
Bitte geben Sie eine Zahl zwischen 1 und 999 ein: 2
Die Wurzel von 2 ist 1.4142135623731
Der ganzzahlige Anteil von 1.4142135623731 lautet: 1
```

### Die Funktionen *chop()* und *chomp()*

Betrachten Sie folgendes Beispielprogramm:

```
#!/usr/bin/perl
print "Bitte machen Sie eine Eingabe: ";
$eingabe1=<STDIN>;
```



## Funktionen



```
print "Bitte machen Sie eine zweite Eingabe: ";
$eingabe2=<STDIN>;
print "Das Ergebnis Ihrer Eingaben lautet: \n";
print "$eingabe1#, $eingabe2#!";
```

Wenn Sie dieses Programm aufrufen und als *eingabe1* „Hallo“ und als *eingabe2* „Welt“ tätigen, führt dies zu folgendem Ergebnis:

```
Bitte machen Sie eine Eingabe: Hallo
Bitte machen Sie eine zweite Eingabe: Welt
Das Ergebnis Ihrer Eingaben lautet:
Hallo
#, Welt
#!
```

Bewusst haben wir hinter den Variablennamen in der Ausgabezeile jeweils ein Doppelkreuz (#) gesetzt, um zu verdeutlichen, das beim Einlesen der Standardeingabe und deren Bestätigung mit Return eben dieses Return *mit* in der Variablen abgespeichert wird, was in der Bildschirmausgabe zu Zeilenumbrüchen führt (die Sie mithilfe dieses Zeichens einfacher erkennen können...).

Diese Returns sollen nun ausgefiltert werden.

### ***chop()***

Die Funktion *chop()* entfernt grundsätzlich das letzte Zeichen einer Eingabe, gleich um welches Zeichen es sich handelt. In vorliegendem Beispiel könnten Sie diese Funktion also verwenden, das in der Variablen abgespeicherten Return-Zeichen auszufiltern.

Dies geschieht mithilfe der Befehlszeile

```
chop($eingabe1, eingabe2);
```

nach dem Einlesen der beiden Variablen:

```
#!/usr/bin/perl
print "Bitte machen Sie eine Eingabe: ";
$eingabe1=<STDIN>;
print "Bitte machen Sie eine zweite Eingabe: ";
$eingabe2=<STDIN>;
chop($eingabe1, eingabe2);
print "Das Ergebnis Ihrer Eingaben lautet: \n";
print "$eingabe1#, $eingabe2#!";
```

Das Programm würde jetzt bei gleicher Eingabe wie zuvor folgendes Ergebnis liefern (was in der Regel wohl auch gewünscht ist):



## Programmierung in Perl

---



```
Bitte machen Sie eine Eingabe: Hallo
Bitte machen Sie eine zweite Eingabe: Welt
Das Ergebnis Ihrer Eingaben lautet:
Hallo#, Welt#!
```

### ***chomp()***

Die Funktion *chomp()* könnte ebenfalls für diese Aufgabe verwendet werden (wie bereits mehrfach betont, führen bei Perl meist mehrere Wege zum gleichen Ziel), da sie ebenfalls das letzte Zeichen des Inhalts einer Variablen entfernt (in diesem Fall also wieder das Return-Zeichen).

Als Besonderheit dieser Funktion soll an dieser Stelle jedoch erwähnt werden, dass man ihr auch einen Variableninhalt übergeben kann, der vom Ende der später zu bearbeitenden Variablen abgeschnitten wird.

Die Zuweisung dieser Variablen geschieht durch Eingabe von:

```
$ /=<WERT>;
```

Diese Funktionalität, die man sicherlich kaum in interaktiven Skripten wie dem dargestellten verwenden wird, soll durch folgendes Beispiel illustriert werden:

```
#!/usr/bin/perl
$ /= "-Team Kiel";
$var="Linux-Team Kiel";
chomp($var);
print "Inhalt der Variablen: $var\n";
```

Die Bildschirmausgabe dieses Skripts ergibt folgendes Ergebnis:

```
Inhalt der Variablen: Linux#
```

Wie also deutlich zu erkennen ist, wird der Inhalt der Variablen *\$var* durch *chomp()* dahingehend modifiziert, dass er durch den Inhalt von *\$/* beschnitten wird.

Wie einleitend in diesem Kapitel versprochen, soll dieses Kapitel mit einem etwas gehaltvollerem Beispiel abgeschlossen werden, um Ihnen den Gebrauch von Perl noch etwas schmackhafter zu machen.

## **Perl als Hilfswerkzeug für die Manipulation von Datenbankinhalten**

Perl stellt mithilfe des Moduls Perl-DBI ein hervorragendes Werkzeug zur Manipulation von Datenbank(inhalt)en dar, weshalb es auch sehr häufig innerhalb des World Wide Web Verwendung findet (ein konkretes Beispiel für die Verwendung von Perl in interaktiven Webseiten werden Sie im folgenden *Kapitel 18 HTML und CGI* finden).



## Perl für die Manipulation von Datenbankinhalten



An dieser Stelle werden einige einfache Beispiele für einen solchen Umgang mit Datenbanken gegeben, wobei (da es sich bei diesem Buch nicht um eine Einführung in Datenbanken handelt) einige Dinge einfach als gegeben vorausgesetzt werden. Wir bitten um Ihr Verständnis, wenn wir diese Grundlagen nur am Rande streifen.

### Das Datenbanksystem PostgreSQL

Das zugrunde liegende Datenbanksystem, welches hier verwendet werden soll, heißt PostgreSQL, die Verwendung anderer (relationaler) Datenbanksysteme (relational database management systems – RDBMS) funktioniert jedoch ähnlich.

PostgreSQL wird detailliert in unserem Buch „Linux im Office – Das Buch“, ebenfalls erschienen beim Sybex-Verlag unter der ISBN-Nummer 3-8155-0310-8 beschrieben.



Hinweis

Die hier verwendete Software ist auf der zweiten CD des Buches im Verzeichnis *postgres* beigelegt, in dem Sie auch Installationshinweise zu den mitgelieferten Paketen finden.

Bei einem relationalem Datenbanksystem wie PostgreSQL (genau genommen handelt es sich hierbei sogar um ein so genanntes „objektrelationales Datenbanksystem“) werden die Daten in so genannten „Tabellen“ (engl. *tables*) gehalten, die miteinander in Beziehung stehen („verknüpft sind“ – aufgrund dieser Beziehung spricht man auch von relationalen Datenbanksystemen).

Die relationale Verknüpfung dient dazu, gleichzeitig auf mehrere Tabellen zugreifen zu können, das Halten „überflüssiger“ (redundanter) doppelter Informationen zu vermeiden, die Konsistenz der Daten zu gewährleisten usw.

Zur Manipulation solcher Datenbanken verwendet man im allgemeinen eine Abfragesprache, die als SQL (*structured query language* – strukturierte Abfragesprache) bezeichnet wird, deren Verwendung jedoch einem einfachen Anwender aus Gründen, die Sie vielleicht bei näherer Darstellung des Beispiels verstehen werden, in der Regel nicht zugemutet werden kann, so dass man sich bemüht, diesem Werkzeuge in die Hand zu geben, die es ihm oder ihr ermöglichen, auch ohne diese Kenntnisse auf die Datenbankinformationen zugreifen und diese gegebenenfalls verändern, ergänzen oder löschen zu können. Solche Werkzeuge sollen an dieser Stelle mit Perl erstellt werden.

Doch zunächst kurz zu der Datenbank – es handelt sich bei ihr um ein „gastronomisches Adressverzeichnis“, sprich: um ein Adressbuch für Restaurants und Kneipen.

### Die Beispieldatenbank

Als Datenbankverwalter *postgres* haben Sie mit

```
createdb gastro
```

eine entsprechende Datenbank angelegt, die drei Tabellen enthalten soll:

## Programmierung in Perl

### Hinweis

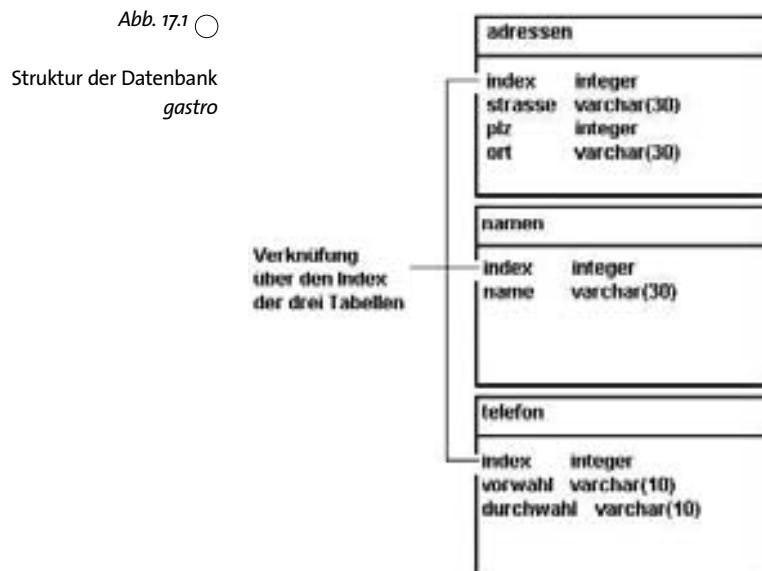


Es wird keinerlei Anspruch darauf erhoben, dass die Aufspaltung der Daten in drei Tabellen an dieser Stelle überhaupt Sinn macht, allerdings soll an dieser Stelle auch keine langatmige Diskussion über Stamm-, Bewegungs- und Änderungsdaten losgetreten werden – lassen Sie das Beispiel einfach Beispiel sein, welches Relationen aufzeigen soll ...

- Eine Tabelle mit den Namen der zu erfassenden gastronomischen Einrichtungen; diese sollen bis zu 30 Zeichen lang sein dürfen. In Postgres verwendet man hierzu den Datentyp *varchar*.
- Eine Tabelle mit Vorwahl- und Rufnummern, die jeweils bis zu zehn Zeichen lang sein und auch Trennzeichen wie das Leerzeichen oder den Bindestrich enthalten dürfen; aus diesem Grunde wird hier ebenfalls den Datentyp *varchar* verwendet.
- Eine Tabelle mit Straßen- und Ortsnamen mit bis zu 30 Zeichen Länge (Datentyp: *varchar*) und einem Tabellenfeld für die Postleitzahl, welches ausschließlich eine Zahl enthalten soll. Da es sich bei diesen ausschließlich um Ganzzahlen handeln kann, wird der Datentyp *integer* verwendet.

Alle drei Tabellen sollen über so genannte „Indexfelder“ miteinander verknüpft sein, so dass die Relationen eindeutig sind (nicht, dass Sie eigentlich bei Ihrem Lieblings-Italiener einen Tisch reservieren wollen und aufgrund falscher Verknüpfungen die Rufnummer Ihres Lieblings-Griechen erhalten ...); hierbei soll es sich um eindeutige (*unique*) Ganzzahlen handeln (*integer*).

Grafisch stellt sich die Datenbankstruktur also wie folgt dar:



## Perl für die Manipulation von Datenbankinhalten



### Anlegen der Tabellen

PostgreSQL bietet einen eigenen Interpreter (*psql*), mit dessen Hilfe die drei Tabellen zunächst angelegt werden sollen.



### Hilfeinformationen zu PostgreSQL

Nach Aufruf des psql-Interpreters stellt Ihnen Postgres ein umfangreiches Hilfesystem zur Verfügung. Die Eingabe von

\h

gibt einen Überblick über die zur Verfügung stehenden SQL-Ausdrücke. Mit

\h ausdruck

können Sie sich Syntaxbeschreibungen zu dem entsprechenden Ausdruck anzeigen lassen, z.B.

\h select

Diese Befehlszeile gäbe Ihnen folgende Informationen:

```
SELECT [ALL | DISTINCT [ON (expression [, ...])]
]
 expression [AS name] [, ...]
 [INTO [TEMPORARY | TEMP] [TABLE] new_table]
 [FROM table [alias] [, ...]]
 [WHERE condition]
 [GROUP BY column [, ...]]
 [HAVING condition [, ...]]
 [{ UNION [ALL] | INTERSECT | EXCEPT } select]
 [ORDER BY column [ASC | DESC | USING operator]
 [, ...]]
 [FOR UPDATE [OF class_name [, ...]]]
 LIMIT { count | ALL } [{ OFFSET | , } start]
```

Die Eingabe von

\?

gibt einen (allerdings ebenfalls in Englisch gehaltenen) Überblick über alle Postgres-internen Befehle.



## Programmierung in Perl

---



Geben Sie als Benutzer *postgres* folgenden Befehl ein, um mit dem Erstellen der Datenbank zu beginnen:

```
psql gastro
```

Die SQL-Anweisung für das Erstellen der Namenstabelle lautet:

Hinweis



Einrückungen und Umbrüche dienen auch hier nur der Übersichtlichkeit – sämtliche folgende Anweisungen hätten auch als „Einzeiler“ formuliert werden können.

```
create table namen
(
 index int unique,
 name varchar(30)
);
```

Die Adresstabelle soll wie folgt angelegt werden:

```
create table adressen
(
 index int unique,
 strasse varchar(30),
 plz int,
 ort varchar(30)
);
```

Und zum Abschluss noch die Anweisung zum Anlegen der Telefontabelle:

```
create table telefon
(
 index int unique,
 vorwahl varchar(10),
 durchwahl varchar(10)
);
```

### Die Datenbanktabellen mit Einträgen versehen

Hinweis



Nun wollen wir zunächst die Tabelle *namen* mit zwei Einträgen „füttern“:

Beachten Sie die Eindeutigkeit der Indexzahl. Hätten wir für das „Zorbas“ ebenfalls eine „1“ eingegeben, wäre die Anweisung mit einem entsprechenden Fehlerkommentar nicht ausgeführt worden.

```
insert into namen values (1, "La Torre");
insert into namen values(2, "Zorbas");
```

Im weiteren Schritt sollen die beiden bisher gültigen Indexeinträgen in die Tabellen *telefon* und *adressen* eingefügt werden:



## Perl für die Manipulation von Datenbankinhalten

```
insert into telefon (index)
 select index from namen where name="La Torre";
insert into telefon (index)
 select index from namen where name="Zorbas";
insert into adressen (index)
 select index from namen where name="La Torre";

insert into adressen (index)
 select index from namen where name="Zorbas";
```

Im nächsten Schritt wird die Tabelle *telefon* mit den entsprechenden Einträgen für die *vorwahl* und die *durchwahl* für die jeweiligen Restaurants aktualisiert:

```
update telefon set vorwahl="0431", durchwahl="93463"
 where index=
 (select index from namen where name="La Torre");

update telefon set vorwahl="0431", durchwahl="96186"
 where index=
 (select index from namen where name="Zorbas");
```

Um die Tabelle *adressen* noch zu ergänzen, geben Sie ein:

```
update adressen
 set strasse="Sandkuhle 4--6", plz=24103, ort="Kiel"
 where index=
 (select index from namen where name="Zorbas");

update adressen
 set strasse="Schlossstrasse", ort="Kiel"
 where index=
 (select index from namen where name="La Torre");
```

Die Adressangabe für das „La Torre“ ist an dieser Stelle bewusst unvollständig.



Hinweis

### Eine Datenbankabfrage

Wenn wir nun in der Datenbank gespeicherte Informationen abfragen wollen, geschieht dies mithilfe des zuvor bereits verwendeten *SELECT*-Ausdrucks.

Möchten wir beispielsweise den Namen, die Vorwahl und die Durchwahl des „La Torre“ in Erfahrung bringen, müssten wir eingeben:

```
select name, vorwahl, durchwahl
 from namen, telefon
 where namen.index=telefon.index and name="La Torre";
```

Das Ergebnis dieses Abfrage sähe wie folgt aus:

709



## Programmierung in Perl

---



```
name | vorwahl | durchwahl
-----+-----+-----+
La Torre | 0431 | 93463
(1 row)
```

Die Suche nach sämtlichen Informationen zum „Zorbas“ (außer des Index-Eintrags) könnte wie folgt aussehen:

```
select name, vorwahl, durchwahl, plz, ort, strasse
 from namen, telefon, adressen
 where namen.index=telefon.index
 and telefon.index=adressen.index
 and name="Zorbas";
```

Das Abgabeergebnis wäre Folgendes:

```
name | vorwahl | durchwahl | plz | ort | strasse
-----+-----+-----+-----+-----+
Zorbas | 0431 | 96186 | 24103 | Kiel | Sandkuhle 4-6
(1 row)
```

Wie Sie an dieser Stelle bereits erkennen werden, wird eine Aufforderung an einen Sachbearbeiter, die Datenbank um weitere Einträge zu ergänzen bzw. bereits bestehende Einträge zu aktualisieren oder auch nur eine Abfrage vorzunehmen vermutlich in einer fristlosen Kündigung enden...

Aus diesem Grunde sollen ihm einige Perl-basierte Anwendungen zur Verfügung gestellt werden, die Eingabe, Aktualisierungen und Abfragen vereinfachen.

### Voraussetzungen

Voraussetzung für die Nutzung von Perl für Datenbankmanipulationen ist das Vorhandensein zweier Softwarekomponenten:

- des DBI-Moduls (database interface module)
- eines Treibers (des database drivers, kurz DBD)

Das DBI-Modul definiert die Programmierschnittstelle, verteilt Methodenaufrufe an den jeweils richtigen Treiber und unterstützt diesen bei der eigentlichen Kommunikation mit der Datenbank

Es geht sogar so weit, dass DBI selbst noch nicht einmal eine Vorstellung von Datenbanktypen hat.

Hinweis



## Perl für die Manipulation von Datenbankinhalten



Da DBI selbst keine datenbankorientierten Aktionen ausführt, sondern nur die Methoden vorgibt, können relativ einfach neue Treiber für Datenbanken entwickelt werden. Mit Hilfe dieser Treiber ist es dann möglich, bereits bestehende (zum Beispiel in Perl geschriebene) Anwendungen nur durch Austausch des Treibers weiter zu verwenden, auch wenn ein Wechsel des dahinter liegenden Datenbanksystems erfolgt sein sollte.

Das Vorhandensein von Perl für verschiedene Betriebssystemplattformen und das Vorhandensein unterschiedlicher Treiber (*DBD::Oracle* für Oracle-Datenbanken, *DBD::Informix* für Informix-Datenbanken, *DBD::Pg* für PostgreSQL-basierte Datenbanken usw.) machen aus dem Zusammenwirken von Perl und DBI ein hervorragendes Werkzeug, für das schnelle Entwickeln von Applikationen (RAD – Rapid Application Development).

*Abbildung 17.2* gibt einen schematischen Überblick über den durch DBI bereitgestellten Datenfluss

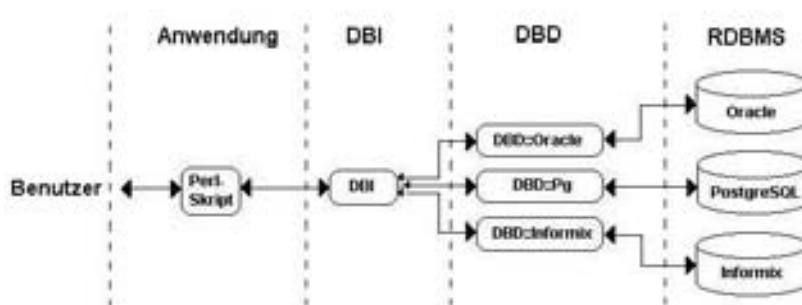


Abb. 17.2

Kommunikation zwischen Benutzer und Datenbank über Perl/DBI

Die Anwendung von DBI wird vorbereitet, indem man die Zeile

```
use DBI;
```

in einem Perl-Skript einfügt.

### Handles

Ohne allzu tief in die Theorie der DBI-Programmierung einsteigen zu wollen, lässt sich jedoch die Einführung einiger Schlüsselbegriffe nicht vermeiden.

Das von DBI verwendete objektorientierte Konzept bedient sich dreier Objekte, die für die Arbeit mit Datenbanken notwendig sind; diese werden als Handles bezeichnet.



## Programmierung in Perl

---



### Treiber-Handle

Bei der Erstellung einfacher Anwendungen benötigen Sie in der Regel keinen expliziten Zugriff auf die Treiber-Handle. Im Rahmen dieser Einführung soll deren Erzeugung und Verwendung durch den schematischen Aufruf von

```
DBI->connect
```

genügen.

### Datenbank-Handle

Der Datenbank-Handle dient vereinfacht formuliert zur Verbindungsherstellung mit der Datenbank. Seine Erzeugung erfolgt für den Postgres-Treiber nach dem Schema:

```
$dbh=DBI->connect
(DBI : Pg : dbname=<DBNAME>, <USERNAME>, <PASSWORD>)
```



### Syntaxunterschiede bei der Definition des Datenbank-Handles unter Verwendung unterschiedlicher DBDs

*Sollten Sie mit einer anderen Datenbank und somit einem anderen DBD arbeiten, beachten Sie bei der Bildung dieses Handles die Unterschiede in der Syntax.*

*Im Fall von Oracle lautet sie beispielsweise*

*DBI:Oracle:<DBNAME>...*

*Es wird also ein Doppelpunkt statt eines Gleichheitszeichens zwischen dem Namen des RDBMS und der Datenbank verwendet!*

Der Verbindungsabbruch erfolgt mit:

```
$dbh->disconnect;
```

### Statement-Handle

Mit der Erzeugung des Statement-Handles definieren Sie die später auszuführenden SQL-Befehle. Dieses geschieht nach dem Muster:



## Perl für die Manipulation von Datenbankinhalten



```
my $sth = $dbh->prepare("<SQL-ANWEISUNGEN>")
```

Die Ausführung erfolgt über die Befehlszeile:

```
$sth->execute;
```

### Beispiele

Mit Hilfe der nunmehr gewonnenen Erkenntnisse sollen einige Hilfsprogramme zur Entlastung des weiter oben erwähnten Sachbearbeiters entwickelt werden, die ihm den Umgang mit der Gastronomie-Datenbank vereinfachen sollen.

Zunächst ein Programm zur Datenbankrecherche; dieses soll folgende Funktionalitäten bieten:

- Aufruf des Programms in der Befehlszeile, wobei dem Programm der Name des zu suchenden gastronomischen Betriebs als Parameter übergeben werden soll; die Eingabe eines Namensfragments soll hierbei möglich sein.
- Ausgabe einer Syntaxbeschreibung, wenn der Programmaufruf ohne Parametrisierung erfolgt mit anschließendem Programmabbruch.
- Verbindungsaufbau mit der Datenbank *gastro*.
- Erzeugung eines Statement-Handles, der nach Namen, Strasse, Postleitzahl, Ort, Vorwahl und Durchwahl derjenigen gastronomischen Betriebe sucht, die dem Suchkriterium entsprechen.
- Ausführen der im Statement-Handle definierten SQL-Anweisung.
- Verbindungsabbau.

```
#!/usr/bin/perl
Ueberpruefung, ob dem Befehl ein Parameter uebergeben
wurde
if("$ARGV[0]" eq "") {
 print "\nSyntax: recherche \"NAME_DES_BETRIEBS\"\n\n";
 exit;
}

use DBI;

Vorbereitung des Datenbank-Handles
my $dbh = DBI->connect("DBI:Pg:dbname=gastro")
 or die "Fehler beim Verbindungsaufbau: $DBI::errstr\n";

Vorbereitung der SQL-Anweisung; beachten Sie die
Konstruktion mit LIKE, um auch Namensfragmente
```

—○  
Beachten Sie bitte  
die im Quellcode  
vorgenommene  
Dokumentierung!



## Programmierung in Perl

---



```
auswerten zu koennen. Ausgewertet wird das erste
Befehlszeilenargument
my $sth = $dbh->prepare("SELECT name, strasse ,plz, ort,
vorwahl, durchwahl
FROM namen, adressen, telefon
WHERE name like '%$ARGV[0]%'
AND telefon.index=namen.index
AND adressen.index=namen.index");

Ausfuehrung der SQL-Anweisung
$sth->execute
 or die "Fehler bei Befehlsausfuehrung: $DBI::errstr\n";

my $row;
Einlesen der Suchergebnisse in der Array row, solange
Eintraege gefunden werden, die dem Suchkriterium
entsprechen
while(@row = $sth->fetchrow_array)
{
 # Ausgabe der jeweiligen Elemente des Arrays
 print "Name: $row[0]\n";
 print "Strasse: $row[1]\n";
 print "PLZ: $row[2]\n";
 print "Ort: $row[3]\n";
 print "Vorwahl: $row[4]\n";
 print "Durchwahl: $row[5]\n";
}

finish $sth,

$dbh->disconnect
 or die "Fehler beim Verbindungsabbau: $DBI::errstr\n";

exit;
```

Die Eingabe von

recherche or

hätte dementsprechend folgendes Ergebnis:

Das folgende Beispielprogramm dient zur Ergänzung der Datenbank um weitere Einträge.

Der Anwender soll hierzu:

- das Programm aufrufen,
- interaktiv zur Eingabe von Namen, Strasse, Postleitzahl, Ort, Vorwahl und Durchwahl aufgefordert werden.



## Perl für die Manipulation von Datenbankinhalten

- seine Eingaben angezeigt bekommen und aufgefordert werden anzugeben, ob alle Eingaben korrekt sind. Wenn er dieses bejaht, soll mit dem Programm fortgefahren werden, ansonsten sollen die Eingaben wiederholt werden.

```
#!/usr/bin/perl

use DBI;
Initialisierung der Antwort-Variablen
$antwort="n";

Beachten Sie in der folgenden Zeile, dass der durch die
Taste Return verursachte Zeilenvorschub beim
Zeichenkettenvergleich berücksichtigt werden muss. Wir
hätten die Variable natürlich auch (wie weiter unten
mit den weiteren Variablen geschehen) mit chop()
bearbeiten können...
until ($antwort eq "j\n" or $antwort eq "J\n")
{
 print "Name des gastronomischen Betriebs: ";
 $name=<STDIN>;
 print "Strasse: ";
 $strasse=<STDIN>;
 print "PLZ: ";
 $plz=<STDIN>;
 print "Ort: ";
 $ort=<STDIN>;
 print "Vorwahl: ";
 $vorwahl=<STDIN>;
 print "Durchwahl: ";
 $durchwahl=<STDIN>;

 print "\nSie haben folgende Eingaben vorgenommen:\n";
 print "Name: ";
 print $name;
 print "Strasse: ";
 print $strasse;
 print "PLZ: ";
 print $plz;
 print "Ort: ";
 print $ort;
 print "Vorwahl: ";
 print $vorwahl;
 print "Durchwahl: ";
 print $durchwahl;
 print "\nSie diese Eingaben korrekt: ";
 $antwort=<STDIN>;
}

Ausfiltern des durch die Taste Return bedingten
Zeilenvorschubs
```



## Programmierung in Perl

---



```
chop($name, $strasse, $plz, $ort, $vorwahl, $durchwahl);

Vorbereitung des Datenbank-Handles
my $dbh = DBI->connect("DBI:Pg:dbname=gastro")
 or die "Fehler beim Verbindungsauflauf: $DBI::errstr\n";

Vorbereitung der SQL-Anweisung
Beachten Sie die automatisierte Vergabe des Index
my $sth = $dbh->prepare(
 insert into namen(index) select max(index)+1 from namen;
 insert into adressen(index) select max(index) from namen;
 insert into telefon(index) select max(index) from namen;

 update namen
 set name="$name"
 where index=(select max(index) from namen);

 update adressen
 set strasse="$strasse",
 plz=$plz,
 ort="$ort"
 where index=(select max(index) from adressen);

 update telefon
 set vorwahl="$vorwahl",
 durchwahl="$durchwahl"
 where index=(select max(index) from telefon);");

Ausfuehrung der SQL-Anweisung
$sth->execute
 or die "Fehler bei Befehlsausfuehrung: $DBI::errstr\n";

finish $sth,
$dbh->disconnect
 or die "Fehler beim Verbindungsabbau: $DBI::errstr\n";

exit;
```

## Zusammenfassung

Wir hoffen, Ihnen einen kleinen Einblick in die Programmiersprache Perl gegeben zu haben, der Sie vielleicht neugierig auf diese gemacht hat.

Die grundlegenden Prinzipien sollten Sie durch diese Einführung erfasst haben, so dass einem weiteren Selbststudium anhand weiterführender Literatur ein guter Grundstein gelegt worden sein müsste.



## Zusammenfassung

---



Im Kapitel 18 *HTML und CGI* soll Ihnen ein Einblick in die Seitengestaltung mit HTML gegeben werden. Diese wird häufig auch als „Programmierung“ bezeichnet, was sicherlich aufgrund fehlender Kontrollstrukturen, der nicht vorhandenen Möglichkeit der Definition von Variablen usw. ein wenig vermassen erscheint.

Wir werden dann noch einmal auf die Datenbank *gastro* zurückkommen und demonstrieren, wie man mithilfe von Perl dynamisch Seiten erstellen kann.







## HTML und CGI



|                                                 |     |
|-------------------------------------------------|-----|
| Holz vs. Bytes                                  | 720 |
| Rechnerwelten                                   | 721 |
| Die HTML-„Programmierung“                       | 722 |
| Kombination der einzelnen                       |     |
| Elemente                                        | 745 |
| Troubleshooting                                 | 745 |
| With a little help from my friends              | 746 |
| Ein Beispiel                                    | 746 |
| Die CGI-Schnittstelle                           | 749 |
| Die Übergabe von Formulardaten an CGI-Programme | 750 |
| Perl und CGI-Skripte                            | 763 |



18

## HTML und CGI

---



HTML ist die Abkürzung für **HyperText Markup Language** und erklärt schon im Namen zwei wesentliche Elemente von HTML.

Zum einen weist „Hypertext“ darauf hin, dass der Text nicht sequenziell – wie in einem Editor oder einem Buch – abgelegt ist, sondern man kann durch Verweise – als „Links“ bekannt – von einem Teil des Dokuments zu einem anderen springen, indem man diese Links einfach anklickt. Dies ermöglicht ein rasches Wechseln der Seiten, so dass man bei gezielter Themensuche schnell und einfach fündig wird, ohne mühselig das gesamte Dokument durcharbeiten zu müssen. Weiterhin ist die Einbettung von Bildern, Videos und Klängen möglich, letztere beiden wegen ihrer großen Datenmengen allerdings in eingeschränktem Umfang.

Zum anderen weist „Markup Language“ darauf hin, dass HTML eine Sprache ist, die Text „auszeichnet“, d.h., bestimmte Textteile in ihrer Darstellung auszeichnet, z.B. kursiv, fett oder als Überschrift. Auch können bestimmte Texte farblich abgehoben oder als Link für eine weitere Seite des Dokuments – oder als Link zu einem Dokument ans andere Ende der Welt – ausgezeichnet werden. Weiterhin kann Text in Listen oder Tabellen dargestellt werden.

Die angeführten Beispiele zeigen aber auch die Beschränkungen von HTML. Da es eine Auszeichnungssprache ist, fehlen ihr auch die minimalsten Bestandteile einer Programmiersprache: HTML kennt weder ein Schleifenkonstrukt, wie z.B. *while* oder *for*, oder eine Auswahl, wie z.B. *if*, noch kann HTML rechnen.

Vielleicht ist es aber gerade die Schlichtheit von HTML, die auch dazu beigetragen hat, dass das World Wide Web derzeit mit monatlichen Steigerungsraten von über 10% förmlich explodiert und in wenigen Jahren zum Standardwerkzeug für Menschen auf der Suche nach Informationen geworden ist.

## Holz vs. Bytes

Wenn man von den klassischen Medien Buch oder Zeitschrift bzw. deren kleinstem Bestandteil, einer Seite, ausgeht und eine HTML-Seite, die mittels eines Browsers betrachtet wird, gegenüberstellt, fallen folgende Gegensätze auf, die für die spätere Gestaltung von Webseiten mittels HTML-Programmierung wichtig sind:

- Papiersetzen besitzen innerhalb eines Dokuments eine klar definierte Seitenbreite und -höhe. HTML-Seiten können beliebig breit und lang sein, und jede Seite kann ihr eigenes Format besitzen.
- Ein Browser stellt in seinem Fenster (meist) nur einen Ausschnitt der HTML-Seite dar, man muss sich durch die Seite „scrollen“ (d.h. den darzustellenden Bildausschnitt verschieben). Dies ist unter anderem auch dadurch bedingt, dass das Fenster eines Browsers in fast allen Fällen ein Querformat aufweist, Zeitschriften oder Bücher hingegen normalerweise im Hochformat erscheinen.



- Es gibt keine linken und rechten Seiten mehr, also auch keine Doppelseiten.
- Bei drucktechnischen Erzeugnissen verursacht eine farbige Darstellung deutliche Mehrkosten, HTML liefert die Farbe sozusagen kostenlos mit.

## Rechnerwelten

Betrachtet man die Vielfältigkeit der angeschlossenen Rechner im Internet, so wird klar, dass HTML als „Internetsprache“ sehr flexibel sein muss: Computer aller Arten, Größen und Kapazitäten sind bei den verschiedensten Endbenutzern angeschlossen. Die Seiten sollen auf einem 14-Zoll-Bildschirm mit Standard-VGA genauso – oder ähnlich – aussehen wie auf einem 23-Zoll-Ganzseiten-Bildschirm. HTML darf nicht mit den verschiedenen Betriebssystemen kollidieren, da die unzähligen Net-surfer sowohl Linux (*sic*) als auch Windows benutzen. Auch die Mac-Benutzer möchten die Seite lesen, die verschiedensten UNIX-Derivate finden sich im Netz, und die letzten Unentwegten bringen es nicht übers Herz, ihren Amiga oder Atari in die Ecke zu stellen.

HTML löst diese Problemstellung auf verschiedene Arten. Zum einen wird zur Beschreibung einer Seite reines 7-Bit-ASCII benutzt, eine Norm, die sich auf allen gängigen Rechnern durchgesetzt hat. Zum anderen muss man sich nicht mit den verschiedenen Normen für Umlaute und Sonderzeichen beschäftigen, die in den verschiedenen Normen und Quasi-Standards zwischen den dezimalen ASCII-Werten 128 und 255 liegen, also 8-Bit-Code sind. Außerdem muss man sich bei der Speicherung als Text nicht um den „Byte-Sex“ kümmern, also ob das höherwertige Byte als erstes oder zweites Byte gespeichert wird – der ewige Krieg Motorola vs. Intel. Text ist auf jedem Rechner und jedem Betriebssystem eben eine Abfolge von Zeichen, die wieder so erscheinen soll, wie sie eingegeben wurde. Auch durch die Beschränkung auf rechnerübergreifendes 7-Bit-ASCII trat übrigens eine andere Seitenbeschreibungssprache ihren Siegeszug an: PostScript.

Für HTML-Dateien hat sich zur Kennung das Suffix *.html* als Standard herauskristallisiert. In der DOS/Windows-3.x-Welt fehlt hierfür natürlich mal wieder ein Buchstabe, hier lautet das Suffix *.htm*.

Auf der anderen Seite geht HTML der Anforderung, auf allen Bildschirmen, Browsern und Betriebssystemen das gleiche Ergebnis zu zeigen, schlüssig aus dem Weg: In Abhängigkeit von der Bildschirmauflösung, des verwendeten Browsers und der aktuellen Größe des Browserfensters wird der Text jeweils neu umgebrochen. Der Textumbruch wird also der vorhandenen Fensterbreite angepasst, und dies ist Aufgabe des Browsers und nicht des HTML-Programmierers, der dieses Verhalten kaum – und wenn, auch nur mit einigen Tricks – beeinflussen kann. Wenn das Browserfenster schmäler wird, wird die Seite also länger. Ein klares Verhalten, aber für den Benutzer von DTP-Programmen sehr gewöhnungsbedürftig: Eine HTML-Seite hat kein klar definiertes Seitenverhältnis. Wenn man also eine HTML-Seite an einem 17-Zoll-Bildschirm entwirft, sollte man sich auch darüber im Klaren sein, dass noch sehr viele 14- und 15-Zöller in Benutzung sind und die Seite dort in einem anderen Verhältnis erscheint.



## HTML und CGI

---



Auch wird die Entscheidung bezüglich der Schriftgröße dem Browser überlassen: HTML kann immer nur angeben, wie groß die Schrift im Verhältnis zu der Basis-schriftgröße erscheinen soll. Dies ist wiederum Angelegenheit des Browsers, dort aber meistens einstellbar.

# Die HTML-„Programmierung“

## TAGS

Sämtliche Steuerungen mittels HTML werden durch so genannte „Tags“ vorgenommen. Tags sind Steuerzeichen, die von spitzen Klammern <...> umschlossen sind. Zwischen Groß- und Kleinschreibung wird bei Tags nicht unterschieden, so dass **<B>** dieselbe Funktion erfüllt wie **<b>**. Allerdings darf zwischen der öffnenden spitzen Klammer und dem Steuerzeichen selbst kein Leerzeichen stehen.

Tags treten meistens paarweise auf, so dass sie als „Schalter an“ und „Schalter aus“ funktionieren.

Die Kennung für den jeweiligen „Schalter aus“ ist identisch mit der für „Schalter an“, wird aber durch einen vorangestellten Slash / kenntlich gemacht. Das Beispiel:

Des Wahnsinns **fette** Beute

wird also in HTML folgendermaßen dargestellt:

Des Wahnsinns **<B>fette</B>** Beute

Wenn Tags geschachtelt werden, müssen sie in der umgekehrten Reihenfolge des Öffnens wieder geschlossen werden:

**<I><B>...</B></I>**

Nicht alle Tags treten gepaart auf, es gibt auch einige, die nur eine einzelne Aktion repräsentieren, wie z.B. der Zeilenumbruch.

Im weiteren Text werden aus Gründen der Übersichtlichkeit alle Tags in Großbuchstaben geschreiben.

## Struktur des Dokuments

Wie eine „richtige“ Programmiersprache benötigt auch HTML Strukturelemente, wenn auch minimale. Diese sehen folgendermaßen aus:

```
<HTML>
<HEAD>
...
</HEAD>
```



## Die HTML-„Programmierung“



```
<BODY>
...
</BODY>
</HTML>
```

Auch hier zeigt sich wieder die oben erwähnte Schachtelung sowie das paarweise Auftreten der Tags. Die Tags `<HTML>` und `</HTML>` sind optional und werden von vielen Browsern ignoriert bzw. nicht benötigt.

Auch die Tags `<HEAD>` und `</HEAD>` sind nicht zwingend erforderlich, können aber eine Menge wichtige Informationen bei komplexeren HTML-Seiten enthalten. Außerdem kann ein Header den Titel der Seite ausweisen.

### Kommentare

Kommentare werden von `<!--` und `-->` umschlossen.

Jegliche Information, die zwischen diesen Tags – in Wahrheit Tag-Hälften – steht, wird ignoriert. Allerdings sei vor der Verwendung eines `>` innerhalb des Kommentars gewarnt: Manche Browser interpretieren dies, als sei der Kommentar bereits an dieser Stelle zu Ende und nicht erst korrekterweise beim ersten Auftreten von `-->`. Das Ignorieren jeglichen Textes zwischen den beiden Tag-Hälften bedeutet aber auch, dass Kommentare nicht geschachtelt werden können. Dies ist keine große Einschränkung, da die Anzahl der Kommentare nicht beschränkt ist und sie an jeder Stelle des Dokuments auftreten können.

Standard bei HTML ist allerdings, dass ein Dokument mit einem Kommentar beginnt, der Standardinformationen enthält wie das Datum der Erstellung, den Autor, den Zweck des Dokuments und eventuell die Versionsnummer des Dokuments. Merke: HTML-Dokumente sind reine Textdokumente, so dass man derartige Informationen manuell erstellen muss; es gibt also keine Möglichkeit zur automatischen Speicherung dieser Informationen.

### Header

Der Header wird von den Tags `<HEAD>` und `</HEAD>` umschlossen und ist optional. Wenn er aber auftaucht, muss der Titel des Dokuments angegeben werden. Der Titel wird von den Tags `<TITLE>` und `</TITLE>` umschlossen.

Er wird zur Kennzeichnung der Seite benutzt und erscheint in der Kopfzeile des Browsers, so dass der Leser von HTML-Seiten auf einen Blick zumindest eine Minimalkauf darüber erhält, welche Art von Information er auf dieser Seite findet. Außerdem gibt es im WWW eine Reihe von Suchmaschinen (so genannte „Search-Bots“), die sich durch alle Webseiten arbeiten, um Stichwortverzeichnisse zu erstellen.



## HTML und CGI

---



Wegen der im Web vorhandenen riesigen Datenmengen beschränken sich die meisten aber darauf, nur die Titelzeile zu durchsuchen und zu registrieren, so dass der Titel sowohl zugunsten der Menschen als auch der Web-Roboter mit Bedacht gewählt werden sollte. Beispiele dafür könnten folgendermaßen aussehen:

```
<TITLE>Die Heimseite von Peter Glinsky</TITLE>
```

oder

```
<TITLE>Desaster Area -- The Greatest Rockband Of The
Universe</TITLE>
```

HTML hält weitere Tags für den Header bereit, die in der Hauptsache zur Vereinfachung der Erstellung von komplexen Dokumenten gedacht sind, deren Erläuterung aber den Rahmen dieses Buches sprengen würden.

## Body

Im Body, dem Körper des Dokuments, sind die eigentlichen Daten enthalten, also alle Texte und Bilder, welche die entsprechende Seite darstellen soll. Die komplette Syntax des Tags *BODY* lautet folgendermaßen:

```
<BODY
BACKGROUND= "URL des Seitenhintergrunds"
BGCOLOR= "Farbe"
LINK= "Farbe"
ALINK= "Farbe"
VLINK= "Farbe"
TEXT= "Farbe" >
...
</BODY>
```

Sämtliche auf *BODY* folgenden Parameter sind optional.

Vorsicht: Die einzelnen Parameter werden durch ein Leerzeichen getrennt, es dürfen also keine weiteren Leerzeichen vor und nach dem Gleichheitszeichen auftreten! Die Parameter im Einzelnen:

### ***BACKGROUND=„URL“***

Hiermit wird dem Hintergrund ein Bild zugewiesen – und zwar die Datei, die durch den „Universal Resource Locator“ (*URL*) spezifiziert wird.

Genauere Angaben zu Bildern und URLs folgen weiter unten.

### **Hinweis**



Für Bilder haben sich zwei Standardformate herausgebildet, die von allen Browersn verarbeitet werden können: *GIF* und *JPEG*. Diese Bilder müssen mit entsprechenden Bildverarbeitungsprogrammen erstellt bzw. bearbeitet werden – HTML ermöglicht nur das *Anzeigen* entsprechender Bilder.



## Die HTML-„Programmierung“



Die URL wird an dieser Stelle in aller Regel der Dateiname des Bildes sein; über die Handhabung von Pfadnamen siehe unten.

Hintergrundbilder sollten kleine Bilder mit einem ansprechenden Muster sein. Klein deshalb, weil alle Browser die Bilder „kacheln“, d.h., sie werden horizontal und vertikal so lange wiederholt, bis der Hintergrund des Dokuments komplett mit einer passenden Anzahl von Exemplaren des Bildes gefüllt ist.

Da dieses Bild der Hintergrund für den Text der Seite sein soll, sollte das Bild nicht zu kräftig oder unruhig sein, da der Text sonst nicht mehr gut lesbar ist.

### ***BGCOLOR = „Farbe“***

Der Parameter *BGCOLOR* steuert die Hintergrundfarbe der HTML-Seite.

*BGCOLOR* und *BACKGROUND* schließen sich gegenseitig aus, d.h., nur einer der beiden Parameter darf aktiviert werden.

Bei der Hintergrundfarbe sollte man immer berücksichtigen, dass das Dokument lesbar bleibt, d.h., der Kontrast zwischen der Farbe des Textes und dem Hintergrund möglichst groß gewählt wird. Der größte Kontrast ist immer noch Schwarzweiß, aber da die Möglichkeit der Farben gegeben ist und Schwarzweiß außerdem langweilig aussieht, wird es kaum genutzt werden. Also sollte man die Hintergrundfarbe möglichst in einer weichen Farbe halten: Gelb, Hellblau oder Hellgrün sind bei schwarzem Text keine schlechte Wahl, während z.B. eine Kombination von Blau und Grün kaum lesbar ist.

Die Umkehrung ist natürlich auch möglich: Einen dunkleren Hintergrund mit hellen Buchstaben kann man ebenfalls erstellen, im Extremfall also weiß/schwarz. Das ist natürlich extrem auffällig, ermüdet die Augen aber auch schnell.

Näheres über Farben erfahren Sie weiter unten.

Wenn weder *BACKGROUND* noch *BGCOLOR* aktiviert sind, wird der Hintergrund in dem bekannten Standardgrau dargestellt.



**Hinweis**

## Die Farben der Links

Die Verweise auf andere Seiten des Dokuments oder andere Seiten im Web heißen „Links“. Sie werden kenntlich gemacht, indem sie sowohl unterstrichen als auch in einer anderen Farbe dargestellt werden. Diese Farben können durch die Parameter *LINK*, *ALINK* und *VLINK* beeinflusst werden.

Mit *LINK* wird die Farbe eines noch nicht aktivierten Links dargestellt, *ALINK* (*activated link*) definiert die Farbe, die es im Moment der Aktivierung annimmt, und *VLINK* (*visited link*) steht für die Farbe, die ein bereits „besuchter“ Link erhält.

Auch hierbei sollte darauf geachtet werden, dass zum einen genügender Kontrast zum Hintergrund vorhanden ist und sich zum anderen die drei Möglichkeiten untereinander genügend abheben. Der Anfänger kann diese drei Optionen ganz weglassen, falls seine Farbgebung nicht allzu sehr aus dem Rahmen fällt.



### Textfarbe

Mit dem Parameter *TEXT=„Farbe“* kann die Farbe des Textes beeinflusst werden. Dies sollte aber nur geschehen, wenn die schwarze Voreinstellung nicht genügt, z.B. der Hintergrund schwarz ist.

Hinweis



Der Abschnitt *Farben* weiter unten in diesem Kapitel enthält eine Tabelle mit Farbwerten.

### Steuerung des Textflusses

Wie bereits oben erwähnt, kann nur beschränkt der Textfluss nur eingeschränkt manipuliert werden: Der Text fließt prinzipiell nur über die Fensterbreite, so dass er von der Bildschirmauflösung und der momentanen Fensterbreite abhängig ist. Auch die Wahl der Schriftart bleibt dem Browser bzw. dessen Einstellungen überlassen. Trotzdem gibt es natürlich Tags, welche die Textgestaltung beeinflussen.

### Überschriften

Überschriften werden in das Tag-Paar *<Hn>...</Hn>* (*H* für *Heading*) eingeschlossen, wobei *n* die Werte zwischen 1 und 6 annehmen kann. Der Wert von *n* kann aber zu Verwirrungen führen: *H1* ist die größte Schrift und *H6* die kleinste, so dass sich die zugrunde liegende Logik wohl eher an der Wertigkeit der Überschriften orientiert. Die Werte sind relative Werte zueinander, die der Browser anhand seiner standardmäßigen Schrift berechnet. Besonders wichtig für Überschriften ist:

`ALIGN = [LEFT | RIGHT | CENTER]`

welches weiter unten noch beschrieben wird.

Abb. 18.1

Die verschiedenen Größen der Überschriften  
in HTML

# Überschrift Größe 1

## Überschrift Größe 2

## Überschrift Größe 2

### Überschrift Größe 4

### Überschrift Größe 5

### Überschrift Größe 6



## Die HTML-„Programmierung“



### Fontgröße, -attribute und -farbe

Die Fontgröße wird – wie die Überschriften – wieder relativ festgelegt. Es gibt hier Werte zwischen 1 und 7, wobei nun im Gegensatz zu den Überschriften – verwirrend genug – 1 den kleinsten und 7 den größten Font darstellt.

Die Syntax lautet:

```
 ...
```

So kann der zwischen den Tags befindliche Text in der gewünschten Größe erscheinen. Größenangaben können nicht nur absolut angegeben werden, sondern mit Vorzeichen auch relativ zu dem zuletzt eingestellten Font. Auf diese Weise kann man mit folgendem Beispiel z.B. den ersten Buchstaben eines Absatzes vergrößert darstellen:

```
Hier beginnt der nächste Absatz!
```

was zu folgendem Ergebnis führt:

Hier beginnt der neue Absatz

Abb. 18.2

Relative Fontgröße

Die Grundeinstellung der Fontgröße ist die relative Einstellung 3, zu der auch automatisch zurückgestellt wird, wenn eine Fonteinstellung mit </FONT> beendet wird. Werden keinerlei Angaben über die Fontgröße gemacht, wird der gesamte Text in dieser Größe ausgegeben.

Achtung: Einige Browser haben eigenartige Vorstellungen von den relativen Schriftgrößen. So ergibt z.B. bei Netscape die größte Fonteinstellung, nämlich 7, größere Buchstaben als bei der größten Überschrift H1!



Hinweis

Farbliche Hervorhebungen von Textteilen werden folgendermaßen getätigten:

```
 ...
```

wobei *Farbe* wieder ein hexadezimales Trippel darstellt (s.u.). Mit farblicher Hervorhebung von Textteilen sollte vorsichtig umgegangen werden, um den Text lesbar zu halten.

- Das Paar <*B*>...</*B*> lässt den Text **fett** (*bold*) erscheinen.
- Das Paar <*I*>...</*I*> überführt den eingeschlossenen Text in *kursive* (*italic*) Schrift.



## HTML und CGI

---



- <TT>...</TT> zeigt den entsprechenden Text in **Schreibmaschinenschrift** (*typewriter*) an.
- <BIG>...</BIG> vergrößert wiederum die Schrift, ohne dass man an den Einstellungen der Fontgröße arbeiten muss.
- <SMALL>...</SMALL> lässt die Schrift schmäler laufen, verringert also den Buchstabenabstand.
- <SUB>...</SUB> bedeutet Subscript, der Text erscheint im Verhältnis zur Zeile tiefer gestellt.
- <SUP>...</SUP> ist das Gegenteil, also Superscript: Der Text wird hoch gestellt.

Weiterhin kann das Aussehen des Textes in seiner jeweiligen Schriftgröße und -farbe durch die Vergabe von Attributen verändert werden.

Dass es auch noch ein sehr unschönes Attribut gibt, sei hier nicht verschwiegen: Mit <U>...</U> können Textteile unterstrichen werden. Da dieses Attribut aber im WWW allgemein für Links gilt, kann es nur Verwirrung stiften: Der Benutzer klickt auf den unterstrichenen Text in der Annahme, es handele sich um einen Link – und nichts passiert. Da Webseiten klar und übersichtlich sein sollten, sollte man dieses Attribut nicht zu seinem aktiven HTML-Sprachschatz hinzufügen.

Mit <BLINK>...</BLINK> führt man hingegen den Benutzer nicht auf eine falsche Fährte, sondern geht ihm auf die Nerven. Der so ausgezeichnete Text blinkt nämlich unaufhörlich. Für die Erhöhung der Aufmerksamkeit des Betrachters kann dies natürlich hilfreich sein, allerdings bewirken permanent blinkende Textteile nach kurzem Betrachten genau das Gegenteil: Man mag sie nicht mehr sehen.

## Zeilenschaltungen und Absätze

HTML überliest bei der Wiedergabe von Text die „White Spaces“, d.h. überflüssige Leerzeichen, Tabulatoren und Zeilenschaltungen, um den Text dann dem jeweiligen Browserfenster anzupassen. Dies ist praktisch für HTML, entzieht aber dem Autor der Seite die Kontrolle über den Textfluss.

Um nun Absätze und Zeilenschaltungen zu erzwingen oder vorformatierten Text so erscheinen zu lassen, wie er eingegeben wurde – z.B. für Quelltexte – kommen die folgenden Tags zum Einsatz.

Die Tags <P>...</P> (*paragraph*) schließen einen Absatz ein, dem eine Leerzeile folgt, so dass eine Gliederung des Textes ermöglicht wird. Diese Tags haben weitere optionale Parameter:

ALIGN=[LEFT | RIGHT | CENTER]

ermöglicht die Ausrichtung des Textes als linksbündige, rechtsbündige bzw. zentrierte Ausgabe.



## Die HTML-„Programmierung“



Die linksbündige Ausgabe ist die Voreinstellung, insofern ist *LEFT* redundant. Man kann also ganze Textblöcke beispielsweise zentriert ausgeben wie im folgenden Beispiel:

```
<P ALIGN=CENTRE>Dieser Text erscheint zentriert</P>
```

Wenn man nach einem Absatz keine Leerzeile anfügen möchte, aber an bestimmten Stellen sichergehen will, dass eine Zeilenschaltung ausgeführt wird, so kommt das Tag *<BR>* (*break*, dt. Zeilenumbruch) zum Tragen.

Es gehört zu den wenigen Tags, die nicht paarweise auftreten und somit an beliebiger Stelle des Textes alleine eingesetzt werden können. Wer die optionalen Parameter der Absatzausrichtung nicht benötigt, aber an bestimmten Textstellen trotzdem eine Leerzeile einfügen möchte, dem kann *<BR>* auch helfen: Durch zweimaliges Auftreten nacheinander wird eine Zeilenschaltung ausgelöst, gefolgt von einer weiteren, nämlich der für eine Leerzeile.

Um die Ausgabe von vorformatiertem Text zu ermöglichen, gibt es die Tags *<PRE>...</PRE>*. Diese erzwingen die Ausgabe des Textes in der Form, wie er in dem HTML-Dokument vorgefunden wird, was z.B. für die Ausgabe von Adressen oder Programmquelltext sinnvoll ist, wie man am folgenden HTML-Fragment erkennen kann:

```
<H1>Mein erstes Pascal-Programm:</H1>
<PRE>
 Program Universe;
 begin
 writeln ("hello universe")
 end.
</PRE>
```

was zu folgendem Ergebnis führt.

### Mein erstes Pascal-Programm:

```
Program Universe;
begin
 writeln ('hello universe')
end.
```

○ Abb. 18.3

Vorformatierter Text



## HTML und CGI

---



### Bilder

Die meisten Browser unterstützen die Bildformate *JPEG* und *GIF*. Bilder erscheinen im Text genau dort, wo sie zwischen den verschiedenen Textblöcken eingefügt werden, d.h., sie werden quasi wie der nächste „Buchstabe“ behandelt. So ist die einfachste Darstellung eines Bildes unter der Angabe des Dateinamens im HTML-Dokument möglich:

```

```

wobei *SRC* für „source“, also Quelle steht.

Das funktioniert allerdings nur, wenn das HTML-Dokument und das Bild im gleichen Verzeichnis liegen.

#### Hinweis



Mehr zu den Pfaden in HTML erfahren Sie in den späteren Abschnitten dieses Kapitels.

Das *IMG*-Tag besitzt diverse weitere Parameter. Als Erstes sei die Ausrichtung erwähnt:

```
ALIGN=[TOP|MIDDLE|BOTTOM|RIGHT|LEFT]
```

Die Ausrichtung bei den Werten *TOP*, *MIDDLE* und *BOTTOM* orientiert sich immer an der aktuellen Textzeile, so dass z.B. *TOP* die Oberkante dieser Textzeile meint. Bei den Werten *RIGHT* und *LEFT* wird das Bild – auf aktueller Höhe der jeweiligen Textzeile – am linken oder rechten Rand platziert und der weitere Text fließt um das Bild herum, so dass hier zumindest ein Layout erreicht werden kann, das dem von Zeitschriften ähnelt.

Damit der Text nicht haarscharf neben, über oder unter dem Bild anschließt, sondern für die Lesbarkeit etwas Abstand gewahrt wird, gibt es die Parameter

```
HSPACE=Pixel
VSPACE=Pixel
```

die den horizontalen bzw. vertikalen Abstand der Schrift in Bildschirmpunkten zu dem Bild angeben.

Weiterhin kann das Bild auf ein gewünschtes Format gebracht werden, falls es nicht entsprechend vorliegt. Mit den Parametern

```
WIDTH=Pixel
HEIGHT=Pixel
```

kann die Größe des Bildes in Bildschirmpunkten angegeben werden, wobei immer beachtet werden muss, dass im WWW natürlich die verschiedensten Bildschirmformate zum Einsatz kommen.



## Die HTML-„Programmierung“



Der Parameter

`BORDER=Pixel`

gibt schließlich die Umrandung des Bildes an. Dies ist nur dann wichtig, wenn es als Link (s.u.) benutzt wird, da ein Bild als Link standardmäßig durch einen blauen Rand gekennzeichnet ist.

Wenn man hier die Rahmenbreite auf 0 setzt, kann dies beim Benutzer zu Verwirrung führen – das sollte also nicht geschehen. Falls das Bild kein Link ist, kann mit dieser Option ein mehr oder minder schöner schwarzer Rahmen erzeugt werden.

Die letzte Option für Bilder wiederum kommt den Menschen entgegen, die immer noch mit einem textorientierten Browser ins Web gehen, die Bilder also nicht sehen können. Mit der Option

`ALT="Text"`

kann hier der Text eingegeben werden, der alternativ (*alt*) anstatt des Bildes steht. Auch bei während der Übertragung zusammengesetzten Verbindungen steht dann dort immerhin schon eine Erklärung, während die Bilder immer noch der Übertragung harren.

Bilder bringen im Allgemeinen größere Datenmengen mit sich und stellen eine große Belastung bei der Übertragung im Netz dar.

Aus diesem Grunde sollte man erstens die Auflösung auf 72 dpi heruntersetzen, sofern dies noch nicht geschehen ist; alles andere ist überflüssiger Datenmüll, da feinere Auflösungen am Bildschirm sowieso nicht zu sehen sind. Zweitens sollte man den Besucher nicht mit einem bildschirmfüllenden Bild begrüßen: Dies kostet auch bei einer guten Verbindung viel Zeit, bei einer schlechten kann es nervenaufreibendes Warten verursachen – oder der Benutzer stoppt kurzerhand die Übertragung und bekommt so die Seite überhaupt nicht zu Gesicht.

Das Tag *IMG* hat deshalb noch eine Erweiterung erfahren, die es ermöglicht, das Bild als „Thumbnail“ (dt. Daumnagel), also als Vorschaubild anzuzeigen, um das eigentliche Bild dann auf Anforderung in besserer Darstellungsqualität – und mit entsprechend größerem Datenumfang – anzuzeigen. Der Parameter heißt

`LOWSRC== "URL"`

und wird für das Vorschaubild genutzt. Der Dateiname des Bildes in besserer Qualität wird dann wie gehabt unter *SRC=„URL“* eingetragen.

## URL

Der **Universal Resource Locator (URL)** beschreibt eine Internetquelle, von oder zu der Daten übertragen werden können. Der Aufbau dieser Adresse geht nach folgendem Schema vonstatten:



## HTML und CGI



Methode: //Server/Pfad/Dateiname

Die hierfür anwendbaren Methoden sind zahlreich, an dieser Stelle seien daher nur die drei gebräuchlichsten genannt: *http* für das WWW, *ftp* für den File Transfer und *mailto* zur Versendung von E-Mail. Im einfachsten Fall ist die URL nur ein Dateiname, sogar ohne Pfad, wenn die jeweilige Datei auf dem lokalen Rechner und im gleichen Verzeichnis liegt. Andererseits können Pfad und Dateiname fehlen, wenn die Homepage einer Institution oder Firma angewählt werden soll und von dort aus zu weiteren Unterdokumenten verzweigt wird. Die URL eines Kieler Stadtmagazins lautet z.B.

<http://www.station.de>

wobei keinerlei Pfade oder Dateinamen angegeben worden sind. *.de* ist hierbei die nationale Kennung, der Server befindet sich in Deutschland. Andere nationale Kennungen sind z.B. *.ca* für Kanada oder *.es* für Spanien. In den USA sind die Kennungen noch weiter aufgespalten, wie aus *Tabelle 18.1* ersichtlich ist.

**URL-Kennung in den USA**



Kennung	Bedeutung
<i>com</i>	kommerziell, meist USA
<i>gov</i>	amerikanische Regierungsbehörden
<i>mil</i>	amerikanisches Militär
<i>edu</i>	amerikanische Erziehungsinstitute
<i>org</i>	für Organisationen ( <a href="http://www.Linux.org">www.Linux.org</a> )

*Tabelle 18.1*



Bei URLs muss Groß- und Kleinschreibung beachtet werden – das Internet in seinen Ursprüngen stammt schließlich aus der UNIX-Welt. Dementsprechend findet hier auch nach der Methode oder innerhalb des Pfadnamens kein Backslash, sondern der Slash / Verwendung.

## Hypertext Links

Bis zu dieser Stelle hat HTML keinerlei Neuigkeiten gebracht: Textfluss kontrollieren und Bilder in den Text einbauen, kursive oder fette Textdarstellung, bunte Buchstaben und farbiger Hintergrund hat heute jede bessere Textverarbeitung zu bieten.

Die wirkliche Neuerung von HTML sind die *Hypertext Links*, auch „Hyperlinks“ oder einfach „Links“ genannt. Links ermöglichen das Anspringen des nächsten Dokumentteils per Mausklick, ohne dass man sequenziell den gesamten Text durcharbeiten muss.



## Die HTML-„Programmierung“



Oder man springt zur nächsten Seite des Dokuments. Von dort kann wieder verzweigt werden – oder zurück zur Homepage gesprungen werden.

Wenn einem andere Adressen im WWW erwähnenswert oder hilfreich zu dem jeweiligen Thema erscheinen, so kann mit nur einem Mausklick dorthin verzweigt werden, ohne dass der Benutzer sofort weiß, auf welchem Server oder in welchem Land er sich befindet – was ja letztlich auch egal ist, wenn es dem jeweiligen Thema dienlich ist.

Außerdem können weitere Dienste angeboten werden: Auf Mausklick kann z.B. ein leeres E-Mail-Formular beim Benutzer erscheinen, das bereits die gewünschte Zieladresse enthält, so dass Benutzer der Seiten mühelos Kontakt aufnehmen können.

Die (verkürzte) Syntax für Links lautet wie folgt:

```
 . . .
```

A steht hier für Anker, der zwischen den beiden Tags befindliche Text stellt den als Link ausgezeichneten Text dar, und für die *URL* kommen mehrere Möglichkeiten in Betracht.

Eine weitere HTML-Seite des Dokumentenbaums soll mittels dieses Links angezeigt werden. In diesem Fall genügt es, dass der Dateiname angegeben wird – vorausgesetzt, er befindet sich im gleichen Verzeichnis wie die Ausgangsseite. Im folgenden Beispiel gibt es weitere Informationen auf Mausklick: Die Datei *WEITERE.HTML* wird geladen. Das HTML-Fragment sieht folgendermaßen aus:

```
Interessierte erhalten hier weitere Informationen!
```

Hier wird also der in den Tags eingeschlossene Text *weitere Informationen* unterstrichen und (in der Regel) blau dargestellt, und bei einem Klick auf diese Textregion wird die HTML-Seite *WEITERE.HTML* zur aktiven Seite des Browsers.

Wer lieber mit einer geringeren Anzahl an Seiten, dafür aber mit längeren Seiten arbeitet, dem kann auch geholfen werden. Solche Seiten werden in der Regel am Kopf mit einem Inhaltsverzeichnis versehen, also mehreren Links, und auf Klick wird zu dem gewünschten Textteil auf dieser Seite gesprungen.

Man könnte sich z.B. eine Seite mit Kurzinformationen der in einer Stadt laufenden Filme denken, die oben auf einer HTML-Seite die Filmtitel mit den entsprechenden Links enthalten, während weiter unten auf der Seite weiterer Text und Bilder zu dem jeweiligen Film vorhanden sind.

Hier dient das erste Auftreten der URL innerhalb des Anker-Tags zur Definition einer Sprungmarke, die mit einem Doppelkreuz # beginnt.

Das zweite Auftreten dieser Sprungmarke, kenntlich gemacht durch die Option *NAME* innerhalb des Anker-Tags – jetzt ohne Doppelkreuz –, dient als Sprungziel. Folgendes Beispiel soll dies verdeutlichen:



## HTML und CGI

---



```
....
Aktuelle Filme
Spiel mit das Lied vom Tod

<!-- weitere Links -->
...
<H3>Spiel mir das Lied vom
Tod</H3>

<!-- weiterer Text -->
```

Durch das Doppelkreuz wird hier die URL als Sprung-Label definiert, und durch Aktivieren dieses Links wird innerhalb derselben Seite zu der Stelle verzweigt, an der das Label in einem Anker-Tag – durch die Option *NAME* gekennzeichnet, aber ohne Doppelkreuz – wieder auftritt.

Außerdem besteht die Möglichkeit, auf eine beliebige Adresse im WWW zu verweisen, wenn man meint, dass diese Seite im Zusammenhang von Interesse ist.

Wer also z.B. auf seinen Seiten seine geliebten Rockstars vorstellt, dazu aber noch weitere Informationen des Internets anbieten möchte, geht wie folgt vor:

```
<H3>NILS LOFGREN</H3>
\6
<!-- Text und Bilder über den Künstler -->
Wer die neuesten Informationen von Nils Lofgren
braucht, wird von Rockhouse versorgt.
...
```

Die letzte hier besprochene Möglichkeit ist, dem Benutzer mittels eines Links schnellen Kontakt zu der eigenen E-Mail-Adresse zu ermöglichen. Dies lässt sich durch URL-Methode *mailto* bewerkstelligen.

Nach dem Aktivieren des Links öffnet sich ein leeres E-Mail-Formular des Browsers oder des entsprechenden Mailprogramms, das bereits die Zieladresse, also die des Autors enthält. Der Benutzer muss nur noch seinen Text eintragen und die E-Mail auf die Reise schicken:

Anregungen und Kritik zu diesem Artikel bitte unter  
<A HREF="mailto://station@station.de">meiner  
E-Mail-Adresse</A>

Auf diese Weise braucht sich der Benutzer nicht um die E-Mail-Adresse des Empfängers zu kümmern und kann trotzdem jederzeit mit dem Autor oder dem/den Verantwortlichen der Seite in elektronischen Kontakt treten.

## Farben

Bei den verschiedensten Tags kann die Farbe bestimmt werden, z.B. bei der Hintergrundfarbe der Seite oder dem Erscheinungsbild von Links.



## Die HTML-„Programmierung“



Farbwerte werden in doppelte Anführungsstriche eingeschlossen und als hexadezimaler (!) Wert des RGB-Modells angegeben. Wer also weiche farbliche Zwischen-töne benötigt, muss schon den hexadezimalen Taschenrechner hervorholen. Die wichtigsten Eck-Farbwerte sind in folgender *Tabelle 18.2* zusammengestellt.

Die wichtigsten RGB-Farbwerte

Hex-Wert	Farbe
000000	schwarz
FFFFFF	weiß
FF0000	rot
00FF00	grün
0000FF	blau
FFFF00	gelb
FF00FF	magenta (rot-lila)
00FFFF	cyan (türkis-blau)
FF7700	orange
777777	grau (50 %)

Tabelle 18.2



Grauwerte werden durch den jeweils gleichen Anteil an R-, G- und B-Werten erstellt, so dass 11111 fast Schwarz ergibt, während EEEEEE einen Hauch von Hell-grau erzeugt.

## Umlaute und Sonderzeichen

Da mit HTML erzeugte Seiten von den verschiedensten Rechnerwelten, die an das Internet angeschlossen sind, gelesen werden sollen, musste auch hier ein neuer Standard geschaffen werden.

Als eine der ersten Normungen der Rechnerwelt wurde der ASCII-Code geschaffen (**American Standard Code for Information Interchange**). Er enthielt das Alphabet in kleiner und großer Schreibweise, die Ziffern, die gängigsten Satz- und Sonderzei-chen sowie 32 Steuerzeichen für z.B. den Tabulator oder die Zeilenschaltung.



## HTML und CGI

---



Hier fing das Elend für die europäische Computerwelt schon an: Die Norm war eine amerikanische und kümmerte sich nicht um die in den meisten europäischen Sprachen benötigten zusätzlichen Satzzeichen, wie z.B. das spanische *ç*, oder um zusätzliche Buchstaben, wie die deutschen Umlaute und das *ß* oder das französische *ë*.

Der PC-Kunde ist aber ja bekanntlich König, und so erfanden die Klein- und Kleinstcomputerhersteller jeder seine eigene Norm, um den europäischen Sprachen gerecht zu werden.

Da von den möglichen 256 Zeichen, die in ein Byte passen, von der ASCII-Norm nur 128 genutzt werden, wurde für die fehlenden Zeichen das achte Bit gesetzt und die Zeichen dort – meist willkürlich und ohne große Struktur – abgelegt. Dies wurde aber von jedem Hersteller wiederum willkürlich nach seiner Haus- und Hofnorm vorgenommen, so dass ein *ß* auf einem Atari ST an anderer Stelle als auf dem IBM-PC liegt, wenn auch immerhin beide im achten Bit.

Die Deutsche Industrienorm musste dann aber auch noch den minimalen Standard der ASCII-Norm aufweichen und die Umlaute anstelle der geschweiften und eckigen Klammern legen – für Programmierer schlicht eine Frechheit.

Im Lauf der Jahre hat sich aber der von IBM geschaffene 8-Bit-Zeichensatz für PCs als allgemein akzeptierter Standard herausgebildet.

Dass Microsoft die in der Zwischenzeit geschaffene ANSI-Norm, die aber kaum genutzt wurde, favorisierte, kann dann nur als weitere Bösartigkeit betrachtet werden. So werden nun bei jedem Textaustausch zwischen verschiedenen Rechnern oder verschiedenen Betriebssystemen Konvertierungsprogramme benötigt, um die zusätzlichen Sonderzeichen und Umlaute richtig darzustellen, was auch nicht immer gelingt: Die PC-Tastatur kennt immer noch keine Anführungszeichen unten – schon vor 20 Jahren auf Satzbelichtern eine Selbstverständlichkeit.

Um nun einen Standard zu schaffen, wurde für HTML von den verschiedenen Normen des achten Bits im Text Abschied genommen und eine neue Norm eingeführt: Die zusätzlichen Buchstaben und Sonderzeichen werden als Zeichenkette dargestellt, die jeweils mit dem Zeichen & beginnen, gefolgt von einem mehr oder minder sinnreichen Kürzel für das Zeichen selbst.

Beendet wird die Zeichenkette durch ein Semikolon. Die Kennung für das kleine *ü* lautet somit

&uuml;

wobei man hier mit gutem Willen U-Umlaut lesen kann. Das deutsche Wort „für“ sieht in HTML also folgendermaßen aus:

f&uuml;r



## Die HTML-„Programmierung“



Da der jeweilige Browser sowieso den gesamten Text der betreffenden Seite nach den verschiedenen Tags parsen muss, ist es kaum Mehrarbeit, wenn nebenbei noch die Sonderzeichen, die jetzt als Zeichenkette auftreten, interpretiert werden.

Diese Schreibweise für Umlaute und Sonderzeichen sollte konsequent angewendet werden – falls sie nicht angewendet wird, kann das Ergebnis auf der lokalen Maschine immer noch richtig aussehen. Aber wehe, wenn diese Seiten auf ein anderes Betriebssystem oder einen anderen Rechntyp stoßen: Jeder Mac-Benutzer wird diese Seite sogleich als unprofessionell identifizieren.

Die Kennungen für Sonderzeichen reagieren – im Gegensatz zu den Tags – empfindlich auf Groß- und Kleinschreibung: `&Uuml;` stellt z.B. das große Ü dar, während `&uuml;` das kleine ü wiedergibt. *Anhang C* enthält eine Tabelle mit den HTML-Sonderzeichen.

Die Entscheidung, Sonderzeichen in HTML nicht durch einen 8-Bit-Zeichensatz darzustellen, sondern durch eine Zeichenkette, hat nicht nur den Vorteil, dass man sich nicht um Normen und Quasi-Standards kümmern muss, sondern ist auch für etwaige Erweiterungen offen: Sollten sich weitere Sonderzeichen als notwendig erweisen, so muss nur die Tabelle erweitert werden, und die Browser der nächsten Generation müssen es darstellen können.

### Zentrieren

Mit den Tags `<CENTER>...</CENTER>` wird alles, was sich dazwischen befindet, zentriert. Diese Tags beziehen sich also nicht nur auf Text, sondern zentrieren auch

- Bilder
- Tabellen
- Listen

Wer nur den Text zentrieren will, muss die Absatz-Tags `<P>...</P>` mit dem entsprechenden Parameter benutzen.

### Der horizontale Strich

In HTML gibt es genau ein (!!!) grafisches Element: den horizontalen Strich. Die Syntax lautet folgendermaßen:

```
<HR
WIDTH=[Pixel|Prozent]
ALIGN=[LEFT|RIGHT|CENTER]
SIZE=Pixel
NOSHADE>
```



## HTML und CGI

---



Alle Parameter sind optional; ohne weitere Parameter wird eine horizontale Linie über die gesamte Breite des Browserfensters gezeichnet. Die Parameter im Einzelnen:

- Die Länge des Striches kann mit *WIDTH* sowohl absolut in Pixeln angegeben werden, also z.B. *WIDTH=100*. Sie kann aber auch als Bezug zur aktuellen Breite des Browserfensters in Prozent angegeben werden, nämlich z.B. *WIDTH=50%*.
- Der Strich kann links, rechts oder zentriert ausgerichtet werden, was der Parameter *ALIGN* bewerkstellt.
- Mit *SIZE* wird die Strichstärke absolut in Pixel angeben. Gängige Werte liegen zwischen 1 und 10.
- Normalerweise wird der horizontale Strich in einer 3D-Darstellung mit Schatten abgebildet. Wer das nicht mag, bemüht den Parameter *NOSHADe*.

## Listen

Ein weiteres Gestaltungselement von HTML bilden Listen, die dem Betrachter eine weitere Ordnung des angebotenen Materials bieten. Nachfolgend sind die drei wichtigsten Listentypen beschrieben: die ungeordnete, die nummerierte und die Definitionsliste.

### Die ungeordnete Liste

Die ungeordnete Liste zeigt die Elemente der Liste am Anfang der Zeile mit einem kleinen Icon – in der Regel einem schwarzen Kreis – an. Sie wird in manchen Textverarbeitungen auch „Bulleted List“ genannt. Die Syntax einer ungeordneten Liste lautet wie folgt:

```

Erster Listenausdruck
Zweiter Listenausdruck
...

```

Mittels des Parameters *TYPE* kann noch die Art des Icons angegeben werden, das am Anfang jeder Zeile angezeigt werden soll. Zulässige Werte sind *CIRCLE*, *DISC* und *SQUARE*. Ein kleines Beispiel soll dies veranschaulichen:

```
<UL TYPE=SQUARE>
<LH>Die Olympiade</LH>
Höher
Schneller
Weiter

```



## Die HTML-„Programmierung“



Das Ergebnis sieht dann folgendermaßen aus.

The screenshot shows a white rectangular area containing the text "Die Olympiade" in a large, bold, black font. Below it is a bulleted list: "■ Höher", "■ Schneller", and "■ Weiter", all in bold black font. To the right of this screenshot, there is a legend-like text block:

○ Abb. 18.4  
Die ungeordnete Liste

### Die nummerierte Liste

Die nummerierte Liste kommt z.B. für wissenschaftliche Inhaltsverzeichnisse in Betracht. Die Struktur ist ähnlich der unnummerierten Liste. Die Tags lauten `<OL>` bzw. `</OL>` für „ordered list“, also geordnete Liste.

Der Parameter `TYPE` erfüllt hier eine andere Funktion: Er gibt die Art der Nummerierung an:

- A steht für Großbuchstaben,
- a steht für Kleinbuchstaben,
- I steht für große römische Zahlen,
- i steht für kleine römische Zahlen,
- 1 steht für arabische Zahlen, wobei dies die Standardeinstellung ist.

An dieser Stelle soll nun auch demonstriert werden, dass die nummerierte Liste – wie alle anderen Listen – geschachtelt werden kann:

```
<OL TYPE="1">
<LH>Inhaltsverzeichnis</LH>
Einleitung 4
Straßennamen7
 <OL TYPE="1">
 Geschichte der Straßennamen7
 Kriterien der Benennung oder Umbenennung einer
 Straße 13
 Der Verwaltungsakt einer Umbenennung14
 Straßenumbenennungen17
```



## HTML und CGI

---



```

Wer nimmt Einfluß auf die Umbenennung von
Straße21
.....

```

Was dann zu folgendem Ergebnis f&uuml;hrt:

Abb. 18.5

Die nummerierte Liste

Inhaltsverzeichnis
1. Einleitung 4
2. Stra&szlig;ennamen7
1. Geschichte der Stra&szlig;ennamen7
2. Kriterien der Benennung oder Umbenennung einer Stra&szlig;e 13
3. Der Verwaltungsakt einer Umbenennung14
4. Stra&szlig;enumbenennungen17
3. Wer nimmt Einflu&szlig; auf die Umbenennung von Stra&szlig;e21
4. ....

### Die Definitionsliste

Als Letztes bleibt noch die Definitionsliste zu erkl&uuml;ren, die f&uuml;r Ubersetzungen oder Erl&uuml;uterungen gebraucht wird. Zu jedem Ausdruck wird eine Definition des Ausdrucks eingerickt dargestellt. Die Syntax lautet:

```
<DL>
<DT>Ausdruck<DD>Die Definition des ersten Ausdrucks
<DT>Ausdruck<DD>Die Definition des zweiten Ausdrucks
</DL>
```

Das Tag-Paar `<DL>...</DL>` schlie&szlig;t also die gesamte Liste ein, w&uuml;hrend am Anfang eines Ausdrucks (*term*) das Tag `<DT>` stehen muss. Die Definition selbst wird durch das Tag `<DD>` eingeleitet.

Ein Beispiel mag dies veranschaulichen:

```
<DL>
<LH>Hackerslang</LH>
<DT>IBM<DD>Internationaler binärer Müll
<DT>Microsoft<DD>klein-weich
<DT>Vobis<DD>Wo ist das Handbuch -- bis wann läuft
die Garantie
</DL>
```



## Die HTML-„Programmierung“



Das Ergebnis dieses Beispiels sieht dann so aus:

Hackerslang	Abb. 18.6
IBM	Die Definitionsliste
Internationaler binärer Müll	
Microsoft	
klein-weich	
Vobis	
Wo ist das Handbuch - bis wann läuft die Garantie	

○ Abb. 18.6

Die Definitionsliste

## Tabellen

Tabellen finden sich inzwischen auch in jeder besseren Textverarbeitung, also wurde dieses Leistungsmerkmal, wenn auch spät, in HTML integriert.

Tabellen werden von den Tags `<TABLE>...</TABLE>` umschlossen, wobei die Anzahl der möglichen Parameter groß ist. Im folgenden wird nur die Möglichkeit beschrieben, die Stärke des Rahmens zu bestimmen, was mit

`BORDER=Pixel`

geschieht, wobei *Pixel* die Anzahl der Bildschirmpunkte angibt.

Falls man eine „frei schwebende“ Tabelle ohne Rahmen möchte, setzt man also `BORDER=0`, was aus optischen Gründen manchmal nützlich sein kann, da Tabellen oft „missbraucht“ werden, um z.B. mehrspaltigen Text zu formatieren.

Jede Tabelle kann eine Überschrift besitzen, die von den Tags `<CAPTION>...</CAPTION>` „umschlossen“ wird. Jede Zeile einer Tabelle wird von den Tags `<TR>...</TR>` (*table row*, dt. Tabellenzeile) „umrahmt“. Innerhalb dieser Zeilen befinden sich wiederum die einzelnen Zellen der Zeile, die in die Tags `<TD>...</TD>` (*table data*, dt. Tabellendaten) eingeschlossen werden.

Innerhalb dieser beiden Tags – und nur hier – befindet sich jetzt der Text oder das Bild – ja, das geht auch –, das die Zelle enthalten soll. Die Zellen-Tags können wiederum den Parameter

`ALIGN = [LEFT | RIGHT | CENTER]`

enthalten, um den Inhalt auszurichten.



## HTML und CGI

---



Wenn man eine symmetrische Tabelle erstellen möchte, so muss darauf geachtet werden, dass zwischen den Zeilen-Tags jeweils die gleiche Anzahl von Zellen-Tags vorhanden ist. Das hört sich trivial an, ist aber gerade noch bei einer Tabelle mit drei mal drei Zellen auf einen Blick zu übersehen, bei größeren Tabellen kann dies hingegen schnell zur Fehlerquelle werden: Der Browser orientiert sich an der Zeile mit den meisten Zellen; in der Zeile, in der Zellen fehlen, erhält die letzte Zelle den überschüssigen Platz.

Die schlichteste Tabelle besteht also aus einer Zeile und einer Reihe – nicht gerade das, was man von einer Tabelle erwartet, aber zur Erläuterung der Syntax hilfreich:

```
<TABLE>
<TR>
 <TD>Ich bin ein Einzeller</TD>
</TR>
</TABLE>
```

Wie beschrieben kann man Tabellen für viele Aufgaben zweckentfremden.

Gern werden sie für die Darstellung eines Kalenders benutzt. Um dies zu demonstrieren, soll hier der Monat April 1997 in der kompletten HTML-Syntax als Beispiel dienen:

```
<HTML>
<!-- P. G., 3/97, V 1.0 -->
<HEAD>
<TITLE>April 1997</TITLE>
</HEAD>
<BODY BGCOLOR="#FFFFFF">
<TABLE BORDER=3>
<CAPTION>APRIL 1997</CAPTION>
<TR>
 <TD>SO</TD>
 <TD>MO</TD>
 <TD>DI</TD>
 <TD>MI</TD>
 <TD>DO</TD>
 <TD>FR</TD>
 <TD>SA</TD>
</TR>
<TR>
 <TD> </TD>
 <TD> </TD>
 <TD ALIGN=ALIGN>1</TD>
 <TD ALIGN=ALIGN>2</TD>
 <TD ALIGN=ALIGN>3</TD>
 <TD ALIGN=ALIGN>4</TD>
 <TD ALIGN=ALIGN>5</TD>
```



## Die HTML-„Programmierung“

```
</TR>
<TR>
 <TD ALIGN=CENTER>6</TD>
 <TD ALIGN=CENTER>7</TD>
 <TD ALIGN=CENTER>8</TD>
 <TD ALIGN=CENTER>9</TD>
 <TD ALIGN=CENTER>10</TD>
 <TD ALIGN=CENTER>11</TD>
 <TD ALIGN=CENTER>12</TD>
</TR>
<TR>
 <TD ALIGN=CENTER>13</TD>
 <TD ALIGN=CENTER>14</TD>
 <TD ALIGN=CENTER>15</TD>
 <TD ALIGN=CENTER>16</TD>
 <TD ALIGN=CENTER>17</TD>
 <TD ALIGN=CENTER>18</TD>
 <TD ALIGN=CENTER>19</TD>
</TR>
<TR>
 <TD ALIGN=CENTER>20</TD>
 <TD ALIGN=CENTER>21</TD>
 <TD ALIGN=CENTER>22</TD>
 <TD ALIGN=CENTER>23</TD>
 <TD ALIGN=CENTER>24</TD>
 <TD ALIGN=CENTER>25</TD>
 <TD ALIGN=CENTER>26</TD>
</TR>
<TR>
 <TD ALIGN=CENTER>27</TD>
 <TD ALIGN=CENTER>28</TD>
 <TD ALIGN=CENTER>29</TD>
 <TD ALIGN=CENTER>30</TD>
 <TD ALIGN=CENTER> </TD>
 <TD ALIGN=CENTER> </TD>
 <TD ALIGN=CENTER> </TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

Das Ergebnis sieht dann folgendermaßen aus:



## HTML und CGI



Abb. 18.7

Der Kalender des Monats April als Tabelle

SO	MO	DI	MI	DO	FR	SA
				1	2	3
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

## Pfade

In den meisten Fällen soll das HTML-Dokument auf einen Server transportiert werden, der nicht im eigenen Besitz ist.

Dies bedeutet, dass man auch nicht unbedingt weiß, an welcher Stelle im Datebaum des Servers die eigenen Daten abgelegt werden. Eine Angabe von festen Pfaden ist demzufolge strikt verboten, da sie auch jegliche Flexibilität bei einer eventuellen Neuorganisation der Dateistruktur des Servers behindert.

Weiterhin stellt sich die Frage, auf welchem Betriebssystem der Server arbeitet. Mit einer DOS-Syntax, die den Pfad mit einem Laufwerksbezeichner versieht, kann ein Linux-Server schon gar nicht arbeiten, und Linux-Server sind im Web sehr verbreitet.

Die schlichteste Möglichkeit – und für kleine Projekte auch ausreichend – besteht darin, dass alle zugehörigen Dateien, also sowohl HTML-Dateien als auch Bilder, in einem Verzeichnis gelagert werden und man sich so die Angabe des Pfades bei Links auf eigene Dokumentteile oder Bilder sparen kann.

Bei größeren Projekten führt das aber schnell zur Unübersichtlichkeit. Hier hilft die *relative Angabe* von Pfadnamen: Die Homepage und eventuelle weitere wichtige Dokumente werden in einem Verzeichnis gehalten. Von diesem Verzeichnis – dafür muss auf dem Server gesorgt werden – gehen bestimmte Unterverzeichnisse ab, die entsprechend ihrer Inhalte etwa *GIF*, *JPEG* oder auch *SEITE2-9* heißen könnten. Um nun etwa ein Bild zu laden, das im Verzeichnis *GIF* liegt, muss folgende Syntax angewandt werden:

```

```

Entsprechend können dann aus anderen Verzeichnissen andere Dokumente geladen werden, und trotzdem bleibt die Übersicht gewahrt.



## Troubleshooting



Übrigens funktioniert diese HTML-Syntax mit dem Slash auch auf Rechnern, die unter DOS/Windows 3.1 laufen, obwohl diese ja sonst den Backslash benötigen.



Hinweis

## Kombination der einzelnen Elemente

Mit den jetzt bekannten, grundlegenden Elementen kann man eine Webseite erstellen, man muss nur wissen, wie. Das Wichtige ist, dass man (fast) alle HTML-Elemente kombinieren kann. Beispiele hierfür sind:

- eine Liste mit Links
- eine Tabelle mit Links
- eine Tabelle mit Bildern

Man könnte z.B. in dem oben vorgestellten Kalender jedes einzelne Datum als Link programmieren, wobei dann auf Mausklick die Termine des jeweiligen Tages geladen werden; eine Möglichkeit, die von vielen Veranstaltungskalendern genutzt wird.

Oder man legt eine Tabelle mit kleinen Vorschaubildern an, z.B. eines Films oder einer Rockband: Die Ladezeiten bleiben so erträglich.

Wer aber Interesse an dem qualitativ hochwertigeren großen Bild hat, der klickt halt das kleine Vorschaubild an und nimmt so die langen Ladezeiten für ein Bild in Kauf, das in der Regel zwischen 50 und 200 KByte umfasst.

## Troubleshooting

An dieser Stelle wird eine der „beliebtesten“ Fehlerquellen für Anfänger und ihre Behebung bzw. Vermeidung aufgezeigt:

Tags, die paarweise auftreten, besitzen in HTML Gültigkeit bis ans Ende des Dokuments. Das bedeutet, dass der Text, wenn möglicherweise das schließende Tag vergessen wurde, bis ans Ende des Dokuments in der ausgezeichneten Art dargestellt wird – auch wenn zwischendurch gleichartige Tags geöffnet und geschlossen werden. Folgendes Beispiel soll dies verdeutlichen:

```
<BODY>
Hier steht etwas Wichtiges <!-- und hier wurde das
Schließen des Fett-
Attributs vergessen-->

<P>
..... <!--Ein Absatz, der in normaler Textart stehen
sollte-->
</P>
```

745



## HTML und CGI

---



```
Eine Zwischenüberschrift

<P>
..... <!--Ein zweiter Absatz, der in normaler Textart
stehen sollte-->
</P>
</BODY>
```

Bei diesem HTML-Fragment wird der gesamte Body in fetter Schrift erscheinen, da in der zweiten Zeile vergessen wurde, die Fettschrift mit `</B>` wieder auszustellen. Auch das Ein- und Ausschalten des Attributs `fett` in der Zwischenüberschrift schaltet die fette Schrift bis Ende des Dokuments nicht ab, da immer noch ein schließendes Tag fehlt.

Abhilfe kann hier nur entweder übersichtliches Programmieren schaffen – lieber jedem Tag seine eigene Zeile spendieren und nicht mit Einrückungen sparen –, oder man erstellt seine HTML-Seiten mithilfe eines Editors, der die verwendete Syntax überprüft. Unter X Window gibt es z.B. die hervorragenden HTML-Editoren *asWedit* und *bluefish*, die über dieses Leistungsmerkmal verfügen.

Weiterhin sollte man komplexere Seiten in Einzelteile zerlegen, um sie separat zu testen. Das ist wesentlich einfacher als bei einer Programmiersprache: Hier bietet sich jedes Ende einer Struktur an, also z.B. einer Liste, eines Absatzes oder einer Tabelle.

Für einen weiteren Fehler gibt es leider keine Abhilfe: In einigen Versionen des Netscape-Browsers sind die typografischen Anführungszeichen („) nicht implementiert und im Text erscheinen in unschöner Weise ihre Bezeichner `&laquo;` bzw. `&raquo;`. Dieses Verhalten tritt im Communicator 4.5 nicht mehr auf. Hier hilft nur das Vermeiden dieser Zeichen.

## With a little help from my friends

Wer sich in HTML vertiefen möchte, sich über den aktuellen Stand der Weiterentwicklungen informieren möchte oder Programmertipps benötigt, der findet unter der Adresse <http://www.w3.org/hypertext/WWW/MarkUp> alles, was das Herz begehrte – und noch viel mehr!

## Ein Beispiel

Im Folgenden wird anhand eines kleinen Beispiels gezeigt, wie man eine HTML-Seite gestalten kann.

```
<HTML>
<HEAD>
<TITLE>Desaster Area -- die größste Rockband
des Universums</TITLE>
</HEAD>
```



## Ein Beispiel



```
<BODY BGCOLOR="FFFF00" TEXT="000000" LINK="0000FF"
VLINK="0000FF">
<HR>
<CENTER>
<H2>DESASTER AREA -- DIE GRÖSSTE ROCKBAND DES
UNIVERSUMS</H2>
<H3>Die offizielle Hompage des Fanclubs, Sektion
Erde</H3>
</CENTER>

<HR>
<IMG SRC="DESASTER.GIF" ALT="Bild des Raumschiffs von
Desaster Area"
LIGN="LEFT" HSPACE="10">

<P>
Willkommen auf der offiziellen
Homepage des Fanclubs von
Desaster Area, Sektion Erde. Desaster Area wurde uns
Erdlingen erstmalig bekannt durch Douglas Adams, der sie
im "Trampführer durch die Galaxis" beschrieb. Auch
wenn sie sich in unserem Sonnensystem in letzter Zeit
etwas rar gemacht haben, so kommen sie doch
demnächst wieder in die Nähe von Erde und
Mars, um dann bei Pluto ihr Abschiedskonzert in diesem
Sonnensystem zu geben und uns dann in Richtung Sirius zu
verlassen. In die Nähe will meinen: Desaster Area
spielen ihre Konzerte immer in ihrem Rauschiff, das sich
jeweils im Orbit des jeweiligen Planeten befindet.
</P>
<P>
Die sich rar gemacht
haben, kommt nicht von ungefähr. Kenner wissen,
daß Desaster Area am Ende eines jeden Konzertes
einen Planten in die jeweilige Sonnen fallen
ließen. Dieses erregte bei den interstellaren
Behörden in letzter Zeit etwas Unwillen, so
daß die Band jetzt neuerdings unübersehbar
Kometen an dem jeweiligen Planeten in Richtung Sonne
fliegen lässt. Der zum dem irdischen Konzert
geplante Komet ist bereits unter dem Namen Hale-Bopp
bekannt geworden.
<P>
<CENTER>
<H2> Die Tourneedaten</H2>
<TABLE BORDER=5>
<CAPTION ALIGN=top>Sonnenstystem</CAPTION>
<TR>
 <TD>Planet</TD><TD>Foto</TD><TD>Datum
(irdisch)</TD>
</TR>
```

## HTML und CGI

---



```
<TR>
<TD>Erde</TD><TD></TD><TD>10.4.1997</TD>
</TR>
<TR><TD>Mars</TD><TD></TD><TD>12.4.1997</TD>
</TR>
</TR>
<TR><TD>Pluto</TD><TD></TD><TD>1.5.1997</TD>
</TR>
</TABLE>

<HR>
<H2>Die Neueste</H2>

Die Konzerte werden auf
Großbildleinwand in folgenden Stadien
übetragen

Volksparkstadion, Hamburg
Weserstadion, Bremen
Parkstadion, Gelsenkirchen
Gottlieb-Daimler-Stadion, Stuttgart
Ruhr-Stadion, Bochum

Weitere Stadien in Vorbereitung

<HR>
Falls wir Dich per E-Mail weiter auf dem Laufenden halten
sollen, bitte ins Formular
eintragen!

<HR>
</CENTER>
</BODY>
</HTML>
```

Dieser HTML-Code führt zu folgendem Ergebnis:

Der Link am Ende der Seite auf die Datei *desaster.html* führt dann direkt zur CGI-Programmierung und damit zu in Verbindung stehenden HTML-Formularen.



## Die CGI-Schnittstelle



Abb. 18.8

Die Homepage von Desaster Area

## Die CGI-Schnittstelle

Zu den häufig in HTML-Dokumenten verwendeten Elementen gehören Formulare, die es dem Anwender ermöglichen, Informationen an den Server zu versenden. Dies geschieht beispielsweise, wenn ein Antrag auf Aufnahme in eine Mailingliste erfolgen soll. In diesem Fall würde der Anwender seine persönlichen Daten (Realname, Adresse, E-Mail-Adresse usw.) in das Formular eintragen, eine Schaltfläche anklicken und somit die Daten an den Server senden, auf dem idealerweise ein Programm läuft, das die Mailingliste automatisch aktualisiert.

Formulare aktivieren also Programme.

Dies geschieht unter Zuhilfenahme des so genannten „Common Gateway Interface“ (CGI). Das CGI ist die Schnittstelle zwischen dem Client (dem Webbrowser, der die Formulareinträge absendet) und dem Server, der diese weiterverarbeitet.

Die Sprache, die Sie für Ihre CGI-Programmierung wählen, bleibt Ihnen dabei weitestgehend freigestellt – ein „Klassiker“ ist hierbei sicherlich C (aufgrund seiner besonderen Bedeutung für UNIX), doch wird diese Sprache mit der fortwährenden Weiterentwicklung von Perl immer mehr durch diese ersetzt, da gerade Zeichenkettenmanipulationen sehr einfach mit Perl durchzuführen sind.



## HTML und CGI



Am Ende dieses Kapitels finden Sie daher einige abschließende Beispiele in Perl, die auf dem vorangehenden *Kapitel 1 Erstinstallation* aufbauen.

Zur Erledigung sehr einfacher Aufgaben genügt es jedoch oftmals, einfache Shell-Skripte zu erstellen, welche die Verarbeitung der übergebenen Formularelemente übernehmen.

In der hier vorgenommenen Darstellung soll deshalb zunächst ein einfaches Beispiel in Form eines Shell-Skripts gegeben werden.

## Die Übergabe von Formulardaten an CGI-Programme

Die Daten, die durch ein CGI-Programm ausgewertet werden, stammen weitestgehend aus drei unterschiedlichen Quellen:

- vom Autor der HTML-Seite,
- vom Benutzer
- und vom Server.

Dem Autor stehen hierfür folgende Formularelemente zur Verfügung.

Formularelemente	
Formularelement	Funktionalität
<i>Check Box</i>	Kontrollfeld
<i>Hidden Box</i>	Verstecktes Feld
<i>Image</i>	Grafik
<i>List Box</i>	Listenfeld (Pulldown)
<i>Password Box</i>	Eingabefeld für ein Passwort
<i>Radio Button</i>	Optionsfeld
<i>Reset Button</i>	Button zum Löschen des Formularinhalts
<i>Submit Button</i>	Button zum Senden der Formulareinträge
<i>Text Area</i>	mehrzeiliges Eingabefeld
<i>Text Box</i>	Eingabefeld

Tabelle 18.3



## Die Übergabe von Formulardaten an CGI-Programme



Nachfolgend eine kurze Beschreibung der Grundlagen für das Erstellen eines Formulars und der Syntax seiner Elemente.

### FORM

Jedes Formular wird durch das Tag `<FORM>` „gerahmt“. Es besitzt folgende Syntax:

```
<FORM
 action="URL"
 method=[get|post]>
 Elemente
</FORM>
```

Mit `action` wird das CGI-Script angegeben, dem die Daten zu übergeben sind. `method` spezifiziert die Art der Übergabe: bei `get` in die Umgebungsvariable `QUERY_STRING` und bei `post` über `stdin`.

### INPUT

Das Tag `<INPUT>` bietet ein Motif-ähnliches Objekt für Benutzereingaben. Seine Syntax lautet:

```
<INPUT
 type=[text | password
 | checkbox | radio
 | submit | reset
 | hidden]
 name ="Name"
 value="Wert"
 size="Zahl"
 maxlength="Zahl"
 checked>
```

Die `INPUT`-Attribute dürfen keine Umlaute enthalten!

- `text` erlaubt Texteingaben der Länge `maxlength` in einem Feld mit `size` Zeichen.
- Bei `password` wird die Eingabe nicht angezeigt.
- Mit `checkbox` wird eine aktivierbare Option geboten. Für mehrere Optionen müssen mehrere `INPUT`-Tags mit gleichem Namen verwendet werden.
- `radio` ist ähnlich wie `checkbox`, erlaubt aber nur eine aktive Auswahl.
- Mit `submit` werden die Werte verschickt und mit `reset` auf die Standardeinstellungen zurückgesetzt.



Hinweis

751





### **SELECT**

<SELECT> bietet eine Motif-ähnliche Auswahlliste, deren Elemente mit dem im Anschluss vorgestellten Tag <OPTION> angegeben werden müssen.

```
<SELECT
 name="Name"
 size="Zahl"
 multiple>
Liste von Optionen
</SELECT>
```

*multiple* erlaubt die Angabe von mehreren Optionen.

### **OPTION**

Ein Element einer SELECT-Liste:

```
<OPTION
 value="Text"
 selected>
```

*selected* aktiviert das Element automatisch.

### **TEXTAREA**

Ein Eingabefeld mit mehreren Zeilen, z.B. für Kommentare:

```
<TEXTAREA
 name="Name"
 rows="Zahl"
 clear=[left | right | all]
 cols="Zahl">
Text
</TEXTAREA>
```

Die meisten Browser fügen automatisch Scrollbars hinzu.

### **Senden von Formulareinträgen**

Durch Anklicken des *Submit*-Buttons werden die Formularwerte an das durch

```
<FORM ACTION="http://www.domain.xxx/cgi-bin/myprog"
METHOD=method>
```

spezifizierte Programm übergeben, wobei, wie bereits zuvor beschrieben, als Methoden zur Verfügung stehen:



## Die Übergabe von Formulardaten an CGI-Programme

- GET – diese Methode hängt die vom Formular übergebenen Daten jeweils an eine Umgebungsvariable *QUERY\_STRING* an.
- POST – diese Methode erlaubt es, den Inhalt des Formulars von *stdin* (aus der Umgebungsvariablen *CONTENT\_LENGTH*) zu lesen.

Außer diesen beiden Umgebungsvariablen stehen noch eine Reihe weiterer Umgebungsvariablen zur Verfügung, die bei Ausführung des Gateway-Programms gesetzt werden.

### Übersicht der wichtigsten Umgebungsvariablen

Variable	Bedeutung
<i>SERVER_SOFTWARE</i>	Der Name und die Version der <i>Information Server Software</i> , welche die Anfrage bearbeitet
<i>SERVER_NAME</i>	Der Hostname des Servers als DNS-Alias oder IP-Adresse, wie er in selbstreferenzierenden URLs verwendet wird
<i>GATEWAY_INTERFACE</i>	Die Revisionsnummer der CGI-Spezifikation, die der Server verwendet
<i>SERVER_PROTOCOL</i>	Der Name und die Revisionsnummer des Informationsprotokolls der Anfrage
<i>SERVER_PORT</i>	Die Nummer des Ports, an den die Anfrage gesendet wurde
<i>REQUEST_METHOD</i>	Die Anfragemethode wie <i>GET, HEAD, POST</i> usw.
<i>PATH_INFO</i>	Pfadinformationen, die durch den Client übergeben werden. Auf Skripte kann mit ihrem virtuellen Pfadnamen zugegriffen werden, gefolgt von weiteren Informationen am Pfadende. Diese Informationen werden als <i>PATH_INFO</i> gesendet. Sie sollten vor der Übergabe an das CGI-Skript vom Server dekodiert werden.
<i>PATH_TRANSLATED</i>	Der Server bietet eine übersetzte Version von <i>PATH_INFO</i> , die den Pfad verwendet und ein Mapping der virtuellen auf physikalische Entsprechungen vornimmt
<i>SCRIPT_NAME</i>	Ein virtueller Pfad zum Skript, das ausgeführt wird; diese Umgebungsvariable wird mit selbstreferenzierenden URLs verwendet



## HTML und CGI

### Übersicht der wichtigsten Umgebungsvariablen (Forts.)

Variable	Bedeutung
<code>QUERY_STRING</code>	Die Information, die dem Fragezeichen in der URL folgt, welche das Skript referenziert. Es handelt sich dabei um die eigentliche Anfrage.
<code>REMOTE_HOST</code>	Der Name des Hosts, der die Anfrage ausführt. Wenn der Server nicht über diese Information verfügt, sollte er die Variable <code>REMOTE_ADDR</code> setzen und diese ungesetzt lassen.
<code>REMOTE_ADDR</code>	Die IP-Adresse des entfernten Hosts, der die Anfrage stellt
<code>AUTH_TYPE</code>	Wenn der Server User-Authentifizierung unterstützt, ist dies die protokollspezifische Authentifizierungsmethode zur Überprüfung des Users.
<code>REMOTE_USER</code>	Wenn der Server User-Authentifizierung unterstützt und das Skript geschützt ist, ist dies der Name des Users, der für den Gebrauch autorisiert ist
<code>REMOTE_IDENT</code>	Wenn der Server die Identifizierungsmethode RFC 931 unterstützt, wird diese Variable auf den User-Namen des entfernten Users gesetzt. Die Nutzung dieser Variablen sollte sich auf Login-Vorgänge beschränken.
<code>CONTENT_TYPE</code>	MIME-Typ der Information, die durch HTTPs <code>POST</code> oder <code>PUT</code> übertragen wird
<code>CONTENT_LENGTH</code>	Länge des Datenblocks, der durch <code>POST</code> oder <code>PUT</code> an die Standardeingabeeinheit der CGI-Anwendung übergeben wird

Tabelle 18.4

Eine vollständige Spezifikation aller Umgebungsvariablen, die für CGI-Anwendungen verfügbar sind, ist unter der WWW-Adresse <http://hoohoo.ncsa.uiuc.edu/cgi/env.html> erhältlich.

Um den Gebrauch dieser Umgebungsvariablen und der zuvor beschriebenen Formularelemente zu veranschaulichen, soll nun gleich in medias res gegangen und eine kleine Beispielapplikation entwickelt werden, die sich auf das im letzten Abschnitt angedeutete Formular für den Antrag zur Aufnahme in die „Desaster Area-Mailingliste“ bezieht.

Hierzu entwickeln Sie die HTML-Seite `desaster.html`:

## Die Übergabe von Formulardaten an CGI-Programme



```
<HTML>
<HEAD>
<TITLE> "Desaster Area"-Mailingleiste </TITLE>
</HEAD>
<BODY>
<H1> "Desaster Area"-Mailingleiste</H1>
<HR>
<H2>Ja, </H2><H3>ja, </H3><H4>ja, </H4><H5>ja,
</H5><H6>ja!!!</H6>
```

Ich bitte um Aufnahme in Ihre Mailingliste.

```
<FORM METHOD=GET
ACTION="http://134.245.35.9/cgi-bin/cgi.sh">
<ISINDEX>
<TABLE>
<TR><TD>Vorname: </TD><TD><INPUT NAME="vname" TYPE="TEXT"
SIZE="20 "></TD></TR>
<TR><TD>Nachname: </TD><TD><INPUT NAME="nname"
TYPE="TEXT"
SIZE="20 "></TD></TR>
<TR><TD>Straße: </TD><TD><INPUT NAME="strasse"
TYPE="TEXT"
SIZE="42 "></TD></TR>
<TR><TD>PLZ: </TD><TD><INPUT NAME="plz" TYPE="TEXT"
SIZE="5 "> Ort: <INPUT
NAME="ort" TYPE="TEXT" SIZE="30 "></TD></TR>
<TR><TD>Planet: </TD><TD><INPUT NAME="planet" TYPE="TEXT"
SIZE="5 "></TD></TR>
</TABLE>

EMail-Adresse: <INPUT NAME="email" TYPE="TEXT" SIZE="39 ">

<CENTER>

<INPUT NAME="los" TYPE="SUBMIT" VALUE="Antrag
abschicken"> <INPUT
NAME="wech" TYPE="RESET" VALUE="Formulareinträge
löschen">

</ISINDEX>
</CENTER>

</FORM>
</BODY>
</HTML>
```



## HTML und CGI



Beim Aufruf dieses Dokuments wird folgende HTML-Seite angezeigt:

Abb. 18.9

Beispielformular für  
eine anschließende CGI-  
Auswertung

The screenshot shows a Netscape browser window with the title "Desaster Area" - Mailingliste. The page content includes several lines of text starting with "Ja," followed by "ja," "ja," "ja," and "ja," each on a new line. Below this, there is a note: "Ich bitte um Anfahrt in Ihre Mailingliste." A form follows with fields for "Vorname" (with an empty input field), "Nachname" (with an empty input field), "Straße" (with an empty input field), "PLZ" (with an empty input field), "Ort" (with an empty input field), and "Postleitzahl" (with an empty input field).

Um die diesem Formular übergebenen Werte auswerten zu können, muss nun noch ein entsprechendes CGI-Programm entwickelt werden, was in diesem einfachen Beispiel in Form eines Shell-Skripts geschehen soll.

Dieses Skript soll folgende Aufgaben erledigen:

- Es soll einige übergebene Variablen auslesen und an den User zurückgeben.
- Es soll eine weitere Skriptdatei generieren, die das Einlesen der eingegebenen Informationen in eine SQL-Datenbank (*POSTGRES*) automatisieren soll.

Auf die einzelnen Mechanismen, die bezüglich der SQL-Syntax zu beachten sind, kann hier aus verständlichen Gründen nicht eingegangen werden – nehmen Sie diese einfach als vorgegeben hin.

- Der Inhalt der neu generierten Datei soll ausgegeben werden.

Hier zunächst des Code des Shell-Skripts. Erklärungen zu den einzelnen Programmzeilen folgen, nachdem Sie die Bildschirmausgaben etwas näher betrachtet haben:



## Die Übergabe von Formulardaten an CGI-Programme

```
#!/bin/sh

echo Content-type: text/plain
echo

echo CGI/1.x Test-Skript-Report:
echo

echo SCRIPT_NAME = "$SCRIPT_NAME"
echo QUERY_STRING = "$QUERY_STRING"
echo REMOTE_HOST = $REMOTE_HOST
echo REMOTE_ADDR = $REMOTE_ADDR
echo "psql maillist <<!!" > db_act
echo "insert into mlist" >> db_act
echo "values (" >> db_act
echo $QUERY_STRING | sed s/+" "/ "/g | \
 sed s/"vname="/"/g | \
 sed s/"&nname="/"/g | \
 sed s/"&strasse="/",\"/g | \
 sed s/"&plz="/",/g | \
 sed s/"&ort="/",\"/g | \
 sed s/"&planet="/",\"/g | \
 sed s/"&email="/",\"/g | \
 sed s/"&los=Antrag \
abschicken"/"\")/g >> db_act
echo "\g" >> db_act
echo "!!" >> db_act
echo Folgende Anweisung zur automatisierten Ergaenzung
echo der Datenbank wurde generiert:

echo
cat db_act
```

Versuchsweise geben Sie folgende Werte in das Formular ein:

## HTML und CGI



Abb. 18.10

Das Probeformular mit den eingegebenen Werten

The screenshot shows a web browser window with the following details:

- Toolbar:** File, Edit, View, Go, Bookmarks, Options, Directory, Window, Help.
- Location Bar:** http://134.245.35.9/dessaster.html
- Menu Bar:** What's New!, What's Cool!, Destinations, Net Search, People, Software.
- Content Area:**
  - Text area:  
Ja,  
ja,  
ja,  
ja,  
ja  
Ich bitte um Aufnahme in Ihre Mailingliste.
  - Form fields:
    - Vorname:
    - Nachname:
    - Strasse:
    - PLZ:  Ort:
    - Platz:
    - E-Mail-Adresse:

Nach dem Anklicken der Schaltfläche *Antrag abschicken* wird folgende Seite angezeigt, deren Informationen im Einzelnen vergleichend mit dem oben beschriebenen Code des CGI-Skripts betrachtet werden sollen.



## Die Übergabe von Formulardaten an CGI-Programme



The screenshot shows a web browser window with the following details:

- Toolbar:** File, Edit, View, Go, Bookmarks, Options, Directory, Window, Help.
- Address Bar:** Location: http://124.245.35.9/cgi-bin/cgi.sh?username=Heinz&name=Du.
- Menu Bar:** What's New!, What's Cool!, Destinations, Net Search, People, Software.
- Content Area:** Displays the output of a CGI script. The script is a Perl script named 'csg/1.0 test script report'. It includes variable assignments for \$REMOTE\_HOST, \$REMOTE\_ADDR, and \$HTTP\_REFERER. It then prints a list of values: 'Mein Bild', 'Am Fier 23', '12345', 'Fischhafen', 'Erde', 'SchlossBoschity.de'.

Abb. 18.11

Bildschirmausgabe des CGI-Skripts

Diese Bildschirmausgabe wurde durch folgende Programmelemente erzeugt:

Die Zeile nach der Angabe *Content-type* muss immer eine Leerzeile sein, darum steht hier ein einsames *echo*.

```
echo Content-type: text/plain
echo
```

Die erste Zeile definiert, welche Textart vom Skript zurückgegeben werden soll (die Alternative wäre „HTML“ gewesen, was aber weiterer Formatierungen bedurft hätte).



Hinweis



## HTML und CGI

---



Die Zeilen

```
echo CGI/1.x Test-Script-Report:
echo
```

dienen rein kosmetischen Zwecken.

```
echo SCRIPT_NAME="$SCRIPT_NAME"
echo QUERY_STRING = "$QUERY_STRING"
echo REMOTE_HOST = $REMOTE_HOST
echo REMOTE_ADDR = $REMOTE_ADDR
```

dienen der Ausgabe der Werte einiger Umgebungsvariablen: der Name des aufgerufenen Skripts, der übergebene Formularstring, der Name des Hosts, der die Seite abrief, und die Adresse dieses Hosts.

```
echo "psql mailing list <!!>db_act
echo "insert into mlist" >> db_act
echo "values (" >> db_act
```

bewirkt das Schreiben der Zeilen

```
psql maillist <!!>
insert into mlist
values (
```

in eine neu anzulegende Datei *db.act*.

Die folgenden Programmzeilen:

```
echo $QUERY_STRING | sed s/\9+\9/\9\9g|\ \
sed s/"vname="/"/g | \
sed s/"&name="/"/g | \
sed s/"&strasse="/", "/g | \
sed s/"&plz="/", /g | \
sed s/"&ort="/, "/g | \
sed s/"&planet="/, "/g | \
sed s/"&email="/", "/g | \
sed s/"&los=Antrag \
abschicken"/"\)/g >> db_act
```

bewirken eine Manipulation des übergebenen *QUERY\_STRING*s. Wenn Sie diesen näher betrachten, erkennen Sie, dass Leerzeichen hier durch das Zeichen + repräsentiert und die einzelnen Formularwerte jeweils mit ihrem Bezeichner durch einen Ampersand (&) aneinander gehängt werden; darüber hinaus befindet sich auch der Bezeichner des *Submit*-Buttons im *QUERY\_STRING*.

All diese Informationen werden durch den Stream-Editor *sed* ausgefiltert, der sie an bestimmten Positionen durch einfache Anführungszeichen ersetzt, da die SQL-Syntax es so verlangt.



## Die Übergabe von Formulardaten an CGI-Programme



Hierdurch entsteht die Zeichenkette

```
"Hein Bloed", "Am Pier
23", 12345, "Fischhafen", "Erde", "hblood@comcity.de")
```

die ebenfalls in die Datei *db\_act* geschrieben wird.

Die Zeilen

```
echo "\g" >> db_act
echo "!" >> db_act
```

schließen die SQL-Anweisung in dieser Datei, wonach sie durch

```
echo Folgende Anweisung zur automatisierten Ergänzung
echo der Datenbank wurde generiert:
echo
cat db_act
```

ausgegeben wird (in einer realen Situation würde man an dieser Stelle natürlich die Datenbank aktualisieren lassen).

### Sonderzeichen

Genau genommen wurde in diesem Beispiel bisher noch einigen weiteren Problemen aus dem Weg gegangen. Der gute Hein Blöd hat freundlicherweise seinen Nachnamen als 7-Bit-Code übergeben, sprich: Er hat auf die Verwendung des Umlauts in seinem Nachnamen verzichtet – „freundlicherweise“, da auch derartige Sonderzeichen in zeichenkettenkodierter Form an die Variable *QUERY\_STRING* übergeben werden. Die Kodierung für die deutschen Sonderzeichen sind in der folgenden Tabelle aufgeführt.

#### Kodierung der deutschen Umlaute durch den Browser

Sonderzeichen	Code
ä	%E4
ö	%F6
ü	%FC
Ä	%C4
Ö	%D6
Ü	%DC
ß	%DF

Tabelle 18.5



## HTML und CGI



Diese Kodierung von Sonderzeichen finden Sie ebenfalls für Interpunktionszeichen und einige andere Sonderzeichen, wie z.B. das §-Zeichen usw., die aus *Tabelle 18.6* ersichtlich sind.

### Sonderzeichen aus HTML-Dokumenten



Zeichen	Kodierung
!	%21
"	%22
§	%A7
\$	%24
%	%25
&	%26
/	%2F
(	%28
)	%29
=	%3D
?	%3F
+	%2B
"	%27
#	%23
;	%3B
:	%3A
,	%2C
{	%7B
[	%5B
]	%5D
}	%7D
\	%5C

Tabelle 18.6





Diese müssten natürlich ebenfalls entsprechend ausgefiltert werden, wobei sich die zuvor beschriebene Technik mit dem Stream-Editor *sed* sicherlich auf die Dauer als etwas sperrig erweisen würde. Jedoch sollte Ihnen dieses erste Beispiel eine Vorstellung davon gegeben haben, was mit HTML-Seiten und den mit ihnen verknüpften CGI-Skripten grundsätzlich möglich ist.

## Perl und CGI-Skripte

Wie bereits im *Kapitel 17 Programmierung in Perl* mehrfach angedeutet, ist die mittlerweile wohl am häufigsten in der CGI-Programmierung eingesetzte Sprache Perl.

Im vorigen Kapitel wurde bereits eine PostgreSQL-basierte Datenbank entwickelt, auf die mithilfe von Perl-Skripten zugegriffen werden konnte.

Auch wenn diese bereits einen ersten Schritt in der Vereinfachung der Handhabung der Datenbank darstellen, kann man dabei immer noch nicht von absolut anwenderfreundlichen Lösungen sprechen. Der Sachbearbeiter muss sich immer noch (z.B. mit *telnet*) auf dem UNIX-basierten Datenbankserver einloggen und dort Befehle am Prompt eingeben.

Die zunehmende Popularität des Internets hat in der Zwischenzeit in vielen Firmen den Ruf nach einem „Intranet“ laut werden lassen – die Verwendung von Internettechniken in der hausinternen Kommunikation, bei der ein Browser als Front-End für Applikationen Verwendung findet.

Um eine Abfrage an die *gastro*-Datenbank zu richten, könnten Sie beispielsweise folgendes Formular entwickeln:

```
<HTML>
<BODY BGCOLOR=#FFFFFF>
<CENTER>
<H1>Adressen-Suchprogramm</H1>
<HR>
<FORM ACTION=". /cgi-bin/abfrage1" METHOD=POST>
Geben Sie den Namen des gesuchten Restaurants ein: <INPUT
TYPE=TEXT NAME="name">

<INPUT TYPE=SUBMIT VALUE="Abfrage starten">
</FORM>
</BODY>
</HTML>
```

## HTML und CGI

Abb. 18.12

Abfrageformular für die Datenbank *gastro*



Dieses Formular greift auf das Perl-Skript *abfrage1* zurück, wenn die Schaltfläche *Abfrage starten* angeklickt wird. Ein solches Skript könnte im Ansatz wie folgt programmiert werden (beachten Sie auch hier wieder die Kommentare zur Erläuterung der einzelnen Programmschritte):

```
#!/usr/bin/perl

use DBI;

Die Ausgabe der Abfrage soll im HTML-Format generiert
werden
print "Content-Type: text/html\n\n";

Einlesen der Formularwerte in die Variable eingabe
read(STDIN, $eingabe, $ENV{ "CONTENT_LENGTH" });

Ausfiltern des Eingabewerts
($fieldname, $eingabe) = split(/=/, $eingabe);

Ersetzen des Plus- durch Lezeichen
$eingabe =~ tr/+/ /;
Umwandeln der codierten Umlaute
$eingabe =~ s/%([a-fA-F0-9][a-fA-F0-9])/pack("C",
hex($1))/eg;

print "<HTML>
<BODY BGCOLOR=#FFFFFF>
";
print "<CENTER><H1>Abfrageergebnis Ihrer Suche nach
Restaurants, die die
Zeichenkette -- $eingabe -- enthalten</H1></CENTER><HR>";
```





```

Der Datenbank-Handle
my $dbh = DBI->connect("DBI:Pg:dbname=gastro")
 or print "Fehler beim
Verbindungsaufbau:

$DBI::errstr
";

Definition des Statement-Handle
my $sth = $dbh->prepare("SELECT
name,strasse,plz,ort,vorwahl,durchwahl FROM
namen,adressen,telefon where name like '%$eingabe%' and
namen.index=adressen.index
and namen.index=telefon.index");

Ausfuehren der zuvor definierten SQL-Anweisung
$sth->execute
 or print "Fehler bei
Befehlsausfuehrung:

$DBI::errstr
";

my $row;

print "<TABLE BORDER=5>";
print "<TR><TD>Name</TD><TD>Strasse</TD><TD>PLZ</TD>
<TD>Ort</TD><TD>Vorwahl</TD><TD>Durchwahl</TD></TR>";

Einlesen der Datenbankeintraege in den Array row,
solange Eintraege gefunden werden, die dem
Suchkriterium entsprechen
while(@row = $sth->fetchrow_array)
{
 # Ausgabe der Datenbankeintraege in Tabellenform durch
 # Zugriff auf die einzelnen Elemente des Array
 print "<TR><TD>$row[0]</TD><TD>$row[1]</TD>
 <TD>$row[2]</TD><TD>$row[3]</TD>
 <TD>$row[4]</TD><TD>$row[5]</TD>
 </TR>";
}
print "</TABLE>";
finish $sth,

Schliessen der Verbindung zur Datenbank
$dbh->disconnect
 or print "Fehler beim Verbindungsabbau:
$DBI::errstr\n";

print "</BODY>\n\n</HTML>";

exit;

```

Die Eingabe der Zeichenkette *or* in dem in Abbildung 18.12 abgebildeten Formular bewirkt folgende Bildschirmausgabe:

## HTML und CGI



Abb. 18.13

Ausgabe des Suchergebnisses der in Abb. 18.12 initiierten Datenbankrecherche

The screenshot shows a Netscape Communicator window with the title bar "Netscape" and menu bar "Datei Bearbeiten Ansicht Gehe zu Communicator Hilfe". The toolbar includes "Zurück" (Back), "Vorwärts" (Forward), "Neu Laden" (Reload), "Home", "Suche" (Search), "Netscape", "Drucken" (Print), and "Sic" (Save). The location bar shows the URL "http://localhost/cgi-bin/abfrage1". The main content area displays a table titled "Abfrageergebnis Ihrer Suche nach Restaurants, die die Zeichenkette - or - enthalten". The table has columns: Name, Straße, PLZ, Ort, Verwaltl, and Deutschland. Two rows of data are shown:

Name	Straße	PLZ	Ort	Verwaltl	Deutschland
Le Tarte	Schlossstrasse	24103	Kiel	0431	93453
Zerber	Saukable 4-6	24103	Kiel	0431	96186

Die im Folgenden entwickelte Intranet-Webseite ergänzt den Rahmen für eine Ergänzung der Datenbank um einen weiteren Eintrag. Ihr HTML-Code lautet wie folgt:

```
<HTML>
<BODY BGCOLOR=#FFFFFF>
<CENTER>
<H1>Ergänung der Datenbank GASTRO</H1>
<HR>
<FORM ACTION="../cgi-bin/abfrage2" METHOD=POST>

<TABLE BORDER=5>
 <TR>
 <TD>Name: </TD>
 <TD><INPUT TYPE=TEXT NAME="name"></TD>
 <TD> </TD>
 <TD> </TD>
 </TR>
 <TR>
 <TD>Strasse: </TD>
 <TD><INPUT TYPE=TEXT NAME="strasse"></TD>
 <TD> </TD>
 <TD> </TD>
 </TR>
 <TR>
 <TD>PLZ: </TD>
 <TD><INPUT TYPE=TEXT NAME="plz"></TD>
 <TD>Ort: </TD>
```



## Perl und CGI-Skripte



```
<TD><INPUT TYPE=TEXT NAME="ort"></TD>
</TR>
<TR>
<TD>Vorwahl: </TD>
<TD><INPUT TYPE=TEXT NAME="vorwahl"></TD>
<TD>Durchwahl: </TD>
<TD><INPUT TYPE=TEXT NAME="durchwahl"></TD>
</TR>
</TABLE>

<INPUT TYPE=SUBMIT VALUE="Datenbank ergänzen">
</FORM>
</BODY>
</HTML>
```

Der Aufruf dieser Seite bietet die in Abbildung 18.14 dargestellte Eingabemaske, in die bereits ein weiterer zukünftiger Datenbankeintrag eingegeben wurde.

Abb. 18.14

Eingabemaske zur Ergänzung der Datenbank

The screenshot shows a Netscape browser window with the title "Ergänzung der Datenbank GASTRO". The form contains the following data:

Name:	Atheni		
Strasse:	Ihlkatenweg 9		
PLZ:	24247	Ort:	Kiel
Vorwahl:	0431	Durchwahl:	69010

Below the form is a submit button labeled "Datenbank ergänzen".



## HTML und CGI

---



Folgendes Perl-Skript verarbeitet die vorgenommenen Einträge:

```
#!/usr/bin/perl

use DBI

Die dynamisch generierte Ausgabe soll im HTML-Format
erfolgen

print "Content-Type: text/html\n\n";

Einlesen der Formularwerte

if($ENV{ "REQUEST_METHOD" } eq "GET")

{

$eingabe = $ENV{ "QUERY_STRING" };

}

else

{

 read(STDIN, $eingabe, $ENV{ "CONTENT_LENGTH" });

}

Ausfiltern des Ampersands (&)
@input = split(/&/, $eingabe);

Aufspalten der Variablen

foreach $field (@input)

{

 ($typ, $value) = split(/=/, $field);

 # Ersetzen der Plus-Zeichen durch Leerzeichen

 $value =~ tr/+/ /;

 # Decodierung der Sonderzeichen

 $value =~ s/%([a-fA-F0-9][a-fA-F0-9])/pack("C", hex($1))/eg;

 # Zuweisung der Elemente der Eingabe zu Variablen

 if($typ eq "name")

 {

 $name=$value;
```





```
}

if($typ eq "strasse")
{
 $strasse=$value;
}
if($typ eq "plz")
{
 $plz=$value;
}
if($typ eq "ort")
{
 $ort=$value;
}
if($typ eq "vorwahl")
{
 $vorwahl=$value;
}
if($typ eq "durchwahl")
{
 $durchwahl=$value;
}

Ueberpruefung, ob alle Felder Werte enthalten
if($name eq "" or $strasse eq "" or $plz eq "" or $ort eq
 "" or $vorwahl eq
 "" or $durchwahl eq "")
{
 print "<H1>Fehler bei der Eingabe</H1><HR>";
 print "Bitte klicken Sie in Ihrem Browser auf
";
 print "die Schaltfläche ZURÜCK und
";
 print "ergänzen Sie die fehlende(n)
Eingabe(n)!
";
 exit;
}

Vorbereitung des Datenbank-Handles
my $dbh = DBI->connect("DBI:Pg:dbname=gastro")
 or die "Fehler beim Verbindungsaufbau: $DBI::errstr\n";

Vorbereitung der SQL-Anweisung
my $sth = $dbh->prepare(
 "insert into namen(index) select max(index)+1 from namen;
 insert into adressen(index) select max(index) from namen;
 insert into telefon(index) select max(index) from namen;
```



## HTML und CGI

---



```
update namen
set name="$name"
where index=(select max(index) from namen);

update adressen
set strasse="$strasse",
plz=$plz,
ort="$ort"
where index=(select max(index) from adressen);

update telefon
set vorwahl="$vorwahl",
durchwahl="$durchwahl"
where index=(select max(index) from telefon);
");

Ausfuehrung der SQL-Anweisung
$sth->execute
 or die "Fehler bei Befehlsausfuehrung: $DBI::errstr\n";

finish $sth,

$dbh->disconnect
 or die "Fehler beim Verbindungsabbau: $DBI::errstr\n";

print "<H1>Bestätigung</H1><HR>";
print "Die Datenbank wurde um folgende Einträge
ergänzt:
";
print "

Name: $name
";
print "Strasse: $strasse
";
print "PLZ: $plz
";
print "Ort: $ort
";
print "Vorwahl: $vorwahl
";
print "Durchwahl: $durchwahl
";

exit;
```

Die Bestätigung der in Abbildung 18.14 vorgenommenen Eingaben durch Anklicken der Schaltfläche *Datenbank ergänzen* führt zu folgender Bildschirmausgabe:

## Perl und CGI-Skripte

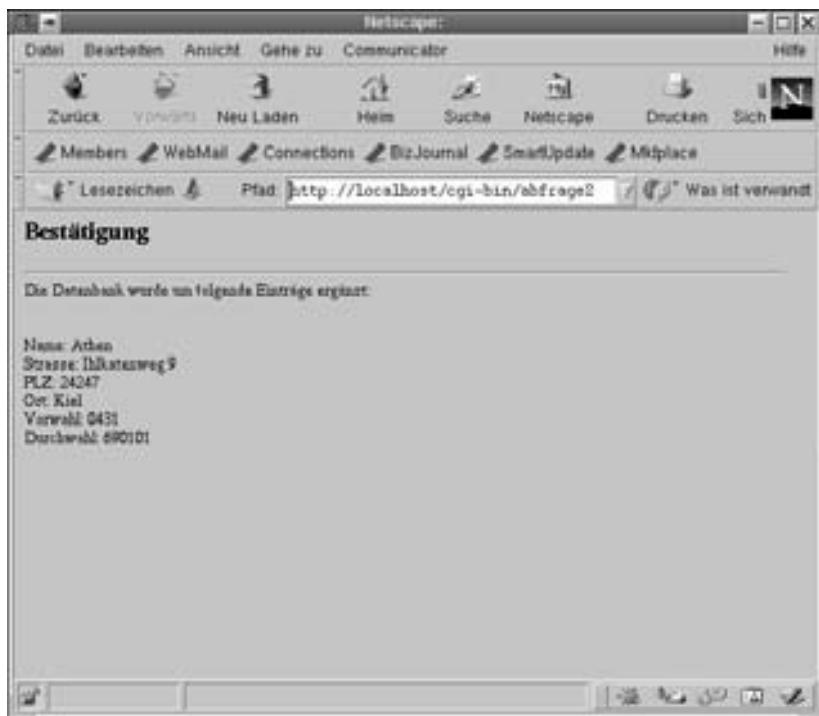


Abb. 18.15

Bestätigung der Ergänzung der Datenbank

Das letzte Beispiel soll die Modifikation bereits vorhandener Einträge demonstrieren. Dies geschieht über eine Reihe von Dialogen. Zunächst soll der zu modifizierende Eintrag wieder gesucht werden können, wozu folgende Eingabemaske kodiert wurde:

```
<HTML>
<BODY BGCOLOR="#FFFFFF">
<CENTER>
<H1>Adressen-Aktualisierung</H1>
<HR>
<FORM ACTION=". /cgi-bin/abfrage3" METHOD=POST>
Geben Sie den Namen des aktualisierenden gastronomischen
Betriebs ein: <INPUT TYPE=TEXT NAME="name">

<INPUT TYPE=SUBMIT VALUE="Abfrage starten">
</FORM>
</BODY>
</HTML>
```

Diese präsentiert sich im Browser wie in Abbildung 18.16 dargestellt.



## HTML und CGI

Abb. 18.16

Eingabemaske für die  
Suche eines zu modifi-  
zierenden Eintrags



Das aus diesem Formular aktivierbare Perl-Skript wurde mit folgendem Inhalt erstellt:

```
#!/usr/bin/perl

use DBI;
Auch hier wird wieder eine neue HTML-Seite durch das
Skript generiert
print "Content-Type: text/html\n\n";

Einlesen des Formulars
read(STDIN, $eingabe, $ENV{ "CONTENT_LENGTH" });

Aufspaltung und Filterung der Formulareingaben
($fieldname, $eingabe) = split(/=/, $eingabe);

$eingabe =~ tr/+/_;
$eingabe =~ s/%([a-fA-F0-9][a-fA-F0-9])/pack("C",
hex($1))/eg;

print "<HTML>
<BODY BGCOLOR=#FFFFFF>
";
print "<CENTER><H1>Abfrageergebnis Ihrer Suche nach --";
$eingabe -- </H1></CENTER><HR>";
```





```
Der Datenbank-Handle
my $dbh = DBI->connect("DBI:Pg:dbname=gastro")
 or print "Fehler beim
Verbindungsaufbau:

$DBI::errstr
";

my $sth = $dbh->prepare("SELECT
name,strasse,plz,ort,vorwahl,durchwahl FROM
namen,adressen,telefon where name like '%$eingabe%' and
namen.index=adressen.index
and namen.index=telefon.index");
$sth->execute
 or print "Fehler bei
Befehlsausfürung:

$DBI::errstr
";

my $row;

print "<H3>Bitte wählen Sie das gewünschte
Lokal und klicken Sie das zu aktualisierende Feld:</H3>";

print "<FORM ACTION=../cgi-bin/abfrage4 METHOD=POST>";

Konstruktion einer Drop-Down-Auswahlliste mit den
gefundenen Namen
print "<SELECT NAME=name>";
while(@row = $sth->fetchrow_array)
{
 $name=$row[0];
 print "<OPTION>$name";
}
print "</SELECT>";

Konstruktion einer Tabelle mit den gefundenen Werten;
das zu aendernde Tabellenfeld soll durch Anklicken
gewaehlt werden koennen
print "<TABLE BORDER=5>";
print "<TR>";
print "<TD><INPUT TYPE=radio NAME=auswahl VALUE=1></TD>";
print "<TD><INPUT TYPE=radio NAME=auswahl VALUE=2></TD>";
print "<TD><INPUT TYPE=radio NAME=auswahl VALUE=3></TD>";
print "<TD><INPUT TYPE=radio NAME=auswahl VALUE=4></TD>";
print "<TD><INPUT TYPE=radio NAME=auswahl VALUE=5></TD>";
print "</TR>";

Schreiben der gefundenen Werte in die Tabelle
while(@row = $sth->fetchrow_array)
{
 print "<TR>";
 print "<TD>$row[1]</TD>";
 print "<TD>$row[2]</TD>";
```

## HTML und CGI



```
 print "<TD>$row[3]</TD>" ;
 print "<TD>$row[4]</TD>" ;
 print "<TD>$row[5]</TD>" ;
 print "</TR>" ;
 }
print "</TABLE>" ;

finish $sth,
$dbh->disconnect
or die "Fehler beim Verbindungsabbau: $DBI::errstr\n";

print "<HR><CENTER><INPUT TYPE=submit
VALUE=Weiter></CENTER>" ;
print "</BODY>\n\n</HTML>" ;

exit;
```

Das zuvor beschriebene Skript bewirkt folgende Anzeige im Browser:



Abb. 18.17

Dynamisch generierte Auswahlliste

Nach Bestätigung der Schaltfläche *Weiter* wird folgendes Skript aufgerufen:





```
#!/usr/bin/perl

print "Content-Type: text/html\n\n";

read(STDIN, $eingabe, $ENV{ "CONTENT_LENGTH" });

@input = split(/&/, $eingabe);

foreach $field (@input)
{
 ($typ, $value) = split(/=/, $field);
 $value =~ tr/+/ /;
 $value =~ s/%([a-fA-F0-9][a-fA-F0-9])/pack("C",
hex($1))/eg;
 if($typ eq "name")
 {
 $name=$value;
 }
 if($typ eq "auswahl")
 {
 $auswahl=$value;
 }
}
print "<HTML>
<BODY BGCOLOR=#FFFFFF>
";
print "<CENTER><H1>Eingabe des zu ändernden
Wertes</H1></CENTER><HR>";
print "<FORM ACTION=../cgi-bin/abfrage5 METHOD=POST>";
print "Bitte geben Sie für das Lokal ";
print "<SELECT NAME=name><OPTION>$name</SELECT> einen
neuen Wert für das zu ändernde Tabellenfeld ";

if($auswahl == 1)
{
 print "<SELECT NAME=tabellenfeld><OPTION>strasse";
}
elsif($auswahl == 2)
{
 print "<SELECT NAME=tabellenfeld><OPTION>plz";
}
elsif($auswahl == 3)
{
 print "<SELECT NAME=tabellenfeld><OPTION>ort";
}
elsif($auswahl == 4)
{
 print "<SELECT NAME=tabellenfeld><OPTION>vorwahl";
}
elsif($auswahl == 5)
{
```



## HTML und CGI



```
 print "<SELECT NAME=tabellenfeld><OPTION>durchwahl";
}

print "</SELECT> ein

";

print "<CENTER><INPUT TYPE=TEXT NAME=neuwert>";
print "<HR><INPUT TYPE=submit VALUE=Weiter>";
print "</CENTER></BODY>\n\n</HTML>";

exit;
```

Dieses Skript bewirkt folgende Anzeige im Browser:

Abb. 18.18



Eingabe des neu in die Datenbank einzulesenden Wertes

Die abermalige Bestätigung der Schaltfläche *Weiter* führt schließlich zur Abarbeitung des folgenden letzten Skripts:

```
#!/usr/bin/perl

use DBI;

print "Content-Type: text/html\n\n";
```





```
read(STDIN, $eingabe, $ENV{ "CONTENT_LENGTH" }) ;

@input = split(/&/, $eingabe);

foreach $field (@input)
{
 ($typ, $value, $newvalue) = split(/=/, $field);
 $value =~ tr/+/ /;
 $value =~ s/%([a-fA-F0-9][a-fA-F0-9])/pack("C",
hex($1))/eg;
 $newvalue =~ tr/+/ /;
 $newvalue =~ s/%([a-fA-F0-9][a-fA-F0-9])/pack("C",
hex($1))/eg;

 if($typ eq "name")
 {
 $name=$value;
 }
 if($typ eq "tabellenfeld")
 {
 $tabellenfeld=$value;
 }
 if($typ eq "neuwert")
 {
 $neuwert=$value;
 }
}

if("$tabellenfeld" eq "strasse" or "$tabellenfeld" eq
"plz" or "$tabellenfeld" eq "ort")
{
 $tabelle="adressen";
}
elsif("$tabellenfeld" eq "vorwahl" or "$tabellenfeld" eq
"durchwahl")
{
 $tabelle="telefon";
}

Vorbereitung des Datenbank-Handles
my $dbh = DBI->connect("DBI:Pg:dbname=gastro")
 or die "Fehler beim Verbindungsaufbau: $DBI::errstr\n";

Vorbereitung der SQL-Anweisung
my $sth = $dbh->prepare(
update $tabelle
set $tabellenfeld="$neuwert"
where index=(select index from namen where
name=\"$name\"));
```

## HTML und CGI



```
Ausfuehrung der SQL-Anweisung
$sth->execute
 or die "Fehler bei Befehlsausfürung: $DBI::errstr\n";
finish $sth,
$dbh->disconnect
 or die "Fehler beim Verbindungsabbau: $DBI::errstr\n";
print "<HTML>
<BODY BGCOLOR=#FFFFFF>
";
print "<CENTER><H1>Bestätigung</H1></CENTER><HR>";
print "Das Tabellenfeld -- $tabellenfeld -- für das
Lokal -- $name -- wurde in -- $neuwert --
geändert!";
print "</BODY>\n\n</HTML>";

exit;
```

Nach der Aktualisierung der Datenbank durch das Skript erfolgt folgende letzte Bildschirmausgabe mit der Bestätigung der vorgenommenen Änderung:



Abb. 18.19

Best&auml;tigung der Datenbankmodifikation

Die hier demonstrierten Beispiele sollten Ihnen bereits einen Vorgeschmack auf die Möglichkeiten Perls in der CGI-Programmierung vermittelt haben.

Wer komplexere Anwendungen schreiben möchte, wird natürlich nicht um die Lektüre weiterführender Literatur herumkommen. Mit zunehmender Sicherheit werden Sie dann auch Nachlässigkeiten der hier als Beispiel verwendeten Seite (z.B. Plausibilitätskontrollen der getätigten Eingaben) beheben können ...

Viel Spaß dabei ;-)





## **Die Tastaturbelegung**



**Belegung der deutschen Tastatur  
ohne Treiber** **780**



**Anhang A**

## Die Tastaturbelegung

Falls kein deutscher Tastaturreiber geladen ist (z.B. während des Urladens von Linux), ist die Tastatur noch nach dem US-Schema belegt.

Diese Tabelle zeigt in der Spalte „US“ das gesuchte Zeichen und unter „D“ die Taste, auf der sich dieses Zeichen auf der deutschen Tastatur befindet.

### Belegung der deutschen Tastatur ohne Treiber

Gegenüberstellung der US-Tastaturbelegung und des deutschen Layouts

US	D	US	D	US	D
~	g	,	^	!	!
@	"	#	§	\$	\$
%	%	^	&	*	(
(	)	)	=	_	?
-	ß	+	,	=	"
{	Ü	[	ü	}	*
]	+	:	Ö	;	ö
"	Ä	"	ä		"
\	#	<	;	,	,
>	:	.	.	?	-
/	-		>	\	<
Z	Y	Y	Z		

Tabelle A.1





## **Kernel-Parameter**



## **Anhang B**



## README- und HOWTO-Dateien

Wir haben nachfolgend alle wichtigen Texte der CD sowie wichtige Adressen aus dem WWW zur Kernel-Konfiguration und Kompilierung tabellarisch aufgeführt. Dass in einigen Fällen auch die C-Quelltexte erwähnt worden sind, liegt daran, dass viele Programmierer nicht einen Readme-Text für den entsprechenden Treiber geschrieben, sondern die Dokumentation innerhalb des Quelltextes, meistens am Anfang, vorgenommen haben. Um die Tabellenform nicht zu sprengen, wurden folgende Abkürzungen eingeführt: Der Pfad der Verzeichnisse *drivers* und *Documentation* ist */usr/src/linux/*. Dies wurde in der Tabelle weggelassen. Außerdem wurde die häufig vorkommende Internetadresse *ftp://sunsite.unc.edu:/pub/Linux/docs/HOWTO* in der Tabelle als *ftp://sunsite...* abgekürzt. Diese Adresse ist übrigens für alle Linux-Fragen, die dieses Buch nicht beantwortet, ein Eldorado.

### 1. Module

CONFIG\_MODVERSIONS  
*ftp://sunsite.unc.edu/pub/LINUX/kernel*

CONFIG\_KMOD  
*Documentation/kmod.txt*

### 2. Prozessoren

CONFIG\_MICROCODE  
*http://www.urbanmyth.org/microcode/*

CONFIG\_MTRR  
*/usr/src/linux/Documentation/mtrr.txt*

CONFIG\_SMP  
*Documentation/SMP.txt*  
*Documentation/smp.txt*

### 3. Allgemeine Einstellungen

CONFIG\_VISWS  
*usr/src/linux/Documentation/sgi-visws.txt*

CONFIG\_PCI  
*ftp://sunsite.unc.edu/pub/LINUX/docs/HOWTO*

CONFIG\_MCA  
*/usr/src/linux/Documentation/mca.txt*

CONFIG\_HOTPLUG  
*http://www.linux-usb.org/policy.html*

CONFIG\_PCMCIA  
*http://www.linuxdoc.org/docs.html#howto*

CONFIG\_BSD\_PROCESS\_ACCT  
*include/linux/acct.h*



## README- und HOWTO-Dateien



CONFIG\_SYSCTL  
*/usr/src/linux/Documentation/sysctl/*

CONFIG\_BINFMT\_MISC  
*/usr/src/linux/Documentation/binfmt\_misc.txt*

CONFIG\_PM  
*http://www.cs.utexas.edu/users/kharker/linux-laptop/*  
*http://www.linuxdoc.org/docs.html#howto*

CONFIG\_ACPI  
*http://www.teleport.com/~acpi/*  
*Documentation/pm.txt*

CONFIG\_APM  
*/usr/doc/howto/en*

### 4. Parallel port support

CONFIG\_PARPORT  
*/usr/src/linux/Documentation/parport.txt*  
*http://www.torque.net/linux-pp.html*

CONFIG\_PARPORT\_PC\_FIFO  
*Documentation/parport.txt*

### 5. Plug and Play configuration

CONFIG\_ISAPNP  
*Documentation/isapnp.txt*

### 6. Block Devices

CONFIG\_BLK\_CPIO\_DA  
*Documentation/cpqarray.txt*

CONFIG\_BLK\_CPIO\_CISS\_DA  
*Documentation/cciss.txt*

CONFIG\_BLK\_DEV\_DAC960  
*Documentation/README.DAC960*

CONFIG\_BLK\_DEV\_NBD  
*http://atrey.karlin.mff.cuni.cz/~pavel*

### 7. Multi-device support (RAID and LVM)

CONFIG\_BLK\_DEV\_MD  
*drivers/block/README.md*

CONFIG\_MD\_RAID1  
*http://luthien.nuclecu.unam.mx/~miguel/raid*

CONFIG\_MD\_RAID5  
*http://luthien.nuclecu.unam.mx/~miguel/raid*

## Kernel-Parameter

---



CONFIG\_BLK\_DEV\_LVM  
*/usr/src/linux/Documentation/LVM-HOWTO*  
*http://linux.msede.com/lvm*

### 8. Networking options

CONFIG\_NETLINK  
*http://snafu.freedom.org/linux2.2/docs/netlink-HOWTO.html*

CONFIG\_FILTER  
*linux/Documentation/networking/filter.txt*

CONFIG\_IP\_MULTICAST  
*http://www.best.com/~prince/techinfo/mbone.html*

CONFIG\_IP\_MULTIPLE\_TABLES  
*http://www.compendium.com.ar/policy-routing.txt*  
*ftp://post.tepkom.ru/pub/vol2/Linux/docs/advanced-routing.tex*  
*ftp://ftp.inr.ac.ru/ip-routing/*

CONFIG\_IP\_ROUTE\_NAT  
*http://www.csn.tu8209;chemnitz.de/~mha/linux8209;ip8209;nat/diplom/nat.html*

CONFIG\_IP\_PNP\_BOOTP  
*Documentation/nfsroot.txt*

CONFIG\_IP\_PIMSM\_V1  
*http://netweb.usc.edu/pim/*

CONFIG\_IPV6  
*http://playground.sun.com/pub/ipng/html/ipng-main.html*

CONFIG\_KHTTPD  
*net/khttpd/README*  
*http://www.fenrus.demon.nl*

CONFIG\_ATM  
*Documentation/networking/atm.txt*

CONFIG\_IPX\_INTERN  
*ftp://metalab.unc.edu/pub/Linux/docs/HOWTO*

CONFIG\_ATALK  
*http://artoo.hitchcock.org/~flowerpt/projects/LINUX-netatalk/*

CONFIG\_DECNET  
*http://linux.dreamtime.org/decnet/*  
*Documentation/networking/decnet.txt*

CONFIG\_BRIDGE  
*Documentation/networking/bridge.txt*

CONFIG\_X25  
*/usr/src/linux/Documentation/networking/x25.txt*  
*/usr/src/linux/Documentation/networking/x25-iface.txt*



## README- und HOWTO-Dateien



CONFIG\_LAPB  
*/usr/src/linux/Documentation/networking/lapb-module.txt*

CONFIG\_NET\_DIVERT  
*http://www.freshmeat.net/projects/etherdivert*  
*http://perso.wanadoo.fr/magpie/EtherDivert.html*

CONFIG\_WAN\_ROUTER  
*/usr/src/linux/Documentation/networking/wan-router.txt*  
*ftp://ftp.sangoma.com*

CONFIG\_NET\_FASTROUTE  
*ftp://ftp.inr.ac.ru/ip-routing/fastroute/fastroute-8390.tar.gz*

CONFIG\_NET\_SCHED  
*ftp://ftp.inr.ac.ru/ip-routing*

### 9. Telephony Support

CONFIG\_PHONE\_IJX  
*Documentation/telephony/ijx.txt*  
*http://www.quicknet.net/*

### 10. ATA/IDE/MFM/RLL support

CONFIG\_IDE  
*Documentation/ide.txt*

CONFIG\_BLK\_DEV\_IDEDMA\_PCI  
*Documentation/ide.txt*

CONFIG\_BLK\_DEV\_AEC62XX  
*drivers/ide/aec62xx.c*

CONFIG\_AEC62XX\_TUNING  
*drivers/ide/aec62xx.c*

CONFIG\_BLK\_DEV\_AMD7409  
*drivers/ide/amd7409.c*

CONFIG\_HPT34X\_AUTODMA  
*drivers/ide/hpt34x.c*

CONFIG\_BLK\_DEV\_VIA82CXXX  
*/usr/src/linux/drivers/block/via82cxxx.c*

### 11. SCSI support

CONFIG\_SCSI\_DEBUG\_QUEUES  
*http://www.andante.org/scsi\_queue.html*

CONFIG\_SCSI\_LOGGING  
*/usr/src/linux/drivers/scsi/scsi.c*

CONFIG\_BLK\_DEV\_3W\_XXXX\_RAID  
*http://www.3ware.com/*



## Kernel-Parameter

---



```
CONFIG_SCSI_AIC7XXX
 /usr/src/linux/drivers/scsi/README.aic7xxx

CONFIG_AIC7XXX_TCQ_ON_BY_DEFAULT
 drivers/scsi/README.aic7xxx

CONFIG_SCSI_CMQFC
 drivers/cmqfc.Readme

CONFIG_SCSI_IBMMCA
 Documentation/mca.txt

CONFIG_SCSI_IPS
 http://www.developer.ibm.com/welcome/netfinity/serveraid.html

CONFIG_SCSI_NCR53C406A
 drivers/scsi/NCR53c406.c
```

### 12. Network device support

```
CONFIG_ARCNET
 /usr/src/linux/Documentation/networking/arcnet.txt

CONFIG_LTPC
 Documentation/networking/ltpc.txt

CONFIG_COPS
 Documentation/networking/cops.txt

CONFIG_IPDDP
 Documentation/networking/ipddp.txt

CONFIG_TUN
 Documentation/networking/tuntap.txt

CONFIG_ETHERTAP
 /usr/src/linux/Documentation/networking/ethertap.txt

CONFIG_NET_SB1000
 http://www.jacksonville.net/~fventuri/
 http://home.adelphia.net/~siglercm/sb1000.html
 http://linuxpower.cx/~cable/

CONFIG_ELPLUS
 Documentation/networking/3c505.txt

CONFIG_VORTEX
 Documentation/networking/vortex.txt
 drivers/net/3c59x.c

CONFIG_SMC9194
 drivers/net/README.smcc9

CONFIG_SK_G16
 drivers/net/sk_g16.c
```



## README- und HOWTO-Dateien



CONFIG\_AC3200  
*drivers/net/cs3200.*

CONFIG\_APRICOT  
*drivers/net/apricot*

CONFIG\_DEC\_ELCP  
*/usr/src/linux/Documentation/networking/tulip.txt*

CONFIG\_DE4X5  
*/usr/src/linux/Documentation/networking/de4x5.txt*

CONFIG\_DGRS  
*/usr/src/linux/Documentation//networking/dgrs.txt*

CONFIG\_DM9102  
*http://www.davicom.com.tw*

CONFIG\_NATSEMI  
*http://www.scyld.com/network/natsemi.html*

CONFIG\_8139TOO  
*Documentation/networking/8139too.txt*

CONFIG\_SIS900  
*Documentation/networking/sis900.txt*

CONFIG\_SUNDANCE  
*http://www.scyld.com/network/sundance.html*

CONFIG\_WINBOND\_840  
*http://www.scyld.com/network/drivers.html*

CONFIG\_ATP  
*drivers/net/atp.c*

CONFIG\_DE600  
*drivers/net/README.DLINK*

CONFIG\_DE620  
*drivers/net/README.DLINK*

CONFIG\_YELLOWFIN  
*http://cesdis.gsfc.nasa.gov/linux/drivers/yellowfin.html*

CONFIG\_SK98LIN  
*Documentation/networking/sk98lin.txt*

CONFIG\_SKFP  
*Documentation/networking/skfp.txt*

CONFIG\_PPP  
*Documentation/networking/ppp.txt*

CONFIG\_PPPOE  
*http://www.math.uwaterloo.ca/~mostrows*



## Kernel-Parameter

---



CONFIG\_STRIP  
<http://mosquitonet.stanford.edu>

CONFIG\_WAVELAN  
</usr/src/linux/Documentation/networking/wavelan.txt>  
[http://www.hpl.hp.com/personal/Jean\\_Tourrilhes/Linux/Wavelan.html](http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Wavelan.html)

CONFIG\_ARLAN  
[www.aironet.com](http://www.aironet.com)  
[www.Telxon.com](http://www.Telxon.com)  
<http://www.ylenurme.ee/~elmer/655>

CONFIG\_AIRONET4500  
[www.aironet.com](http://www.aironet.com)

CONFIG\_IBMTR  
<ftp://metalab.unc.edu/pub/Linux/docs/HOWTO>

CONFIG\_IBMOL  
<ftp://metalab.unc.edu/pub/Linux/docs/HOWTO>  
<http://www.linuxtr.net>

CONFIG\_TMS380TR  
<Documentation/networking/tms380tr.txt>  
<http://www.auk.cx/tms380tr/>

CONFIG\_SMCTR  
<Documentation/networking/smctr.txt>

CONFIG\_SHAPER  
<Documentation/networking/shaper.txt>

CONFIG\_COSA  
<http://www.fi.muni.cz/~kas/cosa>

CONFIG\_COMX  
<Documentation/networking/comx.txt>  
<http://www.itc.hu>

CONFIG\_COMX\_HW\_COMX  
<ftp://ftp.itc.hu/>

CONFIG\_DLCI  
<http://www.ffforum.com>  
</usr/src/linux/Documentation/networking/framerelay.txt>

CONFIG\_WAN\_DRIVERS  
</usr/src/linux/Documentation/networking/wan-router.txt>

CONFIG\_VENDOR\_SANGOMA  
<http://www.sangoma.com>  
</usr/src/linux/Documentation/networking/wanpipe.txt>

CONFIG\_CYCLADES\_SYNC  
<http://www.cyclades.com>



## README- und HOWTO-Dateien



CONFIG\_SBNI  
<http://www.granch.ru>

CONFIG\_PCMCIA\_RAYCS  
*Documentation/networking/ray\_cs.txt*

CONFIG\_ATM\_TCP  
<http://icawww1.epfl.ch/linux-atm/>

CONFIG\_ATM\_IA  
[www.ipphase.com/products/ClassSheet.cfm?ClassID=ATM](http://www.ipphase.com/products/ClassSheet.cfm?ClassID=ATM)  
*Documentation/networking/ipphase.txt*

CONFIG\_ATM\_FORE200E\_MAYBE  
*Documentation/networking/fore200e.txt*

### 13. Amateur Radio support

CONFIG\_HAMRADIO  
<http://www.tapr.org/tapr/html/pkthome.html>

CONFIG\_AX25  
</usr/src/linux/Documentation/networking/ax25.txt>

CONFIG\_6PACK  
</usr/src/linux/Documentation/networking/6pack.txt>

CONFIG\_DMASCC  
<http://hydra.carleton.ca/info/pi2.html>  
<http://www.paccomm.com/gracilis.html>  
*ist:<http://www.nt.tuwien.ac.at/~kkudielk/Linux/>*

CONFIG\_SCC  
</usr/src/linux/Documentation/networking/z853odrv.txt>

CONFIG\_SCC\_DELAY  
</usr/src/linux/Documentation/networking/z853odrv.txt>

CONFIG\_SCC\_DELAY  
*Documentation/networking/z853odrv.txt*

CONFIG\_BAYCOM\_SER\_FDX  
</usr/src/linux/Documentation/networking/baycom.txt>  
<http://www.baycom.de>

### 14. IrDA (infrared) support

CONFIG\_IRDA  
</usr/src/linux/Documentation/networking/irda.txt>

CONFIG\_IRCOMM  
<http://www.pluto.dti.ne.jp/~thiguchi/irda/>



## Kernel-Parameter

---



### 15. ISDN subsystem

CONFIG\_ISDN  
*Documentation/isdn/README*

CONFIG\_ISDN\_PPP  
*Documentation/isdn/README.syncppp*  
*Documentation/isdn/syncPPP.FAQ*

CONFIG\_ISDN\_MPP  
*Documentation/isdn/README.syncppp*

CONFIG\_ISDN\_AUDIO  
*Documentation/isdn/README.audio*

CONFIG\_ISDN\_TTY\_FAX  
*/usr/src/linux/Documentation/isdn/README.fax*

CONFIG\_ISDN\_X25  
*/usr/src/linux/Documentation/isdn/README.x25*

CONFIG\_ISDN\_DRV\_HISAX  
*/usr/src/linux/Documentation/isdn/README.Hisax*

CONFIG\_ISDN\_DRV\_ICN  
*Documentation/isdn/README*  
*Documentation/isdn/README.icn*

CONFIG\_ISDN\_DRV\_PCBIT  
*Documentation/isdn/README*  
*Documentation/isdn/README.pcbit*

CONFIG\_ISDN\_DRV\_SC  
*http://www.spellcast.com*  
*/usr/src/linux/Documentation/isdn/README.sc*

CONFIG\_ISDN\_DRV\_ACT2000  
*/usr/src/linux/Documentation/isdn/README.act2000*

CONFIG\_ISDN\_DRV\_EICON  
*/usr/src/linux/Documentation/isdn/README.eicon*

CONFIG\_ISDN\_CAPI  
*http://www.capi.org/*

CONFIG\_HYSDN  
*Documentation/isdn/README.hysdn*

### 16. Old CD-ROM drivers (not SCSI, not IDE)

CONFIG\_BPCD  
*Documentation/cdrom/bpcd*

CONFIG\_MCDX  
*Documentation/cdrom/mcdx*



## README- und HOWTO-Dateien



CONFIG\_ISP16\_CDI  
*Documentation/cdrom/isp16*

CONFIG\_CDU31A  
*Documentation/cdrom/cdu31a*

**17. Character devices**

CONFIG\_COMPUTONE  
*Documentation/computone.txt*

CONFIG\_DIGIEPCA  
*Documentation/digiepca.txt*

CONFIG\_ESPSERIAL  
*/usr/src/linux/Documentation/hayes-esp.txt*

CONFIG\_RISCOM8  
*Documentation/riscom8.txt*

CONFIG\_SPECIALIX  
*/usr/src/linux/Documentation/specialix.txt*

CONFIG\_SPECIALIX\_RTSCTS  
*/usr/src/linux/Documentation/specialix.txt*

CONFIG\_SX  
*Documentation/sx.txt*

CONFIG\_STALLION  
*drivers/char/README.stallion*

CONFIG\_ISTALLION  
*drivers/char/README.stallion*

CONFIG\_I2C  
*Documentation/i2c/summary*

CONFIG\_I2C\_CHARDEV  
*Documentation/i2c/dev-interface*

CONFIG\_PC110\_PAD  
*http://toy.cabi.net*

CONFIG\_JOYSTICK  
*/usr/src/linux/Documentation/joystick.txt*

CONFIG\_JOY\_ANALOG  
*/usr/src/linux/Documentation/joystick.txt*

CONFIG\_QICo2\_DYNCONF  
*include/LINUX/tpqico2.h*  
*ftp://titius.cfw.com/pub/Linux/util/*

CONFIG\_WATCHDOG  
*Documentation/watchdog.txt*



## Kernel-Parameter

---



```
CONFIG_PCWATCHDOG
 http://www.berkprod.com

CONFIG_RTC
 Documentation/rtc.txt

CONFIG_DTLK
 http://www.rcsys.com/

CONFIG_APPLICOM
 http://www.applicom-int.com/

CONFIG_FTAPE
 /usr/src/linux/drivers/char/ftape/README.PCI
 http://www-math.math.rwth-aachen.de/~LBFM/claus/ftape/
 /usr/src/linux/Documentation/ftape.txt

CONFIG_AGP_INTEL
 http://utah-glx.sourceforge.net/

CONFIG_AGP_VIA
 http://utah-glx.sourceforge.net/

CONFIG_AGP_AMD
 http://utah-glx.sourceforge.net/

CONFIG_AGP_SIS
 http://utah-glx.sourceforge.net/

CONFIG_AGP_ALI
 http://utah-glx.sourceforge.net/

CONFIG_DRM
 http://dri.sourceforge.net
```

### 18. Multimedia devices

```
CONFIG_VIDEO_DEV
 ftp://ftp.uk.linux.org/pb/linux/video4linux
 /usr/src/linux/Documentation/video4linux/API.html

CONFIG_VIDEO_BT848
 /usr/src/linux/Documentation/video4linux/bttv

CONFIG_VIDEO_CPIA
 Documentation/video4linux/README.cvia

CONFIG_VIDEO_ZR36120
 Documentation/video4linux/zr36120.txt

CONFIG_RADIO_CADET
 http://linux.blackhawke.net/cadet.html
 http://roadrunner.swansea.uk.linux.org/v4l.shtml

CONFIG_RADIO_RTRACK
 http://roadrunner.swansea.uk.linux.org/v4l.shtml
 /usr/src/linux/Documentation/video4linux/radiotrack.txt
```



## README- und HOWTO-Dateien

CONFIG\_RADIO\_RTRACK2  
<http://roadrunner.swansea.uk.linux.org/v4l.shtml>  
*/usr/src/linux/Documentation/video4linux/radiotrack.txt*

CONFIG\_RADIO\_AZTECH  
<http://roadrunner.swansea.uk.linux.org/v4l.shtml>

CONFIG\_RADIO\_GEMTEK  
<http://roadrunner.swansea.uk.linux.org/v4l.shtml>

CONFIG\_RADIO\_MIROPCM20  
<http://roadrunner.swansea.uk.linux.org/v4l.shtml>

CONFIG\_RADIO\_SF16FMI  
<http://roadrunner.swansea.uk.linux.org/v4l.shtml>

CONFIG\_RADIO\_TRUST  
<http://roadrunner.swansea.uk.linux.org/v4l.shtml>

CONFIG\_RADIO\_TYphoon  
<http://roadrunner.swansea.uk.linux.org/v4l.shtml>

CONFIG\_RADIO\_ZOLTRIX  
<http://roadrunner.swansea.uk.linux.org/v4l.shtml>

### 19. File systems

CONFIG\_AUTOFS\_FS  
<ftp://ftp.kernel.org/pub/linux/daemons/autofs>

CONFIG\_AUTOFS4\_FS  
<ftp://ftp.kernel.org/pub/linux/daemons/autofs/testing-v4>

CONFIG\_BFS\_FS  
*Documentation/filesystems/bfs.txt*

CONFIG\_UMSDOS\_FS  
*Documentation/filesystems/umsdos.txt*

CONFIG\_VFAT\_FS  
*Documentation/filesystems/vfat.txt*

CONFIG\_EFS\_FS  
<http://aeschi.ch.eu.org/efs>

CONFIG\_JFFS\_FS  
<http://www.developer.axis.com/software/jffs>

CONFIG\_CRAMFS  
*Documentation/filesystems/cramfs.txt*  
*fs/cramfs/README*

CONFIG\_NTFS\_FS  
*/usr/src/linux/Documentation/filesystems/ntfs.txt*

CONFIG\_HPFS\_FS  
*Documentation/filesystems/hpfs.txt*

## Kernel-Parameter

---



CONFIG\_DEVFS\_FS  
*Documentation/filesystems/devfs/README*

CONFIG\_DEVFS\_DEBUG  
*Documentation/filesystems/devfs/boot-options*

CONFIG\_ROMFS\_FS  
*/usr/src/linux/Documentation/filesystems/romfs.txt*

CONFIG\_SYSV\_FS  
*Documentation/filesystems/sysv-fs.txt*

CONFIG\_UDF\_FS  
*Documentation/filesystems/udf.txt*

CONFIG\_CODA\_FS  
*/usr/src/linux/Documentation/filesystems/coda.txt*  
*http://www.coda.cs.cmu.edu*

CONFIG\_ROOT\_NFS  
*Documentation/nfsroot.txt*

CONFIG\_SMB\_FS  
*Documentation/filesystems/smbfs.txt*

CONFIG\_SMB\_NLS\_DEFAULT  
*http://www.hojdpunkten.ac.se/054/samba/index.html*

CONFIG\_NCP\_FS  
*Documentation/filesystems/ncpfs.txt*

### 20. Console drivers

CONFIG\_VGA\_CONSOLE  
*ftp://metalab.unc.edu/pub/Linux/utils/console*

CONFIG\_FB  
*/usr/src/linux/Documentation/fb/framebuffer.txt*  
*http://www.tahallah.demon.co.uk/programming/prog.html*

CONFIG\_FB\_CLGEN  
*Documentation/fb/clgenfb.txt*

CONFIG\_FB\_VESA  
*/usr/src/linux/Documentation/fb/vesafb.txt*

CONFIG\_FB\_MATROX  
*/usr/src/linux/drivers/video/matroxfb.c*  
*/usr/src/linux/Documentation/fb/matroxfb.txt*

CONFIG\_FB\_MATROX\_MAVEN  
*ftp://platan.vc.cvut.cz/pub/linux/matrox-latest*

CONFIG\_FB\_ATY128  
*Documentation/fb/aty128fb.txt*





### 21. Sound

```
CONFIG_SOUND
Documentation/sound/README.modules
ftp://ftp.infradead.org/pub/pcsp/
CONFIG_SOUND_CMPCI
/Documentation/sound/CMI8338
CONFIG_SOUND_ES1370
/usr/src/linux/Documentation/sound/es1370
CONFIG_SOUND_ES1371
/usr/src/linux/Documentation/sound/es1371
CONFIG_SOUND_ESSSOLO1
/usr/src/linux/Documentation/sound/solo1
CONFIG_SOUND_MAESTRO
Documentation/sound/Maestro
CONFIG_SOUND SONICVIBES
/usr/src/linux/Documentation/sound/sonicvibes
CONFIG_SOUND_MSNDCLAS
Documentation/sound/MultiSound
CONFIG_MSNDCLAS_INIT_FILE
Documentation/sound/MultiSound
CONFIG_MSNDCLAS_PERM_FILE
Documentation/sound/MultiSound
CONFIG_SOUND_MSNDPIN
Documentation/sound/MultiSound
CONFIG_MSNDPIN_INIT_FILE
Documentation/sound/MultiSound
CONFIG_MSNDPIN_PERM_FILE
Documentation/sound/MultiSound
CONFIG_SOUND_OSS
http://www.opensound.com/oss.html
CONFIG_SOUND_AD1816
/usr/src/linux/Documentation/sound/AD1816
CONFIG_SOUND_CS4232
Documentation/sound/CS4232
CONFIG_SOUND_GUS
Documentation/sound/ultrasound
CONFIG_SOUND_MPU401
drivers/sound/Readme.cards
```



## Kernel-Parameter

---



```
CONFIG_SOUND_NM256
 /usr/src/linux/Documentation/sound/NM256

CONFIG_SOUND_PAS
 Documentation/sound/PAS16

CONFIG_SOUND_PSS
 Documentation/sound/PSS

CONFIG_PSS_MIXER
 Documentation/sound/PSS

CONFIG_SOUND_SB
 drivers/sound/Readme.cards

CONFIG_SOUND_AWE32_SYNTH
 /usr/src/linux/Documentation/sound/README.awe
 /usr/src/linux/Documentation/sound/AWE32

CONFIG_SOUND_WAVEFRONT
 Documentation/sound/Wavefront
 Documentation/sound/Tropez+

CONFIG_SOUND_YM3812
 Documentation/sound/OPL3

CONFIG_SOUND_OPL3SA1
 Documentation/sound/OPL3-SA

CONFIG_SOUND_OPL3SA2
 Documentation/sound/OPL3-SA2

CONFIG_SOUND_AEDSP16
 /usr/src/linux/drivers/sound/lowlevel/README.aedsp16
 /usr/src/linux/drivers/sound/lowlevel/aedsp16.c
 /usr/src/linux/Documentation/sound/AudioExcelDSP16
```

### 22. USB support

```
CONFIG_USB
 Documentation/usb/
 Documentation/usb/usb-help.txt

CONFIG_USB_DEVICEFS
 Documentation/usb/proc_usb_info.txt

CONFIG_USB_OHCI
 Documentation/usb/ohci.txt

CONFIG_USB_SCANNER
 Documentation/usb/scanner.txt
 Documentation/usb/scanner-hp-sane.txt

CONFIG_USB_MICROTEK
 http://fachschaft.cup.uni-muenchen.de/~neukum/scanner.html
```



## README- und HOWTO-Dateien

CONFIG\_USB\_SERIAL  
*Documentation/usb/usb-serial.txt*

CONFIG\_USB\_SERIAL\_GENERIC  
*Documentation/usb/usb-serial.txt*

CONFIG\_USB\_SERIAL\_VISOR  
*http://usbvisor.sourceforge.net*

CONFIG\_USB\_SERIAL\_FTDI\_SIO  
*http://reality.sgi.com/bryder\_wellington/ftdi\_sio*

CONFIG\_USB\_SERIAL\_KEYSPAN  
*http://www.linuxcare.com.au/hugh/keyspan.html*

CONFIG\_USB\_IBMCAM  
*Documentation/usb/ibmcam.txt*  
*http://roadrunner.swansea.uk.linux.org/v4l.shtml*

CONFIG\_USB\_OV511  
*Documentation/usb/ov511.txt*

CONFIG\_USB\_DC2XX  
*http://www.gphoto.org*  
*Documentation/usb/dc2xx.txt*

CONFIG\_USB\_MDC800  
*http://www.gphoto.org*

CONFIG\_USB\_DABUSB  
*http://dab.in.tum.de*  
*Documentation/usb/URB.txt*

CONFIG\_USB\_PEGASUS  
*Documentation/usb/rio.txt*

CONFIG\_USB\_BLUETOOTH  
*http://developer.axis.com/software/index.shtml*

CONFIG\_USB\_NET1080  
*drivers/usb/net1080.c*

CONFIG\_USB\_HID  
*Documentation/usb/input.txt*

**23. Kernel hacking**

CONFIG\_MAGIC\_SYSRQ  
*Documentation/sysrq.txt*







## **HTML-Sonderzeichen**



Die GPL 801



## **Anhang C**

## HTML-Sonderzeichen



### HTML-Sonderzeichen



Sonderzeichen	HTML	Sonderzeichen	HTML
>	&gt;	"	&quot;
<	&lt;	®	&reg;
©	&copy;	À	&Agrave;
Á	&Aacute;	Â	&Acirc;
Ä	&Auml;	Å	&Aring;
	&Aelig;	Ç	&Ccedil;
È	&Egrave;	É	&Eacute;
Ê	&Ecirc;	Ì	&lgrave;
Í	&iacute;	Î	&lcirc;
Ї	&iuml;	Ñ	&Ntilde;
Ò	&Ograve;	Ó	&Oacute;
Õ	&Otilde;	Ö	&Ouml;
Ø	&Oslash;	Ú	&Uacute;
Ù	&Ugrave;	Û	&Ucirc;
Ü	&Uuml;	ß	&szlig;
á	&aacute;	à	&agrave;
â	&acirc;	ã	&atilde;
ä	&auml;	å	&aring;
æ	&aelig;	ç	&ccedil;
è	&egrave;	é	&eacute;
ê	&ecirc;	ë	&euml;
ì	&igrave;	í	&iacute;
î	&icirc;	ï	&iuml;
ñ	&ntilde;	ó	&oacute;
ò	&ograve;	ô	&ocirc;
õ	&otilde;	ö	&ouml;
ø	&oslash;	ù	&ugrave;



## Die GPL



### HTML-Sonderzeichen (Forts.)

Sonderzeichen	HTML	Sonderzeichen	HTML
ú	&uacute;	û	&ucirc;
ü	&uuml;	ÿ	&yuml;
¡	&iexcl;	¢	&cent;
£	&pound;	¤	&curren;
¥	&yen;		&brvbar;
§	&sect;	“	&laquo;
„	&raquo;	°	&deg;
±	&plusmn;	²	&sup2;
³	&sup3;	‘	&acute;
µ	&micro;	·	&middot;
,	&cedil;	½	&frac12;
¼	&frac14;	¾	&frac34;
¿	&iquest;		

Tabelle C.1



## Die GPL

Die GPL finden Sie auf der Buch-CD.





## Index

### !

&& (bedingte Befehlsausführung) 272  
 \* (Dateinamensexpression) 245, 246  
 > (Datenumleitung) 248, 250, 252,  
 & (Hintergrundprozesse) 274  
 \ (Metazeichen) 262  
 | (Pipelines) 255  
 \$# 633  
 \$ (Gebrauch von Variablen) 279  
 \$\$ 660  
 \${...} (Befehlssubstitution) 273  
 \$o 633  
 \$? 635  
 !-Befehle (History-Mechanismus) 265  
 (...) (Befehlsverkettung) 270  
 -help (Option externer Befehle) 230  
 -version (Option externer Befehle) 230  
 .. (Befehlssubstitution) 273  
 {...} (Befehlsverkettung) 270  
 [...] (Dateinamensexpression) 246  
 "..." (Metazeichen) 262  
 .bash\_history (Datei) 265  
 .inputrc (Datei) 269  
 .profile 83  
 .profile (Datei) 224  
 /dev/lpo 180  
 /dev/null (Datei) 251  
 /dev/ttySo, Gerätedatei 566  
 /etc/auto.home (Datei) 527  
 /etc/auto.master (Datei) 526  
 /etc/auto.misc (Datei) 529  
 /etc/exports (Datei) 518  
 /etc/group (Datei) 226  
 /etc/printcap 536  
 /etc/profile (Datei) 224  
 /etc/rc.d/autofs (Skript) 526, 527  
 /etc/smb.conf 531  
 /etcpasswd (Datei) 220  
 /var/adm/utmp (Datei) 220  
 /var/adm/wtmp (Datei) 220  
 /var/spool/mail (Verzeichnis) 240  
 1TR6 570  
 2>> (Datenumleitung) 250  
 4DOS 74  
 7-Bit-ASCII 721  
 2>&1 (Datenumleitung) 251  
 \; (Befehlsverkettung) 270  
 ? (Dateinamensexpression) 246

### A

a2p (Konvertierungsprogramm) 672  
 a2ps, Konverter 195  
 Abmelden, KDE 478  
 Absätze 728  
 Acrobat Reader 399  
 Adaptec, Host-Adapter 355  
 adduser 214  
 Adresse 43  
 Adressklassen 506  
 afterstep 458  
 Aho 326  
 alias (Befehl) 276  
 Alias-Name 276  
 Alias-Tabelle 83  
 Aliase 83  
 Aliasing 276  
 Joseph H. Allen 306  
 allgemeine Befehlssyntax, eckige Klammern 101  
 amiwm 458  
 amsinck.p 519  
 angeschlossene Geräte, KDE-Kontrollzentrum 482  
 Anmeldungsmanager, KDE-Kontrollzentrum 489  
 anonymous FTP 603  
 ANSI 619  
 ANSI-Norm 736  
 Ansicht, Konqueror 474  
 Applets 623  
 apsfilter 190  
 Arbeiten mit VMware 426  
 Arbeitsfläche, KDE-Kontrollzentrum 483  
 Arbeitsweise der bash 257  
 Archivierung 432  
 Arena 589  
 Argumentwerte 632  
 Armenien 489  
 ARPANET 576  
 ASCII-Code 735  
 Assembler 618  
 at 388  
 ATAPI-CD-ROM-Laufwerke 365  
 ATAPI-IDE-Laufwerk 445  
 ATM 400  
 Audiodaten 344  
 Auflösung 731  
 Ausdrücke (bc) 338  
 Ausdrücke, bedingte 635

### Auswahl des Kernels (Installation)

Auswahllisten mit SELECT	34
Auszeichnungssprache	752
Automounter	517
Automounter Clientkonfiguration	528
Automounter-Prozess	527
Autoprobing	354
autosensing Switches	505
awk (Mustererkennung)	325
Aztech	358

---

B (Programmiersprache) 618  
 Backslash 75, 81  
 Bandlaufwerk 431  
 bash 215, 244  
 BASH\_VERSION Siehe Shellvariable  
 batch 388  
 bc (Taschenrechner) 337  
 BCPL (Programmiersprache) 618  
 bedingte Befehlsausführung 271  
 Befehlssubstitution 272  
 Befehlssyntax 264  
 Befehlsverkettungen 270  
 Befehlszeileneditor 265  
 BEGIN (awk) 331  
 Bell Laboratories 618  
 Benutzeraufforderung 74  
 Benutzereingaben in HTML-Dokumenten 751  
 Benutzerkennnummer Siehe UID  
 Benutzername 373  
 Benutzerrechte 76  
 BeOS 484  
 Berkeley Siehe University of California  
 Besitzergruppe (einer Datei) 226  
 Betriebssystem, intelligentes 91  
 Bezugsquellen (VMware) 408  
 bg (Befehl) 77, 275  
 Bilder 730  
 Bildformat 730  
 Bildschirm, Einstellungen 464  
 Bildschirm, virtueller 476  
 Bildschirmauflösung 721  
 Bildschirmschoner, KDE 484  
 Bildumrandung 731  
 bin86 (Assembler) 624



# Index

---



bind	491	chmod (Befehl)	173	Besitzergruppe	118
bind (Befehl)	268	oktale Notation	174	blockorientierte	
BIOS-Setup	421	chomp(), Perl	701, 702	Gerätedateien	121
bison (Tool)	347	chop(), Perl	701, 702	Dateibesitzer	118
BNC-	500	chown (Befehl)	177	Dateiname	117
Body	724	cifplot-Format	189	Dateityp	118
Bogo-Mips	487	clear (Befehl)	91, 233	Datum der letzten	
Boot-Vorgang	351	Client	749	Änderung	118
Bootblock	199	CLS	91	Datum der letzten	
Bootdiskette	5	Co-Prozessor	354	Änderung de	118
Bootparameter	351, 353	Coherent	204	Datum der letzten	
Bootverfahren	5	com1	566	Benutzung	118
Borland	306, 624	Common Gateway Interface	621,	Drucken	161
Bourne-Shell	243		749	elmrc	580
Brennen von CDs	343	config	396	Ersatz durch	
Broadcast-Adresse	508, 511	control statements	635	Standardein-/Auszabe in der	
Browse, Konqueror	471	Cookies, KDE-Kontrollzentrum	489	Befehlszeile	130
Browser	721	cp (Befehl)	131	Größe in Byte	118
Buffer-Underrun	346	rekursives Kopieren	132	I-Node-Nummer	117
BusLogic-SCSI	356	Sicherungsdateien	134	Jokerzeichen	125
Busmäuse	461	CP/M	74-76, 97	KDE-Kontrollzentrum	483
bwbasic (Basic-Interpreter)	624	create table, PostgreSQL	708	Kennzeichnung von	
Byte-Sex	721	cron	388	ausführbaren durch ls	119
bzImage	398	crond	393	Kennzeichnung von	
		crontab	491	Sicherungsdateien durch ls	134
				Rechtetabelle	118
				zeichenorientierte	
				Gerätedateien	121

## C

---

C (Programmiersprache)	618, 619
C Compiler	411
C-Shell	244, 285
Unterschiede zur bash	285
cal (Befehl)	91, 235
cat (Befehl)	129
Textdateien erstellen	130
cd (Befehl)	111
CD-Rohlinge	344
CD-ROM (Nutzung unter VMware)	418
CD-ROM-Laufwerk	77
CDE	484
CDPATH Siehe Shellvariable	
cdrecord	343
CDs, Brennen von	343
CGI	621, 749
CGI-Programm	756
CGI-Programmierung	748
chfn	380
chggrp (Befehl)	177
Chipsatz-Unterstützung	364
chmod (Ausführbarmachen einer Datei)	176

## D

---

date	489
date (Befehl)	233
Datei-Endemarke	632
Dateibaum	77
Dateübertragungsbefehle	607
Dateideskriptoren	
Dateien	
.	103
..	103
.bash_history	123
.profile	123
.term	123
/etc/apsfilterrc	190, 191
/etc/fstab	207
/etc/magic	105, 145
/etc/mtab	208
/etc/passwd	171
/etc/printcap	181
/etc/rc.config	182
/etc/rc.d/rc.local	181
/sbin/init.d/ldp	182
/var/lib/apsfilter/apsfilter	190
Anzahl der Links	118
Arbeiten mit Dateien	129
Dateien im Spoolbereich	
.seq	182
errs	183
lock	183
status	182
Dateiendezeichen	130
Dateiformate für apsfilter	190
Dateihandle Siehe handle	
Dateinamen	75, 89
anpassen unter SAMBA	543
erlaubte Zeichen	99
Länge	99
Minuszeichen	100
Punkt im Dateinamen	104
Dateinamensexpansion	245
Dateioperationen, Konqueror	474
Dateioperationen, Zwischenablage	
	475
Dateisysteme	
ext	204
ext2	201, 204
iso9660	204
jurnaled filesystem	205
Minix	204
NFS	204
proc	207



## Index



Reiser Filesystem	205		Ersatzdarstellung für Sonderzeichen (awk)	331
Standard-UNIX-Dateisystem	197		Erscheinungsbild,	483
umsdos	204		KDE-Kontrollzentrum	483
xiafs	204		erweiterte Partition	22
<b>Dateitypen</b>			Erziehungsministerium, mexikanisches	492
FIFO	97		Escape-Sequenzen	631
Gerätedateien	97		Ethernet	365
gewöhnliche Dateien	96		Ethernet 1000 MBit	400
Links	99		EUID Siehe Shellvariable	
Named Pipes	97		EULA	410
Sockets	98		Euro-ISDN	570
Verzeichnisse	97		ex (Editor)	297
<b>Datenbank-Handle</b>	712		exit	215
<b>Datensicherung</b>	430, 432		Expertenmenü	26
<b>Datenumleitung</b>	247		export (Befehl)	280
<b>Datum festlegen</b>	235		Exportieren (von Shellvariablen)	280
<b>Datum, KDE-Kontrollzentrum</b>	489		ext	204
<b>DBI-Modul (Perl)</b>	710		ext2	75, 90, 204
DEC	618		extended filesystem	75
Definitionsliste	740		externe Befehle	230
Dekomprimieren	87			
Design, KDE	484			
Designfehler bei Prozessoren	450			
Deutsche Industrienorm	736			
devpts	487			
df 86-86, 491				
df (Befehl)	210			
DHCP-Server	412			
Dialoge	655			
DIP-Schalter	443			
dir (Befehl)	277			
Disk-Navigator, KDE	480			
Diskettenlaufwerk (Nutzung unter VMware)	418			
Diskettenlaufwerke	360			
ditroff-Format	189			
<b>DMA-Kanäle, KDE-Kontrollzentrum</b>	486			
Dämonen	92, 393			
Domain	39			
DOS	74-86, 88-92			
DOS-Bootmenüs	448			
DOS-Emulator	398, 403			
DOSKEY	88			
Druckdämon Siehe lpd				
Drucker, die nicht postscriptfähig sind	192			
Druckerfilter	188			
Druckertreiber	188			
Druckjob	161			
Druckserver	193			
du (Befehl)	210			
DVI-Format	189			

### E

E-Mail	576
E-Mail, KDE-Kontrollzentrum	488
E-Mail-Formular	734
E/A-Adresse	351
echo (Befehl)	233, 630
ed	291
aktuelle Zeile	291
Arbeitsmodi	289
Aufruf	291
Bereich	291
Hilfe	292
Prompt	292
Suchmuster	295
verlassen	291
verlassen, ohne zu Speichern	290
ed (Editor)	289
edit (MS-DOS-Editor)	306
Editor	80
EDSS1	570
egrep (Befehl)	160
zusätzliche Sonderzeichen	160
Ein-/Ausgabe-Ports,	
KDE-Kontrollzentrum	486
Ein-/Ausgabeadresse	443
Eingabefelder in	
HTML-Dokumenten	752
Einrichten eines virtuellen PC's	
(VMware)	414
Einstellungen, Konqueror	474
Eintrag (PATH-Variable)	283
Element einer SELECT-Liste	752
elm (Programm)	579
Beantworten einer Mail	585
header	584
Menüoptionen	583
Nachrichten versenden	583
Standardeditor	581
Weiterleiten einer Nachricht	
elvis (Editor)	347
Emacs	80
emacs (editor)	305, 347
Embedded Systems	400
Emulatoren	79, 403
END (awk)	331
Energiekontrolle,	
KDE-Kontrollzentrum	483
EOF	632
eof (Tastenkombination für)	218

### F

Farbdrucker	192
Farben, KDE	484
FAT	201
fax (Tool)	347
Fehlerquellen in	
HTML-Dokumenten	745
Felder (awk)	327
Felder von /etc/printcap	186
Fenster-Eigenschaften, KDE	485
Fensterleiste, KDE	485
Festfrequenzmonitoren	463
fg (Befehl)	275
fgrep (Befehl)	160
FIFO	96
file (Befehl)	144
File Transfer Protocol	602
find	89
find (Befehl)	145
Aktionen	149
Suchkriterien	147
Suchkriterien und Aktionen	147
finger (Befehl)	220
finger (Tool)	347
FIPS	4
Fire Wire	400
Floppy-Controller	362



# Index



Fontattribute	728	GNU	326, 347, 348, 619	Hintergrund, KDE-Kontrollzentrum	483
Fontfarbe	727	GNU C++-Compiler	620	Hintergrundfarbe, KDE	483
Fontgröße	727	GNU C-Compiler	619	Hintergrundprozesse	274
for (awk)	334	gobc (Objective C-Compiler)	624	HISTCMD	88
for (bc)	341	Goldstar	359	HISTFILE	89
Formulare	749	Gopher	577	HISTFILESIZE Siehe Shellvariable	
Formularelemente	750	gpc (Pascal-Compiler)	624	history	88-89
Fortran	189	gpm	50, 87, 88	HISTSIZE	89
fortune (Befehl)	236	gpp (GNU C++-Compiler)	620	Hochgeschwindigkeitsnetze	400
Framebuffer	459	Grafikkarte	463	HOME (Shellvariable)	280
FreeMWare	404	Grafikkarte, neue	466	HOME (UmgangsvARIABLE)	223,
Fresco	624	grafische Betriebssystemaufsätze		279	
fsck (Befehl)	209	grafisches Login	51	[homes] (SAMBa)	534
fstab	90	Graphite	468	horizontaler Strich in	
FTP	577, 602	grep (Befehl)	156	HTML-Dokumenten	737
FTP-Client	471	Optionen	159	host-only networking	411
ftp-Interpreter	604	Groß- und Kleinschreibung		Host-Rechner	513
ftp-Sitzung	605	bei Umlauten	737	HOSTNAME Siehe Shellvariable	
Full Screen-Modus (VMware)	406	von Tags	722	HOSTTYPE Siehe Shellvariable	
Funktionen (bash)	277	Groß- und Kleinschreibung (bei der		HP Omnibook	361
Funktionen definieren (bc)	340	Eingabe)	215	HPFS	201
Future Domain	357	group	377	HTML	589, 720
fvwm	458	groupadd	377	HTML 3	590
fvwm95	458	groupdel	383	HTML-Sprungmarke	733
		Gruppenidentifikationsnummer	171	HTML-Umlaute	736
		gs Siehe Ghostscript		Hyperlinks	588, 732
		gzip	87	Hypertext	720
		gzip (Tool)	86, 347		

## G

g77 (FORTRAN-Compiler)	624
gas (Tool)	347
Gates, Bill	74
gawk Siehe awk	
gcc (GNU C-Compiler)	619
gcc (Tool)	347
gdb (Tool)	347
Gebrauch von Variablen	279
Gegensätze zwischen HTML- und	
Buchseiten	720
GEM	456
Geräte, KDE-Kontrollzentrum	487
Gerätedateien	96
Gerätename	353
Gerätenummern	118
gewöhnliche Dateien	96
Ghostscript	196
ghostscript (Tool)	347
ghostview (Tool)	347
GID	171, 223, 374
GIF	724
Girlscamp	485
[global] (SAMBa)	534
Globbing	636
gnome	458, 492



## H

handle	248, 711
Hard-Links	139
Hardware-Information,	
KDE-Kontrollzentrum	486
Hauptgerätenummer	118
head (Befehl)	154, 723
Heimatbereiche und Automounter	
	518
Heimatverzeichnis	223
help (Befehl)	230
Here-Dokumente	252
Herr der Ringe	481
Herunterfahren des Systems	92
heterogene Netzwerke	514
high, Kernel	398
Hilfe, KDE-Kontrollzentrum	486
Hilfe, Konqueror	475
Hilfsgerätenummer	118
Hintergrund	77

## I

I-Node	197
I-Node-Liste	198
l4L	570
ibase (bc)	339
IBM	74, 79
IBM L 40 SX	361
IBM Thinkpad	361
Icons auf dem Desktop	478
Icons auf dem Desktop, KDE	490
Icons selbst erzeugen	490
IDE/EIDE/ATAPI	363
idle-time	221
if (awk)	334
if (bc)	341
IFS Siehe Shellvariable	
Im Zeichen des Totenkopfs	484
Indexfeld, PostgreSQL	706
inetd	41, 394
Information, KDE-Kontrollzentrum	
	486
Inode-Dichte	26

## Index



insert, PostgreSQL	708	<b>J</b>	kdf	491	
Installation des Gastbetriebssystems	420	Java (Programmiersprache)	623	keirc	491
Installation via CD-ROM	13	Job	275	Kernel, Konfiguration und Komplizieren	395
Installation via FTP	14	jobs (Befehl)	275	Kernel 2.4.x	399
Installation via NFS	13	joe	80, 83, 85, 87	Kernel automounter support	517
Installation von Festplatte	14	Aufruf	306	Kernel, große	398
Installations-Hauptmenü	19	Blättern	308	Kernel-Parameter	399
Installationsassistent (Vmware)	416	Blockfunktionen	308	Kernel-Quellen	411
Installationsquellen	13	Cursorsteuerung	307	kerneld	394
int(), Perl	701	Dateihandling	310	Kernighan, Brian	95, 326, 619
integer, PostgreSQL-Datentyp	706	jmacs	306	kfirewall	491
Inter Process Communication	97	jpico	306	KFormula	468
interaktive Hilfe, KDE	471	jstar	306	kfstab	491
Internationalisierung, KDE	488	Löschen	308	kheise	491
interne Befehle	230	Onlinehilfe	307	KIllustrator	468
Internet	576	Rechtschreibhilfe	309	kimon	491
Internet per Satellit	578	rjoe	306	Kivio	468
Internet Service Provider	571	Suchen	308, 310	Klang, KDE	486
Internet Software Consortium	412	Verlassen	309	klilo	491
Internet-Wurm	347	joe (editor)	305	Klänge, KDE-Kontrollzentrum	487
interpretierte Programme	621	Jokerzeichen Siehe Dateinamensexpansion	305	klogd	394
Interpunktionszeichen in der CGI-Programmierung	762	journaled filesystem	205	kmodem	491
Interrupt	351	Joystick	491	kmysql	491
Interrupt-Nummer	443	JPEG	724	kmysqld	492
Interrupts, KDE-Kontrollzentrum	487	Jumper	443	knc	492
InterView	624	<b>K</b>	knewmail	492	
Intranet	763	K-Button	470	Kollisionen	507
Iomega ZIP-Laufwerke	431	K Desktop	468	Kommentare in HTML-Dokumenten	723
IP 514		K-Menü	470	Kommentarfeld (in /etc/passwd)	223
IP-Adressen	506	K-Office	467, 468	Komplizieren des Kernels	397
IP-Adressierung	506	Kabelmodems	578	Konfiguration (VMware)	410
IP-Version	514	Kalenderreform	236	Konfiguration des K Desktops	481
IPC	97	kam	491	Konfigurationsdatei von apsfilter	194
IPng	514	Kandalf	481	KonGo	475
IPX	552	Kandidat	467	Konqueror	471, 474, 489, 493
IRC-Chat	577	Kaos	467, 493	Konqueror, als FTP-Client	614
IRQ	351, 443	karchive	491	Konqueror, als Webbrowser	598
IRQ- und Adresskonflikte	443	Kaskadierung	642	Konqueror, Modi	473
ISC Siehe Internet Software Consortium		Katabase	468	Konsole, KDE	475
ISDN SyncPP	571	kcdwrite	491	Konsole, virtuelle	76
ISDN-Karte	566	kcmbind	491	Kontext-Menü, KDE	478
ISDN-Modem	566	kcmclock	491	Kontrollleiste, KDE	485
ISDN-Protokoll	570	kcmjoy	491	Kontrollstrukturen	635
ISO-Standard	79	kcprint	491	Kontrollzentrum, KDE	481, 482
isog66o	204, 345	krontab	491	Konvertierungsprogramme	736
ISP	571	KDE	455, 467, 492	Korn-Shell	244, 285
ispell (Tool)	348	kdesu	491	kfov	492
				KPresenter	467
				kpsqlman	492
				kpstree	492



## Index



Krash	493	logname (Befehl)	219	Rechte-Vergabe beim Erzeugen von Verzeichnissen und
Krayon	468	LOGNAME Siehe Shellvariable		
ksamba	492	logout	215	Dateien
ksms	492	logout, KDE	478	Server-Version
KSpread	467	loop-Device	530	Servername
ktelnet	492	Loopback	40	Supervisor-Login
ktranslator	488	lpc (Befehl)	164	User ID and Group ID
KWord	467	Kommandos	164	User-Logins
kworldwatch	479	lpd	181, 394	Verhalten beim Login und Datei-Operationen
		lpd (Befehl)	163	557
		lpq (befehl)	163	MARS-Server
		lpr (Befehl)	162	Martin Stover (MARtin Stover)
		lprm (Befehl)	164	Maschinennamen in /etc(exports
		ls 77, 80		519
		ls (Befehl)	116	Maskierung der Metazeichen
		Anzeige	122	262
		Option -l	120	Maus
		Optionen	119	87
		rekursive Anzeige	125	Maus, mittlere Taste
		Lynx	589	462
				Maus-Einstellungen
				461
				Mausgeschwindigkeitsmesser
				481
				Mausprokoll
				461
				Mausverhalten, KDE
				485
				MBR
				452
		mc (Midnight Commander)	92, 317	mc (Midnight Commander)
		Eingabezeile	318	92, 317
		FTP mit dem Midnight		
		Commander	325	
		Hilfs-Informationen	319	
		Kopieren und Verschieben	321	
		Menüleiste	324	
		Verlassen	320	
		Verzeichnisfenster	319	
		Viewer	320	
		Mehrbenutzersystem	76	
		Mehrfachauswahl	642	
		Mehrfrequenzwählverfahren	567	
		menuconfig	396	
		mesg (Befehl)	239	
		Metazeichen	157, 257	
		Metazeichen für grep	158	
		Mexiko	492	
		Micro-Pros	306	
		Microsoft	75, 306	
		Microsoft-Maus	461	
		Minix	204	
		minor device numberr	118	
		Mitsumi	359	
		Mitsumi-Treiber	445	
		mkdir (Befehl)	82, 113	
		mkfifo (Befehl)	98	
		mkfs (Befehl)	202	
		Dateisystemtypen	203	
		mkreiserfs (Befehl)	205	
		Mülleimer, K-Desktop	478	
		mlvwm	458	



## Index



Modem	566	NFS Server, Userspace	515	PCI, KDE-Kontrollzentrum	487
Moduldiskette	6	NFS-Client	515	PCI-Boards	361
Module	398	NFS-Dienste	42	PCMCIA, KDE-Kontrollzentrum	487
Monitor	463	NFS-Server	515	PCMCIA-Adapter	12
Monitorfrequenz	464	NFS-Server, kernel-basiert	515	PCMCIA-Kartentypen	399
more (Befehl)	151	nfsd	393, 516	PDF-Datei, Kernel-Parameter	399
more (Befehl) interaktive Benutzung	152	NIS-basiertes Mapping	521	PDP-7-Rechner	618
Mosaic	589	no-hlt	450	Peer-to-Peer-Ansatz	167
Motif	484	no387	451	Perl	672
mount	78, 90	Node	506	Anweisungen	679
mount (Befehl)	78, 90, 206, 391	nohup (Befehl)	275	Anweisungsblöcke	679
funktioniert nicht	446	Norton Commander	92, 317, 471, 492	arithmetische Operatoren	688
Mount-Parameter in /etc(exports	518	Novell	78, 551	Arrays	690
Mount-Points	24, 78	NTFS	204	Ausdrücke	679
mountd	393, 516	nummerierte Liste	739	Dekrementoperator	688
Mounten	205	O	objektorientierte Programmierung	elsif-Anweisung	693
CD-ROM-Laufwerk	90	octet	506	Ersatzdarstellungen in Zeichenketten	685
Mounting	444	Oktett	506	for-Anweisung	697
mountpoint	205	Onlinehilfe	82	foreach-Anweisung	697
mtools (DOS-Tools)	348	Operator, binär	636	formatfreie Sprache	679
Multisession-CD	345	Operator, unär	636	Funktion print()	680
Multitasking	77	Operatoren (awk)	335	Funktion printf()	680
Multitasking-Betriebssystem	76	Operatoren (bc)	338	Funktion printf() –	680
Multiuser-Betriebssystem	76	Optics Storage	360	Formatelemente	681
mv	81	Optionen in /etc(exports	519	Funktionen	701
mv (Befehl)	81, 134	OS/2	79	goto-Anweisung	698
N	logische Operatoren	692, 695			
Named Pipes	96	numerische Datentypen	687		
Kennzeichnung durch ls	119	(Zahlentypen)	683		
Nameserver	43	packet collisions	507		
Netbios Siehe SAMBA	531	Paketabhängigkeiten	32		
NetNews	587	Paketauswahl	31		
Netscape, KDE-Kontrollzentrum	489	Palo Alto	456		
Network File System	204	Panel, KDE	469, 485		
Netzknoten	506	Papierformat	192		
Netzwerk, KDE-Kontrollzentrum	488	Papierkorb, K-Desktop	478		
Netzwerke	496	Parameter	532		
Netzwerkkabel	500	Parameter (SAMBA)	537		
Netzwerkkonfiguration (Installation)	39	Partitionen, KDE-Kontrollzentrum	487		
Netzwerkunterstützung (unter VMware)	419	Partitionieren der Festplatte	15		
newgrp (Befehl)	227	passwd (Befehl)	229, 373, 375		
Newsgroups	587	Passwort	370		
Posten	587	Passwörter, KDE-Kontrollzentrum	489		
Newsreader	588	PATH Siehe Shellvariable			
NeXt	484				
NFS	204, 515				
P	686				
partitionieren der Festplatte	15	unless-Anweisung	693		
partitionieren der Festplatte	15	until-Anweisung	696		
partitionieren der Festplatte	15	Variable	683		
partitionieren der Festplatte	15	Vergleichsoperatoren	687		
partitionieren der Festplatte	15	while-Anweisung	696		
partitionieren der Festplatte	15	Zeichenkettenoperatoren	689		
partitionieren der Festplatte	15	Zeichenkettentypen	685		
partitionieren der Festplatte	15	Perl (Programmiersprache)	621		
partitionieren der Festplatte	15	Perl-Interpreter	680		
partitionieren der Festplatte	15	persönliche Einstellungen, KDE-Kontrollzentrum	488		
partitionieren der Festplatte	15	Pfadangaben in HTML-Dokumenten	744		
partitionieren der Festplatte	15	Pfade			
partitionieren der Festplatte	15	absoluter Pfad	107		
partitionieren der Festplatte	15	relativer Pfad	107		



# Index

---



Philips CM206	360		Router	512
Pi 77			Routing-Tabellen	513
PID	386		RPC-Portmapper	516
Pike, R.	95		Rückgabewerte (von Programmen)	
Pipelines	254		RAD – Rapid Application Development	711
Plattenplatz	516		RAM	354
Platzhalter für printf (awk)	329		rc.inet2	516
plot-Format	189		Rückgabewerte (von Programmen)	
Point-To-Point-Partner	571		reboot	271
POP3	594		rcs (Tool)	348
Portmapper	42		RDBMS	705
PostgreSQL	705		read (Befehl)	632
Power-Management	399		reboot	88, 92
Priebs, Holger	456		Rechte	167
primäre Partition	22		Besonderheiten – das s-Recht	
print (awk)	326		besonderheiten	177
PRINTER (Umgebungsvariable)	188		Besonderheiten – das t-Recht	
[printers] (SAMBA)	535		Ermittlung der Rechte	178
printf (awk)	328		Gruppen und Rechte	172
Pro Audio Spectrum	357		r-Recht	171
Probleme			Rechte für gewöhnliche Dateien	168
beim Floppy Boot	446		Rechte für Verzeichnisse	169
beim Starten mit LOADLIN	447		w-Recht	168
mit dem EIDE-Controller	451		x-Recht	168
mit Laptop-Diskettenlaufwerken	451		x-Recht für Verzeichnisse	170
mit SCSI-Konfigurationen	449		Red Hat	492
proc	487		Referenzieren von Variablen Siehe Gebrauch von Variablen	
PROMOCODE	412		regulärer Ausdruck	156
PROMOCODE (VMware)	409		Reiser Filesystem	205
Prompt	74, 80, 214		reiserfsck (Befehl)	210
propriétaire CD-ROMs	358		relational database management system (RDBMS)	705
Protokolle	496		Remote-Drucker	192
Provider	578		return (bc)	341
Proxies	594		Richards, Martin	618
Proxy, KDE-Kontrollzentrum	490		Richtfunk	579
Prozesse	386		Ritchie, Dennis	618
zeitversetzte	387		rm	81, 82
Prozessor, KDE-Kontrollzentrum	487		rm (Befehl)	82, 135
Prozessoren	399		Löschen von Dateien mit mehreren Links	136
PS1 Siehe Shellvariable			rekursives Löschen	136
PS2 Siehe Shellvariable			rmdir (Befehl)	82, 115
PS4 Siehe Shellvariable			Ränder, KDE	483
pwd (Befehl)	111		ro 91	
PWD Siehe Shellvariable			root	76, 83, 88, 89, 92
QIC-02	431		root squashing Siehe User ID Mapping	
QIC-80	431		root-Dateisystem	205
Queue Siehe Warteschlange			Root-Laufwerk	353
			routen	513
			Router	512
			Routing-Tabellen	513
			RPC-Portmapper	516



QIC-02	431
QIC-80	431
Queue Siehe Warteschlange	



## Index



Shell-Skripte	620	Shifting	654	Subnet-Masken	507, 508
break-Anweisung	648	shutdown	92	Subnetze	511
case-Anweisung	642	Silicon Graphics	515	Suchmaschinen	601
continue-Anweisung	650	Slackware-Setup	87	Suffixe	721
dialog-Anweisung	655	Slash	75, 81	Sun Microsystems	623
Hintergrundtitel	667	smalltalk (Tool)	348	Superblock	199
Infobox	658	SMB Siehe SAMBA	531	SuSE	492
Ja/Nein-Abfragen	659	SMB-Client, Konqueror	473	SuSE-Pakete	27
Menüs	661	smbclient	532	SuSE-Distribution	628
Messagebox	658	smbclient (SAMBA)	545	SWAP-Partition	3
Optionslisten	662	smbd	531	Sylvesternacht	91
Radiolisten	666	smbstatus	532	Symbole, KDE	485
Statusbalken anzeigen	668	smeiffel	624	symbolische Links	139-139
Texteingabe	660	SMP	399	Syquest	431
for-Anweisung	651	Sockets	96	syslogd	393
if-Anweisung	639	Solaris	484	System halted	92
Kommentare	628	Sonderzeichen (für die		System V	204
Positionsparameter	632	Shell-Variablen PS1)	284	System, KDE-Kontrollzentrum	489
Positionsparameter bearbeiten		Sonderzeichen in der		System.map	398
mit shift	654	CGI-Programmierung	761	Systemadministration	350
shift-Anweisung	654	Sony CDU-31A und CDU-33A	358	Systemadministrator	88
Speichern	628	Sound	400	Systemnachrichten, KDE	486
Starten	628	SoundBlaster 16-SCSI	355	Systemverwaltung	386
Starten mithilfe des	629	SoundBlaster Pro	360		
until-Anweisung	650	Soundkarten und LILO	447		
Variablenexport	642	Speicher, KDE-Kontrollzentrum	487		
Verzweigungen	635, 639	speicherresidente Programme	447		
while-Anweisung	645	Spool	161		
Shellvariable	278	Spoolbereich	180	Tabellen	741
BASH_VERSION	281	Sprache, KDE-Kontrollzentrum	488	Tag-Farbe	734
CDPATH	280	SQL	621, 705	Tags	722
EUID	281	sqrt() (bc)	338	tail (Befehl)	154
exportieren	280	sqrt(), Perl	701	Talk-Einrichtung,	
Gebrauch von	279	Standard-Shell	223	KDE-Kontrollzentrum	488
HISTFILE	281	Standard-UNIX-Dateisystem,		tape archive,	432
HISTFILESIZE	281	Einschränkungen	201	tar	431-433, 437-439, 491
HISTSIZE	281	Standardqueue	188	tar (Tool)	348
HOME	280	Start-Menü	470	tar-Datei, Konqueror	475
HOSTNAME	281	START_AUTOFS		Taskbar, KDE	485
HOSTTYPE	281	(Umgebungsvariable)	524	Taskleiste, KDE	477
IFS	280	Statement-Handle	712	Tastatur	462
LOGNAME	281	statischer Mount	517	Tastaturlayout (Installation)	9
MAIL	281	statisches Mapping	521	Tastaturlayout,	
MAILCHECK	281	Statusmeldungen	85	KDE-Kontrollzentrum	489
PATH	280	Steueranweisungen	635	Tastenzuordnung, KDE	486
PS1	281	sticky bit	178	Tcl	622
PS2	281	Sticky Button, KDE	477	Tcl/Tk	622
PS4	281	Structured Query Language (SQL)		tclsh	622
PWD	281		621, 705	tcsh (Shell)	285
Referenzieren	279	Strukturelemente von		tee (Befehl für Pipelines)	256
SHELL	281	HTML-Dokumenten	722	Teekocher	481
TERM	281	stty (Befehl)	217	Telefonanlage	567, 570
TZ	281	su (Befehl)	228	Telefonieren über das Internet	577
UID	281	Sub-Shell	270	Telefonieren über IP	400

## T

Tabellen	741
Tag-Farbe	734
Tags	722
tail (Befehl)	154
Talk-Einrichtung,	
KDE-Kontrollzentrum	488
tape archive,	432
tar	431-433, 437-439, 491
tar (Tool)	348
tar-Datei, Konqueror	475
Taskbar, KDE	485
Taskleiste, KDE	477
Tastatur	462
Tastaturlayout (Installation)	9
Tastaturlayout,	
KDE-Kontrollzentrum	489
Tastenzuordnung, KDE	486
Tcl	622
Tcl/Tk	622
tclsh	622
tcsh (Shell)	285
tee (Befehl für Pipelines)	256
Teekocher	481
Telefonanlage	567, 570
Telefonieren über das Internet	577
Telefonieren über IP	400



# Index



TELNET	577, 615	Universal Ressource Locator	731	/usr/src	109
TELNET-Protokoll	615	University of California	244	/usr/src/linux	109
TERM Siehe Shellvariable		UNIX	74-76, 78-81, 83, 86, 88-90	/var	108
test (Befehl)	635	Unterinhaltsverzeichnis	75	/var/spool	110
testparm	531, 544	update, PostgreSQL	709	/var/spool/cron	110
textorientierte Browser	731	URL	724, 731	/var/spool/cron/crontabs	110
Thin-Ethernet	497, 503	URL-Kennungen	732	/var/spool/mail	110
Thompson, Ken	618	URL-Leiste, Konqueror	473	/var/spool/spool	110
Thumbnail	731	USB	399	/var/tmp	109
Timing-Probleme beim		USENET	594	Arbeiten mit Verzeichnissen	110
CD-ROM/RAM-Zugriff	445	Newsgroups	577	Elternverzeichnis	112
tn3270 (Programm)	577	Usenet-News	587	Kennzeichnung durch ls	112, 113
Toaster	400	User ID Mapping	520	Wurzelverzeichnis	107
Tonwählverfahren	567	User-Account	370	Vesa	459
top	91	User-Agent, Konqueror	490	VFS	201
Torvalds, Linus	487	User-Identifikationsnummer	171	vi	80
touch (Befehl)	183	useradd	371	Arbeitsmodi	297-297
Trantor T128	357	userdel	382	aufrufen	296
Treiber-Handle	712	usermod	378	Befehle zum Dateihandling	304
Trennzeichen bei der Eingabe	216	usermod (Befehl)	285	Bewegen im Text	300
Trennzeichen festlegen (awk)	333	users (Benutzergruppe)	226	Löschen und Kopieren	301
troff-Format	189	Userverwaltung	370	Objekte	299
Troubleshooting	442	uucp (Tool)	348	Puffer	302
true (Befehl)	647			Suchen und Ersetzen	302
Turbo C	306			Suchmuster	303
Turbo Pascal	306			TERM-Variable anpassen	295
Twisted-Pair	504			verlassen	297
type ahead	216			vi (Editor)	295
TZ Siehe Shellvariable				Video for Linux	400
<hr/>					
U				Vietnam	489
Überschriften	726	Variablensubstitution (SAMBA)	536	Virtual Disk	417
ugidd (Daemon)	521	vbox	491	Virtual Filesystem Switch	201
ugidd-basiertes Mapping	521	Verbindungsbefehle	606	Virtual Mode	404
UID	171, 223, 372, 374	Verschlüsselung,		Virtuel Machine Monitor	414
UID Siehe Shellvariable		KDE-Kontrollzentrum	489	virtuelle Arbeitsfläche, KDE	484
umask (Befehl)	175	Versenden einer Nachricht Siehe		virtueller PC	405
Rechte Maske und das		mail		vmlinuz	398
Kopieren von Dateien	175	Verweise (Links)	96	vmtools	406, 413
umask und das x-Recht für		Verzeichnisbefehle	606	VMware	404
gewöhnliche Dateien	176	Verzeichnisse	96	von HTML unterstützte	
Umgebungsvariablen	88	.elm	580	Rechnerwelten	721
in der CGI-Programmierung	753	/	107	vordefinierte Variable (awk)	333
Umlaute und Sonderzeichen	735	/bin	107	Vorschaubild	731
umount	90	/dev	98, 108	<hr/>	
umount (Befehl)	78, 90, 207, 391	/etc	107	W	
umsdos	204	/home	109	w (Befehl)	220
Unesco	492	/lib	108	W3	577
ungeordnete Liste	738	/mnt	108, 206	Wahl der Zeitzone	38
unique, PostgreSQL	706	/proc	109	Waitstates	445, 453
Universal Ressource Loader	724	/sbin	108	wall (Befehl)	239
		/usr	109	Wall, Larry	672



## Index



Warteschlange	180	wvdial	566
Warteschlangen von apsfilter	190	WWW	577
wc (Befehl)	155	WWW-Adressen	595
Web-Browser	589	WWW-Server	596
Web-Browser, KDE-Kontrollzentrum	489-489	www.priebs.de	456
Wechselplatten	431, 432	www.xfree.org	466
Weinberger	326		
Weltzeituhr	479		
whatis	82		
WHEREIS	89		
whitespace character	532	X Window	457
who (Befehl)	218	X-Konfiguration	455
whoami	88	X-Server	457, 459
Wiederholungsanweisungen	635	X-Server, accelerated	457
Win 95	79	X-Server, KDE-Kontrollzentrum	487
Window-Manager	457, 458	X-Server, killen	465
Windows 3.1	91	Xdroast	343
Windows-Arbeitsgruppen	473	xconfig	397
Windows-Ressourcen, KDE-Kontrollzentrum	488	Xenix	204
WINE	403	Xerox-Laboratories	456
WO	89	XF68_FBDev	459
WordStar	306	xf86config	459, 466
World Wide Web	577, 588	XFree86	457, 459
World Wide Web Consortium	598	xftp	610
write (Befehl)	237	xiafs	204
		XT-Festplatte	366



Zeichenketten-Manipulation	621
Zeilenende, UNIX	86
Zeilenschaltungen	728
Zeilenumbruch	729
Zeit, KDE-Kontrollzentrum	489
zentrieren in HTML-Dokumenten	737
zImage	398
zilo	398
zsh (Shell)	285
Zugangserleichterungen, KDE-Kontrollzentrum	489



# **8 • Konfiguration und Kompilieren des Kernels**

Im folgenden Abschnitt möchten wir Ihnen die für die Neu-Konfiguration und -Kompilierung des Kernels notwendigen Schritte und die Optionen, die hierfür zur Verfügung stehen, vorstellen. Dieses zugegebenermaßen etwas trockene

Kapitel werden Sie sicherlich nicht von vorne bis hinten studieren. Sie werden aber, wenn Sie aus einem der im folgenden aufgeführten Gründe einmal in die Situation kommen, Ihren Kernel re-konfigurieren zu wollen (oder gar zu müssen ...), die hier enthaltenen Informationen als eine unersetzliche Fundgrube an Informationen schätzen lernen. Gründe für die Neu-Konfiguration des Kernels Die verschiedenen mitgelieferten Kernel reichen in den meisten Fällen aus, ein LINUX-System zu installieren. Wer jedoch außer Standardkomponenten weitere Hardware im Rechner benutzen will, wie z.B. Netzwerkkarten, Soundkarten, Bandlaufwerke oder ISDN-Karten, der muß die entsprechenden Treiber in seinen Kernel einbinden. Es wird eine riesige Auswahl an Treibern geboten, so daß der Kernel auf die individuellen Hardware-Anforderungen genau maßgeschneidert werden kann. Dieses kann ein vorkonfigurierter Kernel der CD-ROM niemals leisten. Die Situation innerhalb der vielfältigen Software-Umgebungen, in denen LINUX sich befinden kann, ist ähnlich: Auf vielen Rechnern sollen sich neben LINUX auch noch DOS, Windows 9x, NT oder OS/2 befinden und die entsprechenden Festplatten sollen unterstützt werden. Der LINUX-Rechner kann sich innerhalb eines NOVELL- oder APPLE-Netzes befinden und dort als Client oder Server arbeiten. Auch kann der LINUX-Rechner als Firewall oder Server im WWW (siehe Kapitel 13) genutzt werden. Diese Fähigkeiten erlangt LINUX aber erst, nachdem ein Kernel neu konfiguriert und danach kompiliert wird.

## Konfiguration

Es gibt drei Programme, die der Konfiguration des Kernels dienen:

`config`: zeigt Ihnen stur einen Kernel-Parameter mit Kurzbeschreibung nach dem anderen an, wobei schlicht mit `y` oder `n` geantwortet wird. Oft kann auch mit `m` geantwortet werden: der Treiber wird dann als Modul geladen, d.h., er wird nicht fest im Kernel verankert, sondern wird bei Bedarf automatisch hinzugeladen bzw. nach einer bestimmten Zeit ohne Nutzung automatisch wieder entfernt. Module eignen sich gut für bestimmte Bestandteile, die man nicht missen möchte, die aber nicht permanent benötigt werden. `config` eignet sich

zum Neuaufbau eines Kernels und für den Anfänger: alle Möglichkeiten werden angeboten, es kann nichts vergessen werden.

`menuconfig`: bietet in den bekannten semi-grafischen Fenstern (wie z.B. in `yast`) menügesteuert die gleichen Optionen. Die jeweilige Wahl wird angezeigt und kann jederzeit geändert werden, fehlerhafte Eingaben können im Gegensatz zu `config` überschrieben werden. Verschiedene Konfigurationsdateien können geladen werden, so daß verschiedene Kernel konfiguriert werden können. `menuconfig` eignet sich zum einen, wenn nur geringe Änderungen im Kernel vorgenommen werden sollen. Zum anderen wendet es sich an den erfahrenen Benutzer, da es die Abfrage aller Parameter nicht erzwingt. Falls Anfänger mit diesem Programm arbeiten, sollten sie alle Menüpunkte nacheinander abarbeiten, so daß nichts vergessen wird.

`xconfig`: arbeitet unter X Window und leistet das gleiche wie `menuconfig` unter der bekannten grafischen Oberfläche. Alle Aktionen, die die Konfiguration und die Kompilierung des Kernels betreffen, finden in dem Verzeichnis  
`/usr/src/linux`

statt, in das man als erstes wechseln sollte. Dort muß dann eingegeben werden

```
make config
oder
make menuconfig
```

Informationen zu den einzelnen Kernel-Parametern finden Sie in aller Ausführlichkeit auf den nächsten Seiten. Nachdem man sich durch die einzelnen Fragen bzw. Menüpunkte „durchgetastet“ hat, werden die ausgewählten Parameter in die Datei `.config` geschrieben, die zur Steuerung der Kompilierung dient.

## Das Kompilieren

Das Kompilieren des Kernels geschieht in drei Schritten: zuerst werden alle alten temporären Dateien des letzten Kompilationslaufs gelöscht:

```
make clean
```

Als Nächstes werden die einzelnen Abhängigkeiten neu berechnet:

```
make dep
```

Diese beiden Schritte benötigen jeweils nur kurze Zeit. Der eigentliche Vorgang des Kompilierens dauert länger und wird wie folgt gestartet:

```
make zImage
```

Hier kann man sich nun freuen, wenn man neuere Hardware besitzt: Das Kompilieren eines Kernels auf einem Rechner mit neuerem Pentium dauert nur wenige Minuten, auf einem älteren 100-MHz-Pentium-II immerhin schon ca. 10 Minuten. Auf einem 486er mit 50 MHz benötigten wir seinerzeit ca. 35 Minuten. Von 386ern mit wenig RAM wird von stundenlangen Läufen berichtet, was wir aber (glücklicherweise) nicht überprüft haben. Um die Abarbeitung jedes dieser einzelnen Schritte nicht überwachen und den jeweils nächsten Schritt „manuell“ einleiten zu müssen, können Sie diese drei Befehlszeilen, wie in Kapitel 4 (Job-Control) beschrieben, auch wie folgt aufrufen:

```
make clean && make dep && make zImage
```

Selbstverständlich können Sie aber auch (statt einen Kaffee trinken zu gehen, während diese drei Schritte abgearbeitet werden) auf einer anderen virtuellen Konsole weiterarbeiten. Auf jeden Fall sollten diese Hinweise genügen, um bei der Konfiguration des Kernels sorgfältig vorzugehen, damit dann nach dem Kompilieren nicht festgestellt werden muß, daß man etwas vergessen hat. Nach dem Vorgang findet sich der neue Kernel unter dem Namen `zImage` in dem Verzeichnis `/usr/src/linux/arch/i386/boot`. Dieses ist aber nur die „Standardablage“, nach dem der Kernel neu erstellt wurde. Falls man mit LILO arbeitet, liegt der Kernel normalerweise im Verzeichnis `/boot` unter

dem Namen `vmlinuz`. Wenn mit `LOADLIN` gearbeitet wird, liegt der Kernel in dem entsprechenden DOS-Verzeichnis, also z. B. `C:\LOADLIN`. In beiden Fällen sollte zuerst der alte Kernel umbenannt werden, denn es besteht immer die Möglichkeit, daß der neue nicht laufen will. Erst dann sollte das gerade erstellte `zImage` in das gewünschte Verzeichnis (also in `/boot` bzw. in das Verzeichnis `LOADLIN`) kopiert und evtl. umbenannt werden. Wenn Sie mit `LILO` arbeiten, ist noch ein sehr wichtiger Schritt durchzuführen: Der `LILO` muß durch den Aufruf des Befehls `make zlilo` neu installiert werden. Unter bestimmten Umständen, die auf Grund der Vielzahl der Kernel-Parameter, die Auswirkungen auf den Boot-Vorgang haben und die daher für eine genaue Analyse und somit Darstellung an dieser Stelle zu unspezifisch sind, startet das Betriebssystem gegebenenfalls nicht mehr.

## Kernel mit Modulen

Nach Überprüfung der Lauffähigkeit des neuen Kernels können Sie, falls Bestandteile des Kernels als Modul mit der Option `m` in `config` bzw. `menuconfig` definiert worden sind, jetzt noch diese Module erzeugen bzw. installieren. Dieses wird mit den folgenden beiden Befehlszeilen erledigt:

```
make modules
make modules_install
```

Der DOS-Emulator und einige andere Module benötigen die Datei `System.map` im Wurzelverzeichnis, die sich nach dem Kompilieren im Verzeichnis `/usr/src/linux` befindet:

```
cp /usr/src/linux/System.map /
```

Jetzt sollte sich der neue Kernel mittels `LILO` oder `LOADLIN` starten lassen.

## Große Kernel

Wer - aus welchen Gründen auch immer - mit Modulen nicht arbeiten will oder sehr viele Treiber in seinem Kernel benötigt, läuft Gefahr, daß das Kompilieren nicht regulär beendet wird, sondern sich das System mit einer Meldung wie etwa `System is too big` oder `Kernel too big` meldet und den Vorgang abbricht. Hier ist Abhilfe geschaffen worden: statt `make zImage` gibt man ein:

```
make bzImage
```

Die anderen Vorgänge bleiben unverändert. Durch `make bzImage` erhält der Kernel u.a. auch eine Kennung, daß er `high` geladen wird.

## KERNEL-PARAMETER

Sprache der Menüführung Language of Kernel Configuration / Sprache der Kernel Konfiguration (`CONFIG_CONFIGLANG_ENGLISH`)

Hier kann der des englischen ungewohnte Benutzer sich die Abfragen bzw. Menüpunkte sich inzwischen auch in deutscher Sprache anzeigen lassen. Zubeachten ist, daß zwar die Hilfstexte, die mittels der Taste ? angezeigt werden können, sofort in der gewünschten Sprache erscheinen, die Menüführung allerdings erst in der gewählten Sprache erscheint, nachdem das Konfigurationsprogramm verlassen und danach erneut gestartet wird. Diese Option ist nicht bei allen Kernel-Versionen vorhanden und fehlt z.B. bei allen Entwickler-Kernen, weshalb wir uns in diesem Kapitel für die Übernahme der englischen Bezeichnungen entschlossen haben.

## 1. Experimentelle Konfiguration

Prompt for development and/or incomplete code/drivers  
(`CONFIG_EXPERIMENTAL`)

Neben vielen anderen Konfigurationsmöglichkeiten von LINUX werden auch experimentelle Treiber angeboten, die sich in einem Entwicklungsstadium befinden, deren Funktionalität oder Stabilität noch nicht den allgemeinen Ansprüchen genügt. Hierbei kann es sich z. B. um neue Netzwerkprotokolle oder Dateisysteme, aber auch um neue Hardware handeln. Diese Phase wird von Entwicklern „Alpha-Test“ genannt. Wer also an einem stabilen System interessiert ist, sollte hier `n` antworten. Wer an den experimentellen Treibern interessiert ist, sollte etwaige Fehler protokollieren und den Entwicklern zukommen lassen. Wir weisen im weiteren darauf hin, welche Treiber in diesem Stadium sind und auch nur von (`menu`) `config` abgefragt werden, wenn `CONFIG_EXPERIMENTAL` gesetzt ist.

## 2. Module

Enableloadable module support (`CONFIG_MODULES`)

Kernel-Module können zur Laufzeit hinzugeladen und auch wieder entfernt werden, so daß sie sich für zeitlich begrenzte Aufgaben anbieten. Typische Beispiele sind z. B. die ISDN-Anbindung oder das CD-ROM-Filesystem. Durch Module bleibt der Kernel weiterhin relativ klein. Wenn diese Option genutzt wird, gibt es bei vielen der weiteren Abfragen nicht nur die Möglichkeiten `y` oder `n`, sondern auch `m` für die modulare Anbindung des jeweiligen Pakets.

Set Version information on all symbols for modules (`CONFIG_MODVERSIONS`)

Normalerweise müssen alle Module neu kompiliert werden, wenn man eine neue Kernel-Version installiert. Das Setzen dieser Option stellt dieses sicher. Da bei den jeweiligen Distributionen die jeweils zusammengehörenden Module geliefert werden, ist dieses nur bei Experimenten mit dem Kernel notwendig. Für das Erstellen eines neuen Kernels ist dann allerdings das Programm genksyms erforderlich, das als anonymous FTP bei <ftp://sunsite.unc.edu/pub/LINUX/kernel> neben der kompletten Modul-Unterstützung vorhanden ist. Normalerweise ist diese Option also nicht notwendig.

#### Kernel module loader (CONFIG\_KMOD)

Normalerweise obliegt es dem Benutzer die Module, die während des Kompilierens erzeugten Module von Hand mittels insmod zu laden, bevor diese benutzt werden können. Das ist natürlich extrem unpraktisch und deshalb ist der Vorgang automatisiert worden. Bisher war dafür der Daemon kerneld zuständig. Diese Option ist der aktuelle Ersatz für den Kernel-Daemon. Hier sollte also bei jeder Installation mit y antwortet werden. Wer sich in die Details vertiefen möchte, liest unter Documentation/kmod.txt nach.

## 3. Prozessoren

Unter dem Menüpunkt Processor type and features können der Prozessor-Type sowie einige zugehörige Spezialitäten eingestellt werden. Erwähnenswert ist, daß ab Kernel 2.4.x der gute alte 386er nicht mehr zu den Standardeinstellungen zählt.

#### (Pentium-III) Processor family (CONFIG\_M386)

Linux kann für die verschiedenen Prozessoren des PC-Prozessors optimiert werden. Die möglichen Werte sind folgende: 486, 586/K5/5x86/6x86, 6x86MX, Pentium-Classic, Pentium-MMX, Pentium-Pro/Celeron/Pentium-II und Pentium-III.

Wer sich bezüglich seines Prozessors nicht sicher ist, der sollte das Handbuch seines Rechners bzw. seines Mutterbretts zu Rate ziehen. Falls sich etwaige Unsicherheiten nicht ausräumen lassen, so gilt, daß die Wahl einer niedrigeren Prozessorklasse jedenfalls den Betrieb von LINUX gewährt; die neuen Leistungsmerkmale der neueren Prozessoren werden aber natürlich nicht ausgenutzt, so daß LINUX evtl. nicht so schnell läuft, wie es könnte. Auffällig ist, daß auch weiterhin die alten 486er-Prozessoren unterstützt werden, so daß man seine alte Hardware z.B. als Print-, Fax oder Mail-Server in seinem Netz betreiben kann.

Toshiba Laptop support (CONFIG\_TOSHIBA)

Unterstützung der CPUs der Toshiba Laptops.

/dev/cpu/microcode - Intel P6 CPU microcode support  
(CONFIG\_MICROCODE)

Durch diese Option für der Microcode der Intel-P6-Prozessoren ein Update erhalten. Dieses betrifft die CPUs Pentium Pro, Pentium II, Pentium III und Xeon. Dazu muß außerdem die Option /dev file system support aktiviert sein. Der Prozessor-Microcode ist natürlich keine Bestandteil des Linux-Kernels. Die neuesten Informationen bezüglich dieser Option erhält man unter <http://www.urbanmyth.org/microcode/>.

/dev/cpu/\*/msr - Model-specific register support (CONFIG\_X86\_MSR)

Diese Option wird nur unter bestimmten Umständen bei einem Multi-Prozessor-Rechner benötigt. Hierdurch kann bei jeweils einer bestimmten CPU auf die Modell-Spezifischen Register (MSR) privilegiert zugegriffen werden. Dieses Gerät besitzt die Hauptgerätenummer 202 und die Nebennummer von 0 bis 31 für /dev/cpu/0/msr bis /dev/cpu/31/msr.

/dev/cpu/\*/cpuid - CPU information support (CONFIG\_X86\_CPUID)

Dieses Gerät ermöglicht Prozessen direkten Zugriff zu der Anweisung CPUID der X86-Prozessor-Familie. Durch diesen Zugriff können definierte Prozesse auf einem spezifizierten Prozessor ausgeführt werden. Dieses Gerät besitzt die Hauptgerätenummer 203 und die Nebennummern von 0 bis 31 für /dev/cpu/0/cpuid bis /dev/cpu/31/cpuid.

#### High Memory Support (`CONFIG_NOHIGHMEM`)

Linux kann inzwischen auf einem X86-System maximal 64 Gigabyte Arbeitsspeicher verwalten. Die derzeitigen 32-Bit-CPUs können derzeit allerdings nur maximal 4 Gigabyte Arbeitsspeicher verwalten. Falls also mehr als 4 Gb Arbeitsspeicher auf einem Rechner vorhanden sind, kann nicht der gesamte Speicher permanent gemappt sein. Der Speicher, der über dieser Grenze von 4 Gb liegt, wird **High Memory** genannt. Alte DOS-Künstler erinnern sich dunkel. Dort lag die der hochgeladene Speicher allerdings zwischen 640 Kb und 1 Mb.

Das Menü bietet die drei Optionen `Off`, `4 GB` und `64 GB` an. Die Standard-Einstellung ist `Off`, denn dieses ist ausreichend für alle Rechner die 1 GB Speicher oder weniger besitzen - im Moment wohl eher der Normalfall. Der Prozessor verwaltet dann maximal 1 GB Arbeitsspeicher direkt und 3 weitere GB als virtuellen Arbeitsspeicher.

Falls der Rechner mehr als 1 GB bis zu 4 GB Arbeitsspeicher besitzt, so kommt die zweite Option, nämlich `4 GB` zum Einsatz, der gesamte Arbeitsspeicher wird gemappt.

Wer mehr als 4 GB bis maximal 64 GB Arbeitsspeicher auf seinem System besitzt (ihr glücklichen Menschen ...), der aktiviert die dritte Option `64 GB`. Dadurch wird ein Leistungsmerkmal der neueren Intel-Prozessoren aktiviert, das sich PAE (Physical Address Extension) nennt. PAE ist auf Pentium Pro und höheren Intel-Prozessoren vorhanden. PAE stellte ein Paging mittels eines Drei-Level-Mechanismus zur Verfügung, der komplett von Linux unterstützt wird.

Außerdem sei darauf hingewiesen, daß es durchaus noch verschiedene BIOS gibt, die den vorhandenen Speicher nicht richtig melden. Uns sind mehrere BIOS noch von 1997 bekannt, die nicht mehr als 64 MB melden können. Hier muß Linux beim Start angegeben werden, wieviel Speicher das System besitzt, da ansonsten nur 64 MB benutzt werden. Für z.B. 128 MB Arbeitsspeicher lautet der Bootparameter

`mem=128M`

Dieser muß an LILO oder LOADLIN übergeben werden.

Achtung: werden Prozessoren, die keine PAE besitzen, mit dieser Option angesteuert, so bootet der Kernel nicht!

#### MTRR control and configuration (CONFIG\_MTRR)

Wer erstens einen Pentium Pro, einen Pentium II oder einen besseren Prozessor von Intel besitzt und zweitens eine Videokarte auf einem AGP- oder PCI-Bus besitzt, kann mit dieser Option die Transferraten zur Videokarte bis um 250 Prozent beschleunigen. Die Memory Type Range Register (MTRR) dieser Prozessoren werden benutzt, um größere Speicherbereiche in die Videokarte zu laden, bevor sie auf den Bus gesendet werden. Die Cyrix-Prozessoren 6x86, 6x86MX und M II besitzen Address Range Register (ARR), die dieselbe Funktionalität besitzen wie die MTRR. Die ARR werden von diesem Treiber benutzt, um die MTRR zu emulieren. Die AMD-Prozessoren K6-2 (Step 8 und höher) und K6-3 besitzen zwei MTRR, die ebenfalls unterstützt werden. Außerdem behebt diese Option die Fehler von einigen BIOSen, die bei Multiprozessormaschinen eingesetzt werden. Diese setzen nur die MTR-Register der ersten, also der Boot-CPU, aber nicht der weiteren Prozessoren, was zu verschiedensten Fehlern führen kann. Dieser Treiber behebt das Problem. Falls nicht bekannt ist, ob der vorliegende Prozessor MTR-Register besitzt, so kann diese Option bedenkenlos aktiviert werden: falls dieses nicht der Fall ist, so wird dieser Treiber automatisch nicht benutzt, der allerdings nur 3 Kilobyte groß ist. Weitere Informationen erhält man in `/usr/src/linux/Documentation/mtrr.txt`.

#### Symmetric Multi Processing (CONFIG\_SMP)

Falls man stolzer Besitzer eines Rechners mit mehr als einer CPU ist, so sollte man hier mit `y` antworten. Dieses ist derzeit allerdings meistens nur der Fall bei High-End-Servern. Hierbei sind einige Dinge zu beachten: Ein Mutil-Prozessor-Kernel kann evtl. auch auf Single-Prozessor-Rechner laufen; dieses ist aber nicht sichergestellt. Falls ein Multi-Prozessor-Kernel auf einem Single-Prozessor-Board funktioniert, so läuft er langsamer, als ein Single-Prozessor-Kernel. Falls man ein Multi-Prozessor-Board besitzt, es aber mit einem Single-Prozessor-Kernel betreibt, so wird nur die erste CPU genutzt; die weiteren CPUs liegen brach. Bei Multi-Prozessor-Kernel ist die exakte Auswahl des Prozessor-Typs (s.o.) von großer Bedeutung, im Gegensatz zum Single-Prozessor-Kernel. Für Multi-Prozessor-Kernel sollte die Option `Enhanced Real Time Clock Support` aktiviert werden. Durch diese Option wird der Treiber für das Advanced Power Management automatisch deaktiviert. Weitere Hilfe bieten in `/usr/src/linux/` die Dateien `Documentation/SMP.txt`, `Documentation/smp.txt` und `Documentation/IO-APIC.txt`.

APIC and IO-APIC support on uniprocessors (NEW)  
(`CONFIG_X86_UP_IOAPIC`)

APIC (Advanced Programmable Interrupt Controller) ist ein Mechanismus, der die Hardware-Interrupts direkt an die CPU weiterleitet. Wer einen Einzel-Prozessor-Rechner mit APIC besitzt, kann dieses Leistungsmerkmal aktivieren. Falls der Rechner kein APIC besitzt, so ist dieses trotzdem unbedenklich, da keinerlei Verzögerungen eintreten. Wer ein Mehrprozessor-System benutzt, muß diese Option nicht extra aktivieren, denn dieses geschieht automatisch durch den vorangegangenen Menüpunkt.

## 4. Allgemeine Einstellungen

Networking support (`CONFIG_NET`)

Falls man nicht genau weiß, was man hier tut, sollte man `y` antworten, da einige Programme diese Option benötigen, auch wenn der Rechner ansonsten völlig allein, ohne Netzwerkanbindung betrieben wird. Der Druckerspooler benutzt z. B. das Netzwerk im sogenannten Loop-Back-Modus. Die einzelnen Netzwerk-Optionen und -Karten werden in den Sektionen 10 bzw. 16 behandelt.

#### **SGI Visual Workstation support (CONFIG\_VISWS)**

Wer glücklicher Besitzer einer SGI Visual Workstation vom Typ 320 oder 540 ist, der muß diese Option aktivieren. SGI-Kernel und Standard-PC-Kernel sind **nicht austauschbar**. Weitere Informationen enthält `usr/src/linux/Documentation/sgi-visws.txt`.

#### **PCI bios support (CONFIG\_PCI)**

Falls ein PCI-Bus vorhanden ist, erlaubt diese Option die Nutzung von 32-Bit-BIOS-Aufrufen. Einige ältere Systeme haben Fehler im BIOS und arbeiten nicht korrekt mit `CONFIG_PCI`, sehr wohl aber, wenn diese Option ausgeschaltet ist. Falls Probleme auftauchen, gibt es via anonymous FTP eine Positiv-/Negativ-Liste von Rechnern und dieser Option:  
`ftp://sunsite.unc.edu/pub/LINUX/docs/HOWTO`.

#### **PCI access mode (CONFIG\_PCI\_GOBIOS)**

Wie bei jeder Neuerung innerhalb des komplett veralteten PC-Design, schnittzt erstmals jeder Hersteller seine eigene Quasi-Norm - sei es, daß die Spezifikationen nicht ausgereift sind, sei es, daß sie aus Kostengründen nicht genügend beachtet werden, oder sei es, daß die Entwickler nicht genügend nachgedacht haben. So war es auch bei der Einführung von PCI und so gibt es einige frühe Motherboards, die fehlerhaft sind und ein automatisches Erkennen von Peripherie-Geräten nicht schaffen. Hier gibt es drei verschiedene Zugriffsmöglichkeiten für den LINUX-Kernel:

- BIOS : Das BIOS ist **nicht** fehlerhaft; LINUX kann sich auf die Informationen des BIOS verlassen und liest nur diese aus.
- Direct: Der LINUX-Kernel spricht die Geräte direkt an und verläßt sich nicht auf das BIOS. Hierzu müssen evtl. einige Boot-Parameter für Peripherie-Geräte gesetzt werden.
- Any: Hier versucht LINUX zuerst den direkten Zugriff auf Geräte; wenn dieser fehlschlägt, zieht es die BIOS-Werte zu Rate.

Die dritte Option ist natürlich der sicherste Weg, unter Umständen aber auch langsamer, als die beiden vorangehenden. Wer allerdings keine genauen Informationen über sein BIOS besitzt, dem sei dieser Weg empfohlen.

#### PCI device name database (CONFIG\_PCI\_NAMES)

Normalerweise enthält der Kernel eine kleine Datenbank mit den Namen aller bekannter PCI-Geräte um die entsprechenden Informationen in `/proc/pci`, `/proc/ioports` und ähnlichen Dateien im virtuellen Verzeichnis zu erzeugen. Die Kernelgröße wächst hierdurch um 80 KB, was prinzipiell aber kein Problem darstellt, da dieser Speicherplatz nach dem Booten wieder freigegeben wird, da die Daten nicht mehr benötigt werden. Unter speziellen Umständen können diese 80 KB aber dringend benötigt werden. Wer z.B. eine Installations-Diskette erzeugt oder an einem Embedded System arbeitet, bei dem Speicherplatz knapp und kostbar ist, der kann auf diesen Komfort verzichten. In diesem Falle werden im Dateisystem `proc` die Gerätenummern und nicht die Namen angezeigt.

#### EISA support (CONFIG\_EISA)

Als es Ende der 80er Jahre klar wurde, daß die ISA-Architektur an ihre Grenzen stößt, wurden verschiedene Wege in der Bus-Architektur beschritten. IBM entwickelte seinen eigenen MicroChannel (s.u.), während die EISA-Gruppe versuchte zum einen die Leistungsfähigkeit des MicroChannel zu bieten, auf

der anderen Seite aber abwärtskompatibel zum alten ISA-Standard zu bleiben. Seit Mitte der 90er wurden fast nur noch Rechner mit PCI-Bus ausgeliefert. Wer einen älteren Rechner mit EISA (Extended Industry Standard Architecture) besitzt und dafür einen Kernel bauen möchte antwortet mit y, alle anderen sagen n.

#### MCA support (CONFIG\_MCA)

Wenn man zum Jahresende 1999 günstig gebrauchte PCs von Banken oder Versicherungen kaufte, so waren es häufig Rechner vom Typ IBM PS/2 mit MicroChannel-Architektur. MCA ist nichts anderes als eine Bus-Architektur, wie auch ISA, EISA oder PCI. Der MicroChannel hat sich nie richtig durchgesetzt, aber falls man einen dertigen Rechner besitzt, so muß dieser Bus unbedingt aktiviert werden. Weitere Informationen findet man unter /usr/src/linux/Documentation/mca.txt.

#### Support for hot-pluggable devices (CONFIG\_HOTPLUG)

Diese Option ist für die Unmenge von Geräten gedacht, die man während des Betriebs des Rechners einstecken bzw. entfernen kann, ohne das System ausschalten zu müssen. In der Hauptsache teilen sich diese Geräte in zwei Gruppen: zum einen die PCMCIA-Karten, die in Scheckkartengröße als Modem, Netzwerkkarte oder sogar als Festplatte dienen. Die zweite Gruppe sind die USB-Geräte, die langsam zum neuen Peripherie-Standard avancieren. Mit dieser Option HOTPLUG sowie den Optionen USB und KMOD wird der Kernel automatisch die Möglichkeit bieten, einen „policy agent“ zu aktivieren, der die notwendigen Module im User-Modus lädt, die die entsprechenden Geräte benötigen. Weiteres findet man auf den Linux-USB-Seiten unter <http://www.linux-usb.org/policy.html>.

## PCMCIA/CardBus support

Dieses Submenü bietet folgende Möglichkeiten zum Anschluß der Kreditkarten-großen PCMCIA-Karten, die meist im Zusammenhang mit Laptops und Notebooks benutzt werden.

### PCMCIA/CardBus support (CONFIG\_PCMCIA)

Dieses ist die Option, um prinzipiell die Möglichkeit zur Nutzung von PCMCIA-Karten zu aktivieren. Dieses gilt für alle Typen, also sowohl die älteren 16-Bit-Karten als auch die neueren 32-Bit-Karten, die am CardBus hängen, für die allerdings auch noch die nächste Option aktiviert werden muß. Weitergehende Literatur biete das PCMCIA-HOWTO, das unter folgender Adresse erhältlich ist:

<http://www.linuxdoc.org/docs.html#howto>

Dieser Treiber kann auch als Modul kompiliert werden, was im Falle der PCMCIA-Karten ja durchaus sinnvoll ist.

### CardBus support (CONFIG\_CARDBUS)

Der CardBus ist eine Bus-Architektur für neuen 32-Bit PCMCIA-Karten. Die ursprüngliche PCMCIA-Spezifikation erlaubte nur 16-Bit-Karten. Wer unsicher ist, aber neuere PCMCIA-Karten benutzt, sollte diesen Treiber aktivieren.

### i82365 compatible bridge support (CONFIG\_I82365)

Dieser Treiber bietet Unterstützung für die CardBus-Bridge mit dem Intel-Chip i82365, letztendlich fast alle derzeitigen neueren PCMCIA-Brücken. Im Zweifelsfalle y.

### Databook TCIC host bridge support (CONFIG\_TCIC)

Wer in seinem Rechner die selteneren, älteren PCMCIA-Bridge vom Typ Data-book TCIC besitzt, benötigt diesen Treiber.

System V IPC (CONFIG\_SYSVIPC)

Mit Hilfe dieser Option können Programme Informationen austauschen. Einige Programme, wie z. B. der DOS-Emulator, können ohne diese Option nicht arbeiten. Die Antwort heißt also y und kostet auch nur 18 Kbyte.

BSD Process Accounting (CONFIG\_BSD\_PROCESS\_ACCT)

Durch dieses Option wird es einer Applikation ermöglicht, den Kernel mittels eines Systemaufrufs anzugeben, bestimmte Informationen über dieses Programm in eine Datei zu schreiben, wie z.B. die Uhrzeit von Start und Ende von Prozessen, den genutzten Arbeitsspeicher etc. Weitere Informationen über die Möglichkeiten findet man in `include/linux/acct.h`. Im Zweifelsfalle y.

Sysctl support (CONFIG\_SYSCTL)

Dieses ist eine Option für experimentierfreudige Menschen und Systementwickler: hierdurch wird eine Schnittstelle aktiviert, durch die bestimmte Kernel-Parameter und -Variablen während des Betriebes eines Systems geändert werden können, ohne daß der Rechner neu gebootet werden muß. Hierzu muß auf das Verzeichnis `/proc/sys` zugegriffen werden. Genauereres findet man in den Dateien des Verzeichnisses `/usr/src/linux/Documentation/sysctl/`. Dieser Treiber benötigt 8 KB.

Kernel core (/proc/kcore) format (CONFIG\_KCORE\_ELF)

Bevor die sogenannten stabilen Kernel mit den geraden Versionsnummern auf die diversen Rechner losgelassen werden, gibt es viele Versionen, in denen der Systemkern entwanzt werden muß. Für eine spezielle Hilfe hierzu muß das virtuelle Dateisystem `/proc` aktiviert sein, dann findet man unter

/proc/kcore ein Bitimage des Kernels. Dieses kann dann mit Hilfe des GNU Debuggers gdb analysiert werden:

```
cd /usr/src/linux
gdb vmlinux /proc/kcore
```

Bei dieser Option geht es nun um das interne Datenformat dieses Kernel-Abbildes. Die beiden möglichen Formate sind ELF und A.OUT. Das derzeit inzwischen durchweg gängige Format ist ELF. Allerdings gibt es noch einige Werkzeuge, die nur mit dem älteren Format A.OUT umgehen können - für diese ist die zweite Option gedacht. Wer nicht am Kernel hacken möchte beläßt es bei der Standardeinstellung ELF.

#### Kernel support for a.out binaries (CONFIG\_BINFMT\_AOUT)

A.OUT (Assembler.OUTPUT) ist ein Format, das in früheren Versionen von UNIX für Bibliotheken und ausführbare Binärdateien verwendet wurde. Auch LINUX benutzte A.OUT, bis es durch das ELF-Format ersetzt wurde. Alle neuen Programme unter UNIX und LINUX verwenden das ELF-Format, aber es sind natürlich noch diverse im A.OUT-Format vorhanden. Falls man sich nicht sehr sicher ist, daß man nur Programme im neuen Format vorliegen hat, sollte man unbedingt mit y antworten.

#### Kernel support for ELF binaries (CONFIG\_BINFMT\_ELF)

ELF (Executable and Linkable Format) ist ein Format für Bibliotheken und Programme, das inzwischen von verschiedenen Betriebssystemen - auch LINUX - benutzt wird. Daß diese Option verneint werden kann, ist nicht sinnvoll, hat aber historische Gründe. Auch ein Hinzuladen als Modul ist nicht empfehlenswert, da auch der Kernel-Daemon im ELF-Format vorliegt, dieser also nach dem Booten dann nicht geladen werden kann und er also auch nicht dieses Modul laden kann, das benötigt wird, um den Kernel-Daemon zu laden ... Auch als Deadlock bekannt. Also unbedingt y antworten.

#### Kernel support for MISC binaries (CONFIG\_BINFMT\_MISC)

Falls Programme ausgeführt werden sollen, die einen bestimmten Interpreter oder Loader benötigen, so muß hier mit y geantwortet werden. Beispiele hierfür sind Java-, Python- oder Emacs-Programme. Der jeweilige Interpreter muß dazu registriert werden. Details findet man unter `/usr/src/linux/Documentation/binfmt_misc.txt`.

#### Power Management support (CONFIG\_PM)

Wenn der PC „schlafen“ soll, kommt nicht das Sandmännchen, sondern das Power Management. Das PM hilft insbesondere bei Laptops die Arbeitszeit im Akku-Betrieb zu verlängern. Für das PM gibt es inzwischen zwei Standard: APM und ACPI. Wenn einer dieser beiden benutzt werden soll, so muß hier mit y geantwortet werden. Wer mehr über den Betrieb von Laptops unter Linux wissen möchte, kann unter <http://www.cs.utexas.edu/users/kharker/linux-laptop/> nachsehen. Außerdem gibt es ein Mini-Howto für Linux im Batteriebetrieb unter <http://www.linuxdoc.org/docs.html#howto>. Außerdem sei angemerkt, daß Linux auch ohne PM den **Prozessor** schlafen legt, falls nichts zu tun ist.

#### ACPI support (CONFIG\_ACPI)

ACPI (Advanced Configuration and Power Interface) ist die Weiterentwicklung von APM (s.u.) und ermöglicht das automatische Energiesparen nachdem der Rechner eine definierte Zeit untätig war. Dazu muß das Motherboard ACPI unterstützen. Wer mehr über die technischen Details von ACPI lesen möchte, kann das unter <http://www.teleport.com/~acpi/>. Wer diesen Treiber aktivieren möchte, kann außerdem ein kurzes FAQ im Text Documentati-on/pm.txt finden.

#### ACPI interpreter (EXPERIMENTAL) (NEW) (CONFIG\_ACPI\_INTERPRETER)

Mit dieser Option kann der neue, derzeit noch experimentelle ACPI-Interpreter in den Kernel eingebunden werden. Durch diesen kann die gesamte Band-

breite der Möglichkeiten von ACPI ausgeschöpft werden. Da dieser Treiber 120 KB benötigt und außerdem noch im Entwicklungsstadium ist, ist er nur etwas für experimentierfreudige Menschen.

Enter S1 for sleep (EXPERIMENTAL) (CONFIG\_ACPI\_S1\_SLEEP)

Hierdurch kann der niedrigste Level von ACPI aktiviert werden, der den gesamten Rechner schlafen legen kann. Der Treiber ist noch in der Entwicklung befindlich!

Advanced Power Management BIOS support (CONFIG\_APM)

APM ist die ältere Technik, die hilft, Leistung zu sparen und also die Akkus von Laptops zu schonen. Der neue Standard ist ACPI (s.o.). Leider haben sich - besonders in der Frühzeit der Laptops - nicht alle Hersteller an die genaue Spezifikation gehalten (s.u.), wie z.B. bei frühen Travelmates und bei (fast) allen Rechnern, die grüne Bildschirme besitzen. Diese Rechner haben keine APM, die der Spezifikation entspricht, und benutzen Speicherbereiche, die für LINUX reserviert sind. Diese Rechner reagieren mit Kernel Panic während des Hochfahrens. Laptops sind ein leidiges Thema, weil vor lauter Hardwaretricks - um Strom zu sparen - sich leider nicht mehr um einfaches Standard-PC-Design gekümmert wird. Detaillierte Informationen gibt das Laptop-HOWTO in /usr/doc/howto/en. Wegen der oft an Hardware-Design-Fehlern gescheiterten Aktivierung von APM gibt es jetzt einen Boot-Parameter. Man kann also für seinen Laptop APM in den Kernel einbinden, und sehen, ob dieser Treiber mit seinem Rechner-Modell funktionstüchtig ist. Falls nicht, so kann man mittels

apm=off

den Treiber deaktivieren, ohne sich einen weiteren Kernel bauen zu müssen.

Ignore USER SUSPEND (CONFIG\_APM\_IGNORE\_USER\_SUSPEND)

Wenn ein Betriebssystem wirklich gut sein will, muß es sich bemühen, alle Leistungsmerkmale und Ressourcen eines Rechners auszunutzen. Alleine: man stolpert immer wieder über inkompatible Hardware; so auch hier. „User Suspend“ meint, daß ein entsprechendes Programm im User-Modus mittels des APM den Rechner schlafen legen darf. Dieses ist ja prinzipiell erwünscht, funktioniert leider aber bei den Notebooks der Baureihe Versa M von NEC nicht wegen eines Fehlers im BIOS. Alle außer Besitzern von Versa M sagen n.

#### Enable PM at boot time (CONFIG\_APM\_DO\_ENABLE)

Dieser Treiber aktiviert APM während des Hochfahrens des Rechners. Definitiv abgestellt werden muß dieses Merkmal bei den Rechntypen NEC Ultra-Lite Versa 33/C und Toshiba T400CDT, da diese sonst beim Booten hängenbleiben. Dieses kann natürlich auch bei anderen Laptops passieren: wenn der Rechner nicht Hochfahren will, nachdem er sich Schlafen gelegt hat, nicht wieder aufwacht, oder fortwährend piept, so sollte man diesen Treiber deaktivieren.

#### Make CPU Idle calls when idle (CONFIG\_APM\_CPU\_IDLE)

Aktiviert die APM-Aufrufe „CPU Busy“ und „CPU Idle“. Wenn der Kernel in einer Idle-Schleife läuft, also nichts zu tun hat, so kann ein Aufruf von „CPU Idle“ Energie sparen, indem z.B. die Taktrate des Prozessors reduziert wird. Auf Maschinen, die sich nicht sauber an die APM-Spezifikation gehalten haben, kann dieses dazu führen, daß der Rechner hängt. Auf Rechnern mit mehr als einer CPU wird dieser Treiber nicht ausgeführt.

#### Enable console blanking using APM (CONFIG\_APM\_DISPLAY\_BLANK)

Dieses Option für Laptops betrifft den Bildschirm, und zwar im Textmodus. Bekanntlich schaltet sich eine Linux-Text-Konsole nach einiger Zeit automatisch dunkel. Durch diese Option kann zusätzlich die Hintergrundbeleuchtung von Laptops ausgeschaltet werden. Allerdings funktioniert dieses leider nicht

bei allen Laptops. Außerdem wird berichtet, daß die gleichzeitige Benutzung von `gpm` eine größere Anzahl von Fehlermeldungen produzieren kann. Dieser Treiber schaltet nicht die Hintergrundbeleuchtung aus, wenn dort X Window läuft. Hierfür ist der VESA-kompatible Energiesparmodus zuständig.

#### RTC stores time in GMT (`CONFIG_APM_RTC_IS_GMT`)

Hier wird abgefragt, ob die Echtzeituhr (Real Time Clock, RTC) des PC die Uhrzeit in Greenwich Mean Time (GMT) oder in der lokalen Zeit speichern soll. Normalerweise sollte man die Uhrzeit in GMT speichern - den Umrechnung in lokale Zeit inklusive Sommerzeit erledigt Linux automatisch, nachdem man die lokale Zeitzone angegeben hat. Der einzige Grund, die lokale Zeit in der RTC zu speichern ist die abwechselnde Benutzung von verschiedenen Betriebssystemen, von denen eines dann nicht damit umgehen kann, daß man die Weltzeit in seiner Uhr gespeichert hat.

#### Allow interrupts during APM BIOS calls (`CONFIG_APM_ALLOW_INTS`)

Normalerweise werden externe Interrupts deaktiviert während ein Aufruf an das APM-BIOS ausgeführt wird. Dieses wird prinzipiell gemacht, um die Nebeneffekte von diversen schlecht programmierten BIOS zu unterdrücken. Das BIOS sollte normalerweise in der Lage sein die Interrupts selbstständig wieder zu aktivieren. Soweit die schöne Theorie. In der Praxis sind allerdings einige BIOS aufgetaucht, die dieses nicht schaffen. Insbesondere sind dies die neuere IBM Thinkpads. Prinzipiell sollte man im Zusammenhang mit einem APM-BIOS diese Option aktivieren. IBM und andere Ausnahmen bestätigen die Regel.

#### Use real mode APM BIOS call to power off (`CONFIG_APM_REAL_MODE_POWER_OFF`)

Diese Option aktiviert den Real Mode, um mittels des APM-BIOS den Rechner auszuschalten. Auch dieses ist wiederum eine Krücke, um fehlerhaft programmierte BIOS zu einer vernünftigen Arbeitsweise zu überzeugen. Falls der

Rechner beim Ausschalten crasht, statt sich sauber auszuschalten, sollte man es mit diesem Treiber versuchen.

## 5. Memory Technology Devices (MTD)

Diverse Flash- und RAM-Chips werden inzwischen von Linux unterstützt, meist im Zusammenhang mit Embedded Systems. In diesem Menü werden diese Möglichkeiten abgehandelt, die für den Normalverbraucher meist nicht von Interesse sind .

Memory Technology Device (MTD) support (CONFIG\_MTD)

Dieses ist der generische Treiber, der erst einmal prinzipiellen Zugang zu MTD ermöglicht, nämlich die jeweiligen Geräte im Kernel zu registrieren und Zugang zu ihnen zu ermöglichen.

M-Systems Disk-On-Chip 1000 support (CONFIG\_MTD\_DOC1000)

Der Treiber für den DiskOnChip 1000 von M-Systems, der allerdings veraltet ist.

M-Systems Disk-On-Chip 2000 ( CONFIG\_MTD\_DOC2000 )

Dieses ist der Treiber für den DiskOnChip 2000 von M-Systems. Wer dieses Gerät benutzt, benötigt wahrscheinlich auch den Treiber NAND Flash Translation Layer (s.u.) der es ermöglicht ein Dateisystem auf dem Chip zu erzeugen und zu benutzen.

M-Systems Disk-On-Chip Millennium (CONFIG\_MTD\_DOC2001)

Dieses ist der Treiber für den DiskOnChip Millenium von M-Systems. Wer dieses Gerät benutzt, benötigt wahrscheinlich auch den Treiber NAND Flash

Translation Layer (s.u.) der es ermöglicht ein Dateisystem auf dem Chip zu erzeugen und zu benutzen.

Use extra onboard system memory as MTD device (CONFIG\_MTD\_SLRAM)

Falls ungenutzter Onboard-Speicher vorhanden ist, so kann dieser durch diesen Treiben dazu benutzt werden, als MTD-Gerät angesprochen zu werden.

Ramix PMC551 PCI Mezzanine ram card support (CONFIG\_MTD\_PMC551)

Dieses ist derder Treiber für die Ramix PMC551 RAM-Karte.

PMC551 256M DRAM Bugfix (NEW) (CONFIG\_MTD\_PMC551\_BUGFIX)

Einige der Karten PMC551 besitzen falsche Werte für Zeile und Spalte. Dieser Treiber kann den Fehler beheben, allerdings ist die Speicherverwaltung für andere Karten dann nicht mehr zu benutzen.

Debugging RAM test driver (CONFIG\_MTD\_MTDRAM)

Für das Entwanzen des RAM-Test-Treibers, der `vmalloc()` aufruft, um Speicher zur Verfügung zu stellen.

Device size in kB (CONFIG\_MTDRAM\_TOTAL\_SIZE)

Die Größe des MTD-RAMs in Kilobyte.

Size of the erase sectors in kB (CONFIG\_MTDRAM\_ERASE\_SIZE)

Flash-Speicher können nur in bestimmten Blockgrößen gelöscht werden. Die Block-Größe in kB muß hier eingegeben werden.

Common Flash Interface (CFI) support (CONFIG\_MTD\_CFI)

Die Spezification des CFI (Common Flash Interface) von Intel enthält Möglichkeiten, das entsprechende Gerät zu testen. Wer diese Möglichkeiten nutzen möchte, muß diesen Treiber aktivieren.

CFI support for Intel/Sharp Extended Command Set chips  
(CONFIG\_MTD\_CFI\_INTELEXT)

Die Spezifikation des CFI enthält verschiedenen Befehlssätze, die auf den verschiedenen Chips zum Einsatz kommen können. Dieses ist der Befehlsatz, der von Intel und Sharp genutzt wird.

CFI support for AMD/Fujitsu Standard Command Set chips  
(CONFIG\_MTD\_CFI\_AMDSTD)

Der CFI-Befehlssatz zur Unterstützung von Chips der Firmen AMD und Fuji.

Flash chip mapping in physical memory (CONFIG\_MTD\_PHYSMAP)

Flash-Chips können als Arbeitsspeicher der CPU gemappt werden. Dieses ist der Treiber, der mittels CFI den Zugriff ermöglicht. Dazu müssen auch die nächsten beiden Optionen, nämlich Adresse und Größe des Chips angegeben werden.

Physical start location of flash chip mapping (NEW)  
(CONFIG\_MTD\_PHYSMAP\_START)

Die Start-Adresse des Flash-Chips - die Hardware-Dokumentation sollte helfen können.

Physical length of flash chip mapping (NEW) (CONFIG\_MTD\_PHYSMAP\_LEN)

Die Gesamtgröße des gemappten Speichers auf dem Flash-Chip.

Flash chip mapping on Nora (CONFIG\_MTD\_NORA)

Der Treiber für das Mapping des Nora-Flash-Chips.

Flash chip mapping on Photron PNC-2000 (CONFIG\_MTD\_PNC2000)

Der Treiber für das Mapping des Flash-Chips PNC-2000.

Flash chip mapping on RPXLite PPC board (CONFIG\_MTD\_RPXLITE)

Der Treiber für das ungewöhnliche Mapping des Flash-Chips RPXLite.

Direct chardevice access to MTD devices (CONFIG\_MTD\_CHAR)

Dieser Treiber hält für jedes MTD-Gerät einen zeichenorientierten Gerätetreiber bereit, so daß der Benutzer direkt auf das jeweilige Gerät Schreiben bzw. von dem Gerät Lesen kann.

Pseudo-blockdevice access to MTD devices (CONFIG\_MTD\_BLOCK)

Flash-Chips müssen immer eine relativ große Blockgröße gleichzeitig löschen. Dadurch ist es schwierig, sie sinnvoll als Blockgerät einzusetzen. Durch diesen Treiber kann man aber MTM-Geräte auf der Basis von RAM-Chips benutzen. Außerdem benötigt man derzeit auch diesen Treiber für das Journalling Flash File System (JFFS), um einen Handle zu erhalten, wenn es gemounted ist, obwohl das JFFS keinen weiteren Gebrauch von diesem Treiber macht.

FTL (Flash Translation Layer) support (CONFIG\_FTL)

Der Flash Translation Layer (FTL) ist Teil der PCMCIA-Spezifikation. Dieser Treiber enthält die FTL-Unterstützung und benutzt ein Pseudo-Blockgerät mit

Sektoren zu 512 Byte, das über ein normales Dateisystem gelegt werden kann. Der benutzte Algorithmus ist übrigens in den USA patentiert und darf dort nur auf PCMCIA-Hardware benutzt werden.

NFTL (NAND Flash Translation Layer) support (CONFIG\_NFTL)

Dieser Treiber enthält den NAND Flash Translation Layer, der für die DiskOn-Chip-Geräte von M-System benötigt wird. Er stellt ein Pseudo-Blockgerät mit 512-Byte-Sektoren zur Verfügung. Der benutzte Algorithmus ist in den USA patentiert und darf dort nur auf den DiskOnChip-Geräten benutzt werden.

Write support for NFTL (EXPERIMENTAL) (NEW) (CONFIG\_NFTL\_RW)

Treiber zum Schreiben auf dem NAND Flash Translation Layer. Obwohl noch in sehr frühem experimentellen Stadium soll er angeblich arbeiten.

## 6. Parallel port support

Parallel-port support (CONFIG\_PARPORT)

Falls an der parallelen Schnittstelle des Rechners ein oder mehrere Geräte angeschlossen werden sollen, so muß hier mit y geantwortet werden. Typischerweise wird ein Drucker an dem parallelen Port angeschlossen, aber es gibt u.a. auch ZIP-Laufwerke, Bandlaufwerke und CD-ROM-Laufwerke für diese Schnittstelle. In fast allen Fällen ist dieser Treiber also notwendig; wer allerdings z.B. nur übers Netz druckt, benötigt diesen Treiber nicht. Wer mehr als eine parallele Schnittstelle besitzt, oder mehr als ein Gerät an dieser Schnittstelle betreiben möchte, der sollte die Ausführungen unter /usr/src/linux/Documentation/parport.txt lesen. Falls man weitere Information über Gerätetreiber für den parallelen Port benötigt, so gibt es weiterführendes Know How unter <http://www.torque.net/linux-pp.html>. Dieser Treiber kann auch als Modul betrieben werden.

**PC-style hardware (CONFIG\_PARPORT\_PC)**

Falls sie einen Standard-PC mit Standard-Parallel-Port besitzen, und diese Nutzen wollen, so muß hier mit `y` geantwortet werden. Auch einige DEC Alphas besitzen diese Standard-Schnittstelle. Dieser Treiber ist der Standard und ist als Gegensatz zu dem folgenden Treiber zu sehen. Er kann auch als Modul geladen werden.

**Use FIFO/DMA if available (NEW) (CONFIG\_PARPORT\_PC\_FIFO)**

Viele Chipsätze der parallelen Schnittstelle unterstützen ein beschleunigtes Drucken mittels FIFO oder DMA. Allerdings benötigt der Kernel dann die Angabe des IRQ des parallelen Ports. Normalerweise werden für die parallelen Schnittstellen keine IRQs benutzt. Details entält der Text Documentati-on/parport.txt.

**SuperIO chipset support (EXPERIMENTAL) (NEW)  
(CONFIG\_PARPORT\_PC\_SUPERIO)**

Dieser Treiber aktiviert die automatischen Tests für den Super-IO-Chip. Er findet für diesen Chip automatisch die Adresse, den IRQ und den DMA-Kanal heraus. Da der Treiber noch recht neu ist, sollte hier mit `n` geantwortet werden.

**Support foreign hardware (CONFIG\_PARPORT\_OTHER)**

Für Benutzer von Rechnern mit exotischer, proprietärer Hardware. Für den Standard-PC-Benutzer nicht von Interesse, also `n`.

**IEEE 1284 transfer modes (NEW) (CONFIG\_PARPORT\_1284)**

Neuere Drucker unterstützen die bidirektionale Kommunikation laut IEEE 1284. Dieser Treiber unterstützt diese Geräte sowie die erweiterten parallelen Kommunikationsmodi EPP und ECP. Außerdem erzeugt der Treiber Geräte-

informationen, die unter `/proc/sys/dev/parport/*/autoprobe*` gelesen werden können.

## 7. Plug and Play configuration

Plug and Play support (CONFIG\_PNP)

Seit einigen Jahren unterstützen PCs das sogenannte, Plug and Play, das automatische Erkennen von Peripheriegeräten. War PnP am Anfang zu Recht als „Plug and Pray“ verschrien, so funktioniert dieses heutzutage relativ gut und erspart dem Benutzer vieles an Konfiguration und an Bootparametern im Verhältnis zu älteren Rechnern. Da dieses heute Standard ist, wird mit `y` geantwortet.

ISA Plug and Play support (CONFIG\_ISAPNP)

Wer PnP-Unterstützung auch für den ISA-Bus möchte, der findet hier den entsprechenden Treiber. Weitere Information enthält `Documentation/isapnp.txt`. Der Treiber kann auch als Modul kompiliert werden.

## 8. Block Devices

Normal floppy disk support (CONFIG\_BLK\_DEV\_FD)

Notwendig, wenn Diskettenlaufwerke unter LINUX benutzt werden sollen. Wer nicht ausgiebig mit Diskettenlaufwerken arbeitet, sollte dieses als Modul kompilieren.

PS/2 ESDI hard disk support (CONFIG\_BLK\_DEV\_PS2)

Wer einen PS/2-Rechner besitzt, und also einen MCA-Bus und eine ESDI-Festplatte betreibt, der benötigt diesen Treiber.

#### XT harddisk support (CONFIG\_BLK\_DEV\_XD)

Wir hatten schon arge Bedenken, daß auch dieser völlig veraltete Treiber den Renovierungsarbeiten zum Opfer fällt. Sentimental, wie wir sind, sagen wir: wunderbar, daß er noch dabei ist. Obwohl niemand mehr den uralten 8-Bit-Festplatten-Controller, der seinerzeit im IBM-XT-Rechner seinen Dienst absolvierte, benutzt, wird er wohl weiter mit durchgeschleppt, da er die erste Linux-Festplatte betrieb. Im Moment kosten allerdings 10 MByte Festplatte im Fachhandel ca. 0,10 Euro.

#### Compaq SMART2 support (CONFIG\_BLK\_CPQ\_DA)

Dieses ist der Treiber für das Compaq Smart Array SMART2. Weitere Informationen enthält der Text Documentation/cpqarray.txt.

#### Compaq CISS Array support (CONFIG\_BLK\_CPQ\_CISS\_DA)

Dieses ist der Treiber für das Compaq Smart Array CISS. Weitere Informationen enthält der Text Documentation/cciss.txt.

#### Mylex DAC960/DAC1100 PCI RAID Controller support (CONFIG\_BLK\_DEV\_DAC960)

Dieser Treiber unterstützt die PCI-Raid-Controller der Typen Mylex DAC960, AcceleRAID und eXtremeRAID. Weitere Information enthält der Text Documentation/README.DAC960. Der Treiber kann auch als Modul kompiliert werden.

#### Loopback device support (CONFIG\_BLK\_DEV\_LOOP)

Diese Option ermöglicht es, eine Datei als Filesystem zu mounten, sogar als ein verschlüsseltes. Dieses ist z. B. nützlich, wenn man ein ISO-9660-Dateisystem überprüfen möchte, bevor man eine CD brennt. Achtung: Nicht mit dem Loopback-Device verwechseln, das Netzwerkverbindungen mit sich selbst ermöglicht. Siehe dazu unter TCP/IP networking: CONFIG\_INET.

#### Network Block Device support (CONFIG\_BLK\_DEV\_NBD)

Diese Option ist gedacht für Rechner mit wenig oder gar keiner Festplattenkapazität, die also ihre Daten auf einem Server speichern, wie z.B. bei einem diskless Arbeitsplatzrechner. Der Server exportiert Festplatten für den Client, und dieser benutzt sie als seine eigene lokale Platte. Die Kommunikation erfolgt über ein TCP/IP-Netzwerk. Für den Benutzer ist dieses aber kaum ersichtlich; die Festplatten erscheinen als lokale Platten, besitzen allerdings die interne Kennung /dev/ndx. Diese Option hat nichts mit einem NFS-Dateisystem zu tun. Im Gegensatz zu NFS kann die vom Server exportierte Festplatte ein beliebiges Dateisystem enthalten; demnächst soll sogar eine Swap-Partition des Arbeitsplatzrechners auf dem Server ermöglicht werden. Genaueres findet man unter /usr/src/linux/Dokumentation/nbd.txt, außerdem unter <http://atrey.karlin.mff.cuni.cz/~pavel>.

#### RAM disk support (CONFIG\_BLK\_DEV\_RAM)

Hiermit kann ein Teil des RAMs als blockorientiertes Gerät benutzt werden. Diese sogenannte RAM-Disk kann über /dev/ram wie eine Diskette oder Festplatte angesprochen werden. Im Gegensatz zu älteren Kernel-Versionen muß die Größe nicht mehr als Boot-Parameter angegeben werden, sondern die Größe wird automatisch der Speicherumgebung angepaßt. Meistens wird diese Option nur während des Ur-Ladens zur Installation von LINUX genutzt, um den langsamen Diskettenbetrieb zu beschleunigen: die Root-Diskette wird in die RAM-Disk kopiert. Bei den heutigen Festplattengeschwindigkeiten spielt die RAM-Disk im Alltagsbetrieb keine Rolle mehr.

#### Default RAM disk size (NEW) (CONFIG\_BLK\_DEV\_RAM\_SIZE)

Falls die vorangegangene Option aktiviert wurde, kann hier die Größe der RAM-Disk eingegeben werden. Der Standardwert beträgt 4096 kB.

## 9. Multi-device support (RAID and LVM)

Multiple devices driver support (RAID and LVM) (`CONFIG_MD`)

Wer mit RAID oder dem Logischen Volume Manager (LVM) arbeiten möchte sagt hier `y` und wird im Folgenden nach den Details abgefragt.

Multiple devices driver support (`CONFIG_BLK_DEV_MD`)

Dieser Treiber ermöglicht es, mehrere Festplatten bzw. Partitionen als ein logisches Gerät zu behandeln. Weitere Informationen unter `drivers/block/README.md`. Im Zweifelsfall `n`.

Linear (append) mode (`CONFIG_MD_LINEAR`)

Wenn diese Option aktiviert wird, werden die verschiedenen physikalischen Geräte im Gegensatz zum gleich folgenden RAID-0-Modus schlicht linear aneinandergehängt.

RAID-0 (striping) mode (`CONFIG_MD_STRIPED`)

Wenn diese Option aktiviert wird, greift der Treiber im sogenannten RAID-0-Modus auf die Festplatten zu, d.h. sie werden gleichmäßig mit Daten gefüllt, um schneller Zugriffszeiten zu erhalten. Im linearen Modus (s. o.) hingegen wird zuerst Festplatte 1, dann Festplatte 2 etc. beschrieben.

RAID-1 (mirroring) mode (`CONFIG_MD_RAID1`)

Die RAID-1-Norm hält von jeder Festplatte eine exakt gespiegelte Kopie bereit, so daß bei einem Defekt einer Platte jederzeit auf der gespiegelten Platte korrekte und aktuelle Daten vorliegen. Dienstprogramme um RAID 1 einzurichten und zu pflegen findet man unter <http://luthien.nuclecu.unam.mx/~miguel/raid>.

**RAID-4/RAID-5 mode (CONFIG\_MD\_RAID5)**

RAID Level 4 und 5 behandeln eine Anzahl von Festplatten derart, daß für den Benutzer nur ein Laufwerk mit großer Kapazität sichtbar ist. Dabei wird eine Festplatte für etwaige Ausfälle freigelassen, so daß im Fall eines Festplatten-Defekts automatisch auf dieser Sicherheits-Festplatte weitergearbeitet wird. Dienstprogramme um RAID 4 bzw. 5 einzurichten und zu pflegen findet man unter <http://luthien.nuclecu.unam.mx/~miguel/raid>.

**Logical Volume Manager (LVM) support (CONFIG\_BLK\_DEV\_LVM)**

Wer sehr viel Festplatten-Speicherplatz benötigt und diesen auch noch während des Betriebs seines Rechners dynamischen ändern möchte, für den ist dieses die richtige Option. Die Treiber kann mehrere Festplatten zu einer „Volume Group“ zusammengefasst werden. Innerhalb einer Volume Group können logische Volumes angelegt werden, so daß der Benutzer unabhängig von den physikalischen Geräten seine benötigte Struktur zur Datenspeicherung anlegen kann, ohne sich durch die physikalische Größe der vorhandenen Festplatten beschränken zu müssen. Falls der Speicherplatz auf einem logischen Volume nicht ausreicht, so kann - nachdem die entsprechenden zusätzlichen Festplatten eingebaut wurden - das logische Volume dynamisch vergrößert werden, ohne es neu anlegen zu müssen. Für den Privatanwender wohl eher selten von Nutzen, aber irgendewie müssen ja die Unmengen von Multi-Media-Daten, die u.a. im WWW erhältlich sind, gespeichert werden. Details erhält man unter </usr/src/linux/Documentation/LVM-HOWTO>. Die notwendige Software erhält man von <http://linux.msede.com/lvm>.

**LVM information in proc filesystem (CONFIG\_LVM\_PROC\_FS)**

Wer das virtuelle Dateisystem `/proc` aktiviert hat mit `CONFIG_PROC_FS` und diesen Treiber aktiviert, der kann unter `/proc/lvm` die Details des LVM auslesen, wie Information über die physikalischen Volumen und die Gruppen des LVM.

## 10. Networking options

### Packet socket (`CONFIG_PACKET`)

Diese Paket-Protokoll wird von bestimmten Programmen benutzt, die ohne weiteres Netzwerkprotokoll direkt zugreifen, wie z.B. `tcpdump`. Es kann auch als Modul kompiliert werden. Für den normalen Benutzer entbehrlich.

### Packet socket: mmapped IO (NEW) (`CONFIG_PACKET_MMAP`)

Falls der verangehende Treiber aktiviert wurde, kann mit einem beschleunigten Eingabe-/Ausgabe-Mechanismus betrieben werden mittels dieser Option.

### Kernel/User netlink socket (`CONFIG_NETLINK`)

Dieser Treiber ermöglicht eine gegenseitige Kommunikation zwischen bestimmten Teilen des Kernel sowie Benutzerprogrammen. Dieses geschieht mittels Erzeugung der neue Socket-Familie namens `PF_NETLINK`. Dokumentation ist durch die Man-Pages (7) in `netlink` erhältlich, außerdem unter <http://snafu.freedom.org/linux2.2/docs/netlink-HOWTO.html>.

Für folgende Programme wird dieser Treiber derzeit benutzt:

- wenn der Treiber `Routing message` (s.u.) benötigt wird.

- wenn eine Firewall mit dem Parameter IP: `firewall packet netlink device` gewünscht ist.
- wenn `arp`, der ARP-Daemon genutzt werden soll.
- wenn `EtherTap` genutzt wird, welches Benutzerprogrammen ermöglicht, Ethernet Frames zu lesen und zu schreiben.

#### Routing messages (CONFIG\_RTNETLINK)

Wer über die Routing-Informationen auf dem laufenden sein will, benötigt diesen Treiber. Außerdem muß eine Datei `/dev/route` mit der Hauptgerätenummer 36 und der Nebennummer 0 mittels `mknod` erzeugt werden. Dann können hier Informationen gelesen - aber nicht geschrieben - werden.

#### Netlink device emulation (CONFIG\_NETLINK\_DEV)

Dieser Treiber wird nur aus historischen Gründen mitgeführt, um Abwärtskompatibilität zu wahren. Im Moment sollte noch mit `y` geantwortet werden. Demnächst soll diese Option entfernt werden.

#### Network packet filtering (replaces ipchains) (CONFIG\_NETFILTER)

Ein Netzwerkfilter wird benötigt, um die verschiedenen Netzwerkpakete zu überprüfen und nur die gewünschten passieren zu lassen. Unter den gebräuchlichsten Einsatzgebieten eines Netzwerkfilter ist als erstes die Firewall zu nennen. Eine Firewall wird benutzt, um das lokale Netzwerk vor dem Internet zu schützen. Dieser Treiber unterstützt eine Firewall auf Paket-Filter-Basis, d.h. es können individuelle Datenpakete nach Typ, Quelle, Ziel etc. zugelassen bzw. zurückgewiesen werden. Im Gegensatz zu diesem Packetfilter steht die Proxy-basierte Firewalls, die mehr Sicherheit bieten, aber auch deutlich komplexer zu konfigurieren sind. Proxy-basierte Firewalls können den Netzwerkverkehr genauer inspizieren, da sie anhand der Protokolle detaillierteres Wissen über die Datenpakete erhalten, die ein Netzwerkfilter nicht besitzen

kann. Proxy-basierte Firewalls benötigen diesen Netzwerkpaketfilter nicht, werden aber oft mit diesem kombiniert. Auch sollte dieser Treiber aktiviert werden, wenn der Linux-Rechner als Gateway für ein lokales Netz ohne globale IP-Adressen arbeiten soll. Dieses ist auch unter dem Begriff Masquerading bekannt: das Gateway maskiert die lokalen Rechner vor dem globalen Netz sowie umgekehrt. Praktisch bedeutet dieses, daß die beiden Welten füreinander nicht sichtbar sind - jedenfalls auf elektronischer Ebene - aber Kontakt zueinander halten können über das Gateway. Auch kann dieser Treiber eingesetzt werden, um einen lokalen Proxy zu installieren: sämtliche lokalen Rechner greifen auf den Proxy zu, der typischerweise einen großen Cache besitzt. Wenn Anfragen an das Internet von einem Client erfolgen, werden diese alle über den Proxy geleitet, der zuerst nachsieht, ob er die Daten evtl. bereits im Cache liegen hat, im anderen Fall eine transparente Verbindung zum Internet herstellt.

Falls die Option `Fast switching` (s.u.) aktiviert wird, so wird dieser Treiber deaktiviert, da `Fast switching` derzeit den Netzfilter umgeht!

#### Network packet filtering debugging (`CONFIG_NETFILTER_DEBUG`)

Für die Kernel-Künstler, die den vorangehenden Treiber entwanzten möchten, gibt es hiermit den Zugriff auf weitere Informationen.

#### Socket filtering (`CONFIG_FILTER`)

Der Socket-Filter kann im Zusammenhang mit einer Firewall arbeiten. Dieser Treiber ermöglicht es, dass Benutzer-Programme einem beliebigen Socket einen Filter zuweisen und somit den Kernel anweisen, bestimmte Typen von Daten durchzulassen bzw. zurückzuweisen. Das Linux-Socket-Filtrern arbeitet derzeit mit allen Socket-Arten mit der Ausnahme von TCP. Weitere Informationen enthält `linux/Documentation/networking/filter.txt`.

#### Unix domain sockets (`CONFIG_UNIX`)

Die Unix-Sockets wurden ursprünglich entwickelt, um Netzwerkverbindungen aufzubauen und auf sie zuzugreifen. Inzwischen werden sie von diversen Programmen für ihre eigenen Bedürfnisse „mißbraucht“, wie z.B. von X Window oder dem Daemon `syslog`. Also sollte hier jeder mit y antworten.

TCP/IP networking (CONFIG\_INET)

Dieses sind zwei zusammengehörige Protokolle, die im Internet und vielen lokalen Netzwerken benutzt werden. Auch wenn der Rechner beides nicht benötigt, sollte man hier mit y antworten, da viele Programme (X Window, Druckerspooler) sie Rechner-intern benutzen.

IP: multicasting (CONFIG\_IP\_MULTICAST)

Benötigt, um mehreren, vernetzten Computern gleichzeitig Daten zu senden. Diese Option wird auch benötigt, um an MBONE teilzunehmen, einem Breitbandnetz, das Audio und Video überträgt. Weitere Informationen über MBONE sind im Internet unter folgender Adresse erhältlich:  
<http://www.best.com/~prince/techinfo/mbone.html>.

IP: advanced router (CONFIG\_IP\_ADVANCED\_ROUTER)

Falls der Rechner in der Hauptsache als Router, d.h. als „Netzwerk-Verteiler“ arbeiten soll, so muß hier mit y geantwortet werden. Durch diese Antwort wird kein Treiber in den Kernel eingebunden, sondern es wird eine Reihe von weiteren Fragen aktiviert.

Ein Router kann nur korrekt arbeiten wenn auch das IP forwarding aktiviert ist; dazu muß der Treiber für den `/proc filesystem` support und den `Sysctl` support (s.u.) aktiviert sein. Die Standardantwort lautet hier n.

IP: policy routing (CONFIG\_IP\_MULTIPLE\_TABLES)

Normalerweise entscheidet ein Router über die erhaltenen Daten-Pakete allein anhand seiner Zieladresse. Wenn dieser Treiber eingebunden wird, so wird außerdem die Absenderadresse mit in die Entscheidung eingebunden. Noch weitergehende Möglichkeiten bieten die weiter unten stehenden Optionen IP: use TOS value as routing key und IP: fast network address translation. Wer weitergehendes Interesse an dieser Materie hat, kann sich bei folgenden Adressen umsehen:  
<http://www.compendium.com.ar/policy-routing.txt> und  
<ftp://post.tepkom.ru/pub/vol2/Linux/docs/advanced-routing.tex>. Software ist unter folgender Adresse erhältlich:  
<ftp://ftp.inr.ac.ru/ip-routing/>

IP: use netfilter MARK value as routing key (NEW)  
(CONFIG\_IP\_ROUTE\_FWMARK)

Durch diesen Treiber können für spezifizierte Pakete verschiedene Routen definiert werden. Die Man-Page (8) von iptables hilft weiter, ist aber leider nicht bei allen Linux-Distributionen vorhanden.

IP: fast network address translation (NEW) (CONFIG\_IP\_ROUTE\_NAT)

NAT ist die Abkürzung für Network Address Translation. Mittels dieses Treibers können Quell- und Ziel-Adresse der Datenpakete nach den Wünschen des Benutzers modifiziert werden. Genaueres enthält <http://www.csn.tu-chemnitz.de/~mha/linux-ip-nat/diplom/nat.html>.

IP: equal cost multipath (CONFIG\_IP\_ROUTE\_MULTIPATH)

Normalerweise gibt eine Routing-Tabelle genau eine Aktion für ein Datenpaket an. Falls dieser Treiber aktiviert wird, können mehrere verschiedene, gleichwertige Möglichkeiten angegeben werden, von denen der Server zufällig eine auswählt. Die angegeben Möglichkeiten werden als „Wege gleicher Kosten“ betrachtet.

IP: use TOS value as routing key (CONFIG\_IP\_ROUTE\_TOS)

Der Header eines jeden IP-Packet enthält die Information TOS (Type of Service), die Art des Dienstes. Anhand dieser Information kann man verschiedene Behandlungen für verschiedene Datentypen zuweisen, so daß entsprechend den Bedürfnissen des Dienstes hoher Durchsatz (z.B. Video on demand), hohe Verlässlichkeit (z.B. Dateübertragung) oder geringe Priorität (z.B. E-Mail) gewährt sind.

IP: verbose route monitoring (CONFIG\_IP\_ROUTE\_VERBOSE)

Falls man diesen Treiber aktiviert, so wird der Kernel Informationen ausgeben betreffs des Routings. Dieses kann sehr nützlich sein, wenn untypische Pakete ankommen, die z.B. ein Hinweis auf eine Hackerattacke oder ein fehlerhaft konfiguriertes System sein können. Zuständig für die Ausgabe dieser Informationen ist der Daemon `klogd`, der normalerweise aber automatisch gestartet wird.

IP: large routing tables (CONFIG\_IP\_ROUTE\_LARGE\_TABLES)

Falls einzelnen Zonen innerhalb der Routing-Tabelle mehr als 64 Einträge aufweisen, so sollte dieser Treiber aktiviert werden, um den Zugriff zu beschleunigen.

IP: kernel level autoconfiguration (CONFIG\_IP\_PNP)

Dieser Treiber aktiviert die automatische Konfiguration von Geräte sowie der Routing-Tabelle während der Kernel bootet. Die Konfigurations-Informationen werden entweder als Bootparameter übergeben oder werden mittels der Protokolle BOOTP oder RARP übernommen. Dieser Treiber wird nur benötigt, wenn man einen diskless Arbeitsplatzrechner besitzt. Alle anderen Rechner konfigurieren ihr Netzwerk anhand ihrer Skripte, die während des Bootens ausgeführt werden.

**IP: BOOTP support (NEW) (CONFIG\_IP\_PNP\_BOOTP)**

Falls das komplette Datei-System des Rechners von einem anderen Rechner importiert werden soll, der Rechner also eine Diskless Workstation ist, gibt es die Möglichkeit, daß die IP-Adresse dieses Rechners während des Hochfahrens automatisch herausgefunden wird mittels des BOOTP-Protokolls, so muß hier mit `y` geantwortet werden. Dazu muß allerdings zusätzlich ein BOOTP-Server in dem Netzwerk arbeiten. Manche Boot-ROMs der Netzwerkkarten sind in der Lage, das BOOTP-Protokoll zu verstehen, so daß die Hardware die Arbeit erledigt und dieser Treiber nicht nötig ist. Weitere Information enthält Documentation/nfsroot.txt.

**IP: RARP support (NEW) (CONFIG\_IP\_PNP\_RARP)**

Falls eine Diskless Station (Rechner ohne Festplatte und Diskettenlaufwerke) im lokalen Netz vorhanden ist, weiß er nach dem Booten zwar seine Hardware-Adresse auf dem Ethernet, aber die IP-Adresse ist zu diesem Zeitpunkt nicht bekannt. Er sendet also eine Anfrage an den Server, um diese herauszufinden. Um hier auf eine Reverse-Address-Resolution-Protocol-Anfrage antworten zu können, muß diese Option für den LINUX-Server gesetzt sein. Außerdem muß der Daemon `rarpd` laufen. Falls diese Option benötigt wird, ist dieses ein typischer Fall, den Code als Modul laden zu lassen.

**IP: tunneling (CONFIG\_NET\_IPIP)**

„Tunneling“ bedeutet die Kapselung der Daten eines Protokolls innerhalb eines anderen Protokolls, so daß sie über ein Netz geschickt werden können, das das zweite Protokoll beherrscht. Der vorliegende Treiber implementiert die Kapselung von IP innerhalb von IP, was auf den ersten Blick sinnlos erscheint. Dieses kann aber nützlich sein, wenn Rechner auf einem anderen Netzwerk arbeiten müssen, als auf dem, an das sie physikalisch angeschlossen sind. Auch ist dies hilfreich, wenn Laptops nahtlos von einem Netz zu einem anderen überwechseln, ohne ihre IP-Adresse zu ändern. Das Aktivieren dieser

Option erzeugt zwei Module, eins zur Kapselung und eins zur Entkapselung.  
Selten benötigt.

IP: GRE tunnels over IP (CONFIG\_NET\_IPGRE)

„Tunneln“ meint, daß ein Protokoll mitsamt seinen Daten in einem anderen Protokoll verpackt wird, über die entsprechende Leitung verschickt wird, und am anderen Ende wiederum ausgepackt wird. Dieser Treiber ist zuständig für die generische Routen-Kapselung (GRE, Generic Routing Encapsulation) und kann die Protokolle IPv4 oder IPv6 über das derzeit vorherrschende IPv4-Protokoll verschicken. Falls sich am anderen Ende ein Cisco-Router befindet, so ist dieser Treiber hilfreich, das Cisco mit GRE wesentlich besser umgehen kann, als mit dem LINUX-Treiber `Ip tunneling` (s.o.).

IP: broadcast GRE over IP (CONFIG\_NET\_IPGRE\_BROADCAST)

Dieser Treiber ermöglicht mittels GRE über IP ein sogenanntes Broadcast-WAN (Wide Area Network) zu erstellen, das wie ein Standard-Ethernet arbeitet, deren einzelne Stationen aber über das gesamte Internet verteilt sein können. Hierzu muß außerdem der Treiber `IP: multicast routing` aktiviert sein.

IP: multicast routing (CONFIG\_IP\_MROUTE)

Dieses wird benötigt, wenn ein Rechner als Router für IP-Pakete arbeitet, die viele verschiedene Ziel-Adressen besitzen. Genutzt wird diese Fähigkeit in MBONE, einem Breitbandnetz, das Audio- und Video-Daten überträgt.

IP: PIM-SM version 1 support (NEW) (CONFIG\_IP\_PIMSM\_V1)

Dieses ist der Treiber für die Kernel-seitige Unterstützung für PIM (Protocol Independent Multicast) in der Version eins. Dieses Protokoll ist relativ verbreitet, da es von Cisco benutzt wird. Weitere Information über PIM enthält <http://netweb.usc.edu/pim/>.

**IP: PIM-SM version 2 support (NEW) (CONFIG\_IP\_PIMSM\_V2)**

Die Version 2 von PIM ist relativ wenig verbreitet. Um diesen Treiber zu benutzen wird außerdem noch ein Daemon benötigt, entweder `pimd` oder `gated-5`.

**IP: ARP daemon support (CONFIG\_ARPD)**

Normalerweise hält der Kernel eine interne Tabelle, die die IP-Adressen den Hardware-Adressen des lokalen Netzwerks zuweist. Für kleine Netzwerke, die „nur“ einige hundert Adressen oder weniger besitzen, klappt diese Adreßverwaltung (Address Resolution Protocol, ARP) innerhalb des Kernels wunderbar. Die Tabelle innerhalb des Kernel wächst aber unaufhörlich mit jeder neuen Adresse. Dieses widerspricht für sehr große Netzwerke der LINUX-Philosophie und durch Aktivieren dieser Option enthält die Tabelle maximal 256 Einträge, wobei jeweils die ältesten Einträge gelöscht werden. Die Verwaltung dieser Einträge und der Verbindungsaufbau wird dem externen Daemon `arpd` übertragen. Der Treiber ist sowohl noch immer experimentell als auch schon wieder überholt.

**IP: TCP Explicit Congestion Notification support (CONFIG\_INET\_ECN)**

Explicit Congestion Notification (ECN) bedeutet die Erkennung von „Verkehrsstau“ im Paketverkehr. Dieser Treiber aktiviert ECN, was bedeutet, daß weniger Pakete verworfen werden und so der Durchsatz gesteigert wird. Die ECN-Unterstützung kann während des laufenden Betriebs deaktiviert werden. Anzumerken ist, daß derzeit noch viele schlecht konfigurierte Firewalls den Betrieb von ECN zurückweisen und man so leider mit aktiviertem ECN keinen Zugriff auf viele Internetseiten erhält.

**IP: TCP syncookie support (disabled per default) (CONFIG\_SYN\_COOKIES)**

Ein normales TCP/IP-Netzwerk ist anfällig für einen Hack-Angriff, der als SYN-Flooding bekannt ist. Das verschlüsselnde SYN-Cookie-Protokoll verhindert dieses.

## IP: Netfilter Configuration

Dieses ist ein weit ausladendes Submenü, das jede Menge Möglichkeiten zur Konfiguration des Netzfilters ermöglicht.

Connection tracking (required for masq/NAT) (CONFIG\_IP\_NF\_CONNTRACK)

Das sogenannte „Connection Tracking“ führt Buch, welche Pakete von dem Rechner an welchen Empfänger weitergereicht wurden. Dieses ist erforderlich, falls man Masquerading oder andere Formen der NAT benötigt, mit Ausnahme von Fast NAT. Der Treiber kann auch für das erweiterte Paket-Filtern benutzt werden (siehe Connection state match support weiter unten).

FTP protocol support (NEW) (CONFIG\_IP\_NF\_FTP)

Wer auch „Connection Tracking“ für FTP benötigt, braucht diesen Treiber. Er kann auch als Modul kompiliert werden.

Userspace queueing via NETLINK (EXPERIMENTAL)  
(CONFIG\_IP\_NF\_QUEUE)

Der Netzfilter kann nicht nur auf Systemebene, sondern auch auf Benutze-  
r-ebene im sogenannten User Space genutzt werden. Dieses ist der entspre-  
chende Treiber.

IP tables support (required for filtering/masq/NAT)  
(CONFIG\_IP\_NF\_IPTABLES)

IP-Tabellen sind ein mächtiges Hilfmittel, um den Paketfilter, Maskierung und NAT zu konfigurieren. Dieses ist der Treiber dazu.

limit match support (NEW) (`CONFIG_IP_NF_MATCH_LIMIT`)

Dieser Treiber ermöglicht das Einsetzen von definierten Limits, die maximal erreicht werden dürfen. Entweder im Zusammenhang mit `LOG target support` (s.u.) benutzt, oder um bösartige DOS-Attacken (Denial Of Service) abzuwehren.

MAC address match support (NEW) (`CONFIG_IP_NF_MATCH_MAC`)

Dieser Treiber ermöglicht das Filtern von Paketen anhand der eindeutigen MAC-Adresse der Netzwerkkarte.

netfilter MARK match support (NEW) (`CONFIG_IP_NF_MATCH_MARK`)

Mittels dieses Treibers kann anhand des Wertes `nfmark` das Filtern der Pakete entschieden werden. Dieser kann durch `MARK` (s.u.) gesetzt werden.

Multiple port match support (NEW) (`CONFIG_IP_NF_MATCH_MULTIPORT`)

Multiport ermöglicht, daß TCP- oder UDP-Pakete eine Reihe von verschiedenen Quell- oder Ziel-Ports besitzen dürfen; normalerweise ist nur ein einziger Bereich erlaubt.

TOS match support (NEW) (`CONFIG_IP_NF_MATCH_TOS`)

TOS bedeutet in diesem Fall „Type Of Service“. Anhand des in jedem IP-Paket vorhandenen Feldes, das die Art des Dienstes kennzeichnet, kann der Filter konfiguriert werden.

Connection state match support (NEW) (`CONFIG_IP_NF_MATCH_STATE`)

Hier wird der Verbindungsstatus der Pakete überprüft, und zwar anhand ihrer Beziehungen zu den vorher übermittelten Paketen.

Unclean      match      support      (EXPERIMENTAL)      (NEW)  
(CONFIG\_IP\_NF\_MATCH\_UNCLEAN)

Diese Option sieht sich alle relevanten Felder in den Headern der IP-, TCP-, UDP- und ICMP-Pakete an und sortiert ungültige und inkonsistente Daten aus.

Owner      match      support      (EXPERIMENTAL)      (NEW)  
(CONFIG\_IP\_NF\_MATCH\_OWNER)

Filter für Pakete des lokalen Netzwerks, die durch Benutzer, Benutzergruppe oder den zugehörigen Prozeß identifiziert werden.

Packet filtering (NEW) (CONFIG\_IP\_NF\_FILTER)

Hier wird eine Tabelle definiert, die Regeln für einen einfachen Paketfilter enthält für den lokalen Paket-Verkehr.

REJECT target support (NEW) (CONFIG\_IP\_NF\_TARGET\_REJECT)

Dieser Treiber ermöglicht, daß bestimmte Filterregeln definiert werden, und das Paket nicht stillschweigend entfernt wird, sondern daß eine Fehlermeldung an die Gegenseite gesendet wird.

MIRROR      target      support      (EXPERIMENTAL)      (NEW)  
(CONFIG\_IP\_NF\_TARGET\_MIRROR)

Dieser Treiber ermöglicht die Definition von Filter-Regeln. Falls diese nicht erfüllt werden, wird das eingehende Paket zum Absender zurückgeschickt.

Full NAT (NEW) (CONFIG\_IP\_NF\_NAT)

Die volle Network Address Translation ermöglicht das Maskieren, das Forwarden und andere Formen der Übersetzung. Zuständig sind die Einstellungen in der Tabelle `nat` in den IP-Tabellen. Mehr Details ergibt die Man-Page (8) `iptables`, die leider nicht allen Distributionen beigelegt ist.

MASQUERADE target support (NEW)  
`(CONFIG_IP_NF_TARGET_MASQUERADE)`

Das Maskieren ist ein spezieller Fall des NAT: alle nach außen gehenden Verbindungen scheinen von der selben Netzwerkadresse zu kommen. Dieser Treiber ist für den Fall gedacht, daß man TCP/IP über Wählverbindungen benutzt, die dynamische Zuweisungen von IP-Adressen verwenden, wie es z.B. bei vielen Internet-Providern üblich ist.

REDIRECT target support (NEW) `(CONFIG_IP_NF_TARGET_REDIRECT)`

Dieser Treiber leitet alle eingehenden Verbindungen auf eine einzige Adresse, so daß alle Datenpakete auf einer lokalen Maschine landen, statt durchgereicht zu werden. Dieses ist sinnvoll bei einem transparenten Proxy-Server.

Packet mangling `(CONFIG_IP_NF_MANGLE)`

Dieser Treiber fügt die Tabelle `mangle` zur IP-Tabelle hinzu. Die Tabelle wird benutzt, um Datenpakete zu verändern, was wiederum Auswirkungen auf die Route des Pakets haben kann.

TOS target support (NEW) `(CONFIG_IP_NF_TARGET_TOS)`

Diese Option fügt eine Markierung des Typs des Dienstes ein, die es ermöglicht in der Tabelle `mangle` des vorangegangenen Treibers Regeln zu schreiben, die das Feld TOS des IP-Pakets ändern, bevor es weitergeschickt wird.

MARK target support (NEW) `(CONFIG_IP_NF_TARGET_MARK)`

Diese Option fügt die Marke MARK hinzu, die es wiederum ermöglicht in der Tabelle `mangle` Regeln zu erzeugen, die das Feld `nfmark` verändern, bevor das Paket geroutet wird.

**LOG target support (NEW) (CONFIG\_IP\_NF\_TARGET\_LOG)**

Diese Option erzeugt die Marke LOG, die es ermöglicht, Regeln in der IP-Tabelle zu erzeugen, die in `syslog` dokumentiert werden.

**ipchains (2.2-style) support (CONFIG\_IP\_NF\_COMPAT\_IPCHAINS)**

Dieses ist die abwärtskompatible Option, die `ipchains` wieder im Kernel verankert. Dieses sollte nicht bei neuen Kernel geschehen, sondern nur um `ipchains` wie in der Kernelreihe 2.2.x zu benutzen.

**ipfwadm (2.0-style) support (CONFIG\_IP\_NF\_COMPAT\_IPFWADM)**

Dieses ist die abwärtskompatible Option, die `ipfwadm` wieder im Kernel verankert. Dieses sollte nicht bei neuen Kernel geschehen, sondern nur um `ipfwadm` wie in der Kernelreihe 2.0.x zu benutzen.

**The IPv6 protocol (CONFIG\_IPV6)**

Die Planungen für die Verkabelung der Welt laufen rapide, und wir werden es wohl noch erleben, daß Toaster, Waschmaschine, Toilettenspülung und wohlmöglich auch jede Milchbüte ihre eigene IP-Adresse erhalten. Damit ist der aus vier Bytes bestehende Adressraum aber definitiv zu klein: er bietet derzeit theoretisch ca. 4 Millarden Adressen, bei 6 Milliarden Menschen auf dieser Erde definitiv nicht zukunftssicher. Das neue Protokoll IPv6 bietet außer einem erweiterten Adressraum weitere Dienste sowie eine problemlosen Umgang mit dem bisherigen. Weitere Information bietet <http://playground.sun.com/pub/ipng/html/ipng-main.html>.

**IPv6: enable EUI-64 token format (CONFIG\_IPV6\_EUI64)**

Falls man sein Netzwerk auf das neue IPv6-P-Format umgestellt hat, so sollte man diese Option aktivieren.

IPv6: disable provider based addresses (NEW) (`CONFIG_IPV6_NO_PB`)

Linux versucht auch dann korrekt zu arbeiten, wenn das Netzwerk nur teilweise mit dem neuen Format Ipv6 arbeitet. Leider sind das bisherige und das neue Format inkompatibel. Falls Probleme aufgetreten sind, und das Netz komplett auf IPv6 umgestellt worden ist, sollte dieser Treiber eingebunden werden.

### IPv6: Netfilter Configuration

Dieses Submenü bietet analog zu obigen Optionen für Ipv4 auch einige Paketfilter für IPv6 an. Im einzelnen sind das:

IP6 tables support (required for filtering/masq/NAT)  
(`CONFIG_IP6_NF_IPTABLES`)

limit match support (NEW) (`CONFIG_IP6_NF_MATCH_LIMIT`)

netfilter MARK match support (NEW) (`CONFIG_IP6_NF_MATCH_MARK`)

Packet filtering (NEW) (`CONFIG_IP6_NF_FILTER`)

Packet mangling (NEW) (`CONFIG_IP6_NF_MANGLE`)

MARK target support (NEW) (`CONFIG_IP6_NF_TARGET_MARK`)

Kernel httpd acceleration (EXPERIMENTAL) (`CONFIG_KHTTPD`)

Dieser Treiber hat bereits große Diskussionen ausgelöst: ist es sinnvoll einen httpd-Beschleuniger auf Kernel-Ebene zu benutzen? Wie dem auch sei: dieser

Treiber ermöglicht ein teilweise schnelleres Reagieren als Web-Server. Derzeit werden nur Dateien aus dem Dateisystem weitergereicht, Anfragen, wie z.B. CGI-Scripts können nicht bearbeitet werden. Diese Anfragen werden transparent an den Web-Server weitergereicht, in aller Regel also wohl an Apache.

Falls man diesen Treiber als Modul kompiliert, so ist derzeit noch folgender Sicherheitsmechanismus eingebaut: nach dem Laden des Moduls muß außerdem noch eingegeben werden:

```
echo 1 > /proc/sys/net/khttpd/start
```

Folgendes ist derzeit bei dem Umgang mit dem kHTTPd zu beachten:

- Der Treiber ist in einem sehr frühen Stadium!
- Derzeit werden noch keine virtuellen Server unterstützt!
- Es sollte mindestens die Datei net/khttpd/README gelesen werden!
- Weiterführende Information bietet  
<http://www.fenrus.demon.nl>.

#### Asynchronous Transfer Mode (ATM) (EXPERIMENTAL) (CONFIG\_ATM)

ATM ist eine Hochgeschwindigkeits-Technologie für LANs und WANs. Es benutzt eine feste Paketgröße und ist verbindungsorientiert. Für ATM-Netze werden spezielle Netzwerkkarten benötigt. Der zugehörige Hardware-Treiber muß natürlich auch aktiviert werden. Weitere Information enthält Documentation/networking/atm.txt.

#### Classical IP over ATM (NEW) (CONFIG\_ATM\_CLIP)

Wer klassische IP-Kommunikation über ein ATM-Netz benutzen möchte, muß entweder diesen Treiber benutzen oder weiter unten den Treiber LAN Emulation.

Do NOT send ICMP if no neighbour (NEW) (`CONFIG_ATM_CLIP_NO_ICMP`)

Falls ein Nachbar auf einem ATM-Netz nicht erreichbar ist, wird normalerweise die Nachricht ICMP host unr eachable gesendet. Dieser Zustand kann allerdings auch auftreten, falls gerade die Tabelle ATMARP überprüft und reorganisiert wird. Dieser Treiber verwirft in diesem Fall die Pakete ohne weitere Fehlermeldung.

LAN Emulation (LANE) support (NEW) (`CONFIG_ATM_LANE`)

Die LANE-Emulation ermöglicht den normalen IP-Verkehr über ein bestehendes ATM-Netzwerk. Neben dem Betrieb als normaler Client kann mittels dieses Treibers ein Linux-Client auch als Proxy-Client-Brücke zwischen ELAN und den Ethernet-Segmenten fungieren. Wird auch benötigt für Multi-Protocol Over ATM (MPOA) (s.u.).

Multi-Protocol Over ATM (MPOA) support (NEW) (`CONFIG_ATM_MPOA`)

Ermöglicht Geräten, die am Ende des ATM-Netzes arbeiten, wie z.B. Routern oder Bridges, direkte Verbindungen über Subnetz-Grenzen hinweg, was den Netzwerk-Durchsatz erhöht.

The IPX protocol (`CONFIG_IPX`)

Notwendig, um eine Verbindung zu einem NOVELL-Server mit Hilfe des IPX-Protokolls herzustellen. Zum einen ist der Zugriff durch den LINUX-NOVELL-Client `ncpfs` möglich. Zum anderen ermöglicht der LINUX-DOS-Emulator den Zugriff auf den NOVELL-Server. Wer keinen NOVELL-Server in seine Umgebung integrieren möchte, antwortet `n`; wer nur selten auf den Server zugreifen will, bindet dieses mittels `m` als Modul ein.

IPX: Full internal IPX network (NEW) (CONFIG\_IPX\_INTERN)

Auch das Novell-IPX-Netzwerk benutzt Adressen für Netznoten, um jeden Rechner eindeutig zu identifizieren. Normalerweise geschieht dieses automatisch und dynamisch. Falls man eine definierte statische Adresse für seinen LINUX-Rechner im IPX-Netzwerk benötigt, sollte dieser Treiber aktiviert werden. Wer den Mars-Server benutzt, der einen Novell-Server emuliert, darf diese Option **nicht** aktivieren! Details findet man unter <ftp://metalab.unc.edu/pub/Linux/docs/HOWTO>.

Appletalk protocol support (CONFIG\_ATALK)

Appletalk ist das Netzwerk-Protokoll für Apple-Rechner. Ethertalk ist Appletalk via Ethernet und Localtalk ist Appletalk über serielle Schnittstellen. Wenn der LINUX-Rechner in ein Apple-Netzwerk integriert werden soll, muß hier mit y geantwortet werden. Wenn er dort außerdem als File- und Print-Server arbeiten soll, wird auch noch das Paket netatalk benötigt, das unter <http://artoo.hitchcock.org/~flowerpt/projects/LINUX-netatalk/> im WWW erhältlich ist.

DECnet Support (CONFIG\_DECNET)

Das DECnet-Protokoll wird in vielen Erzeugnissen von Digital Equipment – inzwischen übernommen von Compaq - benutzt. Es bietet einen verlässlichen Datenstrom und sequentielle Paket-Kommunikation mit einer großen Anzahl verschiedener Dienste; vergleichbar mit TCP/IP. Weitere Details zur Nutzung dieses Treibers findet man unter <http://linux.dreamtime.org/decnet/>; außerdem unter [Documentation/networking/decnet.txt](#) .

DECnet: router support (EXPERIMENTAL) (NEW)  
(CONFIG\_DECNET\_ROUTER)

Treiber, um einen DECnet-Endknoten in einen Router zu verwandeln (Level 1 oder 2). Er ist derzeit noch nicht voll implementiert und sollte nur von Entwicklern benutzt werden. Wer ihn dennoch benutzen möchte, sollte auch die Optionen `Kernel/User network link driver`, `Routing messages` und `Network packet filtering` aktivieren.

`DECnet: use FWMARK value as routing key (EXPERIMENTAL) (NEW)`  
`(CONFIG_DECNET_ROUTE_FWMARK)`

Hierdurch können in einem DECnet Datenpakete auf verschiedene Routen mittels verschiedener FWMARK (firewalling mark) geschickt werden.

`802.1d Ethernet Bridging (CONFIG_BRIDGE)`

Mit dieser Option kann der LINUX-Rechner als Ethernet-Bridge arbeiten, also verschiedene Ethernet-Segmente zusammenschließen, mit dem Ergebnis, daß der Benutzer nur noch ein Netz sieht. LINUX arbeitet entsprechend der Norm IEEE 802.1, so daß mit mehreren Bridges - auch anderer Hersteller - ein großes Netzwerk gebildet werden kann. Weitere Information bietet `Documentation/networking/bridge.txt`.

`CCITT X.25 Packet Layer (EXPERIMENTAL) (CONFIG_X25)`

X.25 ist ein Standard-Netzwerk-Protokoll um Wide Area Networks (WANs) zu verbinden. Man kann so zu diversen Rechnern, die auch dieses Protokoll fahren, jeweils eine Verbindung aufbauen, um Daten auszutauschen. Dieses Protokoll wird gerne von Regierungsbehörden und Banken für ein WAN genutzt. Das X.25-Protokoll besteht wiederum aus zwei verschiedenen Schichten: das höhere PLP (Packet Layer Protocol) und das LAPB. X.25-Verbindungen können folgendermaßen aufgebaut werden:

- über eine Standard-Telefon-Leitung mittels eines normalen Modems. Hierzu muß der Treiber `X.25 async driver` (s.u.) aktiviert werden

- über das Ethernet mit dem Protokoll 802.2 LLC. Dazu muß der Treiber 802.2 LLC (s.u.) aktiviert werden.
- über das Ethernet mit dem Protokoll LLC. Dafür müssen die Treiber LAPB Data Link Driver und LAPB over Ethernet (s.u.) aktiviert werden.

Bei Bedarf kann der Treiber auch als Modul geladen werden. Weitere Informationen enthalten die Dateien /usr/src/linux/Documentation/networking/x25.txt und /usr/src/linux/Documentation/networking/x25-iface.txt.

#### LAPB Data Link Driver (EXPERIMENTAL) (CONFIG\_LAPB)

Das LAPB-Protokoll (Link Access Procedure, Balanced) ist die tiefere Schicht des Protokolls X.25. Es ermöglicht stabile Verbindungen für den Datenaustausch mit anderen Rechnern. Außerdem ist es der Träger von höheren Transport-Schichten, meistens das X.25 PLP, aber andere sind auch möglich. Genaueres findet man unter /usr/src/linux/Documentation/networking/lapb-module.txt. Dieser Treiber kann auch als Modul kompiliert werden.

#### 802.2 LLC (EXPERIMENTAL) (CONFIG\_LLC)

Diese ist das Link Layer Protokoll für Verbindungen über X.25 und Ethernet mittels normaler Ethernet-Karten (s.o.).

#### Frame Divter (EXPERIMENTAL) (CONFIG\_NET\_DIVERT)

Der Frame Divter ermöglicht das Steuern von Datenpaketen, die nicht direkt zum Internet gelangen sollen. Z.B. kann man hierdurch einen Linux-Rechner, der als Ethernet-Bridge arbeitet, derart erweitern, daß ein echtes transparentes Caching ermöglicht wird, z.B. für einen Squid-Proxy. Dieses ist nützlich, wenn man die Konfiguration des Routers nicht ändern möchte oder kann, weil

man z.B. keinen Zugriff auf ihn besitzt. Weitere Informatiion enthalten die beiden folgenden Web-Adressen:  
<http://www.freshmeat.net/projects/etherdivert> und  
<http://perso.wanadoo.fr/magpie/EtherDivert.html>.

#### Acorn Econet/AUN protocols (EXPERIMENTAL) (CONFIG\_ECONET)

Econet ist ein stabiles, älteres aber langsamens Netzwerk-Protokoll, welches bei Rechnern der Firma Acorn zum Einsatz kam, um auf Server zuzugreifen. AUN ist die höhere Schicht des Econet, die über normale Ethernet-Verbindungen laufen kann, dazu aber das UDP Paket Protokoll benötigt (s.u.). Die Econet-Verbindungen können entweder über Acorns proprietäre Econet-Karten hergestellt werden, aber auch über Standard-Ethernet-Karten. Der Treiber kann auch als Modul kompiliert werden.

#### AUN over UDP (CONFIG\_ECONET\_AUNUDP)

Falls das Econet-Protokoll (s.o.) über normale Ethernet-Verbindungen betrieben werden soll, so wird dieser Treiber benötigt.

#### Native Econet (CONFIG\_ECONET\_NATIVE)

Falls man eine proprietäre Econet-Karte besitzt, muß diese Option aktiviert werden; in aller Regel die Option für Besitzer von Acorn-Rechnern.

#### WAN Router (CONFIG\_WAN\_ROUTER)

WANs (Wide Area Networks) werden benutzt, um mittels Standleitungen, X-25- oder Frame-Relay-Verbindungen lokale Netze (Local Area Networks, LANs) über größere Entfernung zu verbinden. Die Daten-Transfer-Raten liegen hier weit höher, als dieses des Modembetriebs des Privatbenutzers. Meistens wird spezielle Hardware eingesetzt, der WAN-Router, um das lokale Netz mit der weiten Welt zu verbinden. Auch hier bietet LINUX wiederum eine preisgünstige Alternative. Das WAN-Routing kann mit diesem Treiber in den

Kernel integriert werden und mit einer relativ preisgünstigen WAN-Steckkarte kann ein WAN-Router mit einem LINUX-Rechner realisiert werden. Weitere Informationen enthält /usr/src/linux/Documentation/networking/wan-router.txt. Außerdem werden einige Dienstprogramme benötigt, die unter folgender Adresse erhältlich sind: <ftp://ftp.sangoma.com>. Der Treiber kann auch als Modul kompiliert werden.

#### Fast switching (CONFIG\_NET\_FASTROUTE)

Dieser Treiber ermöglicht direkten, schnellen Datentransfer zwischen zwei Netzwerkkarten ohne weiteren Overhead. Auf der anderen Seite ist dieser Treiber definitiv inkompatibel zu vielen wichtigen Betzwerktreibern, wie CONFIG\_FIREWALL, CONFIG\_IP\_ROUTE\_TOS und CONFIG\_IP\_ROUTE\_FWMARK. Mit anderen Optionen von CONFIG\_IP\_ADVANCED\_ROUTER arbeitet dieser Treiber zusammen. Weiterführende Literatur findet man unter <ftp://ftp.inr.ac.ru/ip-routing/fastroute-fastroute-8390.tar.gz>.

#### Forwarding between high speed interfaces (CONFIG\_NET\_HW\_FLOWCONTROL)

Diese Option ermöglicht die Beschleunigung des Netzwerkkarten-Datentransfers, falls sich Daten stauen. Im Moment wird allerdings nur von der Karte Tulip 8390 unterstützt. Außerdem ist dieser Treiber gefährlich, da er langsamere Pentium-Rechner zum Absturz bringen kann. Der Treiber sollte also nicht aktiviert werden, falls er nicht unbedingt benötigt wird.

#### QoS and/or fair queueing (CONFIG\_NET\_SCHED)

Falls mehrere Datenpakete darauf warten, vom Kernel über das Netzwerk versendet zu werden, so muß der Kernel die Entscheidung treffen, in welcher Reihenfolge diese geschickt werden. Dieses kann insbesondere wichtig sein, wenn im Netz Echtzeitanwendungen auf ihre Daten warten. Normalerweise

erledigt dieses der `packet scheduler`. Falls der Standard-Scheduler für spezielle Bedürfnisse nicht ausreichend ist, so kann mit diesem Treiber Abhilfe geschaffen werden. Hier können die verschiedensten Algorithmen den einzelnen Netzwerkgeräten zugezwiesen werden, was allerdings Arbeit für den Experten ist. Eine oder mehrere der folgenden Optionen werden hierzu benötigt. Falls das virtuelle Dateisystem `/proc` aktiviert ist, kann man der Datei `/proc/net/psched` Informationen entnehmen. Weiterführende Information enthält <ftp://ftp.inr.ac.ru/ip-routing>.

#### CBQ packet scheduler (CONFIG\_NET\_SCH\_CBQ)

Der CBQ-Algorithmus (Class-Based Queueing) klassifiziert die Pakete und legt diese Hierarchie in einem Baum ab. Die einzelnen Blätter des Baums wiederum können von bestimmten Algorithmen, die in den folgenden Optionen aufgeführt sind, bearbeitet werden. Dieser Treiber kann auch als Modul kompiliert werden.

#### CSZ packet scheduler (CONFIG\_NET\_SCH\_CSZ)

Dieses ist der Treiber für den CSZ-Packet-Algorithmus (Clark-Shenker-Zhang), und momentan der einzige, der den notwendigen Durchsatz für Echtzeitanwendungen bietet. Dieser Treiber kann auch als Modul kompiliert werden.

#### The simplest PRIO pseudo scheduler (CONFIG\_NET\_SCH\_PRIO)

Dieses ist der Treiber für einen n-Band-Scheduler. Er kann auch als Modul geladen werden.

#### RED queue (CONFIG\_NET\_SCH\_RED)

Der RED-Algorithmus (Random Early Detection) ist ein weiterer Algorithmus zur Verteilung der Datenpakete. Er kann als Modul kompiliert werden.

SFQ queue (CONFIG\_NET\_SCH\_SFQ)

Der SFQ-Algorithmus (Stochastic Fairness Queueing) ist ein weiterer Algorithmus zur Verteilung der Datenpakete. Er kann aus als Modul kompiliert werden.

TEQL queue (CONFIG\_NET\_SCH\_TEQL)

Die TLE-Warteschlange (True Link Equalizer) kann mehrere physikalische Geräte als ein virtuelles Gerät verwalten. Dieser Treiber kann auch als Modul kompiliert werden.

TBF queue (CONFIG\_NET\_SCH\_TBF)

Der TBF-Algorithmus (Token Bucket Filter) ist ein weiterer Algorithmus Verteilung der Datenpakete. Er kann aus als Modul kompiliert werden.

QoS support (CONFIG\_NET\_QOS)

Hiermit kann das Leistungsmerkmal „Quality of Service“ aktiviert werden, womit ein Minimum an Datentransfer für bestimmte Geräte garantiert wird. Dieses ist kein Treiber, sondern nur eine Abfrage, die die Fragen nach den folgenden Treiber aktiviert.

Rate estimator (NEW) (CONFIG\_NET\_ESTIMATOR)

Falls man „Quality of Service“ benutzen möchte, so muß natürlich als erstes Information über den Datendurchsatz von Netzwerk-Geräten besorgt werden. Dieser Treiber erledigt das.

Packet classifier API (CONFIG\_NET\_CLS)

Der CBQ-Algorithmus (s.o.) erfordert, daß Datenpakete von Netzwerkverbindungen klassifiziert werden. Falls dieses gewünscht wird, werden im Folgenden einige Fragen zur Art- und Weise der Klassifizierung aktiviert.

Routing table based classifier (NEW) (`CONFIG_NET_CLS_ROUTE4`)

Hierdurch ist es möglich, Datenpakete anhand eines passenden Eintrags in der Routing-Tabelle zu verschicken.

Firewall based classifier (NEW) (`CONFIG_NET_CLS_FW`)

Hierdurch können Datenpakete anhand der spezifizierten Firewall-Kriterien entsprechend verschickt werden.

U32 classifier (NEW) (`CONFIG_NET_CLS_U32`)

Hierdurch können versendete Datenpakete klassifiziert werden anhand ihrer Zieladresse.

Special RSVP classifier (NEW) (`CONFIG_NET_CLS_RSVP`)

Das RSVP (Resource Reservation Protocol) ermöglicht einem Client die minimale und maximale Datentransferrate zu spezifizieren. Dieses ist wichtig für die Übertragung von Echtzeit-Daten, wie z.B. Video oder Audio.

Special RSVP classifier for IPv6 (NEW) (`CONFIG_NET_CLS_RSVP6`)

Wie vorangehender Treiber, aber für die neuen IPv6-Netze gedacht.

## 11. Telephony Support

In weiser Voraussicht hat man dem Telefonieren via Internet einen ganzen Menüpunkt spendiert. Auch wenn die Auswahl derzeit eher schmal ist: der Anfang ist gemacht.

Linux telephony support (CONFIG\_PHONE)

Dieser Treiber ist gedacht für den Einsatz einer Telefon-Steckkarte für den Linux-Rechner, der das Telefonieren über IP-Verbindungen ermöglicht.

QuickNet Internet LineJack/PhoneJack support (CONFIG\_PHONE\_IJXJ)

Die Firma Quicknet Technologies stellt verschiedene Karten zum Telefonieren über das Internet her. Dieses ist der Treiber u.a. für die Karten Internet Phone-JACK und Internet LineJACK. Es gibt von diesen Karten auch noch ältere Modelle für den ISA-Bus. Auch diese können mit diesem Treiber betrieben werden. Für den Umgang mit diesem Treiber ist der Text [Documentati-](#)  
[on/telephony/ijxj.txt](#) hilfreich. Technische Information über die Karten bietet der Hersteller unter <http://www.quicknet.net/>.

## 12. ATA/IDE/MFM/RLL support

ATA/IDE/MFM/RLL support (CONFIG\_IDE)

Dieses ist der grundlegende Treiber für alle preisgünstigen Massenspeicher, die unter den Bezeichnungen ATA, IDE, EIDE und ATAPI verkauft werden; meistens Festplatten oder CD-ROM-Laufwerke. Diesen Treiber benötigt fast jeder Benutzer, es sei denn, sämtliche Massenspeicher des Rechners arbeiten auf SCSI-Basis oder der Rechner besitzt keine lokalen Massenspeicher.

Die einzelnen Spezifikationen im Einzelnen sind:

- IDE (Integrated Disk Electronics). Der grundlegende Standard, den Western Digital und Compaq 1984 geschaffen haben.
- AT Attachment (ATA) ist eine spätere Obermenge der IDE-Spezifikation.
- ATA-2 wird auch Fast ATA oder Fast IDE genannt.
- ATA-3 ist Enhanced IDE, abgekürzt EIDE. Hiermit können mittels des LBA-Standards maximal 8.4 GB pro Festplatte angesprochen werden. Außerdem kann der Controller jetzt 4 (bisher nur 2) Massenspeicher ansprechen.
- ATA-4 wird auch unter UDMA/33 bzw. UltraDMA/33 geführt und ermöglicht einen schnelleren Datentransfer zwischen Festplatte und CPU mittels schneller DMA-Controller.
- ATAPI ist das ATA Packet Interface, daß in der Hauptsache benutzt wird, um EIDE-CD-ROM-Laufwerke und -Bandlaufwerke anzusteuern.
- SMART IDE ist die Abkürzung für „Self Monitoring, Analysis and Reporting Technology“ und ist gedacht, um etwaige Hardwarefehler rechtzeitig zu erkennen, um die Möglichkeit eines Datenverlustes zu minimieren.

Weitere Information enthält der Text Documentation/ide.txt.

Enhanced IDE/MFM/RLL disk/cdrom/tape support (CONFIG\_BLK\_DEV\_IDE)

Erlaubt die Steuerung von bis zu zehn IDE-Schnittstellen, also bis zu einer Kombination aus zwanzig IDE-Festplatten, -CD-ROM-Laufwerken oder -Bandlaufwerken.

Use old disk-only driver on primary interface (CONFIG\_BLK\_DEV\_HD\_IDE)

Treiber für die erste IDE-Schnittstelle, der keine IDE-/ATAPI-CD-ROM-Laufwerke und keine Festplatten > 504 MByte unterstützt. Dieses kann nützlich sein, um alte MFM- oder RLL-Platten am ersten Controller zu installieren, während neuere IDE-Geräte an der zweiten bis vierten Schnittstelle von dem erweiterten Treiber namens CONFIG\_BLK\_DEV\_IDE betrieben werden.

Include IDE/ATA-2 DISK support (CONFIG\_BLK\_DEV\_IDEDISK)

Dieses ist der neue Treiber für MFM/RLL/IDE-Festplatten sowie CD-ROMs. Falls keine speziellen Gründe vorliegen, den alten Treiber (CONFIG\_BLK\_DEV\_IDE) zu benutzen, sollte dieses der Treiber der Wahl sein, zumal er eine große Anzahl an neuen Leistungsmerkmalen bietet. Details findet man unter /usr/src/linux/Documentation/ide.txt. Falls das Root-Verzeichnis / auf einer IDE-Platte liegt, sollte dieser Treiber keinesfalls als Modul kompiliert werden.

Use multi-mode by default (CONFIG\_IDEDISK\_MULTI\_MODE)

Falls Sie des öfteren eine dieser Fehlermeldungen sehen, verspricht dieser Treiber Abhilfe:

```
hda: set_multmode: status=0x51 {DriveReady SeekComplete Error}
hda: set_multmode: error=0x04 { DriveStatusError }
```

Prinzipiell sollte dieses nicht der Fall sein; die Standard-Antwort ist also n.

PCMCIA IDE support (CONFIG\_BLK\_DEV\_IDECS)

Für den Betrieb von IDE-Geräten über eine PCMCIA-Karte.

Include IDE/ATAPI CDROM support (CONFIG\_BLK\_DEV\_IDECD)

Das ATAPI-Protokoll wird von CD-ROM- und Bandlaufwerken genutzt; die Geräte werden an einem EIDE-Controller angeschlossen. Fast alle neueren CD-ROM-Laufwerke, die nicht mittels SCSI betrieben werden, benutzen dieses Protokoll.

Include IDE/ATAPI TAPE support (CONFIG\_BLK\_DEV\_IDETAPE)

Notwendig für Bandlaufwerke mit ATAPI-Protokoll am EIDE-Controller. Nach dem Booten werden diese Laufwerke wie EIDE-Festplatten identifiziert, also z.B. als `hdc` oder `hdd` etc.

Include IDE/ATAPI FLOPPY support (CONFIG\_BLK\_DEV\_IDEFLOPPY)

Für IDE-Diskettenlaufwerke, die das ATAPI-Protokoll benutzen. Prominente Vertreter dieser Gattung sind z.B. das LS-120 und das ATAPI-ZIP-Laufwerk. Während des Hochfahrens werden diese Laufwerke als weiteres IDE-Laufwerk erkannt, also z.B. als Laufwerk `hdc`.

SCSI emulation support (CONFIG\_BLK\_DEV\_IDESCSI)

Dieser Parameter aktiviert die Emulation eines SCSI-Treibers für ein ATAPI-Laufwerk anstelle eines direkten ATAPI-Treibers. Dieses ist nur nötig, falls kein spezieller ATAPI-Treiber für das Gerät geschrieben worden ist und man so einen vorhandenen SCSI-Treiber benutzen kann.

IDE chipset support/bugfixes

Hier werden jetzt im Folgenden die defekten Chipsätze behandelt, für die es jeweils Treiber gibt, die einen Betrieb trotz der Bugs ermöglichen.

CMD640 chipset bugfix/support (CONFIG\_BLK\_DEV\_CMD640)

Der CMD640-Chipsatz von CMD-Technologies wird auf vielen 486er- und Pentium-Motherboards benutzt. Unglücklicherweise besitzt er einige Designfehler, die zu Datenverlust oder -zerstörung führen können. Liegt ein solcher Chipsatz vor, sollte mit `y` geantwortet werden. Bei Systemen mit PCI-Bus arbeitet diese Option dann automatisch. Wenn allerdings ein VESA-Local Bus auf dem System vorhanden ist, muß auch noch als Boot-Parameter `ide0=cmd640_vlb` übergeben werden.

CMD640 enhanced support (CONFIG\_BLK\_DEV\_CMD640\_ENHANCED)

Diese Option unterstützt das Setzen und Auto-Tuning des PIO-Modus der CMD640 IDE-Schnittstelle. Falls das BIOS dieses nicht bereits erledigt, sollte dieser Treiber eingebunden werden.

ISA-PNP EIDE support (CONFIG\_BLK\_DEV\_ISAPNP)

Falls eine ISA-EIDE-Karte mit Plug and Play betrieben wird und diese konfiguriert werden muß, bevor nach anderen PnP-Geräten gescannt wird, braucht man diesen Treiber.

RZ1000 chipset bugfix/support (CONFIG\_BLK\_DEV\_RZ1000)

Der RZ1000-Chipsatz von PC-Technologies wird auf vielen 486er- und Pentium-Motherboards eingesetzt. Leider ist er fehlerhaft. Falls ein derartiger Chipsatz vorhanden ist, sollte man `y` antworten, was zwar den Durchsatz etwas vermindert, die Fehlerhäufigkeit aber dramatisch senkt.

Generic PCI IDE chipset support (CONFIG\_BLK\_DEV\_IDEPCI)

Falls der Rechner einen PCI-Bus und IDE-Festplatten besitzt, sollte man diesen Treiber aktivieren. Er erkennt automatisch diese Festplatten und spart somit weiteres Konfigurieren.

**Sharing PCI IDE interrupts support (CONFIG\_IDEPCI\_SHARE\_IRQ)**

Einige ATA-/IDE-Chipsätze unterstützen die gemeinsame Benutzung eines einzelnen IRQ mit anderen Karten. Dieser Treiber bietet die Unterstützung von der Software-Seite.

**Generic PCI bus-master DMA support (CONFIG\_BLK\_DEV\_IDEDMA\_PCI)**

Falls ein Rechner mit PCI in der Lage ist, mittels DMA auf die Festplatten zuzugreifen, so sollte man hier mit `y` antworten, da dieses die Zugriffszeiten reduziert. Nachdem man diesen Treiber aktiviert hat, kann man mittels des Hilfsprogramms `hdparm` DMA bei jenen Platten aktivieren, wo dieses nicht automatisch geschehen ist. Normalerweise wird aus Sicherheitsgründen der DMA-Zugriff nicht automatisch aktiviert; dieses kann jedoch mittels der nächsten Option `use DMA by default when available` (s.u.) geschehen. Weitere Information enthält Documentation/ide.txt.

**Boot off-board chipsets first support (CONFIG\_BLK\_DEV\_OFFBOARD)**

Die auf dem Mutterbrett integrierten IDE-Festplatten-Controller erhalten normalerweise die Bezeichnungen `ide0` und `ide1`, während Controller auf zusätzlichen PCI-Steckkarten die Kennungen `ide2` und `ide3` erhalten. Dieses kann zu Konflikten führen falls von dem Steckkarten-Controller und mittels LILO gebootet werden soll. Diese Option vertauscht die Kennungen der Controller: die Steckkarten-Controller erhalten die Kennungen `ide0` und `ide1`, während die Onboard-Controller `ide2` und `ide3` sind. Hierdurch wird auch die Reihenfolge der Festplattenkennungen (`hdx`) verändert, und also muß die Datei `/etc/fstab` entsprechend geändert werden.

**Use PCI DMA by default when available (CONFIG\_IDEDMA\_PCI\_AUTO)**

Vor der Kernel-Version 2.1.112 startete Linux automatisch die Benutzung von DMA für IDE-Laufwerke, wenn der Chipsatz dieses unterstützte. Daraufhin gab es viele Beschwerden über Zwischenfälle, bei denen defekte Chipsätze

Datenverlust und andere Schäden verursachten. Deshalb wird seitdem **nicht** mehr DMA automatisch angeschaltet. Falls man sich sicher ist, daß die Hardware fehlerfrei läuft, kann man diesen Treiber benutzen. Definitiv sollte man diesen Treiber nicht mit dem Chipsatz VIA VP2 benutzen.

#### ATA Work(s) In Progress (EXPERIMENTAL) (CONFIG\_IDEDMA\_PCI\_WIP)

Nicht anfassen, da dieses noch eine Baustelle ist. Stellt sich die Frage, wieso sich dieser Treiber bereits ins Konfigurationsscript verirrt hat.

Good-Bad                  DMA                  Model-Firmware                  (WIP)  
(CONFIG\_IDEDMA\_NEW\_DRIVE\_LISTINGS)

Dieser experimentelle Treiber versucht fehlerhafte DMA-Zugriffe zu vermeiden, indem er die Revisionsnummer von Hardware und Firmware von Festplatten ausliest und mit einer empirischen Liste von „guten“ und „bösen“ Festplatten vergleicht, und danach entscheidet, ob ein DMA-Zugriff stattfinden soll.

#### AEC62XX chipset support (CONFIG\_BLK\_DEV\_AEC62XX)

Diese Steckkarte enthält einen bootfähigen PCI-UDMA-Controller, der maximal vier weitere EIDE-Geräte betreiben kann, die sich alle einen einzigen Interrupt teilen. Außerdem sollte der Treiber `use DMA by default when available` aktiviert werden. Folgende Chipsätze werden unterstützt:

ATP850U/UF ist UltraDMA 33 für PCs. ATP860 ist UltraDMA 66 für PCs. ATP860M ist UltraDMA 66 für Macs. Weitergehend Information enthält der Quelltext `drivers/ide/aec62xx.c`.

#### AEC62XX Tuning support (CONFIG\_AEC62XX\_TUNING)

Zum Tunen des vorangegangenen Treibers. Weiterer Information enthalten die Kommentare im Quelltext `drivers/ide/aec62xx.c`.

**ALI M15x3 chipset support (CONFIG\_BLK\_DEV\_ALI15X3)**

Dieser Treiber unterstützt den DMA-Zugriff der Chipsätze ALI 1533, 1543 und 1543C, die auf einigen Mutterbrettern vorhanden sind. Falls dieser Treiber aktiviert wird, sollte zusätzlich der Treiber CONFIG\_IDEDMA\_AUTO eingebunden werden. Im Zweifelsfalle *n*.

**ALI M15x3 WDC support (DANGEROUS) (CONFIG\_WDC\_ALI15X3)**

Ein gefährlicher Treiber, der UltraDMA-Unterstützung für Ali15x3 bietet, aber die Prüfsummenkontrolle deaktiviert. Gedacht für Hassadeure, die Transfer-Raten von ATA-4/5 auf ATA-2-Strukturen erzielen wollen. Man sollte mit Messer und Gabel essen, und nicht mit Hammer und Meissel! Nein, nein und nochmals nein!

**AMD Viper support (CONFIG\_BLK\_DEV\_AMD7409)**

Dieser Treiber aktiviert den Unterstützung für (U)DMA für den Chipsatz AMD756 Viper. Außerdem sollte noch die Option `Use DMA by default when available` (s.o.) aktiviert werden. Weitere Information enthält der Quelltext `drivers/ide/amd7409.c`.

**AMD Viper ATA-66 Override (WIP) (CONFIG\_AMD7409\_OVERRIDE)**

Diese Option setzt die Kennung für ATA-66. Dieses kann allerdings auch erfolgen durch den Bootparameter `idex=ata66`.

**CMD64X chipset support (CONFIG\_BLK\_DEV\_CMD64X)**

Treiber für IDE-Controller mit einem der folgenden Chipsätze: CMD643, CMD646 oder CMD648.

**CY82C693 chipset support (CONFIG\_BLK\_DEV\_CY82C693)**

Dieser Treiber bietet automatische Erkennung und Unterstützung des Chipsatzes CY82C693, der auf den 164SX-Brettern der PC-Alphas von Digital Equipment. Außerdem muß der Treiber Use DMA by default when available (s.o.) aktiviert werden.

Cyrix CS5530 MediaGX chipset support (CONFIG\_BLK\_DEV\_CS5530)

Der Treiber für UDMA für den Chipsatz Cyrix MediaGX 5530 inklusive automatischer Erkennung und Konfiguration.

HPT34X chipset support (CONFIG\_BLK\_DEV\_HPT34X)

Karten mit diesen Chipsätzen können zusätzlich vier weitere EIDE-Geräte über einen einzigen Interrupt ansteuern. Der Chipsatz HPT343 ist nicht bootfähig, während die Chipsätze HTP345 und HTP363 booten können, dazu allerdings einen BIOS-Patch benötigen. Der Hersteller behauptet, daß mit dieser Karte auch DVD II Laufwerke angesteuert werden können. Selten vorhanden; für den Standardbenutzer also n.

HPT34X AUTODMA support (WIP) (CONFIG\_HPT34X\_AUTODMA)

Dieser Treiber soll einmal die Auto-DMA-Unterstützung bieten. Derzeit noch eine Baustelle. Experimentierfreudige lesen die Kommentare in drivers/ide/hpt34x.c, alle anderen antworten mit n.

HPT366 chipset support (CONFIG\_BLK\_DEV\_HPT366)

Der Treiber für folgende Chipsätze: HPT366 ist der Ultra DMA Chipsatz für ATA-66. HPT368 ist der Ultra DMA Chipsatz für ATA-66 auf RAID-Basis. HPT370 ist der Ultra DMA Chipsatz für ATA-100. Dieser Treiber ermöglicht es bis zu 4 EIDE-Geräte an einem Interrupt zu betreiben.

Der Chipsatz HPT366 ist nur mit einigen Tricks bootfähig. Die eine Möglichkeit ist ihn mit speziellen Befehlen des LILO auf Gerät 0x80 zu referenzieren. Die

andere Möglichkeit ist die Aktivierung des obigen Treibers Boot off-board chipsets first support (CONFIG\_BLK\_DEV\_OFFBOARD). Der Hersteller erwähnt außerdem, daß an dieser Hardware DVD-II-Laufwerke betrieben werden können.

#### Intel PIIxN chipsets support (CONFIG\_BLK\_DEV\_PIIX)

Das BIOS setzt teilweise ungünstige Standardeinstellungen für die PIO-Modi 0 bis 4. Dieser Treiber ermöglicht für die Intel Chipsätze PIIxN das dynamische Einstellen der PIO-Modi mittels des Hilfsprogramms hdparm. Für die meisten Anwender n.

#### PIIxN Tuning support (CONFIG\_PIIX\_TUNING)

Dieser Treiber fügt dem obigen Treiber für die IDE-Geräte von Intel weitere Fähigkeiten hinzu. Da das BIOS in einigen Fällen bestimmte Kombinationen von Geräten und Adapters nicht sauber bearbeiten kann, ist es notwendig geworden, die Geschwindigkeiten der Geräte anzupassen. So muß im Falle PII3 430HX/440FX die Geschwindigkeit von UDMA auf DMA 2 reduziert werden, wenn das BIOS die Initialisierung nicht sauber abwickelt.

#### NS87415 support (EXPERIMENTAL) (CONFIG\_BLK\_DEV\_NS87415)

Der Chipsatz NS87415 wird u.a. in SPARC64-Maschinen verwendet. Dieser Treiber bietet die entsprechende Software-Unterstützung.

#### OPTi 82C621 chipset enhanced support (EXPERIMENTAL) (CONFIG\_BLK\_DEV\_OPTI621)

Dieser Treiber unterstützt den OPTi-Controller 82C621.

#### PROMISE PDC20246/PDC20262/PDC20267 support (CONFIG\_BLK\_DEV\_PDC202XX)

Dieser Treiber unterstützt drei verschiedenen Zusatzkarten:

- Promise Ultra33 mit dem Chipsatz PDC20246. Diese Karte ist ein bootfähiger PCI-UDMA-Controller, der vier zusätzliche EIDE-Geräte über einen Interrupt ansteuern kann. Wenn mehrere dieser Karten installiert werden - maximal drei - kann es zu Konflikten kommen, da es unterschiedliche Revisionsnummern des BIOS gibt. Falls man nicht die Möglichkeit besitzt, eine neues BIOS zu brennen, versucht der Treiber während des Systemstarts die maximale Geschwindigkeit der Laufwerke zu erzeugen. Falls mehr als eine dieser Karten angesteuert werden soll, so ist das Ultra33 BIOS 1.25 oder eine neuere Version erforderlich.
- Promise Ultra66 mit dem Chipsatz PDC20262. Diese Karte ist ein bootfähiger PCI-UDMA-Controller, der vier zusätzliche EIDE-Geräte über einen Interrupt ansteuern kann. Der Treiber versucht während des Bootens den Chipsatz für maximale Geschwindigkeit anzusteuern. Bisher ist maximal UDMA-2 getestet worden. Außerdem muß zusätzlich der Treiber CONFIG\_IDEDMA\_AUTO aktiviert werden.
- Außerdem wird inzwischen auch die Promise Ultra100 mit dem Chipsatz PDC20265/PDC20267 unterstützt.

#### Special UDMA Feature (CONFIG\_PDC202XX\_BURST)

Für den vorangehenden Treiber und die Chips PDC20246, PDC20262, PDC20265 und PDC20267. Ursprünglich entwickelt für den PDC20246 mit Ultra33, der Konflikte mit dem BIOS erzeugte, wenn drei oder mehr Karten benutzt werden.

#### SiS5513 chipset support (CONFIG\_BLK\_DEV\_SIS5513)

Dieser Treiber aktiviert die (U)DMA-Übertragung für Mutterbretter, die auf dem SiS5513 basieren. Die Chips SiS620 und SiS5600/5597 werden nicht unter-

stützt. Der Treiber CONFIG\_IDEDMA\_AUTO muß für diesen Chip aktiviert werden.

#### SLC90E66 chipset support (CONFIG\_BLK\_DEV\_SLC90E66)

Dieser Treiber unterstützt (U)DMA für die Southbridge Victroy66 mit den Intel Northbridges. Der Chipsatz ist für Ultra66 gedacht. Das Praktische an diesem Treiber ist, daß UDMA-, DMA- und PIO-Geräte gleichzeitig betrieben werden können.

Tekram TRM290 chipset support (EXPERIMENTAL)  
(CONFIG\_BLK\_DEV\_TRM290)

Dieser Treiber unterstützt den den DMA-Transfer des Chipsatzes Tekram TRM290 für PCI/IDE.

#### VIA82CXXX chipset support (CONFIG\_BLK\_DEV\_VIA82CXXX)

Für die Besitzer einer Apollo: hiermit können die Chipsätze VIA82Cxxx für den (U)DMA-Transfer optimiert werden, denn oft ist der zweite DMA-Kanal im BIOS nicht aktiviert. Zum einen ist es möglich, einen Boot-Parameter zu übergeben. Falls diese nicht geschieht, so wird der Treiber automatisch versuchen, die beste Konfiguration herzustellen. Falls das virtuelle Dateisystem /proc aktiviert worden ist, so sind Informationen in der Datei /proc/ide/via erhältlich. Dazu muß der Parameter DISPLAY\_APOLLO\_TIMINGS gesetzt sein. Genaueres findet man im Source-Code in /usr/src/linux/drivers/block/via82cxxx.c. Außerdem muß der Treiber CONFIG\_IDEDMA\_AUTO aktiviert sein. Stinknormale PC-Benutzer sagen n.

#### Other IDE chipset support (CONFIG\_IDE\_CHIPSETS)

Falls man bisher seinen exotischen Chipsatz noch nicht gefunden hat, so kann man hier mit y antworten, und weiter suchen. In der Hauptsache geht es hier

um zusätzliche Festplatten-Controller und das Tunen von Chipsätzen. Viele dieser Treiber benötigen Bootparameter während des Startens. Genauereres findet man in /usr/src/linux/Dokumentation/ide.txt.

**Generic 4 drives/port support (NEW) (CONFIG\_BLK\_DEV\_4DRIVES)**

Einige ältere Chipsätze, wie z.B. der Tekram 690CD benutzen den I/O-Port 0x1F0 um bis zu vier Laufwerke anzusteuern. Normalerweise werden hier nur zwei Laufwerke angesteuert. Dieser Treiber kann auch mit derartigen Chipsätzen umgehen. Außerdem muß als Bootparameter `ide0=four` eingegeben werden.

**ALI M14xx support (CONFIG\_BLK\_DEV\_ALI14XX)**

Diese Option unterstützt die zweite IDE-Schnittstelle der Chipsätze ALI M1439/1443/1445/1487/1489 und ermöglicht schnelleren I/O-Transfer. Als Kernel-Parameter muß `ide0=ali14xx` eingegeben werden.

**DTC-2278 support (CONFIG\_BLK\_DEV\_DTC2278)**

Dieser Treiber benötigt den Kernel-Boot-Parameter `ide0=dtc2278`. Er ermöglicht den Zugriff auf die zweite IDE-Schnittstelle der DTC-2278-Karte und erlaubt außerdem schnellere I/O-Geschwindigkeiten.

**Holtek HT6560B support (CONFIG\_BLK\_DEV\_HT6560B)**

Dieser Treiber ermöglicht den Zugriff auf die zweite IDE-Schnittstelle der Holtek-Karte und ermöglicht größeren I/O-Durchsatz. Als Kernel-Boot-Parameter muß `ide0=ht6560b` eingegeben werden.

**PROMISE DC4030 support (CONFIG\_BLK\_DEV\_PROMISE)**

Dieser Treiber ermöglicht den Betrieb eines zweiten Controllers für diesen Chipsatz. Außerdem benutzt er die Cache-Fähigkeiten der Karte. CD-ROM-

und Bandlaufwerke werden derzeit noch nicht unterstützt. Der Kernel-Boot-Parameter lautet `ide0=dc4030`.

`QDI QD6580 support (CONFIG_BLK_DEV_QD6580)`

Dieser Treiber erlaubt höhere I/O-Geschwindigkeiten. Als Kernel-Boot-Parameter muß `ide0=qd6580` eingegeben werden.

`UMC 8672 support (CONFIG_BLK_DEV_UMC8672)`

Dieser Treiber wird mit Hilfe des Boot-Parameters `ide0=umc8672` aktiviert. Er ermöglicht die Benutzung des zweiten Controllers und höhere I/O-Raten.

#### Weitere blockorientierte Geräte

`IGNORE word93 Validation BITS (NEW) (CONFIG_IDEDMA_IVB)`

Dieser Treiber wurde erforderlich aufgrund unklarer Spezifikationen in ATA-4 und ATA-5. Es geht um die klare Erkennung von Signalen.

## 13. SCSI support

`SCSI support (CONFIG_SCSI)`

Falls eine SCSI-Festplatte, ein SCSI-Bandlaufwerk, ein SCSI-CD-ROM-Laufwerk oder ein anderes SCSI-Gerät in dem Rechner vorhanden ist, und unter LINUX genutzt werden soll, muß hier mit `y` geantwortet werden. Außerdem sollte man jetzt Hersteller und Typenbezeichnung des SCSI-Host-Adapters feststellen, da er in Kürze abgefragt wird. Auch die Version des evtl. an der parallelen Schnittstelle angeschlossenen IOMEGA-ZIP-Laufwerks (100 MByte) wird hier behandelt. Falls nur SCSI-Geräte im Einsatz sind, die nicht

permanent genutzt werden, z. B. Bandlaufwerke, besteht die Möglichkeit, die SCSI-Unterstützung und die Karten-Treiber als Modul zu laden.

SCSI disk support (CONFIG\_BLK\_DEV\_SD)

Falls eine SCSI-Festplatte oder ein IOMEGA-ZIP-Laufwerk an der parallelen Schnittstelle betrieben werden soll, muß hier mit y geantwortet werden. Diese Option ist nicht für CD-ROM-Laufwerke zuständig.

Maximum number of SCSI disks that can be loaded as modules (CONFIG\_SD\_EXTRA\_DEVS)

Hier muß die maximale Anzahl von SCSI-Festplatten angegeben werden, für die Platz in entsprechenden Tabellen vorrätig gehalten wird, nachdem der Kernel gebootet wird. Diese Eingabe ist unter Linux immer noch notwendig, da die dynamische Änderung dieses Speicherplatzes eine Fülle von Problemen aufwirft. Die Programmierer versprechen Abhilfe, und zwar „irgendwann“. Die Grundeinstellung ist 40 und sollte normalen Ansprüchen genügen.

SCSI tape support (CONFIG\_CHR\_DEV\_ST)

Für SCSI-Bandlaufwerke unter LINUX. Nicht für CD-ROM-Laufwerke!

SCSI CD-ROM support (CONFIG\_BLK\_DEV\_SR)

Für SCSI-CD-ROM-Laufwerke unter LINUX.

Enable vendor-specific extensions (for SCSI CDROM) (NEW)  
(CONFIG\_BLK\_DEV\_SR\_VENDOR)

Dieser Treiber aktiviert einige Hersteller-spezifische SCSI-Befehle. Dieses ist nur notwendig, falls man Unterstützung für Multisession-CDs benötigt und ältere NEC und Toshiba-Laufwerke besitzt; außerdem für die HP-CD-Brenner. Besitzer anderer SCSI-Laufwerke antworten mit n.

Maximum number of CDROM devices that can be loaded as modules (NEW)  
(CONFIG\_SR\_EXTRA\_DEVS)

Die maximale Anzahl von CD-ROM-Treibern, die als Modul geladen werden können. Die Grundeinstellung ist 2 und sollte normalen Ansprüchen genügen.

SCSI generic support (CONFIG\_CHR\_DEV\_SG)

Falls irgendein SCSI-Gerät angeschlossen werden soll, das nicht in die obigen Kategorien Festplatten, Bandlaufwerke oder CD-ROMs fällt, also z. B. Scanner, Synthesizer oder CD-Brenner, ist dieser Treiber notwendig. Diese Geräte werden nicht direkt vom Kernel unterstützt, sondern benötigen zusätzliche Software, die das SCSI-Protokoll verwaltet.

Enable extra checks in new queueing code  
(CONFIG\_SCSI\_DEBUG\_QUEUES)

Diese Option aktiviert eine Menge zusätzlich Überprüfungen ob Daten in der Queue konsistent sind. Dieses berührt zwar die Performance auf der anderen Seite, aber etwaige Fehler werden früher entdeckt, wenn dieser Treiber aktiviert ist. Weitere Information ist erhältlich unter [http://www.andante.org/scsi\\_queue.html](http://www.andante.org/scsi_queue.html).

Probe all LUNs on each SCSI device (CONFIG\_SCSI\_MULTI\_LUN)

Falls ein SCSI-Gerät vorhanden ist, das mehr als eine LUN (Logical Unit Number) unterstützt, z. B. ein CD-Wechsler („Jukebox“), sollte man hier mit y antworten, damit alle logischen Geräte vom Treiber gefunden werden. Ein SCSI-Gerät mit mehreren LUNs arbeitet dann logisch wie mehrere SCSI-Geräte.

Verbose SCSI error reporting (kernel size +=12K)  
(CONFIG\_SCSI\_CONSTANTS)

Dieser Treiber gibt Fehlermeldungen bezüglich der SCSI-Treiber im Klartext aus. Nur notwendig für Entwickler oder falls unerklärliche Fehler auftreten. Der Komfort der Fehlermeldungen kostet 12 KByte RAM.

#### SCSI logging facility (NEW) (CONFIG\_SCSI\_LOGGING)

Falls man Probleme mit SCSI-Geräten hat, so ist dieses der Treiber zum „Entwanzen“. Nebst diesem Treiber muß zum einen das virtuelle Dateisystem /proc aktiviert sein, zum anderen der Treiber Sysctl support. Mittels des Befehls

```
echo "scsi log token [level]" > /proc/scsi/scsi
```

kann man das System überzeugen, seine SCSI-Geheimnisse preiszugeben. token kann die Werte error, scan, mlqueue, mlcomplete, llqueue, llcomplete, hlqueue und hlcomplete annehmen. Details findet man im Source-Code des Treibers unter /usr/src/linux/drivers/scsi/scsi.c.

### SCSI low-level drivers

Unter diesem Submenü werden ab sofort die Treiber für die einzelnen SCSI-Karten bereitgehalten.

#### 3ware Hardware ATA-RAID support (CONFIG\_BLK\_DEV\_3W\_XXXX\_RAID)

3ware liefert ATA-RAID-Produkte für SCSI. Diese Karte muß im Master-Modus betrieben werden mit 2, 4 oder 8 Kanälen. Weitere Information liefert <http://www.3ware.com/>.

#### 7000FASST SCSI support (CONFIG\_SCSI\_7000FASST)

Treiber für den Western Digital 7000 SCSI-Host-Adapter.

ACARD SCSI support (NEW) (`CONFIG_SCSI_ACARD`)

Unterstützung für den Adapter ACARD 870U/W. Ist auch als Modul ladbar.

Adaptec AHA152X/2825 support (`CONFIG_SCSI_AHA152X`)

Treiber für die Adaptec-Karten AHA-1510, AHA-1520, AHA-1522 und AHA-2825.

Adaptec AHA1542 support (`CONFIG_SCSI_AHA1542`)

Treiber für den Adaptec AHA-1542-Host-Adapter.

Adaptec AHA1740 support (`CONFIG_SCSI_AHA1740`)

Treiber für den Adaptec AHA-1740-Host-Adapter.

Adaptec AIC7xxx support (`CONFIG_SCSI_AIC7XXX`)

Treiber für diverse Adaptec-Karten: die EISA-Karten 274x, Die VLB-Karten 284x, die PCI-Karten 294x, 394x und 3985. Außerdem werden viele verschiedene Versionen der Onboard-Controler von Adaptec auf den verschiedensten Mutterbrettern unterstützt. Details können folgendem Text entnommen werden: `/usr/src/linux/drivers/scsi/README.aic7xxx`.

Enable Tagged Command Queueing (TCQ) by default (NEW)  
(`CONFIG_AIC7XXX_TCQ_ON_BY_DEFAULT`)

Dieser Treiber aktiviert bei allen Geräten der Baureihe AIC7xxx das sogenannte Tagged Command Queueing (TCQ), die dieses unterstützen können. TCQ ist ein Leistungsmerkmal von SCSI-2, das die Leistung erhöht. Der Host-Adapter kann weitere SCSI-Befehle in die Queue senden, auch wenn die vorangegangenen noch nicht abgearbeitet sind. Hierdurch kann das entspre-

chende Gerät - ausgestattet mit eigener Intelligenz - die Ausführung der Befehle optimieren, z.B. indem es die Daten nicht sequentiell liest, sondern wie es die Geometrie des Laufwerks am schnellsten ermöglicht. Allerdings arbeiten nicht alle Geräte korrekt mit dieser Norm. Wer solche Geräte in Benutzung hat, kann trotzdem diesen Treiber aktivieren und die korrekten Platten mit diesem Leistungsmerkmal betreiben, während die inkorrekte Geräte mittels des Bootparameters `tag_info` auf herkömmliche Art betrieben werden. Weitere, ausführliche Information bietet `drivers/scsi/README.aic7xxx`.

Maximum number of TCQ commands per device (NEW)  
(`CONFIG_AIC7XXX_CMDS_PER_DEVICE`)

Diese Option bezieht sich auf den vorangegangenen Treiber. Hier kann die maximale Anzahl von SCSI-Befehlen, die in der Warteschlange Platz haben sollen, definiert werden. Die Standardeinstellung ist 8, der Maximalwert beträgt 24. Falls die Anzahl zu hoch eingestellt wird, wird der Treiber dieses normalerweise nach ca. 10 Minuten Betrieb automatisch nach unten korrigieren. Laufwerke vom Typ Quantum Fireball sollten nicht auf mehr als 8 Befehle eingestellt sein.

Collect statistics to report in /proc (NEW) (`CONFIG_AIC7XXX_PROC_STATS`)

Dieser Treiber bezieht sich wieder auf Geräte der Baureihe AIC7xxx. Es wird Buch geführt über die Anzahl der Befehle, die an das jeweilige Gerät gesendet wurden. Diese sind dann einsehbar über `/proc/scsi/aic7xxx/n`, wobei n die Nummer des gewünschten Controllers angibt. Außerdem muß natürlich das virtuelle Dateisystem `/proc` aktiviert sein. Für Entwickler und sehr neugierige Menschen.

Delay in seconds after SCSI bus reset (NEW)  
(`CONFIG_AIC7XXX_RESET_DELAY`)

Hier wird die Wartezeit für die Geräte der Baureihe AIC7xxx eingegeben, die nach einem Reset vergeht, bis die Karte wieder versucht, mit dem SCSI-Bus

zu kommunizieren. Normalerweise ist hier kein großer Wert notwendig für Festplatten und CD-ROM-Laufwerke, und so ist die Standardeinstellung 5 Sekunden. Allerdings sind einige DAT-Laufwerke dafür bekannt, hier unverschämt viel Zeit zu benötigen. Der Maximalwert beträgt 20 Sekunden, um den Rest des Kernels nicht unnötig zu behindern.

AdvanSys SCSI support (CONFIG\_SCSI\_ADVANSYS)

Treiber für alle SCSI-Karten von AdvanSys.

Always IN2000 SCSI support (CONFIG\_SCSI\_IN2000)

Falls diese Karte für den ISA-Bus als SCSI-Host-Adapter arbeitet, müssen evtl. die Jumper für IRQ und Adresse umgesteckt werden.

AM53/79C974 PCI SCSI support (CONFIG\_SCSI\_AM53C974)

Treiber für die AM 53/79C974 SCSI-Host-Adapter.

AMI MegaRAID support (NEW) (CONFIG\_SCSI\_MEGARAID)

Dieser Treiber unterstützt folgende SCSI-Adapter der Firma AMI: MegaRAID 418, 428, 438, 466, 762, 490 und 467.

BusLogic SCSI support (CONFIG\_SCSI\_BUSLOGIC)

Treiber für die BusLogic-MultiMaster-SCSI-Host-Adapter und inzwischen auch die BusLogic-FlashPoint-SCSI-Host-Adapter.

Omit FlashPoint support (NEW) (CONFIG\_SCSI OMIT\_FLASHPOINT)

Falls für den vorangegangenen Treiber die Unterstützung für den FlashPoint-Adapter deaktiviert werden soll, so muß diese Option aktiviert werden.

Compaq Fibre Channel 64-bit/66Mhz HBA support  
(CONFIG\_SCSI\_CPQFCTS)

Treiber für den Glasfaser-Bus-Adapter von Compaq, der mit 66 Mhz auf 64-Bit-Systemen läuft und den schönen Namen Tachyon erhalten hat. Ausführliche Information enthält die Datei `drivers/cpqfc.Readme`.

DMX3191D SCSI support (CONFIG\_SCSI\_DMX3191D)

Der Treiber für den SCSI-Host-Adapter Domex DMX3191D.

DTC3180/3280 SCSI support (CONFIG\_SCSI\_DTC3280)

Der Treiber für die DTC 3180/3280 SCSI-Host-Adapter.

EATA ISA/EISA/PCI (DPT and generic EATA/DMA-compliant boards) support  
(CONFIG\_SCSI\_EATA)

Dieser Treiber unterstützt alle EATA-/DMA-kompatiblen SCSI-Hostadapter. Vorsicht, es gibt einen weiteren Treiber für diese Funktion, der sich `EATA-DMA support` nennt. Nur einer von diesen beiden Treibern sollt aktiviert sein!

enable tagged command queueing (NEW)  
(CONFIG\_SCSI\_EATA\_TAGGED\_QUEUE)

Hiermit wird ein Leistungsmerkmal von SCSI-2 für den vorangegangenen EATA-Treiber aktiviert: es können bereits weitere SCSI-Befehle verschickt werden, auch wenn die vorangegangenen noch nicht verarbeitet worden sind.

enable elevator sorting (NEW) (CONFIG\_SCSI\_EATA\_LINKED\_COMMANDS)

Treiber zur Reduzierung der Zugriffszeit für EATA-SCSI-Laufwerke.

---

maximum number of queued commands (NEW)  
 (CONFIG\_SCSI\_EATA\_MAX\_TAGS)

Hier kann für den SCSI-EATA-Treiber (s.o.) die maximale Anzahl der SCSI-Befehle, die die Warteschlange aufnehmen soll, eingestellt werden. Das Minimum ist 2, das Maximum ist 62. Die wirklich benutzte Anzahl wird während des Bootens für jedes Laufwerk angezeigt.

EATA-DMA [Obsolete] (DPT, NEC, AT&T, SNI, AST, Olivetti, Alphatronix) support (CONFIG\_SCSI\_EATA\_DMA)

Der veraltete Treiber für das EATA-DMA-Protokoll von SCSI-Adaptoren wie SmartCache III/IV, die SmartRAID-Controller, die Controller DPT PM2011B and PM2012B. Den obestehenden Treiber CONFIG\_SCSI\_EATA benutzen!

EATA-PIO (old DPT PM2001, PM2012A) support  
 (CONFIG\_SCSI\_EATA\_PIO)

Dieser Treiber unterstützt das EATO-PIO-Protokoll von älteren Adaptoren wie DPT PM2001 oder PM2012A.

Future Domain 16xx SCSI/AHA-2920A support  
 (CONFIG\_SCSI\_FUTURE\_DOMAIN)

Future Domain 16xx SCSI support (CONFIG\_SCSI\_FUTURE\_DOMAIN)

Dieser Treiber unterstützt Future Domains 16-Bit-SCSI-Host-Adapter wie TMC-1660/1680, TMC-1650/1670, TMC-3260, TMC-1610M/MER/MEX, sowie andere Adapter, die auf dem Chipsatz von Future Domain basieren, wie Quantum ISA-200S, ISA-250MG und Adaptec AHA-2920A. Die neueren Adaptec AHA-2920C benutzen den Chip AIC-7850, für den der Treiber Adaptec AIC7xxx chipset SCSI controller support.

Future Domain MCS-600/700 SCSI support (CONFIG\_SCSI\_FD\_MCS)

Dieses ist der Treiber für den SCSI-Adapter MCS 600/700 MCA von Future Domain. Einige PS/2-Rechner sind mit dem IBM Fast SCSI Adapter/A ausgestattet, der identisch mit dem MCS 700 und also auch mit diesem Treiber benutzt werden können. Außerdem wird auch die SCSI-Karte Reply SB16 unterstützt. Der Treiber unterstützt auch mehrere Karten dieser Typen in einem Rechner.

GDT SCSI Disk Array Controller support (CONFIG\_SCSI\_GDTH)

Dieses ist der Treiber für alle SCSI Disk Array Controller (EISA/ISA/PCI) der Firma ICP Vortex. Er kann auch als Modul geladen werden.

Generic NCR5380/53c400 SCSI support  
(CONFIG\_SCSI\_GENERIC\_NCR5380)

Dieses ist der generische Treiber der SCSI-Controller. Nicht verwechseln mit den Controllern NCR 53c7 oder 8xx!

Enable NCR53c400 extensions (CONFIG\_SCSI\_GENERIC\_NCR53C400)

Dieser Treiber aktiviert die Optimierungen für die Karten mit dem NCR 53c400. In seiner Standard-Konfiguration testet der Treiber nur, ob eine Trantor T130B-Karte vorliegt. Andere Karten benötigen Kernel-Boot-Parameter.

NCR5380/53c400 mapping method (use Port for T130B)  
(CONFIG\_SCSI\_G\_NCR5380\_PORT)

Die SCSI-Controller NCR 5380 und NCR 53c400 gibt es in zwei Varianten: port-mapped oder memory-mapped. Die weit verbreitete Karte Trantor T130B benutzt den Modus port-mapped.

IBMMCA SCSI support (CONFIG\_SCSI\_IBMMCA)

Dieses ist der Treiber für den IBM-SCSI-Adapter, der in vielen PS/2-Rechnern steckt. Diese Computer besitzen den MCA-Bus, so daß auch der Treiber MCA support auch aktiviert werden sollte. Details dazu in Documentation/mca.txt.

In den Modellen 56, 57, 76 und 77 wird der Adapter während des Bootens oft nicht gefunden. Hier hilft der Bootparameter `ibmmcascsi=<pun>`, wobei `<pun>` die ID-Nummer des SCSI-Subsystems ist; normalerweise ist der Wert 7. Falls dieses nicht arbeitet, bitten auf der IBM-Referenz-Diskette nachsehen.

#### Standard SCSI-order (NEW) (`CONFIG_IBMMCA_SCSI_ORDER_STANDARD`)

Dieser Treiber ist eine zusätzliche Option für die IBM-MCA-SCSI-Treiber (s.o.). DOS hat - neben vieler anderer Verwirrung - auch indirekt Durcheinander in Reihenfolge der Nummerierung der SCSI-Laufwerke gebracht. Es ist leider zur Unsitte geworden, daß die niedrigste physikalische SCSI-Nummer dem Laufwerk C: (in der Terminologie von BIOS und DOS) zugewiesen wird; und weiter in aufsteigender Reihenfolge. Der ANSI-SCSI-Standard hingegen besagt, daß die ID 7 die höchste und die ID 0 die geringste Priorität besitzt. Durch diese Verwirrung wird also der Festplatte C: die geringste Priorität alle SCSI-Geräte zugewiesen. Um die Dinge wieder auf die Füße zu stellen, muß also dieser Treiber aktiviert werden.

#### Reset SCSI-devices at boottime (NEW) (`CONFIG_IBMMCA_SCSI_DEV_RESET`)

Dieser Treiber ist eine zusätzliche Option für die IBM-MCA-SCSI-Treiber (s.o.). Normalerweise wird bei allen SCSI-Peripheriegeräten ein Reset ausgeführt, wenn der Rechner eingeschaltet wird. Nun gibt es zahlreiche ältere SCSI-Geräte, die diese eben nicht erledigen, wie z.B. Drucker, Scanner oder CNC-Maschinen. Durch diesen Treiber wird ein Reset beim Booten erzwungen. Dieses kann aber auf der anderen Seite zu Konflikten mit neueren SCSI-Geräten führen, die dieses nicht mögen und evtl. das System beim Start zum Hängen bringen.

IBM ServeRAID Support (CONFIG\_SCSI\_IPS)

Dieses ist der Treiber für den Hardware Raid-Controller IBM ServeRAID. Weitere Information unter <http://www.developer.ibm.com/welcome/netfinity/serveraid.html>

Initio 9100U(W) support (CONFIG\_SCSI\_INITIO)

Der Treiber für den SCSI-Host-Adapter 91XXU(W) von Initio.

Initio INI-A100U2W support (CONFIG\_SCSI\_INIA100)

Der Treiber für den SCSI-Host-Adapter INI-A100U2W von Initio.

NCR53c406a SCSI support (CONFIG\_SCSI\_NCR53C406A)

Der Treiber für den SCSI-Host-Adapter NCR53c406a. Die durch den Benutzer konfigurierbaren Parameter findet man in `drivers/scsi/NCR53c406.c`.

NCR53c7,8xx SCSI support (CONFIG\_SCSI\_NCR53C7xx )

Dieses ist der Treiber für die NCR 53c7 und 8xx NCR-SCSI-Controller für den PCI-Bus. Bitte nicht mit dem Controller NCR 5380 verwechseln.

always negotiate synchronous transfers (CONFIG\_SCSI\_NCR53C7xx\_sync)

Prinzipiell vorteilhaft, obwohl einige defekte Controller damit eben fehlerhaft umgehen. Die sichere Antwort ist n.

allow FAST-SCSI [10MHz] (CONFIG\_SCSI\_NCR53C7xx\_FAST )

Aktiviert die 10MHz-Fast-SCSI-Übertragung mit dem Host-Adapter. Einige Systeme haben mit dieser Übertragungsrate Probleme, so daß es sicherer ist, mit n zu antworten.

allow DISCONNECT (CONFIG\_SCSI\_NCR53C7xx\_DISCONNECT )

Aktiviert die Fähigkeit des NCR-SCSI-Controllers, die Verbindung zu unterbrechen und wieder aufzubauen. Wenn diese Option aktiviert ist, wird ein langsames SCSI-Gerät während einer Anfrage nicht den Bus blockieren. Das gleichzeitige Arbeiten von mehreren verschiedenen SCSI-Geräten wird ermöglicht und so eine bessere Leistung erzielt, wenn langsame und schnelle SCSI-Geräte gleichzeitig benutzt werden. Einige Geräte arbeiten mit dieser Option nicht zusammen und verursachen so ein Hängen des SCSI-Systems, was einen Systemabsturz zur Folge haben kann. Die sichere Antwort ist also n.

NCR53C8XX SCSI support (CONFIG\_SCSI\_NCR53C8XX)

Dieses ist der Treiber für die PCI-SCSI-Controller-Familie NCR53C8XX. Er unterstützt Paritätsüberprüfung und Fast-SCSI-Transferraten bis 10 MB/s, mit Wide-SCSI-Geräten sogar bis 20 MByte/s.

SYM53C8XX SCSI support (CONFIG\_SCSI\_SYM53C8XX)

Dieser Treiber unterstützt alle Leistungsmerkmale der neueren CHIPS 53C8XX, die in PCI-SCSI-Controllern eingesetzt werden.

Ältere Versionen der Chipreihe 53C8XX werden nicht von diesem Treiber unterstützt. Wer einen 820, 815 oder 825 mit einer Revisionsnummer kleiner als 16 benutzt, der muß den generischen Treiber NCR53C8XX SCSI support (s.o.) benutzen.

default        tagged        command        queue        depth  
(CONFIG\_SCSI\_NCR53C8XX\_DEFAULT\_TAGS)

Diese Option bezieht sich auf den vorangegangenen Treiber. Hier kann die maximale Anzahl von SCSI-Befehlen, die in der Warteschlange Platz haben sollen, definiert werden. Die Standardeinstellung ist 8.

```
maximum number of queued commands
(CONFIG_SCSI_NCR53C8XX_MAX_TAGS)
```

Hiermit kann für den vorangegangenen Treiber die maximale Anzahl der SCSI-Befehle in der Schlange eingegeben werden. Die Standardeinstellung ist 32, das Maximum ist 64, das Minimum ist 2.

```
synchronous transfers frequency in MHz
(CONFIG_SCSI_NCR53C8XX_SYNC)
```

Betrifft wiederum die Adapter der Reihe NCR53C8XX. Der SCSI-2-Standard definiert fünf verschiedene Klassen von Transferraten: FAST-5, FAST-10, FAST-20, FAST-40 und FAST-80, wobei sich die Zahlen auf die Transfrequenz beziehen. Bei einem schlecht arbeitendem System kann null eingegeben werden, so wird allerdings der langsamste Modus aktiviert. Bei normal arbeitenden Systemen kann unbedenklich der höchste Wert, also 80 eingegeben werden, da sich der Treiber automatisch der vorhandenen Hardware anpasst.

```
enable profiling (CONFIG_SCSI_NCR53C8XX_PROFILE)
```

Betrifft wiederum die Adapter der Reihe NCR53C8XX. Hierdurch können statistische Informationen über den Treiber aktiviert werden. Im Normalfall nicht notwendig.

```
use normal IO (CONFIG_SCSI_NCR53C8XX_IOMAPPED)
```

Betrifft wiederum die Adapter der Reihe NCR53C8XX. Hierdurch wird die normale Ein- und Ausgabe aktiviert, die im Gegensatz zur Memory Mapped

Ein- und Ausgabe langsamer ist. Diese Option sollte nur aktiviert werden, wenn man Kommunikationsprobleme mit SCSI-Geräten hat.

```
include support for the NCR PQS/PDS SCSI card
(CONFIG_SCSI_NCR53C8XX_PQS_PDS)
```

Ein spezieller Treiber, der eine Untermenge der Baureihe NCR53C8XX behandelt: wer die SCSI-Adapter von NCR namens PCI Quad SCSI oder PCI Dual SCSI, sollte diesen Treiber aktivieren.

```
assume boards are SYMBIOS compatible (EXPERIMENTAL)
(CONFIG_SCSI_NCR53C8XX_SYMBIOS_COMPAT)
```

Betrifft wiederum die Adapter der Reihe NCR53C8XX. Er aktiviert einige Leistungsmerkmale, die auf der GPIO-Verkabelung beruhen (General Purpose Input/Output).

```
NCR MCA 53C9x SCSI support (CONFIG_SCSI_MCA_53C9X)
```

Einige Rechner mit Microchannel-Architektur, insbesondere die Modelle 35xx von NCR, benutzen einen SCSI-Controller auf Basis NCR 53C94. Dieses ist der Treiber dazu.

```
PAS16 SCSI support (CONFIG_SCSI_PAS16)
```

Dieses ist der Treiber für den PAS16-SCSI-Host-Adapter. Details sind im Kapitel 3.1 des SCSI-HOWTO erhältlich.

```
PCI2000 Unterstützung (CONFIG_SCSI_PCI2000)
```

Dieses ist der Treiber für die EIDE-Steckkarte PCI2000I, die als SCSI-Adapter arbeitet. Der Treiber kann auch als Modul kompiliert werden.

PCI2220i Unterstützung (CONFIG\_SCSI\_PCI2220I)

Dieses ist der Treiber für die EIDE-Steckkarte PCI2220i, die als SCSI-Adapter arbeitet. Der Treiber kann auch als Modul kompiliert werden.

PSI240i Unterstützung (CONFIG\_SCSI\_PSI240I)

Dieses ist der Treiber für die EIDE-Steckkarte PSI240i, die als SCSI-Adapter arbeitet. Der Treiber kann auch als Modul kompiliert werden.

Qlogic FAS SCSI support (CONFIG\_SCSI\_QLOGIC\_FAS)

Dieser Treiber arbeitet nur mit den ISA-, VLB- und PCMCIA-Versionen der Qlogic-FastSCSI!-Karte und unterstützt auch andere Karten, die auf dem Chipsatz FASXX basieren. Für die PCI-Version gibt der folgende Qlogic-ISP-Treiber Unterstützung.

Qlogic ISP SCSI support (CONFIG\_SCSI\_QLOGIC\_ISP)

Dieser Treiber arbeitet mit allen Qlogic-PCI-SCSI-Host-Adaptoren (IQ-PCI, IQ-PCI-10, IQ\_PCI-D) mit Ausnahme der PCI-Basic-Karte, die von dem Treiber AM53/79C974 PCI SCSI unterstützt wird. Wenn dieser Treiber kompiliert werden soll, muß natürlich auch die Option PCI BIOS support aktiviert sein.

Qlogic ISP FC SCSI support (NEW) (CONFIG\_SCSI\_QLOGIC\_FC)

Diese Option aktiviert den SCSI-Adapter QLogic ISP2100 SCSI-FCP; er kann auch als Modul kompiliert werden.

Qlogic QLA 1280 SCSI support (CONFIG\_SCSI\_QLOGIC\_1280)

Der Treiber für die SCSI-Adapter ISP1x80/1x160 von Qlogic.

Seagate ST-02 and Future Domain TMC-8xx SCSI support  
(CONFIG\_SCSI\_SEAGATE)

Dieses ist der Treiber für die 8-Bit-SCSI-Controller; der ST-01 wird auch unterstützt.

Simple 53c710 SCSI support (Compaq, NCR machines)  
(CONFIG\_SCSI\_SIM710)

Dieses ist der Treiber für den NCR53C710-Adapter-Chip, der häufig in NCR- und Compaq-Rechnern gefunden wird.

Symbios 53c416 SCSI support (CONFIG\_SCSI\_SYM53C416)

Dieses ist der Treiber für den SCSI-Adapter sym53c416, der mit einigen HP- Scannern geliefert wird.

Tekram DC390(T) and Am53/79C974 SCSI support  
(CONFIG\_SCSI\_DC390T)

Dieser Treiber unterstützt PCI-Karten, die als SCSI-Adapter mit dem Chip Am53C974A ausgestattet sind. Dieses sind zum Beispiel die Karten Tekram DC390(T), DawiControl 2974 sowie einige Onboard-Controler, wie PCscsi/PCnet (Am53/79C974). Dieser Treiber kann auch als Modul kompiliert werden.

\_omit\_ support for non-DC390 adapters (NEW)  
(CONFIG\_SCSI\_DC390T\_NOGENSUPP)

Der Tekram DC390(T) bezieht normalerweise seine Initialisierungswerte aus seinem EEPROM. Dafür ist dieser Treiber gedacht. Falls aber kein EEPROM gefunden wird, versagt dieser Treiber, also mit Vorsicht zu aktivieren.

Trantor T128/T128F/T228 SCSI support (CONFIG\_SCSI\_T128)

Dieses ist der Treiber für die Trantor-SCSI-Host-Adapter. ACHTUNG: Trantor wurde von Adaptec gekauft und einige ehemalige Trantor-Produkte werden inzwischen unter dem Namen Adaptec verkauft.

UltraStor 14F/34F support (CONFIG\_SCSI\_U14\_34F)

Dieses ist der Treiber für die SCSI-2-Host-Adapter 14F, 24F und 34F von UltraStor. ACHTUNG: Für die gleiche Hardware existiert ein weiterer Treiber mit dem Namen UltraStor SCSI support (s. u.). Es darf nur einer der beiden Treiber aktiviert sein!

enable                  elevator                  sorting                  (NEW)  
(CONFIG\_SCSI\_U14\_34F\_LINKED\_COMMANDS)

Aktiviert für die Karte UltraStor 14F/34F das sogenannte „elevator sorting“. Die Schreib- und Lesebefehle werden nicht nach ihrer zeitlichen Reihenfolge, sondern nach der Lage der Datenblöcke auf dem Speichermedium ausgeführt, um die Schreib- und Lesezugriffe zu beschleunigen. Es wird also versucht, die Bewegungen des Schreib-Lese-Kopfes auf ein Minimum zu reduzieren.

maximum              number              of              queued              commands          (NEW)  
(CONFIG\_SCSI\_U14\_34F\_MAX\_TAGS)

Gibt für die Karte UltraStor 14F/34F die maximale Anzahl der SCSI-Befehle, die in die Warteschlange passen, an. Die Standardeinstellung ist 8, das Minimum 2 und das Maximum 14.

UltraStor SCSI support (CONFIG\_SCSI\_ULTRASTOR)

Dieses ist der Treiber für die SCSI-2-Host-Adapter 14F, 24F und 34F von UltraStor. ACHTUNG: Für die gleiche Hardware existiert ein weiterer Treiber mit dem Namen UltraStor 14F/34F support (s. o.). Es darf nur einer der beiden Treiber aktiviert sein!

**SCSI debugging host simulator (EXPERIMENTAL) (CONFIG\_SCSI\_DEBUG)**

Dieser Treiber emuliert einen SCSI-Adapter, der programmierbar ist. So können eine große Anzahl verschiedenster Situationen erprobt werden, ohne das Risiko eingehen zu müssen, mit realer Hardware arbeiten zu müssen, und dabei Daten zu verlieren. Der Treiber ist gedacht für Entwickler von SCSI-Systemen. Normale Sterbliche antworten mit n.

**PCMCIA SCSI adapter support (CONFIG\_SCSI\_PCMCIA)**

Wer eine PCMCIA-Karte oder eine CardBus-Karte als SCSI-Controller betreibt, der muß hier mit y antworten. Diese Kreditkarten-großen Geräte werden in aller Regel mit Laptops genutzt.

**Adaptec AHA152X PCMCIA support (NEW) (CONFIG\_PCMCIA\_AHA152X)**

Der Treiber für die PCMCIA-SCSI-Karte AHA152X von Adaptec. Kann auch als Modul kompiliert werden.

**Qlogic PCMCIA support (NEW) (CONFIG\_PCMCIA\_QLOGIC)**

Der Treiber für die PCMCIA-SCSI-Karte von Qlogic. Kann auch als Modul kompiliert werden.

**Future Domain PCMCIA support (NEW) (CONFIG\_PCMCIA\_FDOMAIN)**

Der Treiber für die PCMCIA-SCSI-Karte von Future Domain. Kann auch als Modul kompiliert werden.

**Adaptec APA1480 CardBus support (NEW) (CONFIG\_PCMCIA\_AP1480)**

Der Treiber für die CardBus-SCSI-Karte APA1480 von Adaptec. Kann auch als Modul kompiliert werden.

## 14. IEEE 1394 / FireWire support

Die Unterstützung für den „heißen Draht“ wurde inzwischen unter einem eigenen Menüpunkt zusammengefaßt.

IEEE 1394 (FireWire) support (EXPERIMENTAL) (CONFIG\_IEEE1394)

Die Norm IEEE 1394 beschreibt einen seriellen Bus für hohen Datendurchsatz, inzwischen überall bekannt als FireWire. Die derzeit bekannteste Anwendung ist der Anschluß von digitalen Video-Kameras über diesen Bus. Falls FireWire-Hardware genutzt werden soll, so muß hier mit y geantwortet werden. Dieses ist aber nur die prinzipielle Unterstützung für FireWire, die Treiber für die verschiedenen Adapter folgen. Dieser Treiber kann auch als Modul kompiliert werden.

Texas Instruments PCILynx support (CONFIG\_IEEE1394\_PCILYNX)

Falls man einen Controller für IEEE 1394 mit dem Chip PCILynx von Texas Instruments besitzt, so ist dieses der richtige Treiber. Er wurde allerdings für die Revisions-Nummer 2 des Chips geschrieben, und ist mit der älteren Revisions-Nummer 0 nicht getestet worden. Der Treiber kann als Modul kompiliert werden.

Use PCILynx local RAM (CONFIG\_IEEE1394\_PCILYNX\_LOCALRAM)

Einige Test-Boards, die mit dem Chip PCILynx ausgestattet sind, besitzen zudem 64 MB lokalen RAM. Dieser kann benutzt werden, um die Verarbeitungsgeschwindigkeit zu erhöhen, da die Daten nicht über den Bus geschickt werden müssen. Da die Hardware leider nicht verlässlich meldet, ob zusätzlicher lokaler RAM vorhanden ist, vergewissern sie sich bitte, ob dieser RAM vorhanden ist, bevor sie diesen Treiber aktivieren.

---

Support for non-IEEE1394 local ports  
 (CONFIG\_IEEE1394\_PCILYNX\_PORTS)

Dieser Treiber aktiviert den Zugriff auf die RAM-, ROM- und AUX-Ports des PCILynx über Gerätetreiber in /dev.

OHCI (Open Host Controller Interface) support  
 (CONFIG\_IEEE1394\_OHCI1394)

Dieser Treiber ist für die FireWire-Adapter, die den Chipsatz OHCI von Texas Instruments benutzen. Der Treiber ist nur mit dem original TI-Adapter getestet worden; es gibt aber auch einige Dritthersteller, die diesen Chip benutzen. Dieser Treiber kann auch als Modul kompiliert werden.

Video1394 support (CONFIG\_IEEE1394\_VIDEO1394)

Der Treiber für die Video-Unterstützung unter IEEE 1394.

RAW IEEE 1394 I/O support (CONFIG\_IEEE1394\_RAWIO)

Wenn direkter („raw“) Zugriff auf IEEE 1394 Geräte gewünscht wird, so ist dieses der richtige Treiber. Er muß im Moment als Modul kompiliert werden!

Excessive debugging output (CONFIG\_IEEE1394\_VERBOSEDEBUG)

Ein Treiber für Entwickler. Jedes (!) gesendete und empfangene Paket wird dokumentiert. Dieses kann natürlich in kürzester Zeit zu einer großen Datenmenge auf der Festplatte führen, also Vorsicht!

## 15. I2O device support

I2O support (CONFIG\_I2O)

Die sogenannte Intelligente Input/Output Architektur (I2O) ermöglicht das Aufteilen von Hardware-Treiber in zwei Teile: einen betriebssystemspezifischen Teil, der OSM genannt wird sowie einen Hardware-spezifischen Teil, der HDM genannt wird. Der OSM-Teil kann die gesamte Bandbreite von HDMs ansprechen und idealerweise solitem die HDMs betriebssystemunabhängig sein. Dieses ermöglicht, daß derselbe HDM-Treiber von verschiedenen Betriebssystemen benutzt werden kann. Natürlich benötigt man für diesen Treiber eine Adapter-Karte, die eine I2O-Schnittstelle besitzt. Diese Karten bestzen einen speziellen Eingabe-/Ausgabe-Prozessor (IOP), der Hochgeschwindigkeitstransfer ermöglicht, so daß die CPU diese Arbeit nicht leisten muß. Dieses ist die prinzipielle Unterstützung für I2O; im weiteren werden diverse OSM-Schnittstellen für die verschiedenen Geräte angeboten.

#### I2O PCI support (CONFIG\_I2O\_PCI)

PCI ist derzeit der einzige Bus, auf dem I2O unterstützt wird. Wer also eine I2O-Karte betreiben will, muß hier mit y antworten.

#### I2O Block OSM (CONFIG\_I2O\_BLOCK)

Der Treiber für Block-orientierte Geräte, wie z.B. Festplatten und CD-ROM-Laufwerke.

#### I2O LAN OSM (CONFIG\_I2O\_LAN)

Der Treiber für lokale Netzwerke. Wir auch benötigt, falls man mittels I2O-Karten Token-Ring-Netzwerke oder FDDI betreiben will.

#### I2O SCSI OSM (CONFIG\_I2O\_SCSI)

Der Treibr für SCSI-Geräte und auch FibreChannel-Geräte. Für blockorientierte Geräte muß auch der obige Blockgeräte-Treiber aktiviert werden.

#### I2O /proc support (CONFIG\_I2O\_PROC)

Wer über die I2O-Geräte Informationen im virtuellen Dateisystem /proc erhalten möchte, muß diesen Treiber aktivieren und außerdem natürlich den grundlegenden Treiber /proc file system support aktivieren. Danach erhält man Information aus dem virtuellen Verzeichnis /proc/i2o.

## 16. Network device support

Network device support (CONFIG\_NETDEVICES)

Falls der Rechner keinerlei Verbindung zur „Außenwelt“ besitzt, muß hier mit n geantwortet werden. Dieses gilt auch, wenn mit anderen Rechner über UUCP oder term via Telefonleitung kommuniziert wird. Mit y muß geantwortet werden, wenn eine Netzwerkkarte betrieben wird oder eines der folgenden Protokolle für Netz- oder Internet-Betrieb verwendet wird:

SLIP (Serial Line Internet Protocol), CSLIP (Compressed SLIP), PPP (Point to Point Protocol), PLIP (Parallel Line Internet Protocol) oder AX.25/KISS (Protokoll für Internet-Betrieb und anderen Datenaustausch via Amateurfunk). Falls eine Netzwerkkarte unter LINUX laufen soll, suchen Sie schon die entsprechenden Unterlagen; der Kartentyp wird in Kürze abgefragt.

### ARCnet devices

Sämtliche Treiber für das ARCnet haben hier ein Submenü gefunden.

ARCnet support (CONFIG\_ARCNET)

Falls ein Netzwerk des Typs ARCnet vorliegt, so muß dieser Treiber sowie der Treiber für die zugehörige Netzwerkkarte aktiviert werden. Weitere Informationen enthält der Text /usr/src/linux/Documentation/networking/arcnet.txt, der auch

wegen seiner wundervollen lyrischen Ergüsse nicht zu verachten ist. Dieser Treiber kann auch als Modul geladen werden.

Enable standard ARCNet packet format (RFC 1201)  
(CONFIG\_ARCNET\_1201)

Das Standardpaketformat von ARCnet, das eigentlich immer benötigt wird. Ausnahmen sind im folgenden Treiber CONFIG\_ARCNET\_1051 beschrieben.

Enable old ARCNet packet format (RFC 1051) (CONFIG\_ARCNET\_1051)

Dieser Treiber ermöglicht die Nutzung von RF C1051 über die ARCnet-Karte via das virtuelle Gerät arc0s. Er wird nur benötigt, wenn mit ARCnet-Software kommuniziert werden muß, die dem älteren Standard entspricht. Insbesondere gilt dies für den DOS-Treiber arcnet.com, für Amigas, die unter AmiTCP laufen und für einige Arten von NetBSD. Dieser Treiber darf nicht aktiviert werden, wenn die Standard-Implementation RFC 1201 vorliegt, wie z. B. der Treiber arcether.com oder die meisten DOS- oder Windows-ODI-Treiber.

Enable raw mode packet interface (CONFIG\_ARCNET\_RAW)

Die ARCnet-Schnittstelle, die die Paketdaten im „rohen“ Format, also so, wie sie im Netz anliegen, weiterreicht.

ARCnet COM90xx (normal) chipset driver (CONFIG\_ARCNET\_COM90xx)

Dieses ist der Treiber für die weit verbreitete Standard-Karte COM90xx. Der Treiber kann auch als Modul kompiliert werden.

ARCnet COM90xx (IO mapped) chipset driver  
(CONFIG\_ARCNET\_COM90xxIO)

Falls die Standard-Karten COM90xx den Modus „memory mapped“ nicht beherrscht, so sollte man es mit diesem Treiber versuchen, der mit dem Mo-

dus „IO mapped“ arbeitet. Dieses ist natürlich langsamer als der vorangehende Treiber.

ARCnet COM90xx (RIM I) chipset driver (CONFIG\_ARCNET\_RIM\_I)

Dieses ist ein weiterer, noch ungetesteter Treiber für die Karte COM90xx, der den Modus memory mapped benutzt. Doppelentwicklungen bleiben unter LINUX manchmal nicht aus, und der Programmierer würde sich über Informationen bezüglich seinens Treibers freuen: David.Woodhouse@mvhi.com. Der Treiber kann auch als Modul geladen werden.

ARCnet COM20020 chipset driver (CONFIG\_ARCNET\_COM20020)

Dieses ist der Treiber für ARCnet-Karten mit dem neuen Chipsatz COM20020. Der Treiber kann auch als Modul geladen werden.

Support for COM20020 on ISA (CONFIG\_ARCNET\_COM20020\_ISA)

Der Treiber für Karten mit dem Chip COM20020 am ISA-Bus.

Support for COM20020 on PCI (CONFIG\_ARCNET\_COM20020\_PCI)

Der Treiber für Karten mit dem Chip COM20020 am PCI-Bus.

### Appletalk devices

Auch die Apfel-Treiber sind jetzt in einem Untermenü zusammengeführt worden.

Appletalk interfaces support (CONFIG\_APPLETALK)

AppleTalk ist das Protokoll, das die Äpfel-Rechner miteinander reden. Wenn ein Linux-Rechner an ein Appletalk-Netzwerk angeschlossen werden soll, so muß hier mit y geantwortet werden.

[Apple/Farallon LocalTalk PC support \(CONFIG\\_LTPC\)](#)

Dieser Treiber ermöglicht die Benutzung der Appletalk-PC-Karte an einem LocalTalk-Netzwerk. Die Karte ist auch unter dem Namen Farallon PhoneNet bekannt. Falls man diese nicht genau weiß: auf dieser Karte befindet sich ein Chip vom Typ 65C02. Außerdem wird das Paket `netatalk` in der Version 1.3.3 oder höher benötigt. Weitere Information enthält Documentati-on/networking/ltpc.txt.

[COPS LocalTalk PC support \(CONFIG\\_COPS\)](#)

Dieser Treiber ermöglicht die Benutzung der Karte COPS AppleTalk an einem LocalTalk-Netzwerk. Außerdem wird das Paket `netatalk` in der Version 1.3.3 oder höher benötigt. Weitere Information enthält Documentati-on/networking/cops.txt.

[Dayna firmware support \(CONFIG\\_COPS\\_DAYNA\)](#)

Ermöglicht das Betreiben von COPS-kompatiblen AppleTalk-Karten, die Firmware der Firma Dayna benutzen. Dieses sind die Karten Dayna DL2000/ Daynatalk/PC (halbe Bauhöhe), COPS LT-95, Farallon PhoneNET PC III und Farallon PhoneNET PC II.

[Tangent firmware support \(CONFIG\\_COPS\\_TANGENT\)](#)

Ermöglicht das Betreiben von COPS-kompatiblen AppleTalk-Karten, die Firmware der Firma Tangent benutzen. Dieses sind die Karten Tangent ATB\_II, Novell NL-1000 und Daystar Digital LT-200.

[Appletalk-IP driver support \(CONFIG\\_IPDDP\)](#)

Falls nur ein AppleTalk-Netzwerk vorhanden ist, so kann mit diesem Treiber dennoch IP-Netzbetrieb aufgenommen werden, um z.B. als Internet-Gateway für eine Apple-Rechner-Farm zu dienen. Mit diesem Treiber werden die IP-Pakete innerhalb von AppleTalk ein- bzw. ausgepackt. Nur eine dieser beiden Möglichkeiten besteht pro Kernel. Sie werden im Anschluß danach abgefragt. Weitere Information enthält Documentation/networking/ipddp.txt.

IP to Appletalk-IP Encapsulation support (CONFIG\_IPDDP\_ENCAP)

Die IP-Pakete werden innerhalb der AppleTalk-Pakete gekapselt.

Appletalk-IP to IP Decapsulation support (CONFIG\_IPDDP\_DECAP)

Die Appletalk-Pakete werden in IP-Pakete umgewandelt.

Dummy net driver support (CONFIG\_DUMMY)

Dieses ist das Netz-Daten-Nirvana mit einer konfigurierbaren IP-Adresse. Es wird gerne zu Konfigurations- und Testzwecken benutzt und außerdem, um Programmen eine Adresse vorzutäuschen. Da diese Option den Kernel nicht vergößert, ist es ein Leichtes, mit y zu antworten. Wenn mehr als ein Dummy-Treiber gleichzeitig benutzt werden soll, muß diese Option als Modul kompiliert werden.

Bonding driver support (CONFIG\_BONDING)

Mit diesem Treiber können mehrere Ethernet-Kanäle gebunden werden. Cisco nennt diese Möglichkeit „Etherchannel“, Sun nennt es „Trunking“ und bei Linux heißt es „Bonding“ („Bond, James Bond“).

Falls man z.B. zwei Ethernet-Verbindungen zu einem anderen Computer besitzt, so kann man mittels dieses Treibers eine Verbindung mit doppelter Übertragungsgeschwindigkeit herstellen. Auf der anderen Seite wird entweder auch dieser Linux-Treiber erwartet, oder aber ein Cisco 5500 Switch oder ein

SunTrunking-Treiber von SunSoft. Dieser Treiber verhält sich wie der folgende EQL-Treiber, benutzt aber das Ethernet anstatt von seriellen Leitungen.

EQL (serial line load balancing) support (CONFIG\_EQUALIZER)

Falls zwischen diesem Rechner und einem anderen zwei serielle Leitungen (Modems, Telefon Datenleitungen oder Null-Modems) bestehen, kann die Übertragungsgeschwindigkeit verdoppelt werden. Hierzu muß als Protokoll aber entweder SLIP oder PPP gefahren werden. Typischerweise besteht am anderen Ende der Leitungen die gleiche Konfiguration.

Universal TUN/TAP device driver support (CONFIG\_TUN)

TUN/TAP ist ein Mechanismus, der es Benutzerprogrammen ermöglicht, miteinander über Ethernet zu kommunizieren. Dieses kann logisch derart betrachtet werden, daß eine einfache Punkt-zu-Punkt-Verbindung zwischen Benutzerprogrammen besteht. Wenn ein Benutzerprogramm die Gerätedatei /dev/net/tun öffnet, erzeugt der Treiber das zugehörige Netzgerät tunX oder tapX. Nach Beendigung des Programms entfernt der Treiber diese Geräte. Weitere Information enthält Documentati-on/networking/tuntap.txt.

Ethertap network tap (OBSOLETE) (CONFIG\_ETHERTAP)

Wenn vorher die Option Kernel/User network link driver aktiviert wurde, so kann dieser Treiber Benutzerprogramme in die Lage versetzen, Ethernet Frames zu lesen und zu schreiben. Um dieses zu erreichen, muß man eine Gerätedatei /dev/tap0 mit der Hauptgerätenummer 36 und der Nebennummer 16 mittels mknod erzeugt werden. Danach kann von dieser Gerätedatei gelesen werden und auf sie geschrieben werden. /dev/tap0 kann wie jede Ethernet-Karte mittels ifconfig und route konfiguriert werden, es ist allerdings nicht mit dem LAN verbunden: jegliche Kommunikation muß über das Gerät /dev/tap0 erfolgen. Details bietet der Text

/usr/src/linux/Documentation/networking/ethertap.txt. Dieser Treiber kann auch als Modul kompiliert und geladen werden.

#### General Instruments Surfboard 1000 (CONFIG\_NET\_SB1000)

Dieses ist der Treiber für das interne Modem General Instrument SURFboard 1000. Diese ISA-Karte wird benutzt, um Zugriff auf (amerikanische) TV-Kabel zuzugreifen. Derzeit muß dieser Treiber als Modul kompiliert werden. Weitere Informationen bieten folgende Web-Adressen:  
<http://www.jacksonville.net/~fventuri/> und  
<http://home.adelphia.net/~siglercm/sb1000.html> und  
<http://linuxpower.cx/~cable/>.

#### Ethernet (10 or 100Mbit)

Unter diesem Untermenü sind die Treiber für alle 10- oder 100-MBit-Netzwerkkarten zusammengefaßt.

##### Ethernet (10 or 100MBit) (CONFIG\_NET\_ETHERNET)

Ethernet ist das am meisten verbreitete Protokoll für LANs (Local Area Networks). Falls der LINUX-Rechner an ein Ethernet angeschlossen werden soll, muß hier geantwortet werden, was keinen direkten Einfluß auf den Kernel hat: Es werden hier nur die jetzt folgenden Fragen nach dem Netzkartentyp aktiviert.

##### 3COM cards (CONFIG\_NET\_VENDOR\_3COM)

Falls eine Netzkarte der Firma 3COM im Rechner vorhanden ist, sollte hier mit y geantwortet werden; der genaue Typ wird von den nächsten Dialogen abgefragt.

##### 3c501 support (CONFIG\_EL1)

Treiber für die 3COM-Karte 501. Wir empfehlen allerdings die Anschaffung einer neuen Karte, da die 501 sehr langsam ist.

3c503 support (CONFIG\_EL2)

Treiber für die 3COM-Karte 503.

3c505 support (CONFIG\_ELPLUS)

Treiber für die 3COM-Karte 505. Weitere Informationen sind vorhanden unter Documentation/networking/3c505.txt.

3c507 support (CONFIG\_EL16)

Treiber für die 3COM-Karte 507.

3c509/3c579 support (CONFIG\_EL3)

Dieses ist der Treiber für die 3COM-EtherLink III-Baureihe. Die Parameter dieser Karte müssen zuerst mit Hilfe der DOS-Setup-Diskette eingestellt werden.

3c515 ISA "Fast EtherLink" (CONFIG\_3C515)

Der Treiber für die Karte 3Com ISA EtherLink XL 3c515.

3c523 "EtherLink/MC" support (CONFIG\_ELMC)

Der Treiber für die Karte 3Com 3c523 EtherLink/MC.

3c527 "EtherLink/MC 32" support (EXPERIMENTAL) (CONFIG\_ELMC\_II)

Der experimentelle Treiber für die Karte 3Com 3c527 EtherLink/MC 32.

3c590/3c900 series (592/595/597/900/905) "Vortex/Boomerang" support (CONFIG\_VORTEX)

Falls eine 3COM Vortex Karte der Typen 3c590, 3c592, 3c595 oder 3c597 vorhanden ist, ist dies der richtige Treiber. Weitere Informationen sind in Documentation/networking/vortex.txt und in den Kommentaren des Quelltextes drivers/net/3c59x.c enthalten.

AMD LANCE and PCnet (AT1500 and NE2100) support (CONFIG\_LANCE)

Treiber für die Ethernet-Karten AMD LANCE PCnet.

Western Digital/SMC cards (CONFIG\_NET\_VENDOR\_SMC)

Diese Option aktiviert nur das weitere Abfragen der einzelnen Western-Digital-Karten, veranlaßt also keine Code-Erzeugung.

WD80\*3 support (CONFIG\_WD80x3)

Treiber für die Karten 80\*3 von Western Digital.

SMC Ultra MCA support (CONFIG\_ULTRAMCA)

Treiber für die SMC-Ultra-Karte an einem MCA-Bus, also auf einem PS/2-Rechner.

SMC Ultra support (CONFIG\_ULTRA)

Treiber für die SMC-Ultra-Karte. ACHTUNG: es ist mehrfach berichtet worden, daß das gleichzeitige Betreiben einer SMC-Ultra-und einer Adaptec-AHA1542-SCSI-Karte Probleme verursacht hat, und zwar unabhängig vom Betriebssystem.

SMC Ultra32 EISA support (CONFIG\_ULTRA32)

Der Treiber für die EISA-Karte SMC Ultra32 EISA.

SMC 9194 support (CONFIG\_SMC9194)

Treiber für die SMC 9xxx-Baureihe. Falls ein Dell Laptop oder ein anderer Rechner mit dem Chipsatz SMC 9192/9194 vorliegt, ist dieses die richtige Wahl. Weitere Informationen sind unter `drivers/net/README.smc9` vorhanden.

Racal-Interlan (Micom) NI cards (CONFIG\_NET\_VENDOR\_RACAL)

Dieser Menüpunkt aktiviert die Fragen nach den Karten NI5010, NI5210 und NI6210 und erzeugt keinen Kernel-Code.

NI5010 support (EXPERIMENTAL) (CONFIG\_NI5010)

Der Treiber für die Ethernet-Karte NI5010. Dieser Treiber kann auch als Modul geladen werden.

NI5210 support (CONFIG\_NI52)

Der Treiber für die Ethernet-Karte NI5210. Dieser Treiber kann auch als Modul geladen werden.

NI6510 support (CONFIG\_NI65)

Der Treiber für die Ethernet-Karte NI6510. Dieser Treiber kann auch als Modul geladen werden.

AT1700/1720 support (EXPERIMENTAL) (CONFIG\_AT\_1700)

Treiber für die Netzwerkkarte AT 1700.

DECDA, DE10x, DE200, DE201, DE202, DE422 support (CONFIG\_DECDA)

Treiber für die Netzwerkkarten DECDA, die Baureihe DE und die EtherWORKS von Digital Equipment. Für die Karten DE 203, DE 204 und DE 205 ist nicht dieser, sondern der Treiber CONFIG\_EWRK3 zuständig!

HP 10/100VG PCLAN (ISA, EISA, PCI) support (CONFIG\_HP100)

Dieser Treiber ist für folgende Netzwerkkarten von Hewlett-Packard zuständig: J2577, J2573, 27248B, J2577, J2573 und J2585.

Other ISA cards (CONFIG\_NET\_ISA)

Falls der Bus des betreffenden Rechners ISA (im Gegensatz zu EISA, VLB oder PCI) ist, und die Karte der Wahl bisher noch nicht erwähnt wurde, sollte hier mit y geantwortet werden. Diese Option erzeugt keinen Code, sondern aktiviert nur die nächsten Optionen.

Cabletron E21xx support (CONFIG\_E2100)

Treiber für die Karten der Baureihe E 2100 von Cabletron.

EtherExpress 16 support (CONFIG\_EEXPRESS)

Treiber für die Karte EtherExpress 16.

EtherExpressPro support/EtherExpress 10 (i82595)  
(CONFIG\_EEXPRESS\_PRO)

Treiber für die Karte EtherExpressPro.

HP PCLAN+ (27247B and 27252A) support (CONFIG\_HPLAN\_PLUS)

Treiber für Ethernet-Karten von Hewlett-Packard, die unter verschiedenen Modellnummern verkauft wurden, hauptsächlich 27247B und 27252A.

HP PCLAN (27245 and other 27xxx series) support (CONFIG\_HPLAN)

Treiber für die Netzwerkkarten mit den Modellnummern 27245 und 27xxx von Hewlett-Packard. Für die Modellnummer 27252A ist der Treiber CONFIG\_HPLAN\_PLUS (s. o.) zuständig!

ICL EtherTeam 16i/32 support (CONFIG\_ETH16I)

Treiber für die Karte EtherTeam 16i bzw. 32 (EISA) von ICL.

NE2000/NE1000 support (CONFIG\_NE2000)

Treiber für die Karten NE 2000 bzw. NE 1000.

SK\_G16 support (CONFIG\_SK\_G16)

Treiber für die Netzwerkkarte G 16 von Schneider & Koch. Weitere Informationen sind in den Kommentaren des Quelltextes erhältlich:  
drivers/net/sk\_g16.c.

SKnet MCA support (CONFIG\_SKMC)

Der Treiber für die Ethernet-Karten von SKnet am MCA-Bus. Er unterstützt die beiden Karten SKnet Junior MC2 and the SKnet MC2.

NE2 (ne2000 MCA version) support (CONFIG\_NE2\_MCA)

Der Treiber für die Karte NE 2000 am MCA-Bus.

**IBM LAN Adapter/A support (CONFIG\_IBMLANA)**

Der Ethernet-Adapter von IBM für den MicroChannel, also PS/2-Rechner.

**EISA, VLB, PCI and on board controllers (CONFIG\_NET\_PCI)**

Falls die Ethernetkarte am EISA-Bus, am Local Bus oder am PCI-Bus angeschlossen ist, oder Onboard mitgeliefert wurde, so sollte hier mit y geantwortet werden. Eine Unmenge von Menüpunkten zu weiteren Kartentreibern öffnen sich.

**AMD PCnet32 PCI support (CONFIG\_PCNET32)**

Wer eine Ethernet-Karte vom Typ PCnet32 oder PCnetPCI von AMD besitzt, für den ist dieses der richtige Treiber, der auch als Modul geladen werden kann.

**Adaptec Starfire support (EXPERIMENTAL) (CONFIG\_ADAPTEC\_STARFIRE)**

Dieses ist der Treiber für die PCI-Karten Adaptec Starfire oder DuraLAN. Dieses sind beides 64-Bit-PCI-Karten. Ältere 32-Bit-Karten benutzen den Tulip-Treiber.

**Ansel Communications EISA 3200 support (CONFIG\_AC3200)**

Dieses ist der Treiber für die Netzwerkkarte Modell 3200 EISA von Ansel Communications. Weitere Informationen sind im Dokumentationsteil des Quelltextes in `drivers/net/cs3200.c` enthalten.

**Apricot Xen-II on board Ethernet (CONFIG\_APRICOT)**

Dieses ist der Treiber für den Apricot-82596-On-Board-Controller. Weitere Informationen sind im Dokumentationsteil des Quelltextes in drivers/net/apricot.c enthalten.

**CS89x0 support (CONFIG\_EPIC100)**

Der Treiber für die Karte SMC EtherPower II 9432 PCI E, die auf dem Chip SMC83c170 basiert. Dieser Treiber kann auch als Modul geladen werden.

**DECchip Tulip (dc21x4x) PCI support (CONFIG\_DEC\_ELCP)**

Dieser Treiber wurde für die Ethernet-Karten SMC EtherPower entwickelt, die auf den DEC-Chips 21040/21041/21140 aufsetzen, der sogenannten Tulip-Serie. Einige PCI-Karten von LinkSys enthalten auch diesen Chip. Weitere Informationen enthält /usr/src/linux/Documentation/networking/tulip.txt.

**Generic DECchip & DIGITAL EtherWORKS PCI/EISA (CONFIG\_DE4X5)**

Dieses ist der Treiber für die PCI/EISA-Karten der Firma Digital Equipment mit den Typenbezeichnungen DE425, DE434, DE435, DE450 und DE500. Weitere Informationen enthält der Text /usr/src/linux/Documentation/networking/de4x5.txt. Dieser Treiber kann auch als Modul geladen werden.

**Digi Intl. RightSwitch support (CONFIG\_DGRS)**

Dieses ist der Treiber für die PCI/EISA-Ethernet-Karten von Digi International RightSwitch mit den Modellbezeichnungen SE-4 und the SE-6. Weitere Information enthält /usr/src/linux/Documentation//networking/dgrs.txt. Dieser Treiber kann auch als Modul kompiliert werden.

**PCI DM9102 support (CONFIG\_DM9102)**

Dieses ist der Treiber für die Ethernet-PCI-Karte DM9102 von Davicom (<http://www.davicom.com.tw>) und Kompatiblen. Der Treiber kann auch als Modul geladen werden.

EtherExpressPro/100 support (CONFIG\_EEXPRESS\_PRO100)

Wer eine PCI-Netzwerkkarte des Typs Intel EtherExpress PRO/100 besitzt, für den ist dieses der richtige Treiber. Er kann auch als Modul geladen werden.

Enable Power Management (EXPERIMENTAL) (CONFIG\_EEPRO100\_PM)

Viele der Intel PCI-Netzwerkkarten EtherExpress PRO/100 besitzen die Fähigkeit zu Power-Management. Um diese zu aktivieren muß hier mit y geantwortet werden. Achtung: dieses ist immer noch sehr experimenteller Code und sollte nur zum Testen und Entwickeln aktiviert werden.

Mylex EISA LNE390A/B support (CONFIG\_LNE390)

Der Treiber für die EISA-Ethernet-Karten LNE390A und LNE390B. Er kann auch als Modul geladen werden.

National Semiconductor DP83810 series PCI Ethernet support (CONFIG\_NATSEMI)

Der Treiber für die PCI-Ethernet-Karten Baureihe DP83810 von National Semiconductor einschließlich des Chips 83815. Die neuesten Treiber und weitere Information enthält <http://www.scyld.com/network/natsemi.html>.

PCI NE2000 and clones support (see help) (CONFIG\_NE2K\_PCI)

Den Klassiker der Netzwerkkarten gibt es auch in der PCI-Ausführung und dieses ist der Treiber dafür. Dieser Treiber ist auch für folgende Clones zuständig: RealTek RTL-8029, Winbond 89C940, Compex RL2000, KTI

ET32P2, NetVin NV5000SC, Via 86C926, SureCom NE34, Winbond, Holtek HT80232 und Holtek HT80229. Er kann auch als Modul geladen werden.

Novell/Eagle/Microdyne NE3210 EISA support (CONFIG\_NE3210)

Der Treiber für die EISA-Ethernet-Karte NE 3210 der Firmen Novell, Eagle und Microdyne. Er kann auch als Modul kompiliert werden.

Racal-Interlan EISA ES3210 support (CONFIG\_ES3210)

Der Treiber für die EISA-Karte Racal-Interlan ES3210, der auch als Modul geladen werden kann.

RealTek RTL-8139 PCI Fast Ethernet Adapter support (CONFIG\_8139TOO)

Der Treiber für alle Fast Ethernet PCI-Karten, die auf dem Chip RTL8139 basieren. Details enthält Documentation/networking/8139too.txt.

RealTek 8129 (not 8019/8029/8139!) support (EXPERIMENTAL)  
(CONFIG\_RTL8129)

Der Treiber für alle Fast Ethernet PCI-Karten, die auf dem Chip RTL8139 basieren. Achtung: für den Chip RTL8139 ist obiger Treiber zuständig (CONFIG\_8139TOO). Der 8029 hingegen ist ein NE2000-PCI-Clone, man sollte den Treiber NE2K-PCI benutzen.

SiS 900/7016 PCI Fast Ethernet Adapter support (CONFIG\_SIS900)

Dieser Treiber unterstützt die PCI-Netzwerkkarten, die auf den Chips SiS 900, SiS 7016, SiS 630 und SiS 540 beruhen. Weitere Information enthält Documentation/networking/sis900.txt.

SMC EtherPower II (CONFIG\_EPIC100)

Dieses ist der Treiber für die PCI-Ethernetkarte SMC EtherPower II 9432, die den Chip SMC83c170 enthält. Der Treiber kann als Modul kompiliert werden.

Sundance Alta support (CONFIG\_SUNDANCE)

Der Treiber für den Chip Alta von Sundance. Weitere Information und die neuesten Treiber erhält man von <http://www.scyld.com/network/sundance.html>.

TI ThunderLAN support (CONFIG\_TLAN)

Mehrere Karten basieren auf dem CHIP ThunderLAN. Dieses ist der zugehörige Treiber. Er kann auch als Modul geladen werden.

VIA Rhine support (CONFIG\_VIA\_RHINE)

Falls eine VIA-Netzwerkkarte der Typen Rhine-I (3043) oder Rhine-2 (VT86c100A) zum Einsatz kommt, so ist diesesm der richtige Treiber. Er kann auch als Modul kompiliert werden.

Winbond W89c840 Ethernet support (CONFIG\_WINBOND\_840)

Der Treiber für den Chip Winbond W89c840. Er arbeitet auch mit dem Chip TX9882 auf dem Compex RL100-ATX-Board. Weitere Information und die neuesten Treiber erhält man von <http://www.scyld.com/network/drivers.html>.

Sun Happy Meal 10/100baseT PCI support (CONFIG\_HAPPYMEAL)

Dieser Treiber unterstützt die Schnittstelle hme, die auf den meisten Ultra-Systemen sowie auf älteren Systemen mit Sbus vorhanden sind. Der Treiber unterstützt sowohl Sbus als auch PCI.

AT-LAN-TEC/RealTek pocket Adapter support (CONFIG\_ATP)

Diese Karte wird an der parallelen Schnittstelle angeschlossen. Weitere Informationen enthält die Dokumentation im Quelltext `drivers/net/atp.c`. Wer diesen Treiber benutzt, sollte bei der Option `Parallel printer support`, `CONFIG_PRINTER` mit `n` antworten, da die beiden Treiber sich leider gegenseitig behindern.

D-Link DE600 pocket Adapter support (CONFIG\_DE600)

Dieses ist der Treiber für den D-Link-Adapter DE 600. Wenn neben diesem Adapter die parallele Schnittstelle auch für den Drucker benutzt werden soll, sollten sowohl diese als auch die Option `Parallel printer support`, `CONFIG_PRINTER` als Modul kompiliert werden. Weitere Informationen sind unter `drivers/net/README.DLINK` vorhanden.

D-Link DE620 pocket Adapter support (CONFIG\_DE620)

Dieses ist der Treiber für den D-Link-Adapter DE 620. Wenn neben diesem Adapter die parallele Schnittstelle auch für den Drucker benutzt werden soll, sollten sowohl diese als auch die Option `Parallel printer support`, `CONFIG_PRINTER` als Modul kompiliert werden. Weitere Informationen sind unter `drivers/net/README.DLINK` vorhanden.

## Ethernet (1000 Mbit)

Aufgrund der überarbeiteten Sortierung finden wir hier alle Treiber für das Gigabit-Netz.

Alteon AceNIC/3Com 3C985/NetGear GA620 Gigabit support  
(CONFIG\_ACENIC)

Diese ist der Treiber für die Gigabit-Karten Alteon AceNIC und 3Com 3C985 PCI. Dieser Treiber ermöglicht das Senden von sogenannten „Jumbo Frames“, die bis zu 9000 Byte pro Frame aufnehmen können. Dazu muß `ifconfig` mit dem Parameter `add mtu 9000` aufgerufen werden. Dieser Treiber kann auch als Modul geladen werden.

Omit support for old Tigon I based AceNICs  
(`CONFIG_ACENIC OMIT_TIGON_I`)

Durch diese Option kann die Treibergröße um ca. 100 kB verkleinert werden, wenn man sich sicher ist, daß die benutzte Karte **nicht** auf dem älteren Tigon eins basiert, also nicht eine der älteren Karten Alteon AceNIC oder 3Com 3C985 (nicht B) ist. Dieses bedeutet also, daß die Karte auf dem neueren Tigon II basieren muß.

Packet Engines Hamachi GNIC-II support (`CONFIG_HAMACHI`)

Wer solch eine Gigabit-Ethernet-Karte betreiben will, antwortet mit `y`.

Packet Engines Yellowfin Gigabit-NIC support (`CONFIG_YELLOWFIN`)

Dieses ist der Treiber für die Yellowfin PCI-Gigabyte-Ethernet-Karte. Dieses wird u.a. von dem LINUX-Cluster Beowulf genutzt. Details enthält <http://cesdis.gsfc.nasa.gov/linux/drivers/yellowfin.html>. Dieser Treiber kann auch als Modul geladen werden.

SysKonnект SK-98xx support (`CONFIG_SK98LIN`)

Hiermit werden folgende Karten von SysKonnект unterstützt: SK-9841, SK-9842, SK-9843, SK-9844, SK-9821, SK-9822, SK-9861 und SK-9862. Außerdem wird die Karte AT2970 von Allied Telesyn unterstützt. Weitere Information enthält `Documentation/networking/sk98lin.txt`.

FDDI driver support (`CONFIG_FDDI`)

Wer genügend Geld für das Glasfasernetz FDDI (Fiber Distributed Data Interface) besitzt, der sagt hier ja und kann mit 100 Mbps in seinem lokalen Netz arbeiten. Inzwischen geht dieses auch über Kupferkabel. Außerdem muß eine der folgenden FDDI-Netzwerkkarten aktiviert werden.

Digital DEFEA and DEFPA adapter support (CONFIG\_DEFXX)

Dieses ist der Treiber für die FDDI-Karten von Digital Equipment namens DEFEA (ISA) bzw. DEFPA (PCI).

SysKnect FDDI PCI support (CONFIG\_SKFP)

Der Treiber für folgende Adapter von SysKnect: SK-5521, SK-5522, SK-5541, SK-5543, SK-5544, SK-5821, SK-5822, SK-5841, SK-5843, SK-5844, Netelligent 100 FDDI DAS Fibre SC, Netelligent 100 FDDI SAS Fibre SC, Netelligent 100 FDDI DAS UTP, Netelligent 100 FDDI SAS UTP und Netelligent 100 FDDI SAS Fibre MIC. Details enthält Documentati-on/networking/skfp.txt.

High Performance Parallel Interface support (EXPERIMENTAL)  
(CONFIG\_HIPPI)

Das Blödels mit Abkürzungen hat ein neues Opfer: HIPPI. Dieses ist die Abkürzung für High Performance Parallel Interface. HIPPI kann Rechner mit 800Mbit/sec bzw. 1600Mbit/sec verbinden, für kurze Strecken bis 25 m über Kupferkabel, für weitere Entfernungen über Glasfaserkabel. Da HIPPI entweder Cluster verbindet oder aber Supercomputer, werden die meisten Menschen hier mit nein antworten.

Essential RoadRunner HIPPI PCI adapter support (CONFIG\_ROADRUNNER)

Wenn das Protokoll HIPPI heißt, so ist es nur konsequent, eine zugehörige Netzwerkkarte Roadrunner zu taufen. Falls dieses die Netwerkkarte der

Wahl ist, so muß mit y geantwortet werden. Der Treiber kann auch als Modul geladen werden.

#### Use large TX/RX rings (CONFIG\_ROADRUNNER\_LARGE\_RINGS)

Für große Netze kann mittels dieses Treibers bestimmt werden, daß bis zu 2 MB Arbeitsspeicher für die Roadrunner-Karte (s.o.) reserviert werden, um schnelleren Datenverkehr zu ermöglichen. Dazu braucht man natürlich genügend Arbeitsspeicher, um diese 2 MB entbehren zu können.

#### PPP (point-to-point protocol) support (CONFIG\_PPP)

Das PPP (Point to Point Protocol) ist eine Weiterentwicklung des SLIP, daß jedoch die gleiche Aufgabe erfüllt: den Datenverkehr des Internet über Telefon- oder andere serielle Leitungen zu bewegen. Weitere Informationen sind in Documentation/networking/ppp.txt enthalten. Dieser Treiber kann als Modul kompiliert und geladen werden, was sogar geschehen muß, falls in Sektion 2 (Verwendung optional zu ladender Module) die Option Set Version information on all symbols for modules CONFIG\_MODVERSIONS gesetzt wurde.

#### PPP multilink support (EXPERIMENTAL) (CONFIG\_PPP\_MULTILINK)

Das PPP-Multilink-Protokoll (definiert in RFC 1990) ermöglicht das Kombinieren von mehreren logischen oder physikalischen Leitungen zu einer einzigen logischen PPP-Verbindung, um so die Bandbreite zu erhöhen. Dieses erfordert zum einen natürlich den gleichen Mechanismus am anderen Ende der Verbindung sowie den Daemon pppd.

#### PPP support for async serial ports (CONFIG\_PPP\_ASYNC)

Falls PPP asynchron über serielle Schnittstellen betrieben werden soll, so ist dieses der richtige Treiber. Falls also z.B. ein Modem benutzt wird als Verbin-

dung zum Internetprovider, wird diese Option benötigt. Dieses gilt nicht für ISDN-Modems.

PPP support for sync tty ports (CONFIG\_PPP\_SYNC\_TTY)

Falls PPP über synchrone Verbindungen (HDLC) betrieben wird, z.B. über den SyncLink-Adapter, so ist dieses der richtige Treiber. Oft benutzt für Hochgeschwindigkeitsleitungen.

PPP Deflate compression (CONFIG\_PPP\_DEFLATE)

Der Linux-Treiber zur Datenkompression für PPP. Er benutzt den gleichen Algorithmus wie gzip. Der Rechner am anderen Ende der Leitung muß dieses natürlich unterstützen. Andererseits ist es unproblematisch, hier mit y zu antworten.

PPP BSD-Compress compression (CONFIG\_PPP\_BSDCOMP)

Unterstützung für Datenkompression im BSD-Stil für PPP. Die Kompression benutzt den LZW-Algorithmus. Der Rechner am anderen Ende der Leitung muß dieses natürlich unterstützen. Der obige Treiber CONFIG\_PPP\_DEFLATE komprimiert zum einen besser und zum zweiten ist er frei von Patenten.

PPP over Ethernet (EXPERIMENTAL) (CONFIG\_PPPOE)

Unterstützung für PPP über Ethernet. Dieser Treiber benötigt außerdem noch einen speziell gepatchten Daemon pppd, der von folgender Adresse heruntergeladen werden kann: <http://www.math.uwaterloo.ca/~mostrows>.

SLIP (serial line) support (CONFIG\_SLIP)

Falls mittels SLIP (Serial Line Internet Protocol) oder CSLIP (Compressed SLIP) eine Verbindung zum Internet-Provider oder ein Internet-Server aufge-

baut werden soll, muß hier mit y geantwortet werden. Die SLIP-Unterstützung vergrößert den Kernel um 4 KB.

#### CSLIP compressed headers (CONFIG\_SLIP\_COMPRESSED)

Dieses Protokoll ist schneller als SLIP, da es die TCP/IP-Header komprimiert (aber nicht die Daten selbst). CSLIP muß aber natürlich auch am anderen Ende der Kommunikationsleitung unterstützt werden. Falls dies nicht der Fall ist, kann aber in jedem Fall SLIP benutzt werden. Diese Option vergrößert den Kernel nicht.

#### Keepalive and linefill (CONFIG\_SLIP\_SMART)

Fügt dem SLIP-Treiber weitere Kommunikationsfähigkeiten hinzu. Gerne genutzt für analoge Leitungen mit schlechter Übertragung.

#### Six bit SLIP encapsulation (CONFIG\_SLIP\_MODE\_SLIP6)

Falls das serielle Netz über Rechner läuft, die das achte Bit nicht unterstützen, kann diese Option genutzt werden. In diesem Fall werden nur sieben-Bit-ASCII-Daten übertragen, wobei das andere Ende der Verbindung natürlich im gleichen Modus betrieben werden muß.

#### Wireless LAN (non-hamradio) (CONFIG\_NET\_RADIO)

Treiber für Netzwerkbetrieb über Amateurfunk (AX.25), drahtloses Ethernet und andere System. Hiermit werden nur die folgenden Fragen aktiviert.

#### STRIP (Metricom Starmode radio IP) (CONFIG\_STRIP)

Wer ein Funkgerät von Metricom benutzt, hat hier den richtigen Treiber. Dieses ist ein kleines, batteriegespeistes Gerät in der Größe eines Handys. Das genutzte Protokoll heist STRIP (Starmode Radio IP) und ist vom MosquitoNet (<http://mosquitonet.stanford.edu>) entwickelt worden. STRIP kann

mit jedem LINUX-Rechner genutzt werden, der eine serielle Schnittstelle besitzt. Dieser Treiber kann auch als Modul kompiliert werden.

AT&T WaveLAN & DEC RoamAbout DS support (CONFIG\_WAVELAN)

Das Lucent WaveLAN (ehemals NCR, AT&T oder DEC RoamAbout DS) ist ein Funk-LAN, das auf Frequenzen zwischen 900 MHz und 2.4 GHz arbeitet. Dieser Treiber unterstützt die ISA-Version der Karte. Weitere Information enthält  
der Text  
`/usr/src/linux/Documentation/networking/wavelan.txt` sowie  
die Webadresse  
[http://www.hpl.hp.com/personal/Jean\\_Tourrilhes/Linux/Wave](http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Wave)  
lan.html. Dieser Treiber kann auch als Modul geladen werden.

Aironet Arlan 655 & IC2200 DS support (CONFIG\_ARLAN)

Die Arlan-Karte wird von Aironet ([www.aironet.com](http://www.aironet.com)) produziert. Die Karte enthält einen Chip von Telxon ([www.Telxon.com](http://www.Telxon.com)), der sich auf mehreren Karten befindet. Der Treiber wurde mit den Karten 655 und IC2200 getestet. Weitere Information bietet <http://www.ylenurme.ee/~elmer/655>.

Aironet 4500/4800 series adapters (CONFIG\_AIRONET4500)

Der Treiber für die Adapter der Serien 4500/4800. Detaillierte Fragen über den Adaptertyp folgen sofort, wenn man hier mit y antwortet. Weitere Information enthält [www.aironet.com](http://www.aironet.com).

Aironet 4500/4800 ISA/PCI/PNP/365 support  
(CONFIG\_AIRONET4500\_NONCS)

Wer eine Aironet-Karte der Typen 4500/4800 an PNP, ISA, PCI oder PCMCIA betreibt, antwortet hier mit y und muß außerdem die folgenden Fragen beantworten.

Aironet 4500/4800 PNP support (CONFIG\_AIRONET4500\_PNP)

Wer eine ISA-Karte der Typen Aironet 4500/4800 unter PnP betreibt, muß hier mit y antworten. Außerdem muß natürlich der PnP-Jumper auf der Karte aktiviert werden.

Aironet 4500/4800 PCI support (CONFIG\_AIRONET4500\_PCI)

Der Treiber für die PCI-Karten der Typen Aironet 4500/4800 .

Aironet 4500/4800 ISA broken support (EXPERIMENTAL)  
(CONFIG\_AIRONET4500\_ISA)

Experimenteller Treiber, um die ISA-Karte ohne PnP-Modus zu betreiben.  
Derzeit noch nicht recht funktionstüchtig.

Aironet 4500/4800 I365 broken support (EXPERIMENTAL)  
(CONFIG\_AIRONET4500\_I365)

Wer die PCMCIA-Karte Aironet 4500/4800 ohne Standard-PCMCIA-Kartendienste betreiben will, antwortet hier mit y. Derzeit nicht empfohlen.

Aironet 4500/4800 PROC interface (CONFIG\_AIRONET4500\_PROC)

Der Treiber, um Informationen über die Aironet-Karten über das virtuelle Dateisystem /proc zu erhalten, und zwar bei /proc/sys/aironet4500.

## Token Ring devices

Token Ring driver support (CONFIG\_TR)

Token Ring ist die Netzwerk-Topologie von IBM. Sie hat den Vorteil, daß bei nur einem defekten Rechner oder nur einer defekten Netzwerkkarte im Netz

das gesamte Netz ruht. Deshalb benutzen Behörden IBM mit Token-Ring-Topologie, der Rest der Welt aber andere Topologien. Inzwischen hat IBM den physikalischen Aufbau des Token Ring geändert: es ist jetzt kein Ring mehr, sondern das Netz besitzt eine Stern-Topologie.

#### IBM Tropic chipset based adapter support (CONFIG\_IBMTR)

Dieses ist der Treiber für alle IBM-Token-Ring-Karten, die keinen DMA-Zugriff ausführen. Genauere Informationen sind erhältlich unter <ftp://metalab.unc.edu/pub/Linux/docs/HOWTO>. Achtung, dieser Treiber fällt fast immer aus, wenn mehr als eine aktive Token-Ring-Karte im Rechner vorhanden ist. Dieser Treiber kann auch als Modul geladen werden.

#### IBM Olympic chipset PCI adapter support (CONFIG\_IBMOL)

Diese ist der Treiber für folgende PCI Token Ring Karten: alle IBM PCI, PCI Wake On Lan, PCI II, PCI II Wake On Lan und PCI 100/16/4. Weitere Informationen enthält <ftp://metalab.unc.edu/pub/Linux/docs/HOWTO>. Das neueste vom Token-Ring-Projekt findet man unter <http://www.linuxtr.net>.

#### IBM Lanstreamer chipset PCI adapter support (CONFIG\_IBMLS)

Der Treiber für die IBM PCI-Karte Lanstreamer für Token Ring.

#### Generic TMS380 Token Ring ISA/PCI adapter support (CONFIG\_TMS380TR)

Dieser Treiber enthält generische Unterstützung für alle Token-Ring-Karten, die auf dem Chip TMS380 von Texas Instruments basieren. Dieses sind SysKnect TR4/16(+) ISA (SK-4190), SysKnect TR4/16(+) PCI (SK-4590), SysKnect TR4/16 PCI (SK-4591), Compaq 4/16 PCI, Thomas-Conrad TC4048 4/16 PCI und viele Karten von Madge. Die entsprechende Karte wird im Folgenden abgefragt. Weitere Informationen unter Documentati-

on/networking/tms380tr.txt und im Internet unter  
<http://www.auk.cx/tms380tr/>.

#### Generic TMS380 PCI support (CONFIG\_TMSPCI)

Dieser Treiber enthält Unterstützung für folgende PCI-Karten, die auf dem Chip TMS380 basieren: Compaq 4/16 TR PCI, SysKonnect TR4/16 PCI (SK-4590/SK-4591), Thomas-Conrad TC4048 PCI 4/16 und 3Com Token Link Velocity.

#### Madge Smart 16/4 PCI Mk2 support (CONFIG\_ABYSS)

Der Treiber für die Karte Madge Smart 16/4 PCI Mk2.

#### Madge Smart 16/4 Ringnode MicroChannel (CONFIG\_MADGEMC)

Der Treiber für die Karten Madge Smart 16/4 MC16 und MC32 am Micro-Channel-Bus.

#### SMC ISA/MCA adapter support (CONFIG\_SMCTR)

Der Treiber für die ISA- und MCA-Token-Ring-Karten, insbesondere SMC TokenCard Elite (8115T) und SMC TokenCard Elite/A (8115T/A). Details erhält Documentation/networking/smctr.txt.

#### Fibre Channel driver support (CONFIG\_NET\_FC)

„Fibre Channel“ ist ein serielles Hochgeschwindigkeits-Protokoll für Glasfaser, das hauptsächlich benutzt wird, um Rechner mit großen Massenspeichern zu verbinden. Es ist kompatibel mit SCSI und soll dieses eines Tages ersetzen. Wer den Fibre Channel benutzen möchte, antwortet hier mit y und wählt seinen Adapter weiter unten. Außerdem sollten die beiden Treiber SCSI support und SCSI generic support aktiviert sein.

Interphase 5526 Tachyon chipset based adapter support  
(CONFIG\_IPHASE5526)

Der Treiber für Fibre-Channel-Karten, die auf dem Chip Interphase 5526 Tachyon basieren.

Red Creek Hardware VPN (EXPERIMENTAL) (CONFIG\_RCPCI)

Für die Hardware von Red Creek, die ein virtuelles privates Netzwerk (VPN) ermöglicht.

Traffic Shaper (EXPERIMENTAL) (CONFIG\_SHAPER)

Der Treiber ermöglicht die Begrenzung der gesendeten Datenmenge, falls diese über andere Netzwerke von geringerer Bandbreite geschickt werden muß. Weitere Information enthält Documentati-on/networking/shaper.txt.

Wan interfaces support (CONFIG\_WAN)

Wide Area Networks (WANs), wie z.B. X.25, Frame Relay und gemiete Leistungen werden benutzt, um Local Area Networks (LANs) über große Distanzen zu verbinden, und zwar mit einer Datentransferrate, die deutlich höher liegt, als z.B. Modem- oder ISDN-Verbindungen. Normalerweise wird hierzu relativ teure Hardware, ein WAN-Router benötigt. Die Alternative hierzu sind die relativ günstigen WAN-Karten, mit der man einen Linux-Rechner direkt mit dem WAN verbinden kann. Dieses ist der grundlegende Treiber; die Fragen nach den einzelnen Karten folgen sofort.

Comtrol Hostess SV-11 support (CONFIG\_HOSTESS\_SV11)

Diese Karte unterstützt synchrone serielle Verbindungen bis maximal 256 Kbps. Der Treiber unterstützt sowohl PPP als auch Ciscos HDLC. Derzeit muß der Treiber als Modul kompiliert werden.

**COSA/SRP sync serial boards support (CONFIG\_COSA)**

Dieses ist der Treiber für Karten für seriellen synchronen Datenaustausch der Firmen COSA und SRP. Hiermit können die verschiedensten seriellen Geräte nach verschiedenen Normen angesprochen werden, wie X.21, V.24, V.35 oder V.36. Für weitere Informationen und Dienstprogramme sehe man unter folgender Adresse nach: <http://www.fi.muni.cz/~kas/cosa>.

**MultiGate (COMX) synchronous serial boards support (CONFIG\_COMX)**

Dieser Treiber unterstützt die Multigate-Karten. Dieses sind synchrone serielle Karte für PCs, die von der Firma ITConsult-Pro Co in Ungarn hergestellt werden. Weitere Information enthält Documentation/networking/comx.txt und im Internet <http://www.itc.hu>.

**Support for COMX/CMX/HiCOMX boards (CONFIG\_COMX\_HW\_COMX)**

Der Hardware-Treiber für die verschiedenen Typen der Karten der MultiGate-Familie, nämlich CMX, COMX und HiCOMX. Zum Betrieb dieser Karten ist spezielle Firmware erforderlich, die man bei <ftp://ftp.itc.hu/> herunterladen kann.

**Support for LoCOMX board (CONFIG\_COMX\_HW\_LOCOMX)**

Der Treiber für die Karte LoCOMX von MultiGate.

**Support for MixCOM board (CONFIG\_COMX\_HW\_MIXCOM)**

Der Treiber für die Karte MixCOM von MultiGate.

**Support for HDLC and syncPPP protocols on MultiGate boards (CONFIG\_COMX\_PROTO PPP)**

Falls Cisco-HDLC oder synchrones PPP über eine MultiGate-Karte betrieben werden soll, so ist dieses der richtige Treiber.

Support for LAPB protocol on MultiGate boards  
(CONFIG\_COMX\_PROTO\_LAPB)

Falls LAPB über eine MultiGate-Karte betrieben werden soll, so ist dieses der richtige Treiber.

Support for Frame Relay on MultiGate boards (CONFIG\_COMX\_PROTO\_FR)

Falls LFrame Relay über eine MultiGate-Karte betrieben werden soll, so ist dieses der richtige Treiber.

LanMedia Corp. SSI/V.35, T1/E1, HSSI, T3 boards (CONFIG\_LANMEDIA)

Der Treiber für die folgenden seriellen Karten von LanMedia: LMC 1000, LMC 1200, LMC 5200, LMC 5245.

Sealevel Systems 4021 support (CONFIG\_SEALEVEL\_4021)

Der Treiber für die serielle Karte ACB 56 von Sealevel Systems.

SyncLink HDLC/SYNCPPP support (CONFIG\_SYNCLINK\_SYNCPPP)

Für den normalen Gebrauch von PPP für den SyncLink-WAN-Treiber genügt der normale PPP-Treiber. Falls aber Ciscos HDLC/PPP benötigt wird, so wird dieser Treiber gebraucht. Der grundlegende Treiber für die SyncLink-Karte CONFIG\_SYNCLINK muß natürlich auch aktiviert sein.

Frame Relay (DLCI) support (CONFIG\_DLCI)

Dieses ist die Unterstützung für das Frame-Relay-Protokoll, das eine schnelle und kostengünstige Möglichkeit darstellt, sich mit einem Internet-Provider zu verbinden oder ein privates Wide-Area-Netzwerk einzurichten. Die physikalische Verbindung von einem LINUX-Rechner zur lokalen Frame-Relay-Hardware kann mehrere logische Punkt-zu-Punkt-Verbindungen enthalten, die andere Rechner mit dem Frame-Relay-Netzwerk verbindet. Um dieses zu benutzen, benötigt man spezielle Hard- und Software. Genaues findet man unter <http://www.frforum.com> und unter `/usr/src/linux/Documentation/networking/framerelay.txt`.

#### Max open DLCI (CONFIG\_DLCI\_COUNT)

Dieses ist die maximale Anzahl von logischen Punkt-zu-Punkt-Verbindungen, die der Treiber bearbeiten kann. Die Standard-Einstellung beträgt 24.

#### Max DLCI per device (CONFIG\_DLCI\_MAX)

Hier kann eingestellt werden, wie viele logische Punkt-zu-Punkt-Verbindungen von jedem physikalischen Frame-Relay-Gerät bearbeitet werden können. Die Standard-Einstellung ist 8.

#### SDLA (Sangoma S502/S508) support (CONFIG\_SDLA)

Dieses ist der Treiber für die Frame-Relay-Hardware Sangoma S502A, S502E und S508. Obwohl dieses Multi-Protokoll-Karten sind, wird derzeit nur der Zugriff Frame Relay Zugriff vom Treiber unterstützt. Dieser Treiber kann auch als Modul kompiliert werden.

#### WAN router drivers (CONFIG\_WAN\_DRIVERS)

Falls der LINUX-Rechner zu einem WAN-Router ausgebaut werden soll, so muß hier mit y geantwortet werden. Diese Option aktiviert nur eine Liste weitere Menüpunkte. Details findet man unter `/usr/src/linux/Documentation/networking/wan-router.txt`.

Sangoma WANPIPE(tm) multiprotocol cards (CONFIG\_VENDOR\_SANGOMA)

Die WANPIPE-Karten von Sangoma (<http://www.sangoma.com>) unterstützen die verschiedensten Protokolle wie X.25, Frame Relay und PPP. Details findet man unter /usr/src/linux/Documentation/networking/wanpipe.txt.

Die nächsten Optionen stellen Fragen zu den Details dieser Karte.

Maximum number of cards (NEW) (CONFIG\_WANPIPE\_CARDS)

Die Anzahl der WANPIPE-Karten im LINUX-Router. Die Standardeinstellung ist 1; der Treiber unterstützt maximal 8 Karten.

WANPIPE Cisco HDLC support (CONFIG\_WANPIPE\_CHDLC)

Der Treiber um die WANPIPE-Karte unter dem synchronen HDLC-Protokoll von Cisco zu betreiben.

WANPIPE PPP support (NEW) (CONFIG\_WANPIPE\_PPP)

Wer eine Standleitung sein eigen nennt, und die WANPIPE-Karte hierfür einzusetzen möchte, benötigt diesen Treiber. Außerdem muß das PPP (s.o.) aktiviert sein.

Cyclom 2X(tm) cards (EXPERIMENTAL) (CONFIG\_CYCLADES\_SYNC)

Der Treiber für die Multiprotokoll-WAN-Karte Cyclom 2X von Cyclades. Der Treiber kann eine oder mehrere Karten ansteuern. Weitere Information unter <http://www.cyclades.com>. Dieses ist der Hardwaretreiber; die verschiedenen Protokolle werden als nächste abgefragt.

Cyclom 2X X.25 support (CONFIG\_CYCLOMX\_X25)

Der Treiber, um das Protokoll X.25 an einer Cyclom-2X-Karte zu betreiben.

LAPB over Ethernet driver (EXPERIMENTAL) (CONFIG\_LAPBETHER)

Dieses ist der Treiber für ein Pseudo-Gerät - typischerweise /dev/lapb0 - das eine Punkt-zu-Punkt-Verbindung über Ethernet ermöglicht. Falls dieses erwünscht ist, wird außerdem der Treiber LAPB Data Link Driver benötigt.

X.25 async driver (CONFIG\_X25\_ASY)

Dieser Treiber ermöglicht den Datenaustausch mittels des Protokolls X.25 über asynchrone serielle Leitungen, wie z.B. Telefonleitungen über Standardmodems. Der Treiber kann auch als Modul geladen werden.

SBNI12-xx support (CONFIG\_SBNI)

Dieses ist der Treiber für die ISA-Karte SBNI12-xx. Weitere Informationen sind erhältlich in Rußland unter <http://www.granch.ru>. Der Treiber kann auch als Modul geladen werden.

PCMCIA network device support (CONFIG\_NET\_PCMCIA)

Wer seinen Netzwerkadapter auf einer PCMCIA- oder CardBus-Karte hat, der muß diesen Treiber aktivieren und außerdem die gewünschte Karte im Folgenden auswählen. PCMCIA-Karten sind die Geräte von Kreditkartengröße, die normalerweise im Zusammenhang mit mobilen Rechnern benutzt werden. CardBus ist die neuere und schnellere Version von PCMCIA.

3Com 3c589 PCMCIA support (CONFIG\_PCMCIA\_3C589)

Der Treiber für die PCMCIA-Netzwerkkarte 3c589 von 3Com.

3Com 3c574 PCMCIA support (CONFIG\_PCMCIA\_3C574)

Der Treiber für die PCMCIA-Netzwerkkarte 3c574 von 3Com.

Fujitsu FMV-J18x PCMCIA support (CONFIG\_PCMCIA\_FMvj18X)

Der Treiber für die PCMCIA-Netzwerkkarte FMV-J18x von Fujitsu.

NE2000 compatible PCMCIA support (CONFIG\_PCMCIA\_PCNET)

Der Treiber für die PCMCIA-Netzwerkkarte NE2000 oder kompatibel.

New Media PCMCIA support (CONFIG\_PCMCIA\_NMCLAN)

Der Treiber für die PCMCIA-Netzwerkkarten von New Media oder LiveWire.

SMC 91Cxx PCMCIA support (CONFIG\_PCMCIA\_SMC91C92)

Der Treiber für die PCMCIA-Netzwerkkarte 91Cxx von SMC.

Xircom 16-bit PCMCIA support (CONFIG\_PCMCIA\_XIRC2PS)

Der Treiber für die älteren 16-Bit-PCMCIA-(Fast)-Ethernet-Karten von Xircom.

COM20020 ARCnet PCMCIA support (CONFIG\_ARCNET\_COM20020\_CS)

Der Treiber für die PCMCIA-Karte COM20020, um ACRnet zu betreiben.

IBM PCMCIA tokenring adapter support (CONFIG\_PCMCIA\_IBMTR)

IBM bietet auch Unterstützung für Token Ring mittels PCMCIA-Karten. Außerdem muß natürlich der Treiber Token Ring driver support aktiviert sein.

**Xircom Tulip-like CardBus support (CONFIG\_PCMCIA\_XIRTULIP)**

Dieses ist der Treiber für den PCMCIA-Adapter von Digital namens Tulip. Außerdem arbeitet er mit den baugleichen Modellen Lite-On (PNIC), Macronix (MXIC) und ASIX.

**Pcmcia Wireless LAN (CONFIG\_NET\_PCMCIA\_RADIO)**

Der grundlegende Treiber, falls ein drahtloses Netzwerk mittels einer PCMCIA-Karte betrieben werden soll. Die einzelnen Kartentypen werden im Folgenden abgefragt.

**Aviator/Raytheon 2.4MHz wireless support (CONFIG\_PCMCIA\_RAYCS)**

Der Hardware-Treiber für die PCMCIA-Karte von Aviator/Raytheon für den Betrieb eines drahtlosen Netzwerks. Details enthält Documentation/networking/ray\_cs.txt.

**Xircom Netwave AirSurfer wireless support (CONFIG\_PCMCIA\_NETWAVE)**

Der Hardware-Treiber für die PCMCIA-Karte Netwave AirSurfer von Xircom für den Betrieb eines drahtlosen Netzwerks.

**AT&T/Lucent Wavelan wireless support (CONFIG\_PCMCIA\_WAVELAN)**

Der Hardware-Treiber für die PCMCIA-Karte Wavelan von AT&T/Lucent für den Betrieb eines drahtlosen Netzwerks.

**Aironet 4500/4800 PCMCIA support (CONFIG\_AIRONET4500\_CS)**

Der Hardware-Treiber für die PCMCIA-Karten 4500/4800 von Aironet.

## ATM drivers

ATM over TCP (NEW) (`CONFIG_ATM_TCP`)

Der Treiber für ATM über TCP ist immer noch in einem sehr experimentellen Stadium. Wer Genaueres wissen möchte, dem sei folgende Seite empfohlen:  
<http://icaww1.epfl.ch/linux-atm/>.

Efficient Networks ENI155P (NEW) (`CONFIG_ATM_ENI`)

Der Treiber für die beiden Adapter Efficient Networks ENI155p und SMC ATM Power155, die mit 155 Mbps arbeiten.

Fujitsu FireStream (FS50/FS155) (NEW) (`CONFIG_ATM_FIRESTREAM`)

Der Treiber für den Adapter Fujitsu FireStream.

ZeitNet ZN1221/ZN1225 (NEW) (`CONFIG_ATM_ZATM`)

Der Treiber für die Adapter ZeitNet ZN1221 und ZN1225.

IDT 77201 (NICStAR) (ForeRunnerLE) (NEW) (`CONFIG_ATM_NICSTAR`)

Der Treiber für Adapter mit dem Chipsatz NICStAR, der z.B. in den Geräten von IDT und in der Fore ForeRunnerLE vorhanden ist.

Madge Ambassador (Collage PCI 155 Server) (NEW)  
(`CONFIG_ATM_AMBASSADOR`)

Der Treiber für den Adapter Ambassador von Madge Networks.

Madge Horizon [Ultra] (Collage PCI 25 and Collage PCI 155 Client)  
(`CONFIG_ATM_HORIZON`)

Für Adapter mit dem Chipset Horizon von Madge Networks.

#### Interphase ATM PCI x575/x525/x531 (NEW) (CONFIG\_ATM\_IA)

Der Treiber für die Adapter ChipSAR von Interphase, die in verschiedenen Typen hergestellt werden. Informationen über die Karten enthält: [www.ipphase.com/products/ClassSheet.cfm?ClassID=ATM](http://www.ipphase.com/products/ClassSheet.cfm?ClassID=ATM). Weitere Information über den Treiber enthält Documentation/networking/ipphase.txt.

#### FORE Systems 200E-series (NEW) (CONFIG\_ATM\_FORE200E\_MAYBE)

Dieses ist der Treiber für die Baureihe 200E von FORE Systems. Er unterstützt außerdem die Modelle PCA-200E und SBA-200E. Weitere Information enthält Documentation/networking/fore200e.txt.

## 17. Amateur Radio support

#### Amateur Radio support (CONFIG\_HAMRADIO)

Nicht jedes Land ist derart von Kabeln durchzogen, wie z.B. Deutschland. Nun gibt es aber Landstriche und Länder, in denen LINUX-Begeisterte auch am Datenverkehr teilhaben möchten, aber Datenleitungen wurden weit und breit nicht gesichtet. Mittels Amateur-Funk und etwas Software kann man aber auch ein LAN mit Ethernet zusammenbauen. Für die meisten Mitteleuropäer ist dieses kaum von Interesse, da es hier einfachere Wege gibt; für die Tiefen Kanadas, für den dünnbesiedelten mittleren Westen der USA oder für Zimbabwe ist dieses aber durchaus eine gängige Möglichkeit zu kommunizieren. Seitenweise Material findet man unter <http://www.tapr.org/tapr/html/pkthome.html>. Wenn man hier mit y antwortet, so generiert dieses keinen Code für den Kernel, sondern aktiviert die folgenden Fragen.

Amateur Radio AX.25 Level 2 protocol (CONFIG\_AX25)

Dieses ist das Standard-Protokoll für Rechner-Kommunikation via Amateurfunk. Entweder es wird für Punkt-zu-Punkt-Verbindungen benutzt, oder aber um andere Protokolle, wie z.B. TCP/IP zu beherrschen. Um das Protokoll AX.25 zu benutzen benötigt man unterstützende Hardware. Die preiswerte Lösung ist ein TNC (Terminal Node Controller), der an die serielle Schnittstelle angeschlossen wird und das KISS-Protokoll benutzt. Die nächste Möglichkeit ist eine der verschiedenen SCC-Karten, die von den Treibern für Z8530 oder DMA SCC unterstützt werden. Die dritte Möglichkeit besteht darin, die seriellen oder parallelen Baycom-Modem einzusetzen. Die Treiber für die jeweilige Hardware werden weiter unten gefunden. Weitere Information findet man unter `/usr/src/linux/Documentation/networking/ax25.txt`. Dieser Treiber kann auch als Modul geladen werden.

AX.25 DAMA Slave support (CONFIG\_AX25\_DAMA\_SLAVE)

DAMA ist der Mechanismus, um Packet-Kollisionen zu verhindern, während man das Protokoll AX.25 fährt. Ein DAMA-Server (Master) nimmt die eingehenden Datenpakete von den Clients (Slaves) an und verteilt die an die anderen Sklaven. Außer der Einbindung dieses Treibers muß nichts weiter konfiguriert werden; der LINUX-Rechner ist sofort bereit zum Einsatz als DAMA-Slave.

Amateur Radio NET/ROM protocol (CONFIG\_NETROM)

NET/ROM ist eine Netzwerk-Protokoll-Schicht, die über dem AX.25 liegt und viel verwendet wird. Dieser Treiber kann auch als Modul geladen werden.

Amateur Radio X.25 PLP (Rose) (CONFIG\_ROSE)

Das Packet Layer Protokoll (PLP) ist eine Alternative zu NET/ROM (s.o.) und somit eine weitere Möglichkeit, um Datenpakete über eine (A)X.25-Verbindung zu routen.

## AX.25 network device drivers

### Serial port KISS driver (CONFIG\_MKISS)

KISS ist ein Standard-Protokoll, um Daten zwischen einem Rechner und einem TNC auszutauschen. TNCs (Terminal Node Controller) sind Schnittstellen, die die serielle Schnittstelle des Rechners mit dem Mikrofon und den Lautsprechern des Funkgeräts verbinden. Da KISS ein einfaches Protokoll ist, wird es von fast allen TNCs unterstützt. Dieser Treiber kann auch als Modul kompiliert werden.

### Serial port 6PACK driver for AX.25 (CONFIG\_6PACK)

6pack ist die Alternative zu KISS, um eine Verbindung zwischen der seriellen Schnittstelle des Rechners und dem Funkgerät herzustellen. 6pack bietet mehr Möglichkeiten als KISS. Weitere Information enthält der Text `/usr/src/linux/Documentation/networking/6pack.txt`. Der Treiber kann auch als Modul kompiliert werden.

### BPQ Ethernet driver (CONFIG\_BPQETHER)

Dieses ist der Treiber, um Ethernet-Datenpakete über das Amateurfunk-Standardprotokoll AX.25 zu senden.

### High-speed (DMA) SCC driver for AX.25 (CONFIG\_DMASC)

Dieses ist der Treiber für die SCC-Karten, die auf einem Port DMA-Zugriff unterstützen. Diese Karten werden normalerweise benutzt, um den Rechner mit einem Amateurfunk-Modem zu verbinden. Im Moment werden folgende Modems unterstützt: das WA4DSY, das Ottawa PI/PI2 (<http://hydra.carleton.ca/info/pi2.html>) und das Gracilis PackeTwin (<http://www.paccomm.com/gracilis.html>). Sie werden automatisch erkannt. Der Treiber kann mehrere dieser Karten parallel betreiben.

Dabei sollte man die entsprechenden Boot-Parameter für jedes Board übergeben in der folgenden Form:

io=addr1,addr2,...

Bevor jede Schnittstelle mittels `ifconfig` konfiguriert wird, müssen einige Parameter gesetzt werden, wie z.B. der DMA-Kanal. Dieses geschieht mittels eines Dienstleistungsprogramms namens `dmascc_cfg`, das unter folgender Adresse erhältlich ist:

<http://www.nt.tuwien.ac.at/~kkudielk/Linux/>.

#### Z8530 SCC driver (CONFIG\_SCC)

Diese Karten werden genutzt, um Rechner mit Amateurfunkgeräten zu verbinden. Falls dieser Treiber benötigt wird, muß außerdem die Option Amateur Radio AX.25 Level 2 aktiviert werden. Weitere Information enthält der Text `/usr/src/linux/Documentation/networking/z8530drv.txt`. Der Treiber kann auch als Modul kompiliert werden.

additional delay for PA0HZP OptoSCC compatible boards (NEW)  
(CONFIG\_SCC\_DELAY)

Falls der SCC-Treiber nicht sauber arbeitet, sollte man es mit dieser Option versuchen. Details findet man unter `/usr/src/linux/Documentation/networking/z8530drv.txt`.

additional delay for PA0HZP OptoSCC compatible boards  
(CONFIG\_SCC\_DELAY)

Falls Probleme mit dem SCC-Treiber auftreten sollten, so kann diese Option aktiviert werden. Weiterführende Information enthält `Documentation/networking/z8530drv.txt`.

BAYCOM ser12 full duplex driver for AX.25 (CONFIG\_BAYCOM\_SER\_FDX)

Dieses ist einer der zwei Treiber, die Amateurfunk-Modems der Firma Baycom sowie Kompatible mit dem Rechner verbinden. Der Treiber bietet vollen Duplex-Modus sowie Baudraten zwischen 300 und 4800. Dieses ist der neuere Treiber. Falls man Probleme, z.B. wegen älterer Chips in seinem Modem hat, so sollte der nächste, ältere Treiber gewählt werden. Um diesen Treiber zu konfigurieren, benötigt man das Dienstprogramm `sethdlc`, das in dem Standardpaket für AX.25 vorhanden ist. Weitere Informationen enthält `/usr/src/linux/Documentation/networking/baycom.txt`; Information über seine Modems hält Baycom selbst unter <http://www.baycom.de> bereit. Der Treiber kann als Modul kompiliert werden.

BAYCOM ser12 half duplex driver for AX.25 (`CONFIG_BAYCOM_SER_HDX`)

Falls der ver angehende Treiber wegen älterer Hardware nicht sauber arbeitet, so sollte man es mit diesem, älteren Treiber versuchen. Er kann auch als Modul geladen werden.

YAM driver for AX.25 (`CONFIG_YAM`)

Dieser Treiber unterstützt das YAM-Modem am seriellen Port und kann als Modul kompiliert werden.

## 18. IrDA (infrared) support

IrDA subsystem support (`CONFIG_IRDA`)

IrDA ist die Abkürzung für Infrared Data Associations, eine Organisation, die den Standard für drahtlose Infrarot-Kommunikation erstellt hat. Mit entsprechender Hardware und den folgenden Treibern können diverse Peripheriegeräte angesteuert werden, ohne daß der sonst notwendige Kabelsalat entsteht. IrDA wird inzwischen von vielen Laptops genutzt. Um unter LINUX IrDA zu nutzen, werden zusätzliche Dienstprogramme wie `irmanager` und `irattach` benötigt. Weitere Informationen enthält

/usr/src/linux/Documentation/networking/irda.txt. Der Treiber kann auch als Modul kompiliert werden. Und wenn alles korrekt konfiguriert ist, alle Treiber geladen sind, und auch die Hardware intakt ist, aber dann läuft die infrarote Verbindung immer noch nicht? Dann muß man den Stapel Aktenordner zwischen Rechner und Peripheriegerät entfernen, den auch Infrarot ist Licht - wenn auch unsichtbares - wie der Name bereits sagt. Dieser Tip entspringt übrigens nicht unserer Fantasie sondern der leidvollen Praxis - natürlich mit Grafikern!

#### IrLAN Protocol (CONFIG\_IRLAN)

Das Protokoll IrLAN ermöglicht die Ethernet-Kommunikation über Infrarot. Beispiele für die zugehörige Hardware sind HP NetbeamIR oder ESI JetEye NET. Dieser Treiber kann auch als Modul kompiliert werden.

#### IrCOMM Protocol (CONFIG\_IRCOMM)

IrCOMM ist das Protokoll um serielle Schnittstellen über IrDA anzusteuern, d.h. für jegliche seriellen Geräte ist dieser Treiber notwendig. Weitere Information enthält <http://www.pluto.dti.ne.jp/~thiguchi/irda/>. Der Treiber kann auch als Modul kompiliert werden.

#### IrDA protocol options (CONFIG\_IRDA\_OPTIONS)

Dieser Menüpunkt aktiviert nur die folgenden Fragen, erzeugt also keinen Kernel-Code.

#### IrDA Cache last LSAP (CONFIG\_IRDA\_CACHE\_LAST\_LSAP)

Diese Option aktiviert den Cache für IrDA. Da meistens die Empfangs- und Sendedaten über die gleiche Verbindung laufen, ist dieses nützlich. Im Zweifelsfalle y.

#### Fast RRs (CONFIG\_IRDA\_FAST\_RR)

Normalerweise muß eine IrDA-Station eine bestimmte Zeit warten, bis das nächste Datenpaket verschickt werden darf. Wenn größere Datenmengen verschickt werden sollen, so schickt mittels dieses Treibers IrLAP sofort, nachdem es einen Daten-Frame erhalten hat, ein RR (Receive Ready). Der Kommunikationspartner kann also sofort weitere Daten senden, ohne zu warten. Dieses erhöht deutlich die Datentransferrate.

#### Debug information (CONFIG\_IRDA\_DEBUG)

Wenn dieser Treiber aktiviert wird, werden Informationen nach syslog geschrieben. Für den Entwickler.

#### IrTTY (uses Linux serial driver) (CONFIG\_IRTTY\_SIR)

Dieses ist der Treiber, um IrDA über serielle Schnittstellen zu betreiben, die kompatibel zum CHIP 16550 sind, welches der Standard-Chip ist. Die Transferrate ist damit auf 115200 Baud beschränkt.

#### IrPORT (IrDA serial driver) (CONFIG\_IREPORT\_SIR)

Dieses ist der neuere Treiber für den Betrieb von IrDA über serielle Schnittstellen. Er führt manchmal zu besseren Ergebnissen.

#### NSC PC87108/PC87338 (CONFIG\_NSC\_FIR)

Dieser Treiber ist für die IrDA-ISA-Karte ACTiSYS IR2000B mit dem Chipsatz NSC PC87108, die einen maximalen Datentransfer von 4 Mbps ermöglicht. Der Treiber kann auch als Modul kompiliert werden.

#### Winbond W83977AF (IR) (CONFIG\_WINBOND\_FIR)

Dieses ist der Treiber für den IrDA-Chip Winbond W83977AF, der z.B. im Corel NetWinder vorhanden ist. Die maximale Datentransferrate beträgt 4Mbps.

Toshiba Type-O IR Port (CONFIG\_TOSHIBA\_FIR)

Dieses ist der Treiber für den Chip Toshiba Type-O IR, der z.B. in dem Laptop Toshiba Libretto 100CT vorhanden ist.

SMC IrCC (Experimental) (CONFIG\_SMC\_IRCC\_FIR)

Dieses ist der Treiber für den SMC-Infrarot-Controller. Er wird benutzt im Fujitsu Lifebook 635t und im Sony PCG-505TX.

Serial dongle support (CONFIG\_DONGLE)

Hier muß mit y genatwortet werden, wenn ein Infrarot-Gerät am seriellen Port des Rechners angeschlossen wird. Diese werden auch als Dongles bezeichnet. Die Fragen nach den verschiedenen Dongle-Typen folgern sofort.

ESI JetEye PC Dongle (CONFIG\_ESI\_DONGLE)

Der ESI-Dongle wird auf den seriellen Port gesteckt. Um das Dongle zu aktivieren, muß in dem Skript /etc/irda/drivers die Zeile irattach -d esi eingefügt werden.

ACTiSYS IR-220L and IR220L+ dongle (CONFIG\_ACTISYS\_DONGLE)

Dieses ist der Treiber für die Dongle ACTiSYS IR-220L und IR220L+. Die Dongle werden auf die serielle Schnittstelle gesteckt. Um das Dongle zu aktivieren, muß in dem Skript /etc/irda/drivers die Zeile irattach -d actisys eingefügt werden. Dieser Treiber kann auch als Modul kompiliert werden.

Tekram IrMate 210B dongle (CONFIG\_TEKRAM\_DONGLE)

Das Dongle Tekram IrMate 210B wird auf die serielle Schnittstelle gesteckt. Um das Dongle zu aktivieren, muß in dem Skript /etc/irda/drivers die Zeile irattach -d tekram eingefügt werden.

#### Greenwich GIrBIL dongle (CONFIG\_GIRBIL\_DONGLE)

Dieses ist der Treiber für das Dongle Greenwich GIrBIL. Das Dongle wird auf die serielle Schnittstelle gesteckt. Um das Dongle zu aktivieren, muß in dem Skript /etc/irda/drivers die Zeile irattach -d girbil eingefügt werden.

#### Parallax Litelink dongle (CONFIG\_LITELINK\_DONGLE)

Das Dongle Parallax Litelink wird auf die serielle Schnittstelle gesteckt. Um das Dongle zu aktivieren, muß in dem Skript /etc/irda/drivers die Zeile irattach -d litelink eingefügt werden.

#### Old Belkin dongle (CONFIG\_OLD\_BELKIN\_DONGLE)

Der Treiber für die älteren Modelle von Belkin.

#### Adaptec Airport 1000 and 2000 dongle (CONFIG\_AIRPORT\_DONGLE)

Dieses ist der Treiber für die Dongles Adaptec Airport 1000 und 2000. Die Dongles werden auf den seriellen Port gesteckt. Um diese Dongles zu aktivieren, muß in dem Skript /etc/irda/drivers die Zeile irattach -d airport eingefügt werden. Dieser Treiber kann auch als Modul kompiliert werden.

## 19. ISDN subsystem

#### ISDN support (CONFIG\_ISDN)

ISDN (Integrated Services Digital Networks) wird über voll digitalisierte Telefonleitungen über Glasfaserkabel oder hochwertige Kupferkabel realisiert. Die gängigsten Protokolle für die Anbindung eines LINUX-Rechners über ISDN sind die Protokolle SLIP oder PPP.

Der größte Vorteil von ISDN ist die höhere Transferrate im Verhältnis zu Modems. Beide Teilnehmer der Verbindung müssen eine ISDN-Karte besitzen und an das ISDN-Netz angeschlossen sein. Der Treiber unterstützt auch das Wählen über die ISDN-Karte und enthält auch einen Hayes-kompatiblen Modem-Emulator. Die Karten selbst unterstützen bestimmte ISDN-Leistungsmerkmale, wie z. B. Kanal-Bündelung, ohne daß ein Daemon dafür arbeiten muß. Die Protokolle EDSS1 und 1TR6 wie auch das deutsche BTX-Protokoll werden unterstützt. Weitere Informationen enthält Documentation/isdn/README.

#### Support synchronous PPP (CONFIG\_ISDN\_PPP)

Diese Option aktiviert das synchrone PPP via ISDN. Dieses Protokoll wird z. B. von Sun oder Cisco benutzt. Für dieses Protokoll wird eine spezielle Version von pppd namens ippdd gebraucht. Für diese Option muß die Option TCP/IP networking CONFIG\_INET gesetzt sein. Weitere Informationen sind in Documentation/isdn/README.syncppp und Documentation/isdn/syncPPP.FAQ vorhanden.

#### Use VJ-compression with synchronous PPP (CONFIG\_ISDN\_PPP\_VJ)

Aktiviert die Van-Jacobson-Header-Komprimierung bei synchronem PPP.

#### Support generic MP (RFC 1717) (CONFIG\_ISDN\_MPP)

Wenn das synchrone PPP aktiviert ist, ist es möglich, die Transferrate zu verdoppeln, indem zwei ISDN-Verbindungen gebündelt werden. Weitere Informationen enthält die Datei Documentation/isdn/README.syncppp.

**Support audio via ISDN (CONFIG\_ISDN\_AUDIO)**

Wenn diese Option aktiviert ist, unterstützt der Modem-Emulator eine Unter-  
menge der EIA-Class-8-Voice-Befehle. Durch die Benutzung eines getty mit  
Stimm-Unterstützung kann der LINUX-Rechner als ISDN-Anrufbeantworter  
benutzt werden. Natürlich muß dieses auch vom Lowlevel-Treiber unterstützt  
werden, was derzeit nur der Teles-Treiber realisiert. Weitere Informationen  
sind in Documentation/isdn/README.audio enthalten.

**Support AT-Fax Class 1 and 2 commands (CONFIG\_ISDN\_TTY\_FAX)**

Wenn man diesen Treiber aktiviert, unterstützt der Modem-Emulator eine  
Untermenge der Befehle der Fax-Klasse 2. Wenn man die Faxunterstützung  
mit mgetty und sendfax oder mit hylafax kombiniert, kann man den LINUX-  
Rechner als ISDN-FAX-Maschine benutzen. Der jeweilige Gerätetreiber muß  
natürlich zusätzlich hinzugefügt werden. Weitere Information enthält  
`/usr/src/linux/Documentation/isdn/README.fax`.

**X.25 PLP on top of ISDN (CONFIG\_ISDN\_X25)**

Dieser Treiber bietet das Protokoll X.25 via ISDN an. Weitere Information ent-  
hält `/usr/src/linux/Documentation/isdn/README.x25`.

**ISDN feature submodules****isdnloop support (CONFIG\_ISDN\_DRV\_LOOP)**

Dieser Treiber emuliert eine virtuelle ISDN-Karte. Der Zweck ist das Testen  
von Treibern und Programmen, ohne jedesmal Geld für einen Anruf bezahlen  
zu müssen. Wer hiermit arbeiten möchte, benötigt das Dienstprogramm  
`loopctrl` aus dem Paket `isdn4k-utils`. Gedacht für Entwickler.

**ISDN diversion services support (CONFIG\_ISDN\_DIVERSION)**

Diese Option ermöglicht es, in Verbindung mit dem HiSax-Treiber weitere Leistungsmerkmale von ISDN zu nutzen. Derzeit unterstützt werden CD (call deflection), CFU (Call forward unconditional), CFB (Call forward when busy) and CFNR (call forward not reachable).

### Passive ISDN cards

HiSax SiemensChipSet driver support (CONFIG\_ISDN\_DRV\_HISAX)

HiSax ist der Siemens-Chipsatz, der auf vielen Karten zum Einsatz kommt, wie z.B. AVM A1, Elsa, Teles S0-16.0, Teles S0-16.3, Teles S0-8, Teles/Creatix PnP, ITK micro ix1 und vielen Kompatiblen. Wenn sich dieser Chip auf der ISDN-Karte befindet, so sollte mit y geantwortet werden. Der Treiber für die jeweilige Karte muß weiter unten gewählt werden. Der Treiber kann auch als Modul kompiliert werden. Weitere Information enthält /usr/src/linux/Documentation/isdn/README.HiSax.

HiSax Support for EURO/DSS1 (CONFIG\_HISAX\_EURO)

Falls ISDN über das Euro-Protokoll gefahren werden soll, muß dieser Treiber gewählt werden. Dieser Treiber schließt sich gegenseitig mit dem nächsten (dier veralteten deutschen Norm 1TR6) aus. Die neueren deutschen ISDN-Anlagen fahren inzwischen alle Euro-Norm.

Support for german chargeinfo (CONFIG\_DE\_AOC)

Die Deutsche Telekom mußte wieder einige zusätzlichen Leistungsmerkmale an die ISDN-Norm ranflicken, so auch die Gebührenübermittlung während des Rufes sowie nach dem Gespräch. Wer auf diese Informationen zugreifen möchte, benötigt diesen Treiber. Dieses funktioniert nur in Deutschland!

Disable sending complete (CONFIG\_HISAX\_NO\_SENDCOMPLETE)

Wer merkwürdige Probleme während des Datentransfers hat, sollte es mit diesem Treiber versuchen. Außerdem wird er für australisches ISDN benötigt.

Disable sending low layer compatibility (CONFIG\_HISAX\_NO\_LLC)

Noch ein Treiber für den Fall von defekten Daten während der Übertragung.

Disable keypad protocol option (CONFIG\_HISAX\_NO\_KEYPAD)

Falls Sonderzeichen innerhalb der zu wählenden Nummer geschickt werden sollen, wie z.B. \* oder #.

HiSax Support for german 1TR6 (CONFIG\_HISAX\_1TR6)

Die Unterstützung der veralteten deutschen ISDN-Norm 1TR6. Diese findet sich noch selten in älteren ISDN-Telefonanlagen. Die neueren deutschen ISDN-Anlagen fahren inzwischen alle Euro-Norm.

HiSax Support for US NI1 (CONFIG\_HISAX\_NI1)

Für die Benutzung von ISDN in den USA an der Schnittstelle NI1.

Teles 16.0/8.0 (CONFIG\_HISAX\_16\_0)

Dieses ist der Treiber für die Teles-Kartens S0-16.0, S0-8 und viele andere Kompatible.

Teles 16.3 or PNP or PCMCIA (CONFIG\_HISAX\_16\_3)

Dieses ist der Treiber für die Teles Karte S0-16.3, die Teles/Creatix PnP und die Teles PCMCIA.

Teles PCI (CONFIG\_HISAX\_TELESPCI)

Der Treiber für die Teles-PCI-Karte.

Teles S0Box (CONFIG\_HISAX\_S0BOX)

Der Treiber für die Teles/Creatix S0Box, die am parallelen Port angeschlossen wird.

AVM A1 (Fritz) (CONFIG\_HISAX\_AVM\_A1)

Dieses ist der Treiber für die Karte A1 von AVM, in Deutschland besser bekannt als Fritz-Card. Nicht verwechseln mit der B1 Karte von AVM, die eine aktive Karte ist und für die ein anderer Treiber zuständig ist (s.u.).

AVM PnP/PCI (Fritz!PNP/PCI) (CONFIG\_HISAX\_FRITZPCI)

Der Treiber für die neuen Karten von AVM Fritz!PnP und Fritz!PCI.

AVM A1 PCMCIA (Fritz) (CONFIG\_HISAX\_AVM\_A1\_PCMCIA)

Der Treiber für die PCMCIA-Version der Fritz-Karte von AVM.

Elsa cards (CONFIG\_HISAX\_ELSA\_PCC)

Der Treiber für die Karten Elsa Mircolink und die Quickstep-Baureihe, die ISA-Karten und die PCMCIA-Karten von Elsa.

ITK ix1-micro Revision 2 (CONFIG\_HISAX\_IX1MICROR2)

Der Treiber für die Karte ITK ix1-micro (Revision 2).

Eicon.Diehl Diva cards (CONFIG\_HISAX\_DIEHLDIVA)

Der Treiber für die passive ISDN-Karte Eicon.Diehl Diva.

HiSax Support for ASUSCOM cards (CONFIG\_HISAX\_ASUSCOM)

Der Treiber für die passive ISDN-Karte AsusCom und kompatibel.

TELEINT cards (CONFIG\_HISAX\_TELEINT)

Der Treiber für die semiaktive ISDN-Karte TELEINT SA1.

HFC-S based cards (CONFIG\_HISAX\_HFCS)

Der Treiber für die auf HFC-S 2BDS0 beruhenden Karten, wie z.B. die Teles 16.3c.

Sedlbauer cards (CONFIG\_HISAX\_SEDLBAUER)

Der Treiber für die ISDN-Karte von Sedlbauer.

USR Sportster internal TA (CONFIG\_HISAX\_SPORTSTER)

Der Treiber für die ISDN-Karte USR Sportster.

MIC card (CONFIG\_HISAX\_MIC)

Der Treiber für die ISDN-Karte ITH MIC.

NETjet card (CONFIG\_HISAX\_NETJET)

Der Treiber für die ISDN-Karte NetJet von Traverse Technologies.

NETspider U card (CONFIG\_HISAX\_NETJET\_U)

Der Treiber für die ISDN-Karte Netspider U interface von Traverse Technologies.

Niccy PnP/PCI card (`CONFIG_HISAX_NICCY`)

Der Treiber für die ISDN-Karte Dr. Neuhaus Niccy (PnP oder PCI).

Siemens I-Surf card (`CONFIG_HISAX_ISURF`)

Der Treiber für die ISDN-Karte Siemens I-Talk/I-Surf mit dem ISAR-Chip.

HST Saphir card (`CONFIG_HISAX_HSTSAPHIR`)

Der Treiber für die ISDN-Karte HST Saphir.

Telekom A4T card (`CONFIG_HISAX_BKM_A4T`)

Der Treiber für die ISDN-Karte Telekom A4T.

Scitel Quadro card (`CONFIG_HISAX_SCT_QUADRO`)

Der Treiber für die ISDN-Karte Scitel Quadro.

Gazel cards (`CONFIG_HISAX_GAZEL`)

Der Treiber für die ISDN-Karten von Hazel.

HFC PCI-Bus cards (`CONFIG_HISAX_HFC_PCI`)

Der Treiber für ISDN-Karten, die auf dem Chip HFC-S PCI 2BDS0 basieren.

Winbond W6692 based cards (`CONFIG_HISAX_W6692`)

Der Treiber für PCI-ISDN-Karten die auf dem Chip Winbond W6692 basieren.

HFC-S+, HFC-SP, HFC-PCMCIA cards (`CONFIG_HISAX_HFC_SX`)

Der Treiber für die Karten HFC-S+, HFC-SP and HFC-PCMCIA. Derzeit noch eine Baustelle.

### Active ISDN cards

#### ICN 2B and 4B support (CONFIG\_ISDN\_DRV\_ICN)

Dieses ist der Treiber für die Karten 2 B und 4 B der Firma ICN. 2 B ist die Standard-Version für eine einfache ISDN-Leitung mit zwei B-Kanälen, 4 B unterstützt vier Kanäle. Weitere Informationen enthalten die Texte Documentation/isdn/README und Documentation/isdn/README.icn.

#### PCBIT-D support (CONFIG\_ISDN\_DRV\_PCBIT)

Treiber für die PCBIT-ISDN-Karten der portugiesischen Firma Octal. Um diese Karte zu betreiben, muß noch zusätzliche Firmware auf die Karte geladen werden. Weitere Informationen enthalten die Texte Documentation/isdn/README und Documentation/isdn/README.pcbit.

#### Spellcaster support (EXPERIMENTAL) (CONFIG\_ISDN\_DRV\_SC)

Der Treiber für die ISDN-Karten Spellcaster BRI. Dieser Treiber kann derzeit nur als Modul geladen werden. Weitere Information enthält /usr/src/linux/Documentation/isdn/README.sc und <http://www.spellcast.com>.

#### IBM Active 2000 support (EXPERIMENTAL) (CONFIG\_ISDN\_DRV\_ACT2000)

Dieser Treiber unterstützt die aktive ISDN-Karte IBM Active 2000. Zum Betreiben der Karte wird zusätzliche Firmware benötigt, die in die Karte geladen werden muß. Diese Firmware ist Bestandteil des Pakets isdn4k-utils.

Weitere Information enthält  
/usr/src/linux/Documentation/isdn/README.act2000.

Eicon.Diehl active card support (CONFIG\_ISDN\_DRV\_EICON)

Dieser Treiber unterstützt die aktive ISDN-Karte Eicon. Zum Betreiben der Karte wird zusätzliche Firmware benötigt, die in die Karte geladen werden muß. Diese Firmware ist Bestandteil des Pakets `isdn4k-utils`. Weitere Information enthält  
/usr/src/linux/Documentation/isdn/README.eicon.

build eicon driver type standalone  
(CONFIG\_ISDN\_DRV\_EICON\_STANDALONE)

Diese aktiviert die Möglichkeit, den EICON-Treiber ohne Schnittstelle zum ISDN4Linux-Modul zu erstellen.

CAPI2.0 support (CONFIG\_ISDN\_CAPI)

Dieser Treiber unterstützt die CAPI-Schnittstelle (Common ISDN Application Programming Interface). Dieser Standard ermöglicht einen einfacheren, standardisierten Zugriff auf ISDN-Hardware. Wird benötigt, um die aktiven Karten von AVM zu betreiben. Weitere Information enthält  
<http://www.capi.org/>.

AVM B1 ISA support (CONFIG\_ISDN\_DRV\_AVMB1\_B1ISA)

Der Treiber für die ISA-Version der aktiven ISDN-Karte AVM B1.

AVM B1 PCI support (CONFIG\_ISDN\_DRV\_AVMB1\_B1PCI)

Der Treiber für die PCI-Version der aktiven ISDN-Karte AVM B1.

AVM B1 PCI V4 support (CONFIG\_ISDN\_DRV\_AVMB1\_B1PCIV4)

Der Treiber für die PCI-Karte AVM B1 in der Version 4.

AVM T1/T1B ISA support (CONFIG\_ISDN\_DRV\_AVMB1\_T1ISA)

Der Treiber für die aktive ISA-ISDN-Karte AVM T1/T1B, die bis zu 30 B-Kanäle verwalten kann.

AVM B1/M1/M2 PCMCIA support (CONFIG\_ISDN\_DRV\_AVMB1\_B1PCMCIA)

Der Treiber für die PCMCIS-Version der aktiven ISDN-Karte AVM B1.

AVM T1/T1-B PCI support (CONFIG\_ISDN\_DRV\_AVMB1\_T1PCI)

Der Treiber für die aktive PCI-ISDN-Karte AVM T1/T1B, die bis zu 30 B-Kanäle verwalten kann.

AVM C4 support (CONFIG\_ISDN\_DRV\_AVMB1\_C4)

Der Treiber für die aktive PCI-ISDN-Karte AVM C4 PCI. Diese Karte kann 4 ISDN-Leitungen (8 Kanäle) verwalten.

Verbose reason code reporting (kernel size +=7K)  
(CONFIG\_ISDN\_DRV\_AVMB1\_VERBOSE\_REASON)

Falls dieser Treiber zusätzlich zu dem Treiber für die Karte AVM B1 aktiviert wird, werden die Fehlermeldungen betreffs der Verbindung im Klartext ausgegeben.

Hypercope HYSNDN cards (Champ, Ergo, Metro) support (module)  
(CONFIG\_HYSNDN)

Der Treiber für die aktiven ISDN-Karten Champ, Ergo and Metro der Firma Hypercope. Weitere Information enthält Documentati-  
on/isdn/README.hysdn.

#### HYSND CAPI 2.0 support (CONFIG\_HYSDN\_CAPI)

Falls für die Karten von Hypercope die Schnittstelle CAPI 2.0 benutzt werden soll, muß dieser Treiber aktiviert werden.

## 20. Old CD-ROM drivers (not SCSI, not IDE)

#### Support non-SCSI/IDE/ATAPI CD-ROM drives (CONFIG\_CD\_NO\_IDECSI)

Falls ein älteres CD-ROM-Laufwerk vorliegt, das einen eigenen Controller besitzt, also weder am SCSI- noch am IDE-/ATAPI-Bus arbeitet, muß hier mit y geantwortet werden. Diese Option veranlaßt nicht die Erzeugung von Code, sondern aktiviert nur die folgenden Fragen nach den proprietären CD-ROM-Laufwerken. Alle folgenden CD-ROM-Treiber können auch als Modul kompiliert werden, was sich dann empfiehlt, wenn das CD-ROM-Laufwerk nicht ständig im Einsatz ist.

#### Aztech/Orchid/Okano/Wearnes/TXC/CyDROM CDROM support (CONFIG\_AZTCD)

Diese Option aktiviert den Treiber für folgende Laufwerke: Aztech CDA268-01A, Orchid CD-3110, Okano oder Wearnes CDD110, Conrad TXC, CyCDROM CR520 oder CR540.

#### Goldstar R420 CDROM support (CONFIG\_GSCD)

Diese Option aktiviert den Treiber für das Goldstar-R-420-Laufwerk.

Matsushita/Panasonic/Creative, Longshine, TEAC CDROM support  
(CONFIG\_SBPCD)

Dieser Treiber unterstützt die meisten der Laufwerke, die die Panasonic- oder SoundBlaster-Schnittstelle benutzen. Zu diesen gehören die Laufwerke Matsushita CR-521, CR-522, CR-523, CR-562, CR-563, CreativeLabs CD 200, Longshine LCS-7260, das „IBM External ISA CD-ROM“ (in Wahrheit ein Modell CR-56x) und TEAC CD-55A. Einige ähnliche Geräte (Vertos, Genoa und einige Modelle von Funai) werden derzeit nicht unterstützt. Für das Modell H94A von Sanyo ist ein anderer Treiber zuständig (s. u.). Dieser Treiber kann bis zu vier Karten unterstützen, und an jeder Karte können bis zu vier CD-ROM-Laufwerke angeschlossen werden. Wenn hier mit y geantwortet wird, wird auch die Anzahl der vorhandenen Controller abgefragt. Wenn dieser Treiber als Modul kompiliert wird, kann er nur eine Karte (mit maximal vier Laufwerken) verwalten.

MicroSolutions backpack CDROM support (CONFIG\_BPCD)

MicroSolutions CD-ROM ist ein externes Gerät, das an die parallele Schnittstelle angeschlossen wird. Wenn mehrere Geräte an der parallelen Schnittstelle angeschlossen werden, sollte dieser Treiber derzeit als Modul kompiliert und dann von Hand ge- bzw. entladen werden, da er sich mit anderen Treibern für den parallelen Port „beißt“. Weitere Informationen sind vorhanden in Documentation/cdrom/bpcd.

Mitsumi standard [no XA/Multisession] CDROM support (CONFIG\_MCD)

Dies ist der Treiber für die älteren Mitsumi-Modelle LU-005, FX-001 und FX-001D. Dieser Treiber ist nicht zuständig für das Modell FX-001DE und die Triple- oder Quad-Speed-Modelle, da diese alle IDE-/ATAPI-Geräte sind. XA- oder Multi-Session-CDs (Photo-CDs) werden von diesem Treiber nicht unterstützt, der folgende Treiber für diese Laufwerke hat aber diese Fähigkeiten.

Mitsumi [XA/MultiSession] support (CONFIG\_MCDX)

Falls XA- oder Multi-Session-CDs (Photo-CDs) und Standard-CDs von den Mitsumi-Laufwerken LU-005, FX-001 oder FX-001D gelesen werden sollen, ist dieses der richtige Treiber. Ein positiver Nebeneffekt ist, daß er weniger Speicher als der eben erwähnte Treiber benötigt. Weitere Informationen sind in der Datei Documentation/cdrom/mcdx enthalten.

Optics Storage DOLPHIN 8000AT CDROM support (CONFIG\_OPTCD)

Dieses ist der Treiber für das Dolphin-Laufwerk mit einem 34-Pin-Stecker. Er unterstützt auch das Laufwerk Lasermate CR328A. Dieser Treiber ist nicht zuständig für das Laufwerk Optics Storage 8001, da dieses ein IDE-ATAPI-CD-ROM-Laufwerk ist.

Philips/LMS CM206 CDROM support (CONFIG\_CM206)

Treiber für das Philips-CD-ROM-Laufwerk CM 206.

Sanyo CDR-H94A CDROM support (CONFIG\_SJCD)

Treiber für das CD-ROM-Laufwerk CDR-H 94A von Sanyo.

Soft configurable cdrom interface card support (CONFIG\_CDI\_INIT)

Treiber für CD-ROM-Schnittstellen, die während des Hochfahrens des Systems initialisiert werden. Derzeit werden nur die Karten ISP16, MAD16 und Mozart unterstützt.

ISP16/MAD16/Mozart soft configurable cdrom interface support  
(CONFIG\_ISP16\_CDI)

Falls eine dieser CD-ROM-Karten, die auf den Chipsätzen OPTi 82C928 oder 82C929 basieren, erfaßt werden soll, muß hier mit `y` geantwortet werden. Der Text `Documentation/cdrom/isp16` erklärt Details.

Sony CDU31A/CDU33A CDROM support (`CONFIG_CDU31A`)

Diese Option aktiviert den Treiber für die Modelle CDU 31A und CDU 33A von Sony. Dem Treiber müssen beim Booten die notwendigen Parameter übergeben werden, wie in `Documentation/cdrom/cdu31a` beschrieben.

Sony CDU535 CDROM support (`CONFIG_CDU535`)

Dies ist der Treiber für die älteren Sony-Laufwerke CDU-531 und CDU-535.

## 21. Input core support

Input core support (`CONFIG_INPUT`)

Ziel dieser neuen Treibergruppe ist es, für sämtliche Eingabegeräte eine einheitliche Schnittstelle zur Verfügung zu stellen. Derzeit beschränkt sich dieses allerdings auf USB-Geräte.

Keyboard support (`CONFIG_INPUT_KEYBOARD`)

Der Treiber für USB-Tastaturen.

Mouse support (`CONFIG_INPUT_MOUSEDEV`)

Der Treiber für USB-Mäuse.

Horizontal screen resolution (`CONFIG_INPUT_MOUSEDEV_SCREEN_X`)

Nur notwendig, wenn man ein Digitizer oder ein Graphik-Tablett benutzt. In diesem Fall benötigt der Treiber die maximal mögliche X-Auflösung des Geräts.

Vertical screen resolution (CONFIG\_INPUT\_MOUSEDEV\_SCREEN\_Y)

Wie vorangehend, aber für die Y-Achse.

Joystick support (CONFIG\_INPUT\_JOYDEV)

Der Treiber für USB-Joysticks.

Event interface support (CONFIG\_INPUT\_EVDEV)

Falls man auf die USB-Geräte auf Ereignis-gesteuerter Basis zugreifen möchte, so benötigt man diesen Treiber.

## 22. Character devices

Virtual terminal (CONFIG\_VT)

Dieser Menüpunkt muß in aller Regel mit y beantwortet werden, den hiermit wird die Ausgabe am Bildschirm und die Eingabe an der Tastatur aktiviert. Nur wenn das LINUX-System als eingebettetes System genutzt werden soll, kann hier mit n geantwortet werden. Diese Terminals werden als virtuell bezeichnet, da man mehrere von Ihnen parallel laufen lassen kann, wie es bei jeder LINUX-Standard-Installation der Fall ist. Die virtuellen Terminals werden oft auch als virtuelle Konsolen bezeichnet. Zwischen ihnen kann normalerweise mit A!-& im Grafikmodus und im Dos-Emulator mit SA!-& umgeschaltet werden. Mit dem Dienstprogrammsetter kann man die Eigenschaften, wie z.B. die Farben eines virtuellen Terminals verändern. Der Anfänger sollte hier unbe-

dingt mit y antworten, weil er sonst von seinem neuen LINUX-System nichts sieht ....

#### Support for console on virtual terminal (CONFIG\_VT\_CONSOLE)

Die System-Konsole ist das Gerät, das alle (Fehler)-Meldungen und Warnungen des Kernels erhält und das Einloggen im Single-User-Modus erlaubt. Wenn hier mit y geantwortet wird, kann eines der virtuellen Terminals (s.o.) als System-Konsole benutzt werden. In den meisten Fällen sollte diese Option genutzt werden, es sei denn, daß man die Systemmeldungen auf eine serielle Schnittstelle schicken möchte; dann muß man die Option `Console on serial port` (s.u.) aktivieren. Falls man diese Option aktiviert, werden per Standardeinstellung alle System-Meldungen an die jeweils sichtbare virtuelle Konsole gesendet. Dieses Verhalten kann überschrieben werden durch einen entsprechenden Bootparameter `console=ttyx`. Dieses Verhalten ist z.B. von der SuSE-Distribution bekannt, die nach der Erstinstallation nach dem ersten Booten auf der virtuellen Konsole 9 (A) die Systemmeldungen anzeigt.

#### Standard/generic (8250/16550 and compatible UARTs) serial support (CONFIG\_SERIAL)

Dieses ist der Treiber für die Standard-Unterstützung von seriellen Schnittstellen. Nur wer keine Schnittstelle benötigt, sagt hier n. Dieses ist wohl relativ selten der Fall, kann aber auftreten, wenn der Rechner ein reiner Server werden soll oder wenn als Maus eine der Busmäuse zum Einsatz kommt. In der Regel wird aber mit y geantwortet.

#### Support for console on serial port (CONFIG\_SERIAL\_CONSOLE)

Dieser Treiber ist gedacht für den Fall, daß an dem Rechner kein Bildschirm vorhanden ist. Anstatt dessen wird eine serielle Schnittstelle als Systemkonsole betrachtet. Wenn dieser Treiber aktiviert wird, wird die Standardconsole `dev/tty0` als Standardausgabegerät aktiviert, falls eine VGA-Karte installiert

ist. Dieses kann allerdings durch einen Bootparameter, z.B. `console=ttyS1` geändert werden.

Wenn keine VGA-Karte auf dem Rechner vorhanden ist, so wird automatische die erste serielle Schnittstelle `/dev/ttys0` als Systemkonsole benutzt.

Extended dumb serial driver options (`CONFIG_SERIAL_EXTENDED`)

Wer zusätzlich zur den Standardmöglichkeiten des Treibers weitere Leistungsmerkmale benötigt, braucht diesen Treiber. Dieses ist z.B. die Unterstützung von HUB6, gemeinsam genutzte Interrupts für serielle Leitungen oder mehr als vier serielle Schnittstellen (s.u.). Die entsprechenden Optionen folgen sofort.

Support more than 4 serial ports (`CONFIG_SERIAL_MANY_PORTS`)

Wenn man eine Steckkarte für mehr als vier serielle Schnittstellen besitzt, so sollte hier mit `y` geantwortet werden, wenn es sogenannte „dumme“ Karten sind, wie AST FourPort, Accent Async und Boca. Dieser Treiber ist nicht zuständig für die sogenannten „intelligenten“ Karten mit mehreren Schittstellen, wie. z.B. Cyclades und Digiboards, für die weiter unten eigene Treiber aufgeführt sind.

Support for sharing serial interrupts (`CONFIG_SERIAL_SHARE_IRQ`)

Einige seriellen Boards ermöglichen das Betreiben von mehreren Schnittstellen, die sich einen einzigen IRQ teilen. Dieses ist der zugehörige Treiber.

Auto detect IRQ on standard ports (unsafe)  
(`CONFIG_SERIAL_DETECT_IRQ`)

Dieser Treiber versucht die zu den seriellen Schnittstelen zugehörigen IRQs herauszufinden. Dieses ist aber leider keine sichere Option. Der sichere Weg

ist, die IRQs in einem Boot-Skript mittels des Befehls `setserial` zuzuweisen.

#### Support special multiport boards (CONFIG\_SERIAL\_MULTIPORT)

Einige seriellen Multiport-Karten besitzen spezielle Schnittstellen, auf den signalisiert wird, ob eine bestimmte serielle Leitung bedient werden muß. Dieses ist der Treiber für diese spezielle Schnittstelle.

#### Support the Bell Technologies HUB6 card (CONFIG\_HUB6)

Der Treiber für die „dumme“ serielle Multiport-Karte HUB6 von Bell Technologies.

#### Non-standard serial port support (CONFIG\_SERIAL\_NONSTANDARD)

Wer serielle Multiport-Karten mit eigener „Intelligenz“ besitzt, sollte hier mit ja antworten. Beispiele für diese Karten sind z.B. Cyclades und Digiboards. Diese Karten werden normalerweise installiert, um zahlreiche Terminals oder Telefonleitungen zu bedienen. Dieser Menüpunkt erzeugt keine Kernel-Code, sondern aktiviert die folgenden Fragen.

#### Computone IntelliPort Plus serial support (CONFIG\_COMPUTONE)

Dieser Treiber unterstützt die gesamte Produktgruppe Intelliport II/Plus von Computone mit Ausnahme der Karten für den MicroChannel den Karten, die vor Intelliport II erschienen sind. Weitere Information über diese Multiport-Karten enthält `Documentation/computone.txt`.

#### Comtrol Rocketport support (CONFIG\_ROCKETPORT)

Dieses ist der Treiber für die serielle Multiport-Karte Comtrol Rocketport. Dieser Treiber kann auch als Modul geladen werden.

Cyclades async mux support (CONFIG\_CYCLADES)

Dieses ist der Treiber für die PC-Karten von Cyclades die mehrere serielle Schnittstellen enthalten.

Cyclades-Z interrupt mode operation (EXPERIMENTAL)  
(CONFIG\_CYZ\_INTR)

Die Cyclades-Karten erlauben zwei verschiedene Arbeitsmodi: Polling und Interrupt. Falls gepollt wird, überprüft der Treiber in zyklischen Abständen, ob Daten an der jeweiligen Schnittstelle anliegen. Der Polling-Zyklus ist konfigurierbar. Im Modus Interrupt werden eben diese benutzt, um festzustellen, welche Datenleitung gerade aktiviert werden sollte. Der Standard-Modus ist Polling.

Digiboard Intelligent Async Support (CONFIG\_DIGIEPCA)

Dieses ist der Treiber für die Karten Xx, Xeve und Xem von Digiboard, mit denen z.B. zahlreiche Modems an den seriellen Schnittstellen dieser Karten betrieben werden können. Der Treiber unterstützt die ISA-, EISA- und PCI-Versionen. Weitere Information enthält Documentation/digiepca.txt. Achtung: es existiert ein weiterer Treiber für diese Karten mit der Bezeichnung Digiboard PC/Xx Support. Nur einer der beiden Treiber darf aktiviert sein!!

Hayes ESP serial port support (CONFIG\_ESPSERIAL)

Dieser Treiber unterstützt die Karten Hayes ESP für serielle Schnittstellen. Sowohl die Single- als auch die Multiport-Karten werden unterstützt. Weitere Informationen enthält /usr/src/linux/Documentation/hayes-esp.txt. Dieser Treiber kann auch als Modul geladen werden.

Moxa Intellio support (CONFIG\_MOXA\_INTELLIO)

Der Treiber für die Multiport-Karte Intellio von Moxa.

Moxa SmartIO support (CONFIG\_MOXA\_SMARTIO)

Der Treiber für die Multiport-Karte SmartIO von Moxa.

Multi-Tech multiport card support (CONFIG\_ISI)

Dieses ist der Treiber für die serielle Multiport-Karte von Multi-Tech. Der Treiber kann derzeit nur als Modul kompiliert werden.

Microgate SyncLink card support (CONFIG\_SYNCLINK)

Dieser Treiber unterstützt die Multi-Protokoll-Karte Microgate SyncLink in ihrer ISA- und PCI-Version. Dieser Treiber muß als Modul kompiliert werden.

HDLC line discipline support (CONFIG\_N\_HDLC)

Dieser Treiber aktiviert die synchrone HDLC-Kommunikation der Karte Microgate SyncLink. Dieser Treiber muß als Modul kompiliert werden.

SDL RISCom/8 card support (CONFIG\_RISCOM8)

Dieses ist der Treiber für die multi-serielle Karte RISCom/8 von SDL Communications. Weitere Informationen sind in Documentation/riscom8.txt enthalten.

Specialix IO8+ card support (CONFIG\_SPECIALIX)

Dieses ist der Treiber für die serielle Multiport-Karte Specialix IO8+ multiport, und zwar sowohl für die ISA- als auch für die PCI-Karte. Weitere Informationen enthält der Text /usr/src/linux/Documentation/specialix.txt. Der Treiber kann als Modul kompiliert werden.

Specialix DTR/RTS pin is RTS (CONFIG\_SPECIALIX\_RTSCTS)

Die Specialix-Karten können entweder RTS oder DTR zum Handshaking benutzen. Wenn man diesen Treiber aktiviert, wird prinzipiell RTS benutzt. Weitere Informationen enthält der Text /usr/src/linux/Documentation/specialix.txt.

Specialix SX (and SI) card support (CONFIG\_SX)

Dieses ist der Treiber für die seriellen Multiport-Karten SX und SI von Specialix. Mehr Information in Documentation/sx.txt.

Specialix RIO system support (CONFIG\_RIO)

Dieses ist der Treiber für die seriellen Multiport-Karten RIO von Specialix.

Support really old RIO/PCI cards (CONFIG\_RIO\_OLDPCI)

Der Treiber für die älteren seriellen Multiport-Karten des Typs RIO am PCI-Bus von Specialix.

Stallion multiport serial support (CONFIG\_STALDRV)

Dieses sind die Treiber für die multi-seriellen Karten von Stallion wobei die einzelnen Modelle durch die nächsten Fragen spezifiziert werden.

Stallion EasyIO or EC8/32 support (CONFIG\_STALLION)

Dieses ist der Treiber für die Stallion-Karten EasyIO und EasyConnection 8/32. Weitere Informationen enthält drivers/char/README.stallion.

Stallion EC8/64, Onboard, Brumby support (CONFIG\_ISTALLION)

Dieses ist der Treiber für die Karten EasyConnection 8/64, Onboard, Brumby sowie die multi-serielle Karte von Stallion. Weitere Informationen enthält die Datei drivers/char/README.stallion.

#### Unix98 PTY support (CONFIG\_UNIX98\_PTYS)

Ein Pseudo-Terminal ist ein Gerät, das aus zwei Teilen besteht: einem Master und einem Slave. Der Slave verhält sich wie ein physikalisches Terminal, während der Master die Daten von und zum Slave bewegt und außerdem das Terminal emuliert. Typische Programme aus der Seite des Masters sind Telnet-Server und X-Terminals. Linux nannte bisher Master und Slave nach dem BSD-Schema /dev/ptyxx und /dev/ttyxx. Dieses Schema birgt einige Probleme. Seit den GNU C Bibliotheken glibc 2.1 wird den Unix98-Standard unterstützt: Um ein Pseudo-Terminal aufzubauen wird die Gerätedatei /dev/ptmx geöffnet. Die Nummer des Pseudo-Terminals wird übergeben und auf den Slave kann zugegriffen werden über /dev/pts<Nummer>. Was z.B. früher /dev/ttyp2 war ist jetzt /dev/pts/2. Die Dateien in /dev/pts/ werden dynamisch erzeugt durch das virtuelle Dateisystem. Wenn dieser Treiber aktiviert wird, muß auch die Option /dev/pts file-system for Unix98 PTYs eingeschaltet sein. Wenn dieser Treiber benötigt wird, so müssen die C-Bibliotheken glibc 2.1 (oder höher) vorhanden sein.

#### Maximum number of Unix98 PTYs in use (0-2048) (CONFIG\_UNIX98\_PTY\_COUNT)

Hier kann die maximale Anzahl der Unix-Pseudo-Terminals in dem Bereich von 0 bis 2048 angegeben werden. Die Standardeinstellung ist 256. Angemerkt sei, daß jeder Sprung um weitere 256 Terminals den Kernel um zusätzliche 8 KB vergrößert.

#### I2C support (CONFIG\_I2C)

I2C (gesprochen I-square-C) ist ein langsames serielles Bus-Protokoll, das von Phillips entwickelt wurde und für viele Micro-Controler-Anwendungen benutzt wird. SMBus, das Protokoll, das auf vielen modernen Mutterbrettern benutzt wird, ist wiederum eine Untergruppe von I2C. Ausführliche Information enthält die Datei Documentation/i2c/summary.

Sowohl I2C als auch SMBus werden von diesem Treiber unterstützt. Er wird benötigt, um spezielle Hardware-Sensoren zu unterstützen, aber auch, um „Video for Linux“ aktivieren zu können. Wer I2C benötigt, antwortet hier mit y und aktiviert den entsprechenden Treiber für das jeweilige Gerät im Folgenden.

#### I2C bit-banging interfaces (CONFIG\_I2C\_ALGOBIT)

Eine Untergruppe von I2C-Adaptoren stellen die sogenannten Bit-Banging-Adapter dar. Wer einen Adapter dieses Typs aktivieren möchte, antwortet hier mit y und aktiviert im Folgenden den Treiber für das entsprechende Gerät.

#### ELV adapter (CONFIG\_I2C\_ELV)

Dieser Treiber unterstützt den I2C-Adapter für den parallelen Port, der ELV heißt.

#### Velleman K9000 adapter (CONFIG\_I2C\_VELLEMAN)

Dieser Treiber unterstützt den Velleman-K9000-Parallelport-I2C-Adapter.

#### I2C PCF 8584 interfaces (CONFIG\_I2C\_ALGOPCF)

Ermöglicht den Betrieb der sogenannten PCF-Adapter. Die verschiedenen Typen werden stehen in Folge.

#### Elektor ISA card (CONFIG\_I2C\_ELEKTOR)

Dieses ist der Treiber für den PCF8584 am ISA-Bus.

#### I2C device interface (CONFIG\_I2C\_CHARDEV)

Ermöglicht den Zugriff aus Anwenderprogrammen auf die I2C-Gerätedatei, die normalerweise die Bezeichnung `/dev/i2c-*` besitzen. Weitere Information enthält Documentation/i2c/dev-interface.

### Mice

Unter diesem Submenü sind jetzt alle nicht-seriellen Mäuse zu einer Gruppe zusammengefasst.

#### Bus Mouse Support (CONFIG\_BUSMOUSE)

Wenn diese Option ausgewählt wird, werden in Folge die nicht-seriellen Maus-Typen abgefragt, z. B. Bus- oder PS/2-Mäuse. Bei einem Laptop mit eingebautem Trackball sollte die Dokumentation des Rechners zu Rate gezogen werden, um was für einen Maus-Typ es sich handelt, da dieser selten offensichtlich ist.

#### ATIXL busmouse support (CONFIG\_ATIXL\_BUSMOUSE)

Dieser seltene Maus-Typ wird an die ATI-Video-Karte angeschlossen.

#### Logitech busmouse support (CONFIG\_LOGIBUSMOUSE)

Logitech-Busmäuse werden an eine proprietäre Karte mit einem runden, neunpoligen Stecker angeschlossen, im Gegensatz zu den Standard-Mäusen, die Logitech auch herstellt, die aber mit dem seriellen Protokoll arbeiten und also auch an der seriellen Schnittstelle angeschlossen werden.

#### Microsoft busmouse support (CONFIG\_MS\_BUSMOUSE)

Treiber für die Microsoft-Busmäuse, die einem runden, neunpoligen Stecker besitzen.

PS/2 mouse (aka "auxiliary device") support (CONFIG\_PSMOUSE)

Die PS/2-Mäuse werden mit einem sechspoligen Stecker an den Mausport angeschlossen. PS/2-Mäuse werden z. B. mit den Rechnern von Compaq oder IBM ausgeliefert. Auch sind die Trackbälle von einigen Laptops PS/2-Mäuse.

C&T 82C710 mouse port support (as on TI Travelmate)  
(CONFIG\_82C710\_MOUSE)

Um den unzähligen „Standards“ noch einen hinzuzufügen, mußte TI für seinen Laptop Travelmate unnötigerweise viele Mannstunden in die Entwicklung seiner eigenen Busmaus-Schnittstelle investieren. Dieses ist der zugehörige Treiber.

PC110 digitizer pad support (CONFIG\_PC110\_PAD)

Dieses ist der Treiber für das Touchpad an dem Palmtop IBM PC110. Details enthält <http://toy.cabi.net>. Dieser Treiber kann auch als Modul kompiliert werden.

## Joysticks

Joystick support (CONFIG\_JOYSTICK)

Dieses ist der grundlegende Treiber, um die Möglichkeiten eines Freudenstocks zu aktivieren. Die jeweiligen gerätespezifischen Treiber für den entsprechenden Joystick werden im Folgenden aufgeführt. Der Joystick ist danach über die Gerätedatei /dev/jsx ansprechbar. Weitere Informationen enthält die Datei /usr/src/linux/Documentation/joystick.txt, die

auch Internet-Adressen der Hersteller und Informationen über kompatible Modelle enthält. Der Treiber kann auch als Modul kompiliert werden.

`ns558 gameports (CONFIG_INPUT_NS558)`

Der Treiber für den Standard-Gameport bei ISA-, ISAPnP- oder PCI-Rechnern.

`PDPI Lightning 4 gamecards (CONFIG_JOY_LIGHTNING)`

Dieses ist der Treiber für die PDPI Lightning 4 Daddelkarten, die mit einem analogen Joystick verbunden werden muß.

`Aureal Vortex and Trident 4DWave gameports (CONFIG_INPUT_PCIGAME)`

Der Joysticktreiber für die Karten Trident 4DWave DX/NX und Aureal Vortex 1/2.

`Classic PC analog joysticks and gamepads (CONFIG_JOY_ANALOG)`

Dieses ist der Treiber für den klassischen analogen Joystick, der an dem Gameport des PC angeschlossen wird. Viele neuere Joysticks mit verschiedensten zusätzlichen Schaltern, Knöpfen und Reglern benutzen immer noch dieses Protokoll. Wir verweisen auf `/usr/src/linux/Documentation/joystick.txt`.

`Assasin 3D and MadCatz Panther devices (CONFIG_JOY_ASSASIN)`

Dieses ist der Treiber für die Joysticks FPGaming Assasin 3D, MadCatz Panther oder MadCatz Panther XL. Er kann auch als Modul kompiliert werden.

`Logitech ADI digital joysticks and gamepads (CONFIG_INPUT_ADI)`

Für den Logitech-Controller, der das ADI-Protokoll am PC-Gamepot benutzt.

Creative Labs Blaster Cobra gamepad (CONFIG\_INPUT\_COBRA)

Für die Benutzer des Gamepads Blaster Cobra von Creative Labs.

Genius Flight2000 Digital joysticks and gamepads (CONFIG\_INPUT\_GF2K)

Der Treiber für Genius Flight2000 und MaxFighter.

Gravis GrIP joysticks and gamepads (CONFIG\_JOY\_GRAVIS)

Dieses ist der Treiber für die Joysticks Gravis GamePad Pro, Gravis Xterminator oder Gravis Blackhawk Digital. Er kann auch als Modul geladen werden.

InterAct digital joysticks and gamepads (CONFIG\_INPUT\_INTERACT)

Der Treiber für die Gerät von InterAct.

ThrustMaster DirectConnect joysticks and gamepads  
(CONFIG\_JOY\_THRUSTMASTER)

Dieses ist der Treiber für die Joysticks ThrustMaster Millenium 3D Inceptor und ThrustMaster 3D Rage Pad. Er kann auch als Modul kompiliert werden.

Microsoft SideWinder digital joysticks and gamepads  
(CONFIG\_JOY\_SIDEWINDER)

Dieses ist der Treiber für sinnvolle Microsoft-Produkte. Folgende Joysticks werden unterstützt: Microsoft SideWinder 3d Pro, Microsoft SideWinder Precision Pro, Microsoft SideWinder Force Feedback Pro, Microsoft Sidewinder GamePad und Genius Flight2000 F-23 Digital.

Serial port input line discipline (CONFIG\_INPUT\_SERPORT)

Der grundlegende Treiber für alle Spielgeräte, die an der seriellen Schnittstelle angeschlossen werden.

Logitech WingMan Warrior joystick (CONFIG\_INPUT\_WARRIOR)

Der Treiber für den Logitech WingMan Warrior an der seriellen Schnittstelle.

LogiCad3d Magellan/SpaceMouse 6dof controller  
(CONFIG\_INPUT\_MAGELLAN)

Der Treiber für die Geräte Magellan und Space Mouse 6DOF am seriellen Port.

SpaceTec SpaceOrb/Avenger 6dof controller (CONFIG\_INPUT\_SPACEORB)

Der Treiber für SpaceOrb 360 und SpaceBall Avenger 6DOF an der seriellen Schnittstelle.

SpaceTec SpaceBall 4000 FLX 6dof controller  
(CONFIG\_INPUT\_SPACEBALL)

Der Treiber für den Controller SpaceBall 4000 FLX von SpaceTec an der seriellen Schnittstelle.

I-Force/Serial controllers (CONFIG\_INPUT\_IFORCE\_232)

Für den Joystick und das Lenkrad von I-Force an der seriellen Schnittstelle.

I-Force/USB controllers (CONFIG\_INPUT\_IFORCE\_USB)

Für den Joystick und das Lenkrad von I-Force am USB-Port.

**QIC-02 tape support (CONFIG\_QIC02\_TAPE)**

Treiber für QIC-02-Bandlaufwerke, die keine SCSI-Schnittstelle besitzen.

**Do you want runtime configuration for QIC-02 (CONFIG\_QIC02\_DYNCONF)**

Die Parameter des QIC-02-Bandlaufwerks können endgültig in der Header-Datei `/include/LINUX/tpqic02.h` vor dem Kompilieren eingetragen werden. Wer aber die Parameter des Treibers jeweils zur Laufzeit einstellen möchte, muß hier ja wählen und sich das dazu notwendige Programm `qic02conf` von `ftp://titius.cfw.com/pub/Linux/util/` herunterzuladen.

## Watchdog Cards

**Watchdog Timer Support (CONFIG\_WATCHDOG)**

Wenn diese Option aktiviert wird, muß außerdem eine zeichenorientierte Gerätedatei namens `/dev/watchdog` mittels `mknod` mit der Hauptgerätenummer 10 und der Hilfsgerätenummer 130 erzeugt werden. Danach kontrolliert ein „Wachhund“ (engl. `watchdog`), ob der Rechner noch unter den Lebenden weilt: wenn seit mehr als einer Minute nichts automatisch in diese Datei geschrieben wurde, geht der Watchdog davon aus, daß der Rechner hängt und veranlaßt ein Runter-/Hochfahren des Systems. Dieses ist z.B. wichtig für Netzwerk-Server, auf die die Benutzer nicht warten sollten. Weitere Informationen sind in der Datei `Documentation/watchdog.txt` enthalten.

**Disable watchdog shutdown on close (CONFIG\_WATCHDOG\_NOWAYOUT)**

Normalerweise kann der Watchdog-Prozeß von außerhalb angehalten werden; wenn diese Option aktiviert ist, kann er nur noch durch Ziehen des Netzteckers, herausdrehen der Sicherung oder Sabotage des örtlichen Stromanbieters deaktiviert werden.

**Software Watchdog (CONFIG\_SOFT\_WATCHDOG)**

Dieser Watchdog ist eine reine Software-Lösung. Zwar wird diese Lösung nicht alle Problemsituationen erkennen, die ein Hardware-Watchdog registriert, aber sie ist deutlich billiger und einfacher zu installieren.

**WDT Watchdog timer (CONFIG\_WDT)**

Dieses ist der Treiber für die Watchdog-Hardware WDT500P und WDT501P. Während des Bootens kann bei diesem Treiber kein Autoprobing durchgeführt werden, so daß der IO-Port und der IRQ mittels des Boot-Parameters `wdt=` mitgeteilt werden muß.

**WDT PCI Watchdog timer (CONFIG\_WDTPCI)**

Dieses ist der Treiber für die PCI-Watchdog-Karte WDT500/501. Während des Bootens kann bei diesem Treiber kein Autoprobing durchgeführt werden, so daß der IO-Port und der IRQ mittels des Boot-Parameters `wdt=` mitgeteilt werden muß.

**WDT501 features (CONFIG\_WDT\_501)**

Falls man die Watchdog-Hardware WDT 501P installiert hat, kann man mit dieser Option ein Thermometer realisieren. Dazu muß außerdem eine zeichenorientierte Gerätedatei namens `/dev/temperature` mittels `mknod` mit der Hauptgerätenummer 10 und der Hilfsgerätenummer 131 erzeugt werden. Das einsame Byte in dieser Datei gibt die Temperatur in Fahrenheit (!!!) an.

**Fan Tachometer (CONFIG\_WDT\_501\_FAN)**

Aktiviert den Drehzahlmesser des Lüfter an dem WDT501.

**Berkshire Products PC Watchdog (CONFIG\_PCWATCHDOG)**

Dieses ist der Treiber für die Watchdog-PC-Karte von Berkshire Products. Die Karte überwacht, ob der Kernel noch arbeitet. Ist dieses nicht der Fall, wird der Rechner nach einer bestimmten Zeit wieder hochgefahren. Weitere Informationen bietet <http://www.berkprod.com>.

Acquire SBC Watchdog Timer (CONFIG\_ACQUIRE\_WDT)

Dieses ist der Treiber für den Watchdog von Acquire Inc. Der Treiber besitzt die gleiche Funktionalität wie der vorangegange für den WDT 501, aber eben für andere Hardware.

SBC-60XX Watchdog Timer (CONFIG\_60XX\_WDT)

Der Treiber für den Watchdog SBC-60XX.

Mixcom Watchdog (CONFIG\_MIXCOMWD)

Dieses ist der Treiber für den Watchdog von Mixcom. Der Treiber besitzt die gleiche Funktionalität wie der vorangegange für den WDT 501, aber eben für andere Hardware.

Intel i810 TCO timer / Watchdog (CONFIG\_I810\_TCO)

Der Hardware-Treiber für den TCO-Timer, der in die Intel-Chips i810 und i815 integriert ist. Der Ablauf des Timers kann konfiguriert werden durch das Befehlszeilenargument `i810_margin=<n>`, wobei `<n>` den Startwert darstellt. Dieser wird alle 0.6 Sekunden erhöht. Die Standardeinstellung ist 50, so daß es einen Timeout von einer Minute und 23 Sekunden bis zum Reset gibt.

Intel i8x0 Random Number Generator support (CONFIG\_INTEL\_RNG)

Dieses ist die Kernel-seitige Unterstützung für den Hardware-basierten Zufallszahlen-Generator, der auf den Intel-Mutterbrettern voranden ist, die auf den i8x0-Chips basieren.

**/dev/nvram support (CONFIG\_NVRAM)**

In den Tiefen des Rechners gibt es einen Speicherbereich auf der Echtzeituhr, der 50 Byte umfaßt und bei den PCs normalerweise CMOS RAM genannt wird, während Atari diesen Bereich NVRAM taufte. Um hierauf zuzugreifen, muß dieser Treiber aktiviert werden. Außerdem muß eine Gerätedatei mittels `mknod` angelegt werden, die die Hauptgerätenummer 10 und die Nebennummer 144 trägt. Auf Atari-Rechnern wird `/dev/nvram` automatisch aktiviert; hier muß dieser Treiber nicht aktiviert werden. Dieser Treiber kann auch als Modul geladen werden.

**Enhanced Real Time Clock Support (CONFIG\_RTC)**

Falls diese Option aktiviert wird, muß außerdem eine zeichenorientierte Gerätedatei namens `/dev/rtc` mittels `mknod` mit der Hauptgerätenummer 10 und der Hilfsgerätenummer 135 erzeugt werden. Die so erzeugte Echtzeituhr kann u. a. eine Weckfunktion erhalten. Die Statusinformationen sind in der Datei `/proc/rtc` abrufbar, während diverse Parameter der Uhr in der Datei `/dev/rtc` eingestellt werden können. Weitere Informationen enthält Documentation/rtc.txt.

**Double Talk PC internal speech card support (CONFIG\_DTLK)**

Dieses ist der Treiber für DoubleTalk PC, einen Sprach-Synthesizer, der von RC Systems (<http://www.rcsys.com/>) hergestellt wird. Der Treiber kann auch als Modul kompiliert werden.

**Siemens R3964 line discipline (CONFIG\_R3964)**

Dieser Treiber ermöglicht synchrone Kommunikation mit Geräten über das Paket-Protokoll R 3964 von Siemens.

**Applicom intelligent fieldbus card support (CONFIG\_APPLICOM)**

Der Treiber für die intelligenten Feldbus-Karten von Applicom International.  
Weitere Information unter <http://www.applicom-int.com/>.

## Ftape, the floppy tape device driver

### Ftape (QIC-80/Travan) support (CONFIG\_FTAPE)

Wer ein Bandlaufwerk an seinem Controller für das Diskettenlaufwerk angeschlossen hat, der findet hier den richtigen Treiber. Ein Bandlaufwerke werden mit eigenem Controller geliefert, wie z.B. Seagate Tape Store 3200, Iomega Ditto 3200" und Exabyte Eagle TR-3. Auch diese Laufwerke und die zugehörigen Controller werden unterstützt. Einige Controller benötigen außerdem spezielle Einstellungen in einem der folgenden Menüs. Hierzu gehören die Modelle CMS FC-10, FC-20, Mountain Mach-II, sowie alle Controller, die mit dem Chip Intel 82078 FDC arbeiten, wie z.B. die schnellen Controller von Seagate und Exabyte sowie Iomega Ditto Dash. Wer ein Bandlaufwerk auf PCI-Basis besitzt, sollte folgenden Text lesen:  
`/usr/src/linux/drivers/char/ftape/README.PCI`. Der Treiber kann auch als Modul geladen werden. Da das Ftape-HOWTO derzeit überholt ist, wird folgende Internet-Seite empfohlen: <http://www-math.math.rwth-aachen.de/~LBFM/claus/ftape/>. Für Benutzer, die schon länger mit Bandlaufwerken arbeiten, sei darauf hingewiesen, daß sich die System-Schnittstelle im Verhältnis zu den Vorgänger-Versionen geändert hat. Genaues bietet `/usr/src/linux/Documentation/ftape.txt`.

### Zftape, the VFS interface (CONFIG\_ZFTAPE)

Dieser Treiber ist eine Neuerung. Bei vorherigern Versionen war die Schnittstelle zum virtuellen File-System (VFS) des Kernel direkt im Treiber CONFIG\_FTAPE untergebracht. Diese Schnittstelle ist jetzt in diesem Treiber untergebracht. Deshalb muß dieses Modul aktiviert werden; im anderen Falle kann man keine Bänder lesen oder schreiben. Bänder, die mit dem alten Treiber geschrieben wurden, kann man sehr wohl mit diesem neuen Treiber lesen,

während mit diesem Treiber geschriebene Bänder für den alten Treiber unleserlich sind.

Default block size (CONFIG\_ZFT\_DFLT\_BLK\_SZ)

Hiermit wird die Standard-Block-Größe von Ftape eingestellt, der voreingestellte Wert ist 10240. Dieser kann allerdings zur Laufzeitgeändert werden mittels des Befehls

```
mt -f /dev/qft0 setblk #Blockgröße
```

Number of ftape buffers (EXPERIMENTAL) (CONFIG\_FT\_NR\_BUFFERS)

Dieser Wert stellt die Anzahl der DMA-Puffer ein und sollte in seiner Voreinstellung belassen werden (3).

Enable procfs status report (+2kb) (CONFIG\_FT\_PROC\_FS)

Dieser Treiber erzeugt die Datei /proc/ftape. Dieser Datei können diverse Status-Informationen entnommen werden, z.B. mittels

```
cat /proc/ftape/history
```

Achtung: wenn CONFIG\_FTAPE als Modul kompiliert wurde und dieses Modul gerade nicht geladen ist, so bringt die Abfrage den Kernel zum Stillstand. Dieses kann leider nicht beseitigt werden.

(Normal) Debugging output (CONFIG\_FT\_NORMAL\_DEBUG)

Diese Level-Abfrage kontrolliert die Menge an Debug-Informationen, die der Treiber produzieren kann. Die Standardeinstellung normal sollte so belassen bleiben.

(Standard) Floppy tape controllers (CONFIG\_FT\_STD\_FDC)

Die Standardeinstellung Standard unterstützt Bandlaufwerke am Floppy-Controller. Wer eine eigene Steckkarte für den Controller in sein System eingebaut hat, muß diesen Wert wie folgt ändern:

- MACH-2 für Mountain Mach-2.
- FC-10/FC-20 für Colorado FC-10 oder FC-20
- Alt/82078 für alles Intel 82078 FDC basierten Controller wie die Highspeed-Controller von Seagate, Exabyte and Iomega.

Wenn eine andere Option als Standard gewählt wurde, müssen die Parameter, insbesondere DMA-Kanal, IRQ und E/A-Adresse korrekt eingestellt sein.

Default FIFO threshold (EXPERIMENTAL) (CONFIG\_FT\_FDC\_THR)

Setzt den Schwellwert für die FIFO des Controllers des Bandlaufwerks. Die Standardeinstellung ist 8.

Maximal data rate to use (EXPERIMENTAL) (CONFIG\_FT\_FDC\_MAX\_RATE)

Mit einigen langsameren Motherboards ist es möglich, daß das Bandlaufwerk nicht seine maximale Geschwindigkeit erzielen kann. Dieses erkennt man an der Fehlermeldung `reduced data rate because of excessive overrun errors`. Damit das Laufwerk nicht immer wieder anlaufen muß, sondern wirklich „streamt“ kann hier dem Laufwerk mitgeteilt werden, daß es nicht mit höchster Geschwindigkeit arbeiten soll. Der Standarwert beträgt 2000 Bits/sec.

/dev/agpgart (AGP Support) (CONFIG\_AG)

AGP (Accelerated Graphics Port) ist ein Bus-System, um Grafikkarten mit dem Rest des Systems zu verbinden. Dieses ist der grundlegende Treiber für die

AGP-Videokarten. Der Treiber für die jeweilige Karte muß zusätzlich aktiviert werden.

Intel 440LX/BX/GX and I815/I840 support (CONFIG\_AGP\_INTEL)

Dieses ist der Treiber für die GLX-Komponente von XFree86 4.x für die Intel-Chips 440LX/BX/GX, 815 und 840. Der Treiber ist noch in der Entwicklung. Die neuesten Versionen können unter <http://utah-glx.sourceforge.net/> heruntergeladen werden.

Intel I810/I815 (on-board) support (CONFIG\_AGP\_I810)

Dieses ist der Treiber für die Intel-Onboard-Chips 810 und 815.

VIA chipset support (CONFIG\_AGP\_VIA)

Dieses ist der Treiber für die GLX-Komponente von XFree86 4.x für die Chips VIA MPV3/Apollo Pro. Der Treiber ist noch in der Entwicklung. Die neuesten Versionen können unter <http://utah-glx.sourceforge.net/> heruntergeladen werden.

AMD Irlongate support (CONFIG\_AGP\_AMD)

Dieses ist der Treiber für die GLX-Komponente von XFree86 4.x für den Chip AMD Irlongate. Der Treiber ist noch in der Entwicklung. Die neuesten Versionen können unter <http://utah-glx.sourceforge.net/> heruntergeladen werden.

Generic SiS support (CONFIG\_AGP\_SIS)

Dieses ist der Treiber für die GLX-Komponente von XFree86 4.x für die Chips von SiS (Silicon Integrated Systems). Der Treiber ist noch in der Entwicklung. Die neuesten Versionen können unter <http://utah-glx.sourceforge.net/> heruntergeladen werden.

[glx.sourceforge.net/](http://utah-glx.sourceforge.net/) heruntergeladen werden. Achtung: Die AGP-Chips SiS 5591/5592 werden von diesem Treiber nicht unterstützt.

**ALI M1541 support (CONFIG\_AGP\_ALI)**

Dieses ist der Treiber für die GLX-Komponente von XFree86 4.x für den Chipsatz ALi M1541. Dieser Chipsatz unterstützt sowohl AGP 1.x als auch 2.x. Allerdings gibt es bekannte Inkompatibilitäten mit der Matrox-Karte G 200 aufgrund von Timing-Problemen mit AGP 2.x. Dieses ist ein Hardware-Problem, das von dem Treiber nicht beseitigt werden kann. AGP 1.x funktioniert mit dieser Karte. Der Treiber ist noch in der Entwicklung. Die neuesten Versionen können unter <http://utah-glx.sourceforge.net/> heruntergeladen werden.

**Direct Rendering Manager (XFree86 DRI support) (CONFIG\_DRM)**

Der Kernel-Treiber für die Unterstützung von Direct Rendering Infrastructure (DRI), die ab XFree86 4.0 eingeführt wurde. Außerdem muß zusätzlich das entsprechende Modul für die jeweilige Graphikkarte aktiviert werden (s.u.). Weitere Information enthält <http://dri.sourceforge.net>.

**3dfx Banshee/Voodoo3+ (CONFIG\_DRM\_TDFX)**

Der Treiber für die Karten 3dfx Banshee und Voodoo3 oder höher.

**3dlabs GMX 2000 (CONFIG\_DRM\_GAMMA)**

Der Treiber für die Karte 3dlabs GMX 2000.

**ATI Rage 128 (CONFIG\_DRM\_R128)**

Der Treiber für die Karte ATI Rage 128.

Intel I810 (CONFIG\_DRM\_I810)

Der Treiber für die Karte Intel I810.

Matrox g200/g400 (CONFIG\_DRM\_MGA)

Der Treiber für die Karte Matrox g 200 or g 400.

PCMCIA character device support

PCMCIA serial device support (CONFIG\_PCMCIA\_SERIAL\_CS)

Dieses ist der Treiber für die älteren 16-Bit-PCMCIA-Karten für verschiedenen serielle Geräte, wie serielle Schnittstellen, Modems sowie für die Modem-Funktionen der Multifunktions-Ethernet-Modem-Karten.

CardBus serial device support (CONFIG\_PCMCIA\_SERIAL\_CB)

Dieses ist der Treiber für die neueren CardBus-Karten für verschiedenen serielle Geräte, wie serielle Schnittstellen, Modems sowie für die Modem-Funktionen der Multifunktions-Ethernet-Modem-Karten.

## 23. Multimedia devices

Video For Linux (CONFIG\_VIDEO\_DEV)

Dieser Treiber enthält Grundlegendes für Audio- und Video-Karten sowie für UKW-Radio-Karten. Die Treiber für die einzelnen Geräte stehen in den nächsten Optionen. Details und Werkzeuge findet man unter <ftp://ftp.uklinux.org/pub/linux/video4linux>. Wer selber Treiber oder Anwenderprogramme für dieses relativ neue Gebiet schreiben möchte, dem sei folgender Text ans Herz gelegt:

/usr/src/linux/Documentation/video4linux/API.html. Der Treiber kann auch als Modul kompiliert werden.

#### V4L information in proc filesystem (CONFIG\_VIDEO\_PROC\_FS)

Dieser Treiber ermöglicht den Zugriff auf Geräte-Informationen über das virtuelle Dateisystem /proc/video. Dazu muß außerdem der Treiber CONFIG\_PROC\_FS (s.o.) aktiviert sein.

#### BT848 Video For Linux (CONFIG\_VIDEO\_BT848)

Dieses ist der Treiber für Frame-Grabber/Overlay-Karten, die auf dem Chip BT848 basieren, wie z.B. die Karten von Miro, Hauppauge und STB. Weitere Informationen enthält /usr/src/linux/Documentation/video4linux/bttv. Dieser Treiber kann auch als Modul geladen werden.

#### Mediavision Pro Movie Studio Video For Linux (CONFIG\_VIDEO\_PMS)

Der Treiber für Mediavision Pro Movie Studio. Er kann auch als Modul geladen werden.

#### CPiA Video For Linux (CONFIG\_VIDEO\_CPIA)

Dieses ist der Treiber für Video-Kameras, die auf CPiA (Colour Processor Interface ASIC) von Vision basieren, wie z.B. die Creative Labs Video Blaster Webcam II. Dieser Treiber arbeitet nicht mit der Creative Webcam III. Weitere Information enthält Documentation/video4linux/README.cpi.a.

#### CPiA USB Lowlevel Support (CONFIG\_VIDEO\_CPIA\_USB)

Wer Video-Kameras, die auf CPiA (Colour Processor Interface ASIC) von Vision basieren, am USB-Port betreibt, benötigt diesen Treiber.

**SAA5249 Teletext processor (CONFIG\_VIDEO\_SAA5249)**

Unterstützung für Videotext mittels des Chips SAA5249. Dieser befindet sich auf einigen WinTV-Karten. Der Treiber kann auch als Modul kompiliert werden.

**SAB3036 tuner (CONFIG\_TUNER\_3036)**

Der Treiber für den Tuner Philips SAB3036 und kompatibel.

**Stradis 4:2:2 MPEG-2 video driver (EXPERIMENTAL) (CONFIG\_VIDEO\_STRADIS)**

Der experimentelle Treiber für die MPEG-2-Karte von Stradis.

**Zoran ZR36057/36060 Video For Linux (CONFIG\_VIDEO\_ZORAN)**

Der Treiber für alle Video-Karten, die auf den Chips ZR36057/36060 basieren.

**Include support for Iomega Buz (CONFIG\_VIDEO\_BUZ)**

Unterstützung für die Videokarte Iomega Buz.

**Zoran ZR36120/36125 Video For Linux (CONFIG\_VIDEO\_ZR36120)**

Der Treiber für alle Framegrabber- und Overlay-Boards, die auf den Chips ZR36120/ZR36125 basieren. Dieses sind z.B. Victor II, WaveWatcher, Video Wonder, Maxi-TV und Buster. Weitere Information erhält Documentati-on/video4linux/zr36120.txt.

## Radio Adapters

**ADS Cadet AM/FM Radio Tuner Card (CONFIG\_RADIO\_CADET)**

Der Treiber für die Mehrbereichs-Radio-Karte ADS Cadet. Dokumentation über die Karte bietet <http://linux.blackhawke.net/cadet.html>. Programme, um seinem LINUX-Rechner den Radiosound zu entlocken findet man unter <http://roadrunner.swansea.uk.linux.org/v41.shtml>. Dieser Treiber kann auch als Modul kompiliert werden.

#### AIMSlab RadioTrack (aka RadioReveal) support (CONFIG\_RADIO\_RTRACK)

Dieses ist der Treiber für die Besitzer der UKW-Radio-Karten AIMSlab RadioTrack und RadioReveal. Die neueren Radio-Karten von AIMSlab besitzen einen anderen Chip, der nicht von diesem Treiber unterstützt wird. Für diese neueren Karten ist der nächste Treiber (s.u.) zuständig. Programme, um seinem LINUX-Rechner den Radiosound zu entlocken findet man unter <http://roadrunner.swansea.uk.linux.org/v41.shtml>. Weitere Information findet man unter </usr/src/linux/Documentation/video4linux/radiotrack.txt>. Der Treiber kann auch als Modul kompiliert werden.

#### AIMSlab RadioTrack II support (CONFIG\_RADIO\_RTRACK2)

Dieses ist der Treiber für das neuere Modell AIMSlab RadioTrack II. Programme, um seinem LINUX-Rechner den Radiosound zu entlocken findet man unter <http://roadrunner.swansea.uk.linux.org/v41.shtml>. Weitere Information findet man unter </usr/src/linux/Documentation/video4linux/radiotrack.txt>. Der Treiber kann auch als Modul kompiliert werden.

#### Aztech/Packard Bell Radio (CONFIG\_RADIO\_AZTECH)

Der Treiber für die Rechner-Radios von Aztech und Packard Bell. Programme, um seinem LINUX-Rechner den Radiosound zu entlocken findet man unter <http://roadrunner.swansea.uk.linux.org/v41.shtml>.

#### GemTek Radio Card (CONFIG\_RADIO\_GEMTEK)

Dieses ist der Treiber für die Rechner-Radiokarte von GemTek. Programme, um seinem LINUX-Rechner den Radiosound zu entlocken findet man unter <http://roadrunner.swansea.uk.linux.org/v41.shtml>. Dieser Treiber kann auch als Modul kompiliert werden.

#### Miro PCM20 Radio (CONFIG\_RADIO\_MIROPCM20)

Der Treiber für die UKW-Radiokarte PCM20 von Miro. Zusätzlich wird der Treiber ACI mixer (miroPCM12/PCM20) benötigt. Programme, um seinem LINUX-Rechner den Radiosound zu entlocken findet man unter <http://roadrunner.swansea.uk.linux.org/v41.shtml>. Dieser Treiber kann auch als Modul kompiliert werden.

#### SF16FMI Radio (CONFIG\_RADIO\_SF16FMI)

Der Treiber für die Rechner-Radiokarte SF16FMI. Programme, um seinem LINUX-Rechner den Radiosound zu entlocken findet man unter <http://roadrunner.swansea.uk.linux.org/v41.shtml>. Dieser Treiber kann auch als Modul kompiliert werden.

#### TerraTec ActiveRadio ISA Standalone (CONFIG\_RADIO\_TERRATEC)

Der Treiber für die ISA-Karte ActiveRadio von TerraTec. Der Treiber befindet sich noch in der Entwicklung.

#### Trust FM Radio Card (CONFIG\_RADIO\_TRUST)

Dieses ist der Treiber für die UKW-Radio-Rechner-Karte von Trust. Programme, um seinem LINUX-Rechner den Radiosound zu entlocken findet man unter <http://roadrunner.swansea.uk.linux.org/v41.shtml>. Dieser Treiber kann auch als Modul kompiliert werden.

#### Typhoon Radio (a.k.a. EcoRadio) (CONFIG\_RADIO\_TYphoon)

Der Treiber für die Rechner-Radiokarte Typhoon. Programme, um seinem LINUX-Rechner den Radiosound zu entlocken findet man unter <http://roadrunner.swansea.uklinux.org/v41.shtml>. Dieser Treiber kann auch als Modul kompiliert werden.

#### Support for /proc/radio-typhoon (CONFIG\_RADIO\_TYPOON\_PROC\_FS)

Dieser Treiber bietet Unterstützung für das virtuelle Dateisystem /proc/radio-typhoon. Hier können Status-Informationen über Frequenz, Lautstärke etc. abgelesen werden.

#### Zoltrix Radio (CONFIG\_RADIO\_ZOLTRIX)

Der Treiber für die Rechner-Radiokarte von Zoltrix. Programme, um seinem LINUX-Rechner den Radiosound zu entlocken findet man unter <http://roadrunner.swansea.uklinux.org/v41.shtml>. Dieser Treiber kann auch als Modul kompiliert werden.

## 24. File systems

#### Quota support (CONFIG\_QUOTA)

Dieser Treiber ermöglicht die Festlegung des maximalen Festplattenspeichers pro Benutzer, auch Diskquota genannt. Derzeit wird dies nur vom Dateisystem ext2 unterstützt; außerdem wird spezielle Software benötigt, die per Anonymous FTP unter <ftp.funet.fi/pub/LINUX/PEOPLE/Linus/subsystems/quota/> erhältlich ist. Diese Option ist nur in einer größeren Multiuser-Umgebung nützlich.

#### Kernel automounter support (CONFIG\_AUTOFS\_FS)

Der Automounter ist ein Werkzeug, das automatisch Server-Dateisysteme für den Benutzer mountet. Wer nicht in einem komplexen, verteilten Netz arbeitet, der wird dieses nicht benötigen. Um mit dem Automounter zu arbeiten sind Dienstprogramme nötig, die man unter folgender Adresse findet: <ftp://ftp.kernel.org/pub/linux/daemons/autofs>. Außerdem muß die Option NFS filesystem support (s.u.) aktiviert sein.

Kernel automounter version 4 support (also supports v3)  
(CONFIG\_AUTOFS4\_FS)

Der Treiber für den die Automounter-Version 4. Wer hiermit arbeiten möchte, benötigt spezielle Werkzeuge, die man unter <ftp://ftp.kernel.org/pub/linux/daemons/autofs/testing-v4> findet.

ADFS file system support (CONFIG\_ADFS\_FS)

Das Advanced Disk File System ist das von Acorn für seine Rechner benutzte Dateisystem, sowohl für Disketten als auch für Festplatten. Da sich noch im Entwicklungsstadium befindet, sind derzeit nur Lesezugriffe erlaubt. Die Festplatte mit dem Acorn-Dateisystem sollte die erste Festplatte im Rechner sein.

ADFS write support (DANGEROUS) (CONFIG\_ADFS\_FS\_RW)

Der experimentelle Treiber, der auch Schreibzugriff auf Festplatten und Disketten mit ADFS erlaubt. Achtung: da der Treiber noch in der Entwicklung ist, kann es zu Datenverlusten kommen.

Amiga FFS filesystem support (CONFIG\_AFFS\_FS)

Seit der Version 1.3 des AmigaOS ist das Fast File System (FFS) Standard. Dieser Treiber unterstützt das Lesen und Schreiben von Amiga-Festplatten. Aufgrund der unterschiedlichen Hardware ist das Lesen von Amiga-Disketten leider nicht möglich.

**Apple Macintosh filesystem support (experimental) (CONFIG\_HFS\_FS)**

Hiermit können Disketten und Festplatten mit dem Dateisystem des Apple Macintosh gelesen und geschrieben werden. Der Treiber kann auch als Modul geladen werden.

**BFS file system support (EXPERIMENTAL) (CONFIG\_BFS\_FS)**

Das Boot File System (BFS) ist ein Dateisystem, das unter SCO UnixWare benutzt wird um dem Bootlader Zugriff auf das Kernel-Image und andere wichtige Dateien zu ermöglichen. Normalerweise wird es auf /stand gemountet. Wer diese Option benötigt, muß auch den Treiber UnixWare slices support aktivieren. Weitere Information enthält Documentation/filesystems/bfs.txt.

**DOS FAT fs support (CONFIG\_FAT\_FS)**

Falls ein Dateisystem installiert werden soll, das auf einer FAT (File Allocation Table) basiert, wie MS-DOS, Windows'95 oder UMSDOS, so muß hier mit y geantwortet werden. Dieser Treiber ist nicht das Dateisystem selbst, sondern legt die Grundlagen für die verschiedenen Systeme mit FAT. Der Kernel wird hierdurch um 24 kB vergrößert.

**MSDOS fs support (CONFIG\_MSDOS\_FS)**

Dieser Treiber ermöglicht das Mounten von DOS-Festplatten und -Diskettenlaufwerken unter LINUX. Die DOS-Dateien und Verzeichnisse können somit bearbeitet werden, als ob es LINUX-Dateien wären. Falls komprimierte DOS-Laufwerke benutzt werden (Stacker o.ä.), ist dieses leider kein gangbarer Weg. In diesem Fall ist der Zugriff über spezielle Optionen des DOS-Emulators möglich. Wer LINUX mit dem UMSDOS-Dateisystem benutzt, muß hier mit y antworten, da UMSDOS auf das DOS-Dateisystem aufsetzt. Benutzer von Windows 95 oder NT sollten besser die folgende Option vfat fs support CONFIG\_VFAT\_FS verwenden, die im Gegensatz zu dieser

auch die langen Dateinamen behandeln kann. Falls auf das DOS-Dateisystem nicht permanent zugegriffen wird, sollte man diesen Treiber als Modul kompilieren. Für Benutzer, die nur sehr selten mit der DOS-Welt in Berührung kommen, gibt es die `mtools`, mit denen z.B. Dateien von LINUX nach DOS und vice versa kopiert werden können, und die diesen Treiber nicht benötigen. Der Treiber für das MSDOS-Dateisystem vergrößert den Kernel um 7 kB.

`umsdos: Unix like fs on top of std MSDOS fs (CONFIG_UMSDOS_FS)`

Dieses ist der Treiber für das UMSDOS-Dateisystem. Das gesamte UMSDOS-Dateisystem tritt dem DOS-Benutzer als eine einzige, natürlich größere DOS-Datei entgegen. Innerhalb dieser Datei ist ein komplettes LINUX-Dateisystem mit allen gewohnten Vorzügen, wie z.B. Benutzerrechten oder Links, das von LINUX aus genutzt werden kann. Man braucht also nicht die Festplatte zu neu partitionieren und mit LILO das Master Boot Record verändern, sondern kann mittels UMSDOS LINUX auf einer bestehenden DOS-Partition installieren. UMSDOS eignet sich sehr gut, um eine schnelle Installation für den LINUX-Neuling zu ermöglichen. Da alle UMSDOS-Dateizugriffe aber indirekt über das bekannterweise langsame DOS-Dateisystem laufen, sind hier nicht die Zugriffszeiten des Standard-LINUX-Dateisystems ext2 zu erreichen. Um das UMSDOS-Dateisystem benutzen zu können, müssen die beiden Optionen `fat fs support CONFIG_FAT_FS` und `msdos fs support CONFIG_MS DOS_FS` (s.o.) aktiviert sein. Dieser Treiber vergrößert den Kernel um 25 kB. Weitere Informationen sind unter `Documentation/filesystems/umsdos.txt` vorhanden.

`VFAT (Windows-95) fs support (CONFIG_VFAT_FS)`

Dieser Treiber ermöglicht das Mounten von MSDOS-Festplatten, die mit den langen Dateinamen von Windows 95 oder NT arbeiten. Komprimierte Festplatten werden allerdings nicht unterstützt. Das VFAT-Dateisystem kann nicht als Root-Partition benutzt werden; für diesen Zweck muß das UMSDOS-System `umsdos CONFIG_UMSDOS_FS` eingesetzt werden. Dieser Treiber

vergrößert den Kernel um 10 kB. Weitere Informationen sind unter Documentation/filesystems/vfat.txt vorhanden.

EFS file system support (read only) (EXPERIMENTAL) (CONFIG\_EFS\_FS)

EFS ist das Dateisystem, das für die SGI IRIX-Systeme benutzt wird. Weitere Information enthält <http://aeschi.ch.eu.org/efs>.

Journalling Flash File System (JFFS) support (EXPERIMENTAL) (CONFIG\_JFFS\_FS)

Das ist ein Dateisystem, daß für den Gebrauch mit Flash-Speicher-Geräten entwickelt wurde. Ursprünglich wurde es mit dem Kernel 2.0 von der Firma Axis Communications herausgegeben, um ihre Linux-Produkte benutzen zu können. Das der Treiber der GPL unterliegt, wurde er „geborgt“ unter weiterentwickelt für die neuen Speicher-Technologien. Weitere Information über das JFFS enthält <http://www.developer.axis.com/software/jffs>.

JFFS debugging verbosity (0 = quiet, 3 = noisy) (CONFIG\_JFFS\_FS\_VERBOSE)

Für Entwickler: der Grad der „Geschwätzigkeit“ für das Entwanzeln des JFFS-Treibers.

Compressed ROM file system support (CONFIG\_CRAMFS)

Das CramFs (Compressed ROM File System) ist ein einfaches, kleines, komprimierendes Dateisystem, das für Eingebettete Systeme auf ROM-Basis entwickelt wurde. CramFs kann nur gelesen werden, die Gesamtgröße ist auf 256 MB beschränkt, die Dateigröße auf 16 MB. Weitere Information enthalten Documentation/filesystems/cramfs.txt und fs/cramfs/README.

Simple RAM-based file system support (CONFIG\_RAMFS)

Ramfs ist ein Dateisystem, das seine Dateien im Arbeitsspeicher hält. Schreib- und Lesezugriff sind möglich. Im Gegensatz zur RAM-Disk, die einen festen Teil des RAM benutzt, kann das Ramfs entsprechend der enthaltenen Dateien wachsen und schrumpfen. Bevor das Ramfs benutzt werden kann, muß es wie jedes andere Dateisystem gemountet werden. Wenn z.B. das Verzeichnis `/ramfiles` als Mount-Punkt benutzt werden soll, so muß eingegeben werden

```
mount -t ramfs ramfs /ramfiles
```

Es muß unbedingt beachtet werden, daß hier alle Dateien sich nur im flüchtigen Speicher befinden und beim Herunterfahren des Systems für immer verloren sind!

#### ISO9660 cdrom filesystem support (CONFIG\_ISO9660\_FS)

ISO 9660 ist das Standard-Dateisystem für CD-ROMs. Es ist auch bekannt unter dem Namen „High Sierra Filesystem“, abgekürzt hsfs. Ohne diesen Treiber können keine CD-ROMs gelesen werden. Da kaum jemand permanent CDs liest, ist hier das Kompilieren als Modul empfohlen. Der ISO-9660-Treiber vergrößert den Kernel um 27 kB.

#### Microsoft Joliet cdrom extensions (CONFIG\_JOLIET)

Nachdem die Welt sich auf ISO 9660 als Format für CD-ROMs geeinigt hatte, meinte Microsoft wieder einmal mehr, eigene Standards errichten zu müssen und fügte ISO 9660 eigene Erweiterungen hinzu. Dieser Spuk dauerte wenige Monate, dann konnten auch andere Systeme - wie z.B. Linux - diese Schnittstellen lesen. Dieses ist der zugehörige Treiber.

#### Minix fs support (CONFIG\_MINIX\_FS)

Minix ist das von Andrew S. Tanenbaum entwickelte „Mini-Unix“, das gerne zu Lehrzwecken eingesetzt wird und sein eigenes Filesystem besitzt. Auch das erste Dateisystem von LINUX war Minix, so daß es manchmal aus histori-

schen Gründen, z.B. für alte Disketten, noch eingesetzt werden muß. Das Standard-Dateisystem von LINUX ist inzwischen ext2. Für jemanden, der frisch mit LINUX beginnt, hat das Minix-Dateisystem keine Bedeutung mehr.

#### NT NTFS support (read only) (CONFIG\_NTFS\_FS)

NTFS ist wieder mal ein Dateisystem von Microsoft, diesmal für WinDOS NT. Wer Zugriff auf seine NTFS-Partitionen haben möchte, der aktiviert diesen Treiber. Es ist allerdings nur der Lesezugriff möglich. Weitere Information enthält `/usr/src/linux/Documentation/filesystems/ntfs.txt`. Der Treiber kann auch als Modul geladen werden.

#### NTFS write support (DANGEROUS) (CONFIG\_NTFS\_RW)

Dieses ist der experimentelle Treiber um auf dem NTFS schreiben zu können. Der Schreibzugriff ist noch lange nicht komplett und außerdem derzeit noch nicht gut getestet. Alles erfolgt also auf eingegebene Gefahr! Man sollte also zuerst eine Sicherheitskopie von seinen NTFS-Laufwerken erstellen. Nach der Nutzung dieses Treibers sollte man auch vorsichtshalber das Windows-Dienstprogramm `chkdsk` benutzen. Unterstützt werden derzeit nur Windows NT 4 und frühere Versionen, also noch nicht Windows Null Null.

#### OS/2 HPFS filesystem support (CONFIG\_HPFS\_FS)

OS/2 („OS Halbe“) ist IBMs Betriebssystem für PCs und HPFS ist das dazugehörige Dateisystem. Da IBM die Spezifikation ihres Filesystems nicht offenlegt, ist dieser Treiber mittels fleißiger Arbeit mit dem Diskeditor entstanden, aber niemand weiß genau, ob alle Möglichkeiten des Dateisystems erfaßt wurden. Der Treiber ist jetzt aber auch - im Gegensatz zu älteren Versionen - mit Schreibzugriff ausgestattet. Weitere Informationen enthält die Datei `Documentation/filesystems/hpfs.txt`.

#### /proc filesystem support (CONFIG\_PROC\_FS)

Dieses virtuelle Dateisystem - auch als „process information pseudo file-system“ bezeichnet - stellt Informationen über den Systemstatus bereit, und zwar über die Interna des Kernels. „Virtuell“ bedeutet in diesem Fall, daß dieses Dateisystem keinen Platz auf der Festplatte benötigt. Zum einen ist es natürlich interessant und oft auch hilfreich, dem Kernel auf die Finger zu sehen, z.B. bei Interruptkonflikten o.ä. Zum anderen wird dieses Dateisystem von vielen Programmen benutzt, so daß man es in seinen Kernel aufnehmen sollte. Weitere Information halten die man-Pages bereit. Der Kernel wird durch diesen Treiber um 18 kB vergrößert.

#### /dev file system support (EXPERIMENTAL) (CONFIG\_DEVFS\_FS)

Dieses ist der Treiber für das virtuelle Dateisystem devfs in /dev, das die Schnittstelle für die Gerätedateien bildet. Der Vorteil besteht darin, daß der System-Administrator nicht mehr per `mknod` manuell Gerätedateien erzeugen muß. Dieser Treiber befindet sich noch im Aufbau. Wer ihn benutzen möchte, sollte tunlichst die Dateien im Verzeichnis `Documentation/filesystems/devfs/` lesen, insbesondere die Datei `README`.

#### Automatically mount at boot (CONFIG\_DEVFS\_MOUNT)

Diese Option bezieht sich auf den vorangegangenen Greifer für das devfs. Wenn hier mit `y` geantwortet wird, so wird das devfs automatisch während des Hochfahrens des Rechners aktiviert. Dieses Verhalten kann wiederum deaktiviert werden durch den Boot-Parameter `devfs=nomount`.

#### Debug devfs (CONFIG\_DEVFS\_DEBUG)

Für Entwickler. Wer diese Option aktiviert, erhält diverse Informationen über das devfs. Details enthält `Documentation/filesystems/devfs/boot-options`.

#### /dev/pts filesystem for Unix98 PTYs (CONFIG\_DEVPTS\_FS)

Falls weiter oben die Option `Unix98_PTY_support` aktiviert worden ist, so sollte man hier auch das zugehörige Dateisystem aktivieren. Dieses ist zuständig für Pseudo-Terminals nach dem Unix98-Standard.

**QNX filesystem support (EXPERIMENTAL) (CONFIG\_QNX4FS\_FS)**

Dieses ist das Dateisystem des Betriebssystems QNX 4 für Disketten und Festplatten. Dieser Treiber kann auch als Modul geladen werden.

**QNX4FS write support (DANGEROUS) (CONFIG\_QNX4FS\_RW)**

Der experimentelle Treiber für den Schreibzugriff für das QNX4-Dateisystem.

**ROM filesystem support (CONFIG\_ROMFS\_FS)**

Dieses ist ein sehr platzsparendes Dateisystem, von dem nur gelesen werden kann. Es wurde entwickelt, um Installations-Disketten und RAM-Disks zur Initialisierung des Systems klein zu halten. Details enthält der Text `/usr/src/linux/Documentation/filesystems/romfs.txt`. Der Treiber kann auch als Modul geladen werden.

**Second extended fs support (CONFIG\_EXT2\_FS)**

Dieses ist der Treiber für das derzeitige LINUX-Standard-Dateisystem namens ext2. Für fast alle Benutzer ist also y die richtige Antwort mit zwei Ausnahmen: falls nur das UMSDOS-Dateisystem auf einer DOS- Partition installiert werden soll, so ist ext2 nicht unbedingt notwendig, wobei die UMSDOS-Installationen mehr für das schnelle „Reinriechen“ in LINUX taugen, als für den täglichen Gebrauch. Die zweite, seltene Ausnahme ist ein Diskless Computer, der über das Netz bootet und also nur das NFS-Dateisystem aktiviert haben muß. Alle anderen müssen hier mit y antworten, auch wenn es den Kernel um 41 kB vergrößert.

**System V and Coherent filesystem support (CONFIG\_SYSV\_FS)**

Die kommerziellen Unix-Systeme SCO, Xenix und Coherent für PCs benutzen das Dateisystem des SystemV. Wer auf eine lokale Festplatte mit diesem Dateisystem zugreifen möchte, benötigt diesen Treiber. Weitere Informationen sind in Documentation/filesystems/sysv-fs.txt enthalten. Dieser Treiber vergrößert den Kernel um 34 kB.

SYSV file system write support (DANGEROUS) (CONFIG\_SYSV\_FS\_WRITE)

Der Treiber für den Schreibzugriff auf SYSV. Vorsicht, noch in der Entwicklung befindlich!

UDF file system support (read only) (CONFIG\_UDF\_FS)

Dieses ist das neue Dateisystem, das auf einigen CDROMs sowie DVDs benutzt wird. Wenn man DVDs lesen möchte, so muß hier mit y genantwortet werden. Dieser Treiber ermöglicht nur das Lesen des UDF-Dateisystems. Wer CDs oder DVDs brennen möchte, muß auch den folgenden Treiber aktivieren. Weitere Information enthält Documentation/filesystems/udf.txt.

UDF write support (DANGEROUS) (CONFIG\_UDF\_RW)

Zum Schreiben des UDF-Dateisystems. Noch in der Entwicklung!

## Network File Systems

Coda file system support (advanced network fs) (CONFIG\_CODA\_FS)

Coda ist ein Netzwerk-Dateisystem, genau wie NFS. Es hat aber einige Leistungsmerkmale mehr als NFS. Wenn dieser Treiber aktiviert wird, so kann man als Client an einem Server mit Coda-Dateisystem arbeiten. Weitere Information enthält der Text /usr/src/linux/Documentation/filesystems/coda.txt die Web-

Seite <http://www.coda.cs.cmu.edu>. Der Treiber kann auch als Modul geladen werden.

NFS file system support (CONFIG\_NFS\_FS)

Falls der Rechner im Netz betrieben wird und man die Dateien eines NFS-Servers in das LINUX-Filesystem einbinden will, ist dieses der richtige Treiber. Die Dateien können dann behandelt werden, als ob sie zum lokalen LINUX-System gehören. Weitere Auskunft geben die man-Pages. Dieser Treiber vergrößert den Kernel um 27 kB.

Provide NFSv3 client support (CONFIG\_NFS\_V3)

Der Treiber für die neuere Version 3 des NFS-Protokolls.

Root file system on NFS (CONFIG\_ROOT\_NFS)

Falls in einer Diskless Station das gesamte Dateisystem vom Server importiert wird, so wird dieser Treiber benötigt. In diesem Fall ist es wahrscheinlich, daß auch der Treiber `IP: kernel level autoconfiguration` benötigt wird, um die Netzwerkadresse des Client automatisch zu erfassen. Details enthält `Documentation/nfsroot.txt`.

NFS server support (CONFIG\_NFSD)

Wenn der LINUX-Rechner als NFS-Server arbeiten soll, so muß dieser Treiber aktiviert werden. Dieser Treiber kann auch als Modul geladen werden.

Provide NFSv3 server support (CONFIG\_NFSD\_V3)

Falls der Linux-Rechner als NFS-Server arbeiten soll und die Versionen NFSv3 oder NFSv2 unterstützen muß, so ist dieses der richtige Treiber.

**SMB file system support (to mount WfW shares etc.) (CONFIG\_SMB\_FS)**

SMB (Server Message Buffer) ist das Protokoll, das Windows for Workgroups (WfW), Win 9x, Win Me, Windows NT und 00 benutzen, um via Ethernet zu kommunizieren. Das Aktivieren dieser Option ermöglicht das Mounten der erwähnten Dateisysteme in das LINUX-Filesystem; sie verhalten sich dann wie LINUX-Dateien. Falls der LINUX-Rechner nicht als SMB-Client, sondern als SMB-Server arbeiten soll, so ist dieser Treiber nicht notwendig, sondern diese Aufgabe wird von dem Programm samba erledigt. Weitere Informationen sind unter Documentation/filesystems/smbfs.txt vorhanden.

**Use a default NLS (CONFIG\_SMB\_NLS\_DEFAULT)**

Dieser Treiber aktiviert die NLS-Übersetzung (s.u.) als Standardeinstellung. Dazu muß außerdem der gewünschte Zeichensatz CONFIG\_NLS\_DEFAULT aktiviert werden und außerdem muß dem SMB-Server die Standardeinstellung an CONFIG\_SMB\_NLS\_REMOTE übergeben werden. Die NLS-Einstellung können auch während des Mountens verändert werden, allerdings wird dieses derzeit noch nicht unterstützt, aber in Zukunft ... Es gibt allerdings zur Zeit einen inoffiziellen Patch für Samba 2.0.7 unter <http://www.hojdpunkten.ac.se/054/samba/index.html>.

**Default Remote NLS Option: "cp437" (CONFIG\_SMB\_NLS\_REMOTE)**

Mit dieser Einstellung wird definiert, welche Codepage der Server als Standardeinstellung benutzt. Die Voreinstellung ist cp437.

**NCP filesystem support (to mount NetWare volumes) (CONFIG\_NCP\_FS)**

NCP (NetWare Core Protocol) ist das Protokoll, das von Novell für die Kommunikation zwischen Client und Server benutzt wird. Das Aktivieren dieser Option ermöglicht das Mounten eines Novell-Servers in das LINUX-Dateisystem. Wer dieses möchte, muß auch in der Sektion 6. Netzwerk die

Option `The IPX protocol CONFIG_IPX` aktivieren. Weitere Informationen enthält `Documentation/filesystems/ncpfs.txt`.

**Packet signatures (CONFIG\_NCPFS\_PACKET\_SIGNING)**

Um größere Sicherheit zu erzielen, wird hiermit ermöglicht, den NCP-Daten-Paketen eine Signatur zu geben. Dieses ist erst möglich ab Version 2.0.12 von ncpfs.

**Proprietary file locking (CONFIG\_NCPFS\_IOCTL\_LOCKING)**

Ermöglicht das Record-Locking auf NCPFS-Volumes. Nicht benötigt, solange man nicht entsprechende Software benutzt, die dieses unterstützt.

**Clear remove/delete inhibit when needed (CONFIG\_NCPFS\_STRONG)**

Ermöglicht die Manipulation von Dateien, obwohl sie Flags gesetzt haben, die dieses verbieten. Dazu muß allerdings `ncpmount` mit dem Parameter `-s` aufgerufen werden.

**Use NFS namespace when available (CONFIG\_NCPFS\_NFS\_NS)**

Ermöglicht die Benutzung von Novells Namespace, der Erweiterung, die ertens Dateinamen zuläßt, die länger als 12 Zeichen sind und zweiten zwischen Groß- und Kleinschreibung unterscheidet.

**Use OS2/LONG namespace when available (CONFIG\_NCPFS\_OS2\_NS)**

Die Art und Weise, um auf Novells Namespace von OS/2-Rechnern aus zugreifen: die Dateinamen dürfen bis zu 255 Zeichen lang sein, aber auf Groß- und Kleinschreibung wird nicht geachtet.

**Lowercase DOS filenames (CONFIG\_NCPFS\_SMALLDOS)**

Falls der Namespace OS2/LONG (s.o.) aktiviert wurde, werden alle Dateinamen, die unter DOS erzeugt wurden, in Kleinbuchstaben umgewandelt. Wenn dieser Treiber nicht aktiviert wird, erscheinen sie in Großbuchstaben. Dieses ist eigentlich nur eine kosmetische Operation, da der OS/2-Namespace nicht zwischen Groß- und Kleinbuchstaben unterscheidet. Der Grund ist also nur, eine Abwärtskompatibilität zu erhalten, wenn man auf einem Novell-Server den Namespace von DOS auf OS/2 erweitert. Lange Dateinamen, wie sie z.B. von Win95 erzeugt werden, sind nicht betroffen. Diese Option löst übrigens nicht das Problem, daß Windows und Linux verschiedene Vorstellungen von Sonderzeichen in Dateinamen haben, da Windows auf der Clientseite seine eigenen Umwandlungen vornimmt.

Allow mounting of volume subdirectories (CONFIG\_NCPFS\_MOUNT\_SUBDIR)

Ermöglicht, daß nicht nur ganze Novell-Volumes, sondern auch deren Unterverzeichnisse an den Client exportiert werden können.

NDS authentication support (CONFIG\_NCPFS\_NDS\_DOMAINS)

Dieses ermöglicht das Speichern des privaten Schlüssels für NDS im Kernel, der benötigt wird, um andere Server zu autentifizieren. Dieses funktioniert erst ab Version 2.0.12.1 von NCPFS.

Use Native Language Support (CONFIG\_NCPFS\_NLS)

Zur Übersetzung von Zeichensätzen für Dateinamen von Novell-Servern für fremde Systeme, wie z.B. Windows '95. Die entsprechenden Codepages stehen weiter unten.

Enable symbolic links and execute flags (CONFIG\_NCPFS\_EXTRAS)

Ermöglicht die Benutzung von symbolischen Links auf NCPFS-Servern.

## Partition Types

Advanced partition selection (CONFIG\_PARTITION\_ADVANCED)

Hier muß mit y geantwortet werden, wenn Festplatten benutzt werden sollen, die mittels eines anderen Betriebssystems als Linux partitioniert worden sind. Diese Option löst keinen Kernel-Code aus, sondern aktiviert nur die nächsten Fragen.

Acorn partition support (CONFIG\_ACORN\_PARTITION)

Zur Unterstützung des Acorn-Schemas zur Partitionierung von Festplatten.

Alpha OSF partition support (CONFIG\_OSF\_PARTITION)

Zur Unterstützung des Alpha-Schemas zur Partitionierung von Festplatten.

Amiga partition table support (CONFIG\_AMIGA\_PARTITION)

Für die nimmermüden Amiga-Fans die entsprechende Unterstützung der Festplatten-Partitionierung.

Atari partition table support (CONFIG\_ATARI\_PARTITION)

Für die letzten überlebenden Atari-Fans die entsprechende Unterstützung der Festplatten-Partitionierung.

Macintosh partition map support (CONFIG\_MAC\_PARTITION)

Zur Unterstützung der Partitionstabelle von Apfel-Rechnern.

PC BIOS (MSDOS partition tables) support (CONFIG\_MSDOS\_PARTITION)

Der Treiber für die Partitionstabelle von PCs - der Normalfall.

BSD disklabel (FreeBSD partition tables) support  
(CONFIG\_BSD\_DISKLABEL)

FreeBSD benutzt ein eigenes System, um Festplatten zu partitionieren. Es belegt nur einen Eintrag in der primären Partitionstabelle der Festplatte und verfährt mit den erweiterten Partitionen wie DOS. Dieser Treiber ermöglicht das Lesen der Partitionstabelle entsprechend dem FreeBSD-Schema und also auch das Mounten dieser Festplatten.

Solaris (x86) partition table support (CONFIG\_SOLARIS\_X86\_PARTITION)

Auch Solaris benutzt seine eigenen System für die Festplatten-Partitions-Tabelle. Wer diese Festplatten lesen möchte, benötigt diesen Treiber.

UnixWare slices support (CONFIG\_UNIXWARE\_DISKLABEL)

Auch UnixWare benutzt seine eigene Partitionstabelle. Wer von derartigen System Daten lesen möchte, muß diesen Treiber aktivieren.

SGI partition support (CONFIG\_SGI\_PARTITION)

Zur Unterstützung der Partitionstabelle von Rechnern der Firma SGI.

Ultrix partition table support (CONFIG\_ULTRIX\_PARTITION)

Zur Unterstützung der Partitionstabelle von Ultrix-Rechnern.

Sun partition tables support (CONFIG\_SUN\_PARTITION)

Zur Unterstützung der Partitionstabelle von Sun-Sparc-Rechnern.

Native Language Support (CONFIG\_NLS)

Am Anfang gab es nur 7 Bit und keine nationalen Sonderzeichen, wie z.B. ä,ö,ü. Als die Amerikaner entdeckten, daß sie nicht alleine auf der Welt sind, war es zu spät: jedes Betriebssystem hatte inzwischen seine eigenen und sehr verschiedenen Vorstellungen entwickelt, wo und wie Sonderzeichen abgelegt sind. Dieses schafft immer wieder Verwirrung in Dateinamen mit solchen Sonderzeichen, wenn Linux mit anderen Betriebssystemen in Berührung kommt, z.B. wenn es als Samba-Server dient und also mit WinDOS-Clients kommunizieren muß. Um die entsprechenden Sonderzeichen korrekt zu übersetzen werden die sogenannten Codepages benutzt. Die Übersetzung findet nur bei den Dateinamen statt, nicht beim Inhalt der Datei!

Default NLS Option: "iso8859-1" (CONFIG\_NLS\_DEFAULT)

Hier kann die Standard-Codepage für den jeweiligen Sprachraum eingesetzt werden. Gültige Werte sind:

big5, cp437, cp737, cp775, cp850, cp852, cp855, cp857,  
cp860, cp861, cp862, cp863, cp864, cp865, cp866, cp869,  
cp874, cp932, cp936, cp949, cp950, euc-jp, euc-kr,  
gb2312, iso8859-1, iso8859-2, iso8859-3, iso8859-4,  
iso8859-5, iso8859-6, iso8859-7, iso8859-8, iso8859-9,  
iso8859-14, iso8859-15, koi8-r, sjis.

Falls ein unkorrekter Wert eingesetzt wird, so wird automatisch der Wert iso8859-1 für mitteleuropäische Sprachen eingesetzt.

Codepage 437 (CONFIG\_NLS\_CODEPAGE\_437)

Die Codepage für USA und das englischsprachige Kanada.

Codepage 737 (CONFIG\_NLS\_CODEPAGE\_737)

Die Codepage für Griechenland.

Codepage 775 (CONFIG\_NLS\_CODEPAGE\_775)

Die Codepage für die baltischen Sprachen.

Codepage 850 (CONFIG\_NLS\_CODEPAGE\_850)

Die Codepage für Mittel- und Westeuropa, z.B. UK, Spanien, Italien, Frankreich und Deutschland.

Codepage 852 (CONFIG\_NLS\_CODEPAGE\_852)

Die Codepage für Osteuropa, z.B. Albanien, Ungarn, Polen und Rumänien.

Codepage 855 (CONFIG\_NLS\_CODEPAGE\_855)

Die Codepage für kyrillische Buchstaben.

Codepage 857 (CONFIG\_NLS\_CODEPAGE\_857)

Die Codepage für die Türkei.

Codepage 860 (CONFIG\_NLS\_CODEPAGE\_860)

Die Codepage für Portugal.

Codepage 861 (CONFIG\_NLS\_CODEPAGE\_861)

Die Codepage für Island.

Codepage 862 (CONFIG\_NLS\_CODEPAGE\_862)

Die Codepage für Israel.

Codepage 863 (CONFIG\_NLS\_CODEPAGE\_863)

Die Codepage für das französisch-sprachige Kanada.

Codepage 864 (CONFIG\_NLS\_CODEPAGE\_864)

Die Codepage für Arabisch.

Codepage 865 (CONFIG\_NLS\_CODEPAGE\_865)

Die Codepage für Skandinavien.

Codepage 866 (CONFIG\_NLS\_CODEPAGE\_866)

Die Codepage für Rußland.

Codepage 869 (CONFIG\_NLS\_CODEPAGE\_869)

Die Codepage für Griechenland.

Codepage 874 (CONFIG\_NLS\_CODEPAGE\_874)

Die Codepage für Thailand.

NLS ISO 8859-1 (CONFIG\_NLS\_ISO8859\_1)

Der als Latin-1 bekannte Zeichensatz, der in Nord-, West- und Südeuropa benutzt wird, also auch im deutschsprachigen Raum.

NLS ISO 8859-2 (CONFIG\_NLS\_ISO8859\_2)

Dieses ist der Zeichensatz Latin-2, der in Osteuropa benutzt wird.

NLS ISO 8859-3 (CONFIG\_NLS\_ISO8859\_3)

Dieses ist der Zeichensatz Latin-3, der Für Malta und die Türkei, sowie für Esperanto (!) benutzt wird.

NLS ISO 8859-4 (CONFIG\_NLS\_ISO8859\_4)

Dieser als Latin-4 bekannte Zeichensatz wird im Baltikum - Estland, Lettland und Litauen - benutzt.

NLS ISO 8859-5 (CONFIG\_NLS\_ISO8859\_5)

ISO8859-5 ist der kyrillische Zeichensatz, der in Russland, Weißrussland, Bulgarien, Serbien und der Ukraine benutzt wird.

NLS ISO 8859-6 (CONFIG\_NLS\_ISO8859\_6)

Der arabische Zeichensatz.

NLS ISO 8859-7 (CONFIG\_NLS\_ISO8859\_7)

Der neugriechische Zeichensatz.

NLS ISO 8859-8 (CONFIG\_NLS\_ISO8859\_8)

Der hebräische Zeichensatz.

NLS ISO 8859-9 (CONFIG\_NLS\_ISO8859\_9)

Für die Türkei.

NLS KOI8-R (CONFIG\_NLS\_KOI8\_R)

Der bevorzugte russische Zeichensatz.

## 25. Console drivers

VGA text console (CONFIG\_VGA\_CONSOLE)

Dieser Treiber ermöglicht die Anzeige im Text-Modus auf dem Bildschirm. Jeder möchte das, es sei denn, er konfiguriert ein eingebettetes System. Das Programm `SVGATextMode` kann die Möglichkeiten der jeweiligen Karte im Text-Modus ausschöpfen. Es ist erhältlich von <ftp://metalab.unc.edu/pub/Linux/utils/console>.

Video mode selection support (CONFIG\_VIDEO\_SELECT)

Dieser Treiber ermöglicht die Auswahl der verschiedenen Text-Modi, die die Grafik-Karte erlaubt, während des Bootens. Falls die Karte andere Modi als den Standard 80 x 25 erlaubt, so kann man Bootparameter `vga=` mit der entsprechenden Option eingeben. Auch die Angabe von `video=ask` ist möglich, hiermit wird das Video-Modus-Menü aktiviert während des Bootens.

MDA text console (dual-headed) (CONFIG\_MDA\_CONSOLE)

Dieses ist der Treiber für ältere MDA oder monochrome Hercules-Karten. Diese können hierdurch als zweite Video-Karte im LINUX-System arbeiten. Falls die MDA-Karte der primäre Hardware-Treiber im Rechner ist, sollte hier mit `n` geantwortet werden, den der Standard-VGA-Treiber kann diese Situation behandeln. Der Treiber kann auch als Modul geladen werden.

## Frame-buffer support

Support for frame buffer devices (EXPERIMENTAL) (CONFIG\_FB)

Das virtuelle Frame-Buffer-Gerät ist eine Abstraktion der Grafikkarte. Mit Hilfe dieser Schnittstelle braucht die Anwender-Software sich nicht um Details der Hardware kümmern. Frame-Buffer arbeiten Hardware-System-übergreifend und so muß sich ein Programmierer nicht um verschiedene Architekturen kümmern. Auf das Gerät wird über die Gerätedateien /dev/fbX zugegriffen. Genaueres entält der Text /usr/src/linux/Documentation/fb/framebuffer.txt sowie das HOWTO unter <http://www.tahallah.demon.co.uk/programming/prog.html>. Inzwischen existiert auch ein X-Server, der explizit auf das Frame-Buffer-Gerät zugreift. Auf vielen Rechnern außerhalb der PC-Welt ist der Frame-Buffer oft die einzige Möglichkeit, um auf die Grafik zugreifen zu können.

nVidia Riva support (EXPERIMENTAL) (CONFIG\_FB\_RIVA)

Der Treiber für Graphikkarten mit dem Chip nVidia Riva, der auch TNTx genannt wird.

Cirrus Logic support (EXPERIMENTAL) (CONFIG\_FB\_CLGEN)

Dieses ist der Treiber für Graphikkarten für den Amiga, die auf dem Chips Cirrus Logic GD542x/543x basieren. Hierzu gehören SD64, Piccolo, Picasso II/II+, Picasso IV und EGS Spectrum. Weitere Information enthält Documentation/fb/clgenfb.txt.

Permedia2 support (EXPERIMENTAL) (CONFIG\_FB\_PM2)

Der Treiber für die Graphikkarte Permedia 2.

Cyber2000 support (CONFIG\_FB\_CYBER2000)

Der Treiber für die VGA-Chips Integrations CyberPro 20x0 und 5000, die u.a. im Rebel.com Netwinder benutzt werden.

#### VESA VGA graphics console (CONFIG\_FB\_VESA)

Dieses ist der Treiber für das Frame-Buffer-Gerät für Grafikkarten, die dem Standard VESA 2.0 entsprechen. Der ältere Standard VESA 1.2 wird hierdurch nicht unterstützt. Kostenlos gibt es außerdem während des Bootens einen Pinguin dazu. Weitere Information enthält der Text /usr/src/linux/Documentation/fb/vesafb.txt.

#### VGA 16-color graphics console (NEW) (CONFIG\_FB\_VGA16)

Dieses ist der Treiber für das Frame-Buffer-Gerät der VGA-Karte mit 16 Farben.

#### Hercules mono graphics console (EXPERIMENTAL) (CONFIG\_FB\_HGA)

Während in manchen Internet-Cafes die berühmt-berüchtigte monochrome Hercules-Graphikkarte als Wandschmuck dient, schreiben andere Menschen hierfür noch Linux-Treiber. Ja, die Dinger sind halt sperrig, aber unverwüstlich.

#### Matrox acceleration (EXPERIMENTAL) (CONFIG\_FB\_MATROX)

Dieses ist der Treiber für folgende Matrox-Grafikkarten: Matrox Millennium, Matrox Millennium II, Matrox Mystique, Matrox Mystique 220, Matrox Productiva G100, Matrox Mystique G200, Matrox Millennium G200 oder Matrox Marvel G200. Derzeit sind allerdings die Karten G100, Mystique G200 und Marvel G200 noch nicht getestet. Es können diverse Bootparameter in der Form

`video=matrox:XXX"`

übergeben werden, wobei XXX verschiedene Werte annehmen kann. Dieses sind am Ende des Quelltextes aufgeführt: /usr/src/linux/drivers/video/matroxfb.c. Weitere Information

enthält die Datei `/usr/src/linux/Documentation/fb/matroxfb.txt`. Dieser Treiber kann auch als Modul geladen werden.

#### **Millennium I/II support (CONFIG\_FB\_MATROX\_MILLENIUM)**

Dieses ist der Treiber für die Grafikkarten Matrox Millennium und Matrox Millennium II. Wenn weiter unten die Option Advanced lowlevel driver options aktiviert wird, so muß außerdem eine der folgenden Optionen eingeschaltet sein: 4 bpp packed pixel, 8 bpp packed pixel, 16 bpp packed pixel, 24 bpp packed pixel oder 32 bpp packed pixel.

#### **Mystique support (CONFIG\_FB\_MATROX\_MYSTIQUE)**

Dieses ist der Treiber für die Grafikkarten Matrox Mystique und Matrox Mystique 220. Wenn weiter unten die Option Advanced lowlevel driver options aktiviert wird, so muß außerdem eine der folgenden Optionen eingeschaltet sein: 8 bpp packed pixel, 16 bpp packed pixel, 24 bpp packed pixel oder 32 bpp packed pixel.

#### **G100/G200/G400 support (CONFIG\_FB\_MATROX\_G100)**

Dieses ist der Treiber für die Grafikkarten Matrox Productiva G100, Matrox Mystique G200, Matrox Marvel G200 oder Matrox Millennium G200. Wenn weiter unten die Option Advanced lowlevel driver options aktiviert wird, so muß außerdem eine der folgenden Optionen eingeschaltet sein: 8 bpp packed pixel, 16 bpp packed pixel, 24 bpp packed pixel oder 32 bpp packed pixel.

#### **Matrox I2C support (CONFIG\_FB\_MATROX\_I2C)**

Dieses ist der Treiber für den I2C-Bus, der auf allen Matrox-Karten vorhanden ist.

G400 second head support (CONFIG\_FB\_MATROX\_MAVEN)

Wer zwei Monitore parallel an einer Matrox-Karte G 400 oder an einer G 200 mit MGA-TVO betreiben möchte, benötigt diesen Treiber. Achtung: der Betrieb des zweiten Bildschirms ist nicht kompatibel mit den beschleunigten SVGA-Servern XFree 3.3.x! Mit XFree 3.9.17 kann man beide Monitore betreiben, indem man SVGA mit dem Treiber `fbdev` benutzt.

Der Treiber startet im Monitor-Modus und man benötigt das Werkzeug `matroxset` um nach PAL oder NTSC zu schalten oder um die Ausgabe der beiden Monitore zu vertauschen. Das Werkzeug erhält man unter `ftp://platan.vc.cvut.cz/pub/linux/matrox-latest`. Der Zweite Monitor startet immer im Modus 640 x 480; man muß `fbset` benutzen, um die Auflösung zu ändern. Auch sei darauf hingewiesen, daß der zweite Monitor nur 16 oder 32 bpp unterstützt; also sollte der entsprechende Treiber (s.u.) eingebunden werden.

Multihead support (CONFIG\_FB\_MATROX\_MULTIHEAD)

Wenn mehr als eine der unterstützten, also weiter oben aufgeführten Matrox-Karten im Rechner steckt, und alle genutzt werden sollen, so ist dieses der richtige Treiber. Wer nur eine Matrox-Karte besitzt, sollte hier mit `n` antworten. Man kann mehrere Instanzen dieses Moduls laden mit dem Parameter `dev=N` wobei `N` die Werte 0, 1, etc. für die verschiedenen Matrox-Karten annehmen kann.

ATI Mach64 display support (EXPERIMENTAL) (CONFIG\_FB\_ATY)

Dieses ist der Treiber für Grafikkarten mit dem Chip ATI Mach64.

ATI Rage 128 display support (EXPERIMENTAL) (CONFIG\_FB\_ATY128)

Der Treiber für Graphikkarten mit dem Chip ATI Rage 128. Weitere Information enthält `Documentation/fb/aty128fb.txt`.

3Dfx Banshee/Voodoo3 display support (EXPERIMENTAL)  
(CONFIG\_FB\_3DFX)

Der Treiber für Graphikkarten mit den Chips 3Dfx Banshee bzw. Voodoo 3.

SIS 630/540 display support (EXPERIMENTAL) (CONFIG\_FB\_3DFX)

Der Treiber für Graphikkarten mit den Chips SiS 630 bzw. 540.

Virtual Frame Buffer support (ONLY FOR TESTING!)  
(CONFIG\_FB\_VIRTUAL)

Dieser Treiber wurde für Testzwecke geschrieben und sollte von normalen Benutzer nicht aktiviert werden.

Advanced low level driver options (CONFIG\_FBCON\_ADVANCED)

Normalerweise kümmert sich ein Low-Level-Treiber automatisch, wie die Pixel im Video-RAM im Text-Modus organisiert werden. Dieses ist für den normalen Sterblichen auch ausreichend. Wer hier aber hier weiteres Tuning vornehmen möchte, der kann diesen Treiber aktivieren und muß einen der passenden Low-Level-Frame-Buffer-Treiber aktivieren. Dieser Treiber kann auch als Modul geladen werden.

Monochrome support (CONFIG\_FBCON\_MFB)

Dieses ist der Low-Level-Frame-Buffer-Treiber für monochrome Karten (1 Bit Farbtiefe).

2 bpp packed pixels support (CONFIG\_FBCON\_CFB2)

Dieses ist der Low-Level-Frame-Buffer-Treiber für Karten mit 4 Farben (2 Bit Farbtiefe).

4 bpp packed pixels support (CONFIG\_FBCON\_CFB4)

Dieses ist der Low-Level-Frame-Buffer-Treiber für Karten mit 16 Farben (4 Bit Farbtiefe).

8 bpp packed pixels support (CONFIG\_FBCON\_CFB8)

Dieses ist der Low-Level-Frame-Buffer-Treiber für Karten mit 256 Farben (8 Bit Farbtiefe).

16 bpp packed pixels support (CONFIG\_FBCON\_CFB16)

Dieses ist der Low-Level-Frame-Buffer-Treiber für Karten mit 32k oder 64k Farben (15 oder 16 Bit Farbtiefe), auch bekannt als Hicolor.

24 bpp packed pixels support (CONFIG\_FBCON\_CFB24)

Dieses ist der Low-Level-Frame-Buffer-Treiber für Karten mit 16M Farben (24 Bit Farbtiefe), auch bekannt als Truecolor.

32 bpp packed pixels support (CONFIG\_FBCON\_CFB32)

Dieses ist der Low-Level-Frame-Buffer-Treiber für Karten mit 16M Farben (32 Bit Farbtiefe), auch bekannt als Truecolor.

Amiga bitplanes support (CONFIG\_FBCON\_AFB)

Dieses ist der Low-Level-Frame-Buffer-Treiber für 2 bis 256 Farben (1 bis 8 Bitebenen) für den Amiga.

Amiga interleaved bitplanes support (CONFIG\_FBCON\_IIBM)

Dieses ist der Low-Level-Frame-Buffer-Treiber für 2 bis 256 Farben (1 bis 8 Bitebenen) für den Amiga im Modus Interleave.

Atari interleaved bitplanes (2 planes) support (`CONFIG_FBCON_IPLAN2P2`)

Dieses ist der Low-Level-Frame-Buffer-Treiber für 4 Farben (2 Bitebenen) für den Atari im Modus Interleave.

Atari interleaved bitplanes (4 planes) support (`CONFIG_FBCON_IPLAN2P4`)

Dieses ist der Low-Level-Frame-Buffer-Treiber für 16 Farben (4 Bitebenen) für den Atari im Modus Interleave.

Atari interleaved bitplanes (8 planes) support (`CONFIG_FBCON_IPLAN2P8`)

Dieses ist der Low-Level-Frame-Buffer-Treiber für 256 Farben (8 Bitebenen) für den Atari im Modus Interleave.

Mac variable bpp packed pixels support (`CONFIG_FBCON_MAC`)

Dieses ist der Low-Level-Frame-Buffer-Treiber für 1 bis 32 Bits für den Mac. Die variable Fontgröße wird unterstützt.

VGA 16-color planar support (`CONFIG_FBCON_VGA_PLANES`)

Dieses ist der Frame-Buffer-Treiber für uralte VGA, die nicht VESA-2-kompatibel sind.

VGA characters/attributes support (NEW) (`CONFIG_FBCON_VGA`)

Dieses ist der Low-Level-Frame-Buffer-Treiber für den VGA-Text-Modus. Hierzu muß zusätzlich weiter oben der Treiber VGA chipset support (text only) aktiviert sein.

HGA monochrome support (EXPERIMENTAL) (`CONFIG_FBCON_HGA`)

Dieses ist der Gerätetreiber für die (sehr alten) monochromen Hercules Graphikkarten.

Support only 8 pixels wide fonts (`CONFIG_FBCON_FONTWIDTH8_ONLY`)

Durch diese Option wird die Kernel-Unterstützung für Fonts auf solche mit einer Pixelgröße von 8 x 8 reduziert.

Select compiled-in fonts (`CONFIG_FBCON_FONTS`)

Falls man andere als die Standard-Fonts der Frame-Buffer-Konsole benutzen möchte, so muß hier mit y antwortet werden. Die verschiedenen möglichen Fonts werden im Folgenden abgefragt.

VGA 8x8 font (`CONFIG_FONT_8x8`)

Dieses ist der Font für die sogenannte hohe Auflösung im Textmodus, also für 80 Zeichen zu 50 Zeilen oder höher.

VGA 8x16 font (`CONFIG_FONT_8x16`)

Der VGA-Standard-Font für eine Darstellung von 80 Zeichen zu 25 Zeilen.

Sparc console 8x16 font (`CONFIG_FONT_SUN8x16`)

Der Standard-Font für Sun-Sparc-Rechner.

Pearl (old m68k) console 8x8 font (`CONFIG_FONT_PEARL_8x8`)

Der Font 8 x 8 für ältere Motorola 68xxx-Systeme.

Acorn console 8x8 font (CONFIG\_FONT\_ACORN\_8x8)

Der Font 8 x 8 für Acorn-Systeme.

## 26. Sound

Sound card support (CONFIG\_SOUND)

Falls eine Soundkarte im entsprechenden Rechner vorhanden ist, sollte hier mit `y` geantwortet werden. Auch sollte man das entsprechende Handbuch griffbereit haben, da auch nach der Auswahl der Soundkarte Informationen wie E/A-Adresse oder Interrupt zu der angegebenen Karte abgefragt werden.

Falls einer der folgenden Treiber als Modul kompiliert werden soll, so gibt es spezielle Information unter `Documentation/sound/README.modules`. Wer keine Soundkarte hat und trotzdem nicht auf Sound verzichten möchte, der findet Treiber, um dem PC-Lautsprecher Töne zu entlocken unter `ftp://ftp.infradead.org/pub/pcsp/`.

C-Media PCI (CMI8338/8378) (CONFIG\_SOUND\_CMPCI)

Der Treiber für die PCI-Karten von C-Media mit den Chips CMI 8338 oder 8378. Weitere Information enthält `/Documentation/sound/CMI8338`.

Enable S/PDIF loop for CMI8738 (CONFIG\_SOUND\_CMPCI\_SPDIFLOOP)

Ermöglicht bei Karten mit dem Chip 8738 das direkte Durschleifen von S/PDIF-in nach S/PDIF-out.

Enable 4 channel mode for CMI8738 (CONFIG\_SOUND\_CMPCI\_4CH)

Aktiviert den Vier-Kanal-Modus für Karten mit dem Chip CMI 8738.

Separate rear out jack (CONFIG\_SOUND\_CMPCI\_REAR)

Aktiviert den unabhängigen Ausgang Rear-out für die CMI-Karten.

Creative SBLive! (EMU10K1) (CONFIG\_SOUND\_EMU10K1)

Der Treiber für Soundkarten mit dem Chipsatz EMU 10K1, wie z.B. die Karten Creative SBLive!, SB PCI512 und Emu-APS.

Crystal SoundFusion (CS4280/461x) (CONFIG\_SOUND\_FUSION)

Der Triber für die Karten SoundFusion (CS 4280/46xx) von Crystal. Falls dieser Treiber nicht funktioniert, sollte der Treiber für den CS 4232 versucht werden.

Crystal Sound CS4281 (CONFIG\_SOUND\_CS4281)

Für Karten mit dem Soundchip CS 423x von Crystal. Weitere Information enthält Documentation/sound/CS4281.

Ensoniq AudioPCI (ES1370) (CONFIG\_SOUND\_ES1370)

Wer eine Soundkarte besitzt, die auf dem Chip ES1370 basiert, für den ist dieses der richtige Treiber. Weitere Information enthält der Text /usr/src/linux/Documentation/sound/es1370.

Creative Ensoniq AudioPCI 97 (ES1371) (CONFIG\_SOUND\_ES1371)

Wer eine Soundkarte besitzt, die auf dem Chip ES1371 basiert, wie z.B. die Ensoniq AudioPCI97, für den ist dieses der richtige Treiber. Weitere Information enthält der Text /usr/src/linux/Documentation/sound/es1371.

ESS Technology Solo1 (CONFIG\_SOUND\_ESSSOLO1)

Wer eine PCI-Soundkarte besitzt, die auf dem Chip Solo1 von ESS Technology basiert, für den ist dieses der richtige Treiber. Weitere Information enthält der Text /usr/src/linux/Documentation/sound/solo1.

#### ESS Maestro, Maestro2, Maestro2E driver (CONFIG\_SOUND\_MAESTRO)

Der Treiber für die PCI-Soundkarten Maestro von ESS. Dieses beinhaltet die Karten Maestro 1, Maestro 2 und Maestro 2E. Weitere Information enthält Documentation/sound/Maestro.

#### S3 SonicVibes based PCI sound cards (CONFIG\_SOUND\_SONICVIBES)

Wer eine PCI-Soundkarte besitzt, die auf dem Chip S3 von SonicVibes basiert, für den ist dieses der richtige Treiber. Weitere Information enthält der Text /usr/src/linux/Documentation/sound/sonicvibes.

#### Trident 4DWave DX/NX, SiS 7018 or ALi 5451 PCI Audio Core (CONFIG\_SOUND\_TRIDENT)

Dieses ist der Treibr für die Soundkarte Trident 4DWave-DX/NX. Außerdem unterstützt er die Onboard-Soundchips SiS 7018 und ALi 5451. Der Soundchip SiS 7018 befindet sich auf der Super South Bridge SiS960 und auch in den Chips SiS 540/630. Der Soundchip ALi 5451 PCI befindet sich auf den South Bridges ALi M1535, M1535D, M1535+ und M1535D+. Weitere Informationen enthalten die Kommentare im Quelltext driver/sound/trident.c.

#### Support for Turtle Beach MultiSound Classic, Tahiti, Monterey (CONFIG\_SOUND\_MSNDCLAS)

Der Treiber für folgende Soundkarten von Turtle Beach: MultiSound Classic, Tahiti und Monterey, aber nicht für die Pinnacle oder die Fiji. Weitere Information enthält Documentation/sound/MultiSound.

Full pathname of MSNDINIT.BIN firmware file  
(CONFIG\_MSNDCLAS\_INIT\_FILE)

Die MultiSound-Karten benötigen zur Initialisierung eine spezielle Firmware, die nicht in diesem Treiber enthalten ist. Hier muß der Pfad und der Dateiname dieser Firmware angegeben werden. Wie man die Firmware erhält, wird in der Datei Documentation/sound/MultiSound beschrieben.

Full pathname of MSNDPERM.BIN firmware file  
(CONFIG\_MSNDCLAS\_PERM\_FILE)

Die MultiSound-Karten benötigen zum Betrieb eine weitere spezielle Firmware, die nicht in diesem Treiber enthalten ist. Hier muß der Pfad und der Dateiname dieser Firmware angegeben werden. Wie man die Firmware erhält, wird in der Datei Documentation/sound/MultiSound beschrieben.

Support for Turtle Beach MultiSound Pinnacle, Fiji  
(CONFIG\_SOUND\_MSNDPIN)

Der Treiber für die Soundkarten MultiSound Pinnacle und Fiji von Turtle Beach. Weitere Information in der Datei Documentation/sound/MultiSound .

Full pathname of PNDSPINI.BIN firmware file  
(CONFIG\_MSNDPIN\_INIT\_FILE)

Die MultiSound-Karten benötigen zur Initialisierung eine spezielle Firmware, die nicht in diesem Treiber enthalten ist. Hier muß der Pfad und der Dateiname dieser Firmware angegeben werden. Wie man die Firmware erhält, wird in der Datei Documentation/sound/MultiSound beschrieben.

Full pathname of PNDSPERM.BIN firmware file  
(CONFIG\_MSNDPIN\_PERM\_FILE)

Die MultiSound-Karten benötigen zum Betrieb eine weitere spezielle Firmware, die nicht in diesem Treiber enthalten ist. Hier muß der Pfad und der Dateiname dieser Firmware angegeben werden. Wie man die Firmware erhält, wird in der Datei Documentation/sound/MultiSound beschrieben.

#### VIA 82C686 Audio Codec (CONFIG\_SOUND\_VIA82CXXX)

Dieses ist der Treiber für die Audio Codec, die auf den Chips VIA 82Cxxx basiert. Typischerweise sind diese Onboard-Chips auf den Mutterbrettern.

#### OSS sound modules (CONFIG\_SOUND\_OSS)

Das Open Sound System kann viele verschiedene Soundkarten ansteuern und ist relativ einfach zu konfigurieren. Dazu muß der Treiber für die jeweilige Soundkarte aktiviert sein. Der SuSE-Installation liegt eine kostenlose, ältere Version bei. Die neuste Version kann für schmales Geld bei <http://www.opensound.com/oss.html> erworben werden. Der Treiber kann auch als Modul kompiliert werden.

#### Verbose initialisation (CONFIG\_SOUND\_TRACEINIT)

Gibt einige Informationen über die Initialisierung des OSS aus.

#### Persistent DMA buffers (CONFIG\_SOUND\_DMAP)

Linux hat oft Probleme, die DMA-Buffer für ISA-Soundkarten zu allozieren, wenn der Rechner weniger als 16 MB RAM besitzt. Um das Problem zu umgehen wird durch diesen Treiber beim Start des Systems ein DMA-Puffer von 64kB angelegt, der bis zum Herunterfahren des Systems bestehen bleibt.

#### AD1816(A) based cards (EXPERIMENTAL) (CONFIG\_SOUND\_AD1816)

Dieses ist der Treiber für alle Soundkarten, die auf dem Chip AD1816(A) von Analog Devices basieren. Weitere Information bietet `/usr/src/linux/Documentation/sound/AD1816`.

**Aztech Sound Galaxy (non-PnP) cards (CONFIG\_SOUND\_SGALAXY)**

Dieses ist der Treiber für die ältere Variante der Soundkarte Aztech Sound Galaxy ohne PnP-Unterstützung. Außerdem werden die Karten Waverider Pro 32 - 3D und Galaxy Washington 16 unterstützt.

**Adlib Cards (CONFIG\_SOUND\_ADLIB)**

Der Treiber für die AdLib-Soundkarten.

**ACI mixer (miroPCM12) (CONFIG\_SOUND\_ACI\_MIXER)**

Das Protokoll ACI (Audio Command Interface) wird benutzt, um mit einigen Karten von Miro zu kommunizieren, wie z.B. miroSOUND PCM12 und PCM20.

**Crystal CS4232 based (PnP) cards (CONFIG\_SOUND\_CS4232)**

Dieses ist der Treiber für Karten, die auf dem Chipsatz Crystal CS4232 aufbauen. Weitere Information enthält `Documentation/sound/CS4232`.

**Ensoniq SoundScape support (CONFIG\_SOUND\_SScape)**

Falls die Karte auf dem Ensoniq-SoundScape-Chipsatz basiert, ist dieses der richtige Treiber. Derartige Karten werden außer von Ensoniq auch von Spea und Reveal hergestellt.

**Gravis Ultrasound support (CONFIG\_SOUND\_GUS)**

Dieser Treiber unterstützt alle Karten der Firma Gravis Ultrasound, einschließlich der GUS und der GUS MAX. Weitere Information enthält Documentati-on/sound/ultrasound.

**16 bit sampling option of GUS (\_NOT\_ GUS MAX) (CONFIG\_SOUND\_GUS16)**

Aktiviert das Sampling mit 16 Bit für die GUS-Karte, aber nicht die GUS-MAX-Karte.

**GUS MAX support (CONFIG\_SOUND\_GUSMAX)**

Der Treiber für die zusätzlichen Funktionen der Karte GUS MAX.

**Intel ICH audio support (CONFIG\_SOUND\_ICH)**

Der Treiber für den Intel Chip i810, geschrieben von Meister Cox persönlich. Weitere Information enthält driver/sound/i810\_audio.c.

**MediaTriX AudioTriX Pro support (CONFIG\_TRIX)**

Dieses ist der Treiber für die Karte AudioTriX Pro, die von Media TriX hergestellt wird.

**Microsoft Sound System support (CONFIG\_SOUND\_MSS)**

Falls eine original Windows-Sound-System-Karte von Microsoft oder eine Aztech SG 16 Pro oder NX16 Pro vorhanden ist, ist dieses der richtige Treiber. Falls der Hersteller der Karte behauptet, daß sie kompatibel zum Microsoft Sound System ist, kann dieser Treiber auch funktionieren. Falls für die Karte in dieser Sektion ein eigener Treiber vorhanden ist, sollte aber dieser aktiviert und an dieser Stelle mit n geantwortet werden.

**MPU-401 support (NOT for SB16) (CONFIG\_SOUND\_MP401)**

Bei dieser Option ist folgendes Wissen notwendig: die MPU-401-Schnittstelle wird von fast allen Soundkarten unterstützt. Einige Karten haben in ihrem eigenen Treiber bereits die Unterstützung dieser Schnittstelle, so daß das Aktivieren beider Treiber Konflikte verursachen wird. Auch falls der Schnittstellen-Standard nicht hunderprozentig unterstützt wird, kann es Probleme mit diesem Treiber geben. Genauere Informationen über die jeweilige Kompatibilität gibt die Datei `drivers/sound/Readme.cards` file.

#### NM256AV/NM256ZX audio support (CONFIG\_SOUND\_NM256)

Dieses ist der Treiber für die Chips NeoMagic 256AV/256ZX. Diese werden in folgenden Geräten gefunden: Sony Z505S/SX/DX, einigen der Sony F-Serie und den Laptops von Dell mit den Typenbezeichnungen Latitude CPi und CPt. Weitere Information enthält `/usr/src/linux/Documentation/sound/NM256`.

#### OPTi MAD16 and/or Mozart based cards (CONFIG\_SOUND\_MAD16)

Falls sich auf der Soundkarte der Mozart-Chipsatz OAK OTI-601 oder der MAD16-Chipsatz (OPTi 82C928 oder 82C929) befindet, ist dieses der richtige Treiber. Diese Chipsätze sind recht verbreitet, so daß einige Noname-Hersteller sie verwenden. Außerdem wird der MAD16-Chip auch in einigen Modellen der Firmen Turtle Beach (Tropez), Reveal und Diamond verwendet.

#### Support MIDI in older MAD16 based cards (requires SB) (CONFIG\_MAD16\_OLDCARD)

Dieses ist der MIDI-Treiber für die älteren Modelle, die auf den Chipsätzen C 928 oder Mozart basieren. Wer diese Option aktiviert muß außerdem die Unterstützung für den Sound Blaster aktivieren.

#### ProAudioSpectrum 16 support (CONFIG\_SOUND\_PAS)

Dieses ist der Treiber für die Karten ProAudio Spectrum 16, ProAudio Studio 16 oder Logitech SoundMan 16. Andere Karten von Media Vision oder Logitech sind nicht PAS16-kompatibel; für sie sind andere Treiber zuständig. Weitere Information enthält Documentation/sound/PAS16.

PSS (AD1848, ADSP-2115, ESC614) support (CONFIG\_SOUND\_PSS)

Der Treiber für die Karten Orchid SW32, Cardinal DSP16, Beethoven ADSP-16 oder andere Karten, die auf dem PSS-Chipsatz basieren (AD1848 und ADSP-2115 DSP und Echo ESC614 ASIC). Weitere Information enthält Documentation/sound/PSS.

Enable PSS mixer (Beethoven ADSP-16 and other compatible) (CONFIG\_PSS\_MIXER)

Zur erweiterten Kontrolle der Beethoven ADSP-16 und anderer kompatibler Karten. Hiermit können Lautstärke, Höhen und Bässe geregelt werden. Weitere Information enthält Documentation/sound/PSS.

Have DSPxxx.LD firmware file (CONFIG\_PSS\_HAVE\_BOOT)

Falls man eine Datei mit Firmware für die PPS-Karten besitzt, so sollte hier mit y geantwortet werden.

Full pathname of DSPxxx.LD firmware file (CONFIG\_PSS\_BOOT\_FILE)

Der volle Pfad der Firmware-Datei, die normalerweise DSPxxx.LD oder SYNTH.LD heißen.

100% Sound Blaster compatibles (SB16/32/64, ESS, Jazz16) support (CONFIG\_SOUND\_SB)

Falls die original SoundBlaster-Karte von Creative Labs oder ein hundertprozentig kompatibler Clone wie Thunderboard oder SM Games vorhanden ist, ist

dieses der richtige Treiber. Weitere Informationen sind in drivers/sound/Readme.cards enthalten. Falls der Hersteller einer unbekannten Karte garantiert, daß sie SoundBlaster-kompatibel ist, sollte dieser Treiber funktionieren.

#### AWE32 synth (CONFIG\_SOUND\_AWE32\_SYNTH)

Falls man eine der folgenden Karten besitzt, benötigt man diesen Treiber: SoundBlaster SB32, AWE32-PnP, SB AWE64 oder Kompatible. Folgende Texte enthalten weitere Information: /usr/src/linux/Documentation/sound/README.awe und /usr/src/linux/Documentation/sound/AWE32.

#### Full support for Turtle Beach WaveFront (Tropez Plus, Tropez or Maui) (CONFIG\_SOUND\_WAVEFRONT)

Der Treiber für die Karten Tropez Plus, Tropez oder Maui von Turtle Beach. Weitere Information enthalten die Dateien Documentation/sound/Wavefront und Documentation/sound/Tropez+.

#### Limited support for Turtle Beach Wave Front (Maui, Tropez) (CONFIG\_SOUND\_MAUI)

Der Synthesizer-Treiber für die Karten Turtle Beach Wave Front, Maui und Tropez.

#### Yamaha FM synthesizer (YM3812/OPL-3) support (CONFIG\_SOUND\_YM3812)

Der Treiber für Soundkarten mit FM-Chips von Yamaha (OPL2/OPL3/OPL4). Für den Chip OPL3 gibt es weitere Information in Documentation/sound/OPL3.

#### Yamaha OPL3-SA1 audio controller (CONFIG\_SOUND\_OPL3SA1)

Der Treiber für den Soundchip OPL3-SA1 von Yamaha, die üblicherweise Obboard auf dem Mutterbrett untergebracht sind. Details enthält Documentation/sound/OPL3-SA.

Yamaha OPL3-SA2, SA3, and SAx based PnP cards  
(CONFIG\_SOUND\_OPL3SA2)

Der Treiber für die PnP-Karten von Yamaha mit den Chips OPL3-SA2, SA3 und SAx. Weitere Information unter Documentation/sound/OPL3-SA2.

6850 UART support (CONFIG\_SOUND\_UART6850)

Diese Option aktiviert den Treiber für die Midi-Schnittstelle des 6850 UART-Chips. Sehr selten.

Gallant Audio Cards (SC-6000 and SC-6600 based)  
(CONFIG\_SOUND\_AEDSP16)

Der Treiber für die Karte Gallant Audio Excel DSP 16. Weitere Information enthält /usr/src/linux/drivers/sound/lowlevel/README.aedsp16 sowie den Kopf des Quelltextes /usr/src/linux/drivers/sound/lowlevel/aedsp16.c und /usr/src/linux/Documentation/sound/ExcelDSP16.

SC-6600 based audio cards (new Audio Excel DSP 16) (CONFIG\_SC6600)

Auf den neueren Karten Audio Excel DSP findet sich der Chip SC-6600. Dieses ist der richtige Treiber für diese Karte.

Activate SC-6600 Joystick Interface (CONFIG\_SC6600\_JOY)

Falls man die Joystick-Schnittstelle der Karte Audio Excel DSP 16 benutzt, muß man hier mit y antworten.

SC-6600 CDROM Interface (4=None, 3=IDE, 1=Panasonic, 0=Sony)  
(CONFIG\_SC6600\_CDROM)

Dieser Treiber wird benutzt, um die CD-ROM-Schnittstelle der Karte Audio Excel DSP zu aktivieren. Gültige Werte sind:

- 0 : Sony
- 1 : Panasonic
- 2 : IDE
- 4 : kein CD-Rom-Laufwerk.

SC-6600 CDROM Interface I/O Address (CONFIG\_SC6600\_CDROMBASE)

Die E-/A-Adresse für die CD-ROM-Schnittstelle des SC-6600.

Audio Excel DSP 16 (SBPro emulation) (CONFIG\_AEDSP16\_SBPRO)

Wenn man die Karte Sound Blaster Pro emulieren möchte, so ist dieses der richtige Treiber.

Audio Excel DSP 16 (MPU401 emulation) (CONFIG\_AEDSP16\_MPU401)

Falls man die vorhandenen Karte überzeugen möchte, daß sie die Midi-Schnittstelle MPU-401 emuliert, so ist dieses der richtige Treiber. Außerdem muß man die Option MPU-401 support aktivieren.

TV card (bt848) mixer support (CONFIG\_SOUND\_TVMIXER)

Der Treiber für Mixer-Funktionen der TV-Karte bt 848.

## 27. USB support

### Support for USB (CONFIG\_USB)

Der Universal Serial Bus (USB) ist die Spezifikation für ein serielles Subsystem, das größere Geschwindigkeit und mehr Möglichkeiten bietet als die serielle Schnittstelle. USB versorgt u.a. die Geräte mit Strom und ermöglicht außerdem das sogenannte „Hot Swapping“, also das Einsticken von Geräten im laufenden Betrieb des Rechners. Maximal 127 Geräte können an einem USB-Port angeschlossen werden. Dieses werden in einer Baumstruktur organisiert, wobei die Geräte die Blätter des Baums darstellen, während die Knoten die Hubs sind. Viele neuere Peripherie-Geräte besitzen USB-Anschlüsse, wie z.B. Tastaturen, Mäuse, Scanner, Modems und Drucker.

Falls man unter Linux mit USB-Geräten arbeiten möchte, so muß zum einen entweder mindestens einer der Treiber UHCI support oder OHCI support aktiviert sein. Zum zweiten muß natürlich der entsprechende Treiber für das jeweilige Gerät aktiviert sein. Weitere Information enthalten die Dateien im Verzeichnis Documentation/usb/, als Einstieg ist Documentation/usb/usb-help.txt empfohlen.

### USB verbose debug messages (CONFIG\_USB\_DEBUG)

Falls der Anschluß von USB-Geräten nicht wie erhofft funktioniert, so produziert dieser Treiber jede Menge Meldungen in der Datei /var/log/messages. Mehr für Entwickler gedacht.

### Preliminary USB device filesystem (CONFIG\_USB\_DEVICEFS)

Dieser Treiber nimmt die USB-Geräte in das virtuelle Dateisystem /proc auf. Dazu muß natürlich auch der Treiber /proc file system support (a.o.) aktiviert sein. Die jeweils am USB aktivierte Geräte findet man dann unter /proc/usb/devices. Die jeweilige Verbindung findet man außerdem unter

/proc/usb/xxx/yyyy, wobei xxx die Bus-Nummer darstellt und yyyy die Gerätenummer. Weitere Information dazu enthält Documentation/usb/proc\_usb\_info.txt. Die Standardeinstellung ist y.

Enforce USB bandwidth allocation (EXPERIMENTAL)  
(CONFIG\_USB\_BANDWIDTH)

Dieser experimentelle Treiber überwacht die Bandbreite des USB-Subsystems derart, daß einzelne Geräte gehindert werden, mehr als 90% der Bandbreite zu belegen, und somit andere Geräte an der korrekten Funktion hindern.

UHCI Alternate Driver (JE) support (CONFIG\_USB\_UHCI\_ALT)

Auch bei USB wurde das Rad wieder zweimal erfunden: es gibt den UHCI-sowie den OHCI-Standard. Dieses ist der Treiber für den UHCI-Standard (Universal Host Controller Interface) von Intel. Alle derzeitigen Host-Controller mit Chips von Intel basieren auf diesem Standard. Dazu gehören die Chips 430TX, 440FX, 440LX, 440BX, i810 und i820. Außerdem arbeiten die VIA-PCI-Chipsätze mit diesem Standard, wie z.B. VIA VP2, VP3, MVP3, Apollo Pro, Apollo Pro II und Apollo Pro 133. Falls der Controller hier nicht aufgeführt ist, sollte man sich die nächste Option ansehen.

OHCI (Compaq, iMacs, OPTi, SiS, ALi, ...) support (CONFIG\_USB\_OHCI)

Dieses ist der Treiber für den OHCI-Standard (Open Host Controller Interface), der von Compaq, Microsoft und National erfunden wurde. Diesen Standard findet man auf den meisten Rechnern, die keine Intel-Architektur besitzen, sowie auf vielen PCs, der USB-Adapter keinen Intelchip besitzt, sondern z.B., die Chips wie SiS 610 oder ALi IV, ALi V, Aladdin Pro. Weiter Information enthält Documentation/usb/ohci.txt.

USB Printer support (CONFIG\_USB\_PRINTER)

Der Standard-Treiber für Drucker am USB-Port. Er kann auch als Modul geladen werden.

#### USB Scanner support (CONFIG\_USB\_SCANNER)

Der Standard-Treiber für den Anschluß von Scannern am USB-Port. Weitere Information enthalten `Documentation/usb/scanner.txt` und `Documentation/usb/scanner-hp-sane.txt`.

#### Microtek X6USB scanner support (EXPERIMENTAL) (CONFIG\_USB\_MICROTEK)

Der Treiber, um den Scanner Microtek X6 USB und baugleiche Scanner am USB-Port anzuschließen. Um den Scanner mit SANE zu betreiben, braucht man von diesem Programm eine gepatchte Version, die man erhält bei <http://fachschaft.cup.uni-muenchen.de/~neukum/scanner.html>.

#### USB Audio support (CONFIG\_USB\_AUDIO)

Der Treiber zur Unterstützung von Audio via USB, wie z.B. Lautsprechern.

#### USB Modem (CDC ACM) support (CONFIG\_USB\_ACM)

Dieser Treiber unterstützt USB-Modems und ISDN-Adapter, die die Schnittstelle CDC ACM (Communication Device Class Abstract Control Model) unterstützen.

#### USB Serial Converter support (CONFIG\_USB\_SERIAL)

Der Treiber für die USB-Adapter, die den Anschluß von Geräten mit normaler serieller Schnittstelle ermöglichen. Der Treiber kann maximal 256 serielle Geräte verwalten. Unterstützte Geräte sind folgende: ConnectTech White-HEAT 4, HandSpring Visor, Keyspan PDA für iMacs, die Keyspan USA-

Baureihe, FTDI, ZyXEL omni.net lcd plus ISDN TA, Digi AccelePort und Belkin F5U103. Außerdem muß der Treiber für das jeweilige Gerät (s.u.) aktiviert werden. Falls es keinen speziellen Treiber für das gewünschte Gerät gibt, kann evtl. der generische Treiber (USB Generic Serial Driver, s.u.). Weitere Information enthält Documentation/usb/usb-serial.txt.

#### USB Generic Serial Driver (CONFIG\_USB\_SERIAL\_GENERIC)

Dieses ist der generische Treiber für Konverter zwischen seriell Anschluß und USB. Außerdem wird der obenstehen Treiber CONFIG\_USB\_SERIAL benötigt. Weitere Information enthält Documentation/usb/usb-serial.txt.

#### USB Handspring Visor Driver (CONFIG\_USB\_SERIAL\_VISOR)

Der Treiber für den HandSpring Visor an seiner USB-Docking-Station. Weitere Information enthält <http://usbvisor.sourceforge.net>.

#### USB Digi International AccelePort USB Serial Driver (CONFIG\_USB\_SERIAL\_DIGI\_ACCELEPORT)

Der Treiber für die verschiedenen Gerät der Baureihe Digi AccelePort. Falls ein paralleler Anschluß vorhanden ist, so erscheint dieser unter Linux als dritter serieller Anschluß. Digi Acceleport USB 8 wird derzeit von diesem Treiber noch nicht unterstützt.

#### USB ConnectTech WhiteHEAT Serial Driver (EXPERIMENTAL) (CONFIG\_USB\_SERIAL\_WHITEHEAT)

Der Treiber für den Konverter ConnectTech WhiteHEAT.

#### USB FTDI Single Port Serial Driver (EXPERIMENTAL) (CONFIG\_USB\_SERIAL\_FTDI\_SIO)

Der Treiber für die Single-Port-Konverter von FTDI. Weitere Information enthält [http://reality.sgi.com/bryder\\_wellington/ftdi\\_sio](http://reality.sgi.com/bryder_wellington/ftdi_sio).

USB Keyspan PDA Single Port Serial Driver (EXPERIMENTAL)  
(CONFIG\_USB\_SERIAL\_KEYSPAN\_PDA)

Der Treiber für die Single-Port-Konverter Keyspan PDA. Der Treiber benutzt die komplett neu entwickelte Firmware.

USB Keyspan USA-xxx Serial Driver (EXPERIMENTAL)  
(CONFIG\_USB\_SERIAL\_KEYSPAN)

Der Treiber für die Konverter der Baureihe Keyspan USA-xxx. Dieser benutzt die offizielle Firmware von Keyspan. Für das jeweilige Gerät muß die entsprechende Firmware aktiviert werden (s.u.). Weitere Information enthält <http://www.linuxcare.com.au/hugh/keysan.html>.

USB Keyspan USA-28 Firmware (CONFIG\_USB\_SERIAL\_KEYSPAN\_USA28)

Die Firmware für den Konverter Keyspan USA-28.

USB Keyspan USA-28X Firmware (CONFIG\_USB\_SERIAL\_KEYSPAN\_USA28X)

Die Firmware für den Konverter Keyspan USA-28X.

USB Keyspan USA-19 Firmware (CONFIG\_USB\_SERIAL\_KEYSPAN\_USA19)

Die Firmware für den Konverter Keyspan USA-19.

USB Keyspan USA-18X Firmware (CONFIG\_USB\_SERIAL\_KEYSPAN\_USA18X)

Die Firmware für den Konverter Keyspan USA-18X.

USB              Keyspan              USA-19W              Firmware  
(CONFIG\_USB\_SERIAL\_KEYSPAN\_USA19W)

Die Firmware für den Konverter Keyspan USA-19W.

USB    ZyXEL    omni.net    LCD    Plus    Driver    (EXPERIMENTAL)  
(CONFIG\_USB\_SERIAL\_OMNINET)

Der Treiber für ZyXEL omni.net LCD ISDN TA.

USB Belkin and Peracom Single Port Serial Driver (EXPERIMENTAL)  
(CONFIG\_USB\_SERIAL\_BELKIN)

Der Treiber für die Single-Port-Konverter von Belkin und Peracom.

USB Serial Converter verbose debug (CONFIG\_USB\_SERIAL\_DEBUG)

Wer Meldungen zum Entwanzten der Konverter zwischen USB und serieller Schnittstelle benötigt, antwortet mit y. Für Entwickler.

USB IBM (Xirlink) C-it Camera support (CONFIG\_USB\_IBMCAM)

Dieses ist der Treiber für die Kamera C-it, auch bekannt unter dem Namen Xirlink PC Camera. Wer dieses Kamera von IBM am USB-Port betreiben möchte, benötigt diesen Treiber. Weitere Information enthält Documentati-on/usb/ibmcam.txt. Außerdem muß der Treiber für v4l (Video For Linux) (s.o.) aktiviert werden. Weitere Information über v4l enthält <http://roadrunner.swansea.uk.linux.org/v4l.shtml>.

USB OV511 Camera support (CONFIG\_USB\_OV511)

Der Treiber für Webcams, die die CCDs OV511/OV511+ oder OV7610/20/20AE benutzen. Weitere Information enthält Documentation/usb/ov511.txt. Der Treiber für Video For Linux (s.o.) muß natürlich auch aktiviert sein.

#### USB Kodak DC-2xx Camera support (CONFIG\_USB\_DC2XX)

Der Treiber für die digitalen Kameras von Kodak mit folgenden Typenbezeichnungen: DC-240 und DC-280 über USB. Außerdem werden die Typen DC-220, DC-260, DC-265 und DC-290 unterstützt. Für die ersten beiden Typen sollte man sich die neueste Version von gPhoto besorgen (<http://www.gphoto.org>), da dies auch USB-Unterstützung bietet. Weitere Information enthält Documentation/usb/dc2xx.txt.

#### USB Mustek MDC800 Digital Camera support (EXPERIMENTAL) (CONFIG\_USB\_MDC800)

Der Treiber für die digitale Kamera MDC 800 von Mustek am USB-Port. Der Treiber kann zusammen mit gPhoto 0.4.3 oder einer höheren Version benutzt werden (<http://www.gphoto.org>). Um den Treiber zu benutzen muß man mit folgendem Befehl eine Gerätedatei erzeugen:

```
mknod /dev/mustek c 180 32
```

#### USB Mass Storage support (CONFIG\_USB\_STORAGE)

Für die Benutzung von von Massenspeichern über USB.

#### USB Mass Storage verbose debug (CONFIG\_USB\_STORAGE\_DEBUG)

Wer Meldungen zum Entwirren des Treibers für Massenspeicher über USB Schnittstelle benötigt, antwortet mit y. Für Entwickler.

#### DABUSB driver (CONFIG\_USB\_DABUSB)

Der Treiber für Empfänger von DAB (Digital Audio Broadcasting) am USB-Port. Weitere Information über Linux und digitales Audio enthält <http://dab.in.tum.de>. Weitere Information über den Treiber unter Documentation/usb/URB.txt.

PLUSB Prolific USB-Network driver (EXPERIMENTAL)  
(CONFIG\_USB\_PLUSUSB)

Der Treiber für das Netzwerkgerät PL-2302 USB-to-USB von Prolific. Hiermit kann eine Punkt-zu-Punkt-Verbindung über USB mit 5 Mbit/s erzielt werden. Nachdem man die zwei Rechner verbunden hat, muß man die Treiber verbinden durch die Eingabe von

```
ifconfig plusb0 10.0.0.1 pointopoint 10.0.0.2
```

bzw. mit getauschten Netzwerkadressen auf der zweiten Seite.

USB ADMtek Pegasus-based ethernet device support (EXPERIMENTAL)  
(CONFIG\_USB\_PEGASUS)

Der Treiber für Ethernet über USB. Bisher werden folgende Geräte unterstützt:

ADMtek AN986 Pegasus, ADMtek ADM8511 Pegasus II, Accton 10/100, Billington USB-100, Corega FEter USB-TX, MELCO/BUFFALO LUA-TX, D-Link DSB-650TX, DSB-650TX-PNA, DSB-650, DU-E10, DU-E100, Linksys USB100TX, USB10TX , LANEED Ethernet LD-USB/TX, SMC 202 und SO-HOWare NUB Ethernet.

USB Diamond Rio500 support (EXPERIMENTAL) (CONFIG\_USB\_RIO500)

Der Treiber für den Anschluß des MP3-Spieler Diamond Rio 500 am USB-Port des Rechners. Weitere Information enthält Documentation/usb/rio.txt, besonders über Gefahren bei schwachen Batterien.

D-Link USB FM radio support (EXPERIMENTAL) (CONFIG\_USB\_DSBR)

Der Treiber für die UKW-Radios von D-Link am USB-Port. Zu beachten ist, daß der Audio-Ausgang nicht digital ist, und also muß Line-Out entweder mit einer Soundkarte oder mit Aktivboxen verbunden werden. Außerdem muß auch der Treiber für Video For Linux (s.o.) aktiviert werden.

#### USB Bluetooth support (EXPERIMENTAL) (CONFIG\_USB\_BLUETOOTH)

Für den Anschluß von Geräten mit Bluetooth-Protokoll an USB. Wer Blauzahn-Geräte betreiben möchte, benötigt außerdem noch den entsprechenden Stack, der unter <http://developer.axis.com/software/index.shtml> zu erhalten ist.

#### NetChip 1080-based USB Host-to-Host Link (EXPERIMENTAL) (CONFIG\_USB\_NET1080)

Für alle Geräte, die auf dem NetChip 1080 beruhen. Mit diesem Treiber kann eine Punkt-zu-Punkt-Verbindung aufgebaut werden, übrigens auch mit Windows-Rechnern. Weitere Information enthält der Kommentar im Quelltext `drivers/usb/net1080.c`.

#### USB Human Interface Device (full HID) support (CONFIG\_USB\_HID)

Der grundlegende Treiber für alle Geräte am USB-Port, bei denen Menschen Eingaben am Rechner tätigen. Dieses umfaßt u.a. Tastaturen, Mäuse und Graphik-Tablets. Weitere Information enthält `Documentation/usb/input.txt`.

#### USB HIDBP Keyboard (basic) support (CONFIG\_USB\_KBD)

Der Treiber für Tastaturen am USB-Port. Dieser Treiber ist deutlich kleiner als der HID-Treiber (s.o.).

#### USB HIDBP Mouse (basic) support (CONFIG\_USB\_MOUSE)

Der Treiber für Mäuse am USB-Port. Dieser Treiber ist deutlich kleiner als der HID-Treiber (s.o.).

Wacom Intuos/Graphire tablet support (`CONFIG_USB_WACOM`)

Der Treiber für die Graphik-Tablets von Wacom am USB-Port. Außerdem müssen die Treiber `CONFIG_INPUT_MOUSEDEV` und `CONFIG_INPUT_EVDEV` (s.o.) aktiviert sein.

## 28. Kernel hacking

Magic SysRq key (`CONFIG_MAGIC_SYSRQ`)

Die Option für Kernel-Hacker. Hiermit wird folgendes Verhalten aktiviert: der Benutzer erhält auch nach einem System-Crash noch ein gewisses Maß an Kontrolle und kann z.B. die Puffer auf die Festplatte zurückschreiben und das System wieder hochfahren. Dieses wird ermöglicht durch Drücken von `SysRq`, auf der PC-Tastatur also A und D sowie die entsprechende Taste für den jeweiligen Befehl. Weitere Information für Hacker enthält der Text `Documentation/sysrq.txt`.

# A Die Tastaturbelegung

Falls kein deutscher Tastaturtreiber geladen ist (z.B. während des Urladens von LINUX), ist die Tastatur noch nach dem US-Schema belegt.

Diese Tabelle zeigt in der Spalte „US“ das gesuchte Zeichen und unter „D“ die Taste der deutschen Tastatur, auf der sich dieses Zeichen befindet.

### **Belegung der deutschen Tastatur ohne Treiber**

US	D	US	D	US	D
~		'	^	!	!
@	"	#	§	\$	\$
%	%	^	&	*	(
(	)	)	=	-	?
-	ß	+	`	=	'
{	Ü	[	ü	}	*
]	+	:	Ö	:	ö
"	Ä	'	ä	"	'
\	#	<	;	,	,
>	:	.	.	?	-
/	-		>	\	<
Z	Y	Y	Z		

Tabelle A.1: Gegen-überstellung der US-Tastaturbelegung und des deutschen Layouts

## **B • Kernel-Parameter**

## README- und HOWTO-Dateien

Wir haben nachfolgend alle wichtigen Texte der CD sowie wichtige Adressen aus dem WWW zur Kernel-Konfiguration und Kompilierung tabellarisch aufgeführt. Daß in einigen Fällen auch die C-Quelltexte erwähnt worden sind, liegt daran, daß viele Programmierer nicht einen Readme-Text für den entsprechenden Treiber geschrieben haben, sondern die Dokumentation innerhalb des Quelltextes, meistens am Anfang, vorgenommen haben. Um die Tabellenform nicht zu sprengen, wurden folgende Abkürzungen eingeführt: der Pfad der Verzeichnisse `drivers` und `Documentation` ist `/usr/src/linux/`. Dieses wurde in der Tabelle weggelassen. Außerdem wurde die häufig vorkommende Internet-Adresse `ftp://sunsite.unc.edu:/pub/Linux/docs/HOWTO` in der Tabelle als `ftp://sunsite...` abgekürzt. Diese Adresse ist übrigens für alle Fragen betreffs LINUX, die dieses Buch nicht beantwortet, ein Eldorado.

## README- und HOWTO-Dateien

### 2. Module

`CONFIG_MODVERSIONS`

`ftp://sunsite.unc.edu/pub/LINUX/kernel`

`CONFIG_KMOD`

`Documentation/kmod.txt`

### 3. Prozessoren

`CONFIG_MICROCODE`

`http://www.urbanmyth.org/microcode/`

`CONFIG_MTRR`

`/usr/src/linux/Documentation/mtrr.txt`

`CONFIG_SMP`

`Documentation/SMP.txt`

`Documentation/smp.txt`

### 4. Allgemeine Einstellungen

`CONFIG_VISWS`

`usr/src/linux/Documentation/sgi-visws.txt`

`CONFIG_PCI`

`ftp://sunsite.unc.edu/pub/LINUX/docs/HOWTO`

`CONFIG_MCA`

`/usr/src/linux/Documentation/mca.txt`

CONFIG\_HOTPLUG

<http://www.linux-usb.org/policy.html>

CONFIG\_PCMCIA

<http://www.linuxdoc.org/docs.html#howto>

CONFIG\_BSD\_PROCESS\_ACCT

include/linux/acct.h

CONFIG\_SYSCTL

/usr/src/linux/Documentation/sysctl/

CONFIG\_BINFMT\_MISC

/usr/src/linux/Documentation/binfmt\_misc.txt

CONFIG\_PM

<http://www.cs.utexas.edu/users/kharker/linux-laptop/>

<http://www.linuxdoc.org/docs.html#howto>

CONFIG\_ACPI

<http://www.teleport.com/~acpi/>

Documentation/pm.txt

CONFIG\_APM

/usr/doc/howto/en

## 6. Parallel port support

CONFIG\_PARPORT

/usr/src/linux/Documentation/parport.txt

<http://www.torque.net/linux-pp.html>

CONFIG\_PARPORT\_PC\_FIFO

Documentation/parport.txt

## 7. Plug and Play configuration

CONFIG\_ISAPNP

Documentation/isapnp.txt

## 8. Block Devices

CONFIG\_BLK\_CPQ\_DA

Documentation/cpqarray.txt

CONFIG\_BLK\_CPQ\_CISS\_DA

Documentation/cciss.txt

CONFIG\_BLK\_DEV\_DAC960  
Documentation/README.DAC960

CONFIG\_BLK\_DEV\_NBD  
<http://atrey.karlin.mff.cuni.cz/~pavel>

**9. Multi-device support (RAID and LVM)**

CONFIG\_BLK\_DEV\_MD  
drivers/block/README.md

CONFIG\_MD\_RAID1  
<http://luthien.nuclecu.unam.mx/~miguel/raid>

CONFIG\_MD\_RAID5  
<http://luthien.nuclecu.unam.mx/~miguel/raid>

CONFIG\_BLK\_DEV\_LVM  
/usr/src/linux/Documentation/LVM-HOWTO  
<http://linux.msede.com/lvm>

**10. Networking options**

CONFIG\_NETLINK  
<http://snafu.freedom.org/linux2.2/docs/netlink-HOWTO.html>

CONFIG\_FILTER  
linux/Documentation/networking/filter.txt

CONFIG\_IP\_MULTICAST  
<http://www.best.com/~prince/techinfo/mbone.html>

CONFIG\_IP\_MULTIPLE\_TABLES  
<http://www.compendium.com.ar/policy-routing.txt>  
<http://post.tepkom.ru/pub/vol2/Linux/docs/advanced-routing.tex>  
<ftp://ftp.inr.ac.ru/ip-routing/>

CONFIG\_IP\_ROUTE\_NAT  
<http://www.csn.tu-chemnitz.de/~mha/linux-ip-nat/diplom/nat.html>

CONFIG\_IP\_PNP\_BOOTP  
Documentation/nfsroot.txt

CONFIG\_IP\_PIMSM\_V1  
<http://netweb.usc.edu/pim/>

CONFIG\_IPV6

<http://playground.sun.com/pub/png/html/png-main.html>

CONFIG\_KHTTPD

net/khttpd/README

<http://www.fenrus.demon.nl>

CONFIG\_ATM

Documentation/networking/atm.txt

CONFIG\_IPX\_INTERN

<ftp://metalab.unc.edu/pub/Linux/docs/HOWTO>

CONFIG\_ATALK

<http://artoo.hitchcock.org/~flowerpt/projects/LINUX-netatalk/>

CONFIG\_DECNET

<http://linux.dreamtime.org/decnet/>

Documentation/networking/decnet.txt

CONFIG\_BRIDGE

Documentation/networking/bridge.txt

CONFIG\_X25

/usr/src/linux/Documentation/networking/x25.txt

/usr/src/linux/Documentation/networking/x25-iface.txt

CONFIG\_LAPB

/usr/src/linux/Documentation/networking/lapb-module.txt

CONFIG\_NET\_DIVERT

<http://www.freshmeat.net/projects/etherdivert>

<http://perso.wanadoo.fr/magpie/EtherDivert.html>

CONFIG\_WAN\_ROUTER

/usr/src/linux/Documentation/networking/wan-router.txt

<ftp://ftp.sangoma.com>

CONFIG\_NET\_FASTROUTE

<ftp://ftp.inr.ac.ru/ip-routing/fastroute/fastroute-8390.tar.gz>

CONFIG\_NET\_SCHED

<ftp://ftp.inr.ac.ru/ip-routing>

## 11. Telephony Support

CONFIG\_PHONE\_IXJ

Documentation/telephony/ixj.txt

<http://www.quicknet.net/>

## 12. ATA/IDE/MFM/RLL support

CONFIG\_IDE

Documentation/ide.txt

CONFIG\_BLK\_DEV\_IDEDMA\_PCI

Documentation/ide.txt

CONFIG\_BLK\_DEV\_AEC62XX

drivers/ide/aec62xx.c

CONFIG\_AEC62XX\_TUNING

drivers/ide/aec62xx.c

CONFIG\_BLK\_DEV\_AMD7409

drivers/ide/amd7409.c

CONFIG\_HPT34X\_AUTODMA

drivers/ide/hpt34x.c

CONFIG\_BLK\_DEV\_VIA82CXXX

/usr/src/linux/drivers/block/via82cxx.c

## 13. SCSI support

CONFIG\_SCSI\_DEBUG\_QUEUES

[http://www.andante.org/scsi\\_queue.html](http://www.andante.org/scsi_queue.html)

CONFIG\_SCSI\_LOGGING

/usr/src/linux/drivers/scsi/scsi.c

CONFIG\_BLK\_DEV\_3W\_XXXX\_RAID

<http://www.3ware.com/>

CONFIG\_SCSI\_AIC7XXX

/usr/src/linux/drivers/scsi/README.aic7xxx

CONFIG\_AIC7XXX\_TCQ\_ON\_BY\_DEFAULT

drivers/scsi/README.aic7xxx

CONFIG\_SCSI\_CPQFCTS

drivers/cpqfc.Readme

CONFIG\_SCSI\_IBMMCA

Documentation/mca.txt

CONFIG\_SCSI\_IPS

<http://www.developer.ibm.com/welcome/netfinity/serveraid.html>

CONFIG\_SCSI\_NCR53C406A

drivers/scsi/NCR53c406.c

## 16. Network device support

CONFIG\_ARCNET

/usr/src/linux/Documentation/networking/arcnet.txt

CONFIG\_LTPC

Documentation/networking/ltpc.txt

CONFIG\_COPS

Documentation/networking/cops.txt

CONFIG\_IPDDP

Documentation/networking/ipddp.txt

CONFIG\_TUN

Documentation/networking/tuntap.txt

CONFIG\_ETHERTAP

/usr/src/linux/Documentation/networking/ethertap.txt

CONFIG\_NET\_SB1000

<http://www.jacksonville.net/~fventuri/>

<http://home.adelphia.net/~siglercm/sb1000.html>

<http://linuxpower.cx/~cable/>

CONFIG\_EPLUS

Documentation/networking/3c505.txt

CONFIG\_VORTEX

Documentation/networking/vortex.txt

drivers/net/3c59x.c

CONFIG\_SMC9194

drivers/net/README.smc9

CONFIG\_SK\_G16

drivers/net/sk\_g16.c

CONFIG\_AC3200

drivers/net/cs3200.

```
CONFIG_APRICOT
 drivers/net/apricot

CONFIG_DEC_ELCP
 /usr/src/linux/Documentation/networking/tulip.txt

CONFIG_DE4X5
 /usr/src/linux/Documentation/networking/de4x5.txt

CONFIG_DGRS
 /usr/src/linux/Documentation//networking/dgrs.txt

CONFIG_DM9102
 http://www.davicom.com.tw

CONFIG_NATSEMI
 http://www.scyld.com/network/natsemi.html

CONFIG_8139TOO
 Documentation/networking/8139too.txt

CONFIG_SIS900
 Documentation/networking/sis900.txt

CONFIG_SUNDANCE
 http://www.scyld.com/network/sundance.html

CONFIG_WINBOND_840
 http://www.scyld.com/network/drivers.html

CONFIG_ATP
 drivers/net/atp.c

CONFIG_DE600
 drivers/net/README.DLINK

CONFIG_DE620
 drivers/net/README.DLINK

CONFIG_YELLOWFIN
 http://cesdis.gsfc.nasa.gov/linux/drivers/yellowfin.html

CONFIG_SK98LIN
 Documentation/networking/sk98lin.txt

CONFIG_SKFP
 Documentation/networking/skfp.txt
```

• Kernel-Parameter

---

9

CONFIG\_PPP

Documentation/networking/ppp.txt

CONFIG\_PPPOE

<http://www.math.uwaterloo.ca/~mostrows>

CONFIG\_STRIP

<http://mosquitonet.stanford.edu>

CONFIG\_WAVELAN

/usr/src/linux/Documentation/networking/wavelan.txt

[http://www.hpl.hp.com/personal/Jean\\_Tourrilhes/Linux/Wavelan.html](http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Wavelan.html)

CONFIG\_ARLAN

[www.aironet.com](http://www.aironet.com)

[www.Telxon.com](http://www.Telxon.com)

<http://www.ylenurme.ee/~elmer/655>

CONFIG\_AIRONET4500

[www.aironet.com](http://www.aironet.com)

CONFIG\_IBMTR

<ftp://metalab.unc.edu/pub/Linux/docs/HOWTO>

CONFIG\_IBMOL

<ftp://metalab.unc.edu/pub/Linux/docs/HOWTO>

<http://www.linuxtr.net>

CONFIG\_TMS380TR

Documentation/networking/tms380tr.txt

<http://www.auk.cx/tms380tr/>

CONFIG\_SMCTR

Documentation/networking/smctr.txt

CONFIG\_SHAPER

Documentation/networking/shaper.txt

CONFIG\_COSA

<http://www.fi.muni.cz/~kas/cosa>

CONFIG\_COMX

Documentation/networking/comx.txt

<http://www.itc.hu>

CONFIG\_COMX\_HW\_COMX  
ftp://ftp.itc.hu/  
CONFIG\_DLCI  
<http://www.frforum.com>  
</usr/src/linux/Documentation/networking/framerelay.txt>  
CONFIG\_WAN\_DRIVERS  
</usr/src/linux/Documentation/networking/wan-router.txt>  
CONFIG\_VENDOR\_SANGOMA  
<http://www.sangoma.com>  
</usr/src/linux/Documentation/networking/wanpipe.txt>  
CONFIG\_CYCLADES\_SYNC  
<http://www.cyclades.com>  
CONFIG\_SBNI  
<http://www.granch.ru>  
CONFIG\_PCMCIA\_RAYCS  
Documentation/networking/ray\_cs.txt  
CONFIG\_ATM\_TCP  
<http://icawww1.epfl.ch/linux-atm/>  
CONFIG\_ATM\_IA  
[www.ipphase.com/products/ClassSheet.cfm?ClassID=ATM](http://www.ipphase.com/products/ClassSheet.cfm?ClassID=ATM)  
Documentation/networking/ipphase.txt  
CONFIG\_ATM\_FORE200E\_MAYBE  
Documentation/networking/fore200e.txt

**17. Amateur Radio support**

CONFIG\_HAMRADIO  
<http://www.tapr.org/tapr/html/pkthome.html>

CONFIG\_AX25  
</usr/src/linux/Documentation/networking/ax25.txt>

CONFIG\_6PACK  
</usr/src/linux/Documentation/networking/6pack.txt>

CONFIG\_DMASCC  
<http://hydra.carleton.ca/info/pi2.html>

<http://www.paccomm.com/gracilis.html>  
[ist:<http://www.nt.tuwien.ac.at/~kkudielk/Linux/>](http://www.nt.tuwien.ac.at/~kkudielk/Linux/)

CONFIG\_SCC

/usr/src/linux/Documentation/networking/z8530drv.txt

CONFIG\_SCC\_DELAY

/usr/src/linux/Documentation/networking/z8530drv.txt

CONFIG\_SCC\_DELAY

Documentation/networking/z8530drv.txt

CONFIG\_BAYCOM\_SER\_FDX

/usr/src/linux/Documentation/networking/baycom.txt

<http://www.baycom.de>

## 18. IrDA (infrared) support

CONFIG\_IRDA

/usr/src/linux/Documentation/networking/irda.txt

CONFIG\_IRCOMM

<http://www.pluto.dti.ne.jp/~thiguchi/irda/>

## 19. ISDN subsystem

CONFIG\_ISDN

Documentation/isdn/README

CONFIG\_ISDN\_PPP

Documentation/isdn/README.syncppp

Documentation/isdn/syncPPP.FAQ

CONFIG\_ISDN\_MPP

Documentation/isdn/README.syncppp

CONFIG\_ISDN\_AUDIO

Documentation/isdn/README.audio

CONFIG\_ISDN\_TTY\_FAX

/usr/src/linux/Documentation/isdn/README.fax

CONFIG\_ISDN\_X25

/usr/src/linux/Documentation/isdn/README.x25

CONFIG\_ISDN\_DRV\_HISAX

/usr/src/linux/Documentation/isdn/README.HiSax

CONFIG\_ISDN\_DRV\_ICN  
    Documentation/isdn/README  
    Documentation/isdn/README.icn

CONFIG\_ISDN\_DRV\_PCBIT  
    Documentation/isdn/README  
    Documentation/isdn/README.pcbit

CONFIG\_ISDN\_DRV\_SC  
    <http://www.spellcast.com>  
    /usr/src/linux/Documentation/isdn/README.sc

CONFIG\_ISDN\_DRV\_ACT2000  
    /usr/src/linux/Documentation/isdn/README.act2000

CONFIG\_ISDN\_DRV\_EICON  
    /usr/src/linux/Documentation/isdn/README.eicon

CONFIG\_ISDN\_CAPI  
    <http://www.capi.org/>

CONFIG\_HYSDN  
    Documentation/isdn/README.hysdn

**20. Old CD-ROM drivers (not SCSI, not IDE)**

CONFIG\_BPCD  
    Documentation/cdrom/bpcd

CONFIG\_MCDX  
    Documentation/cdrom/mcdx

CONFIG\_ISP16\_CDI  
    Documentation/cdrom/isp16

CONFIG\_CDU31A  
    Documentation/cdrom/cdu31a

**22. Character devices**

CONFIG\_COMPUTONE  
    Documentation/computone.txt

CONFIG\_DIGIEPCA  
    Documentation/digiepca.txt

CONFIG\_ESPSERIAL

/usr/src/linux/Documentation/hayes-esp.txt

CONFIG\_RISCOM8

Documentation/riscom8.txt

CONFIG\_SPECIALIX

/usr/src/linux/Documentation/specialix.txt

CONFIG\_SPECIALIX\_RTSCSTS

/usr/src/linux/Documentation/specialix.txt

CONFIG\_SX

Documentation/sx.txt

CONFIG\_STALLION

drivers/char/README.stallion

CONFIG\_ISTALLION

drivers/char/README.stallion

CONFIG\_I2C

Documentation/i2c/summary

CONFIG\_I2C\_CHARDEV

Documentation/i2c/dev-interface

CONFIG\_PC110\_PAD

<http://toy.cabi.net>

CONFIG\_JOYSTICK

/usr/src/linux/Documentation/joystick.txt

CONFIG\_JOY\_ANALOG

/usr/src/linux/Documentation/joystick.txt

CONFIG\_QIC02\_DYNCONF

include/LINUX/tpqic02.h

<ftp://titius.cfw.com/pub/Linux/util/>

CONFIG\_WATCHDOG

Documentation/watchdog.txt

CONFIG\_PCWATCHDOG

<http://www.berkprod.com>

CONFIG\_RTC

Documentation/rtc.txt

```
CONFIG_DTLK
 http://www.rcsys.com/
CONFIG_APPLICOM
 http://www.applibcom-int.com/
CONFIG_FTape
 /usr/src/linux/drivers/char/ftape/README.PCI
 http://www-math.math.rwth-aachen.de/~LBFM/claus/ftape/
 /usr/src/linux/Documentation/ftape.txt
CONFIG_AGPG_INTEL
 http://utah-glx.sourceforge.net/
CONFIG_AGPG_VIA
 http://utah-glx.sourceforge.net/
CONFIG_AGPG_AMD
 http://utah-glx.sourceforge.net/
CONFIG_AGPG_SIS
 http://utah-glx.sourceforge.net/
CONFIG_AGPG_ALI
 http://utah-glx.sourceforge.net/
CONFIG_DRM
 http://dri.sourceforge.net

23. Multimedia devices

CONFIG_VIDEO_DEV
 ftp://ftp.uk.linux.org/pb/linux/video4linux
 /usr/src/linux/Documentation/video4linux/API.html
CONFIG_VIDEO_BT848
 /usr/src/linux/Documentation/video4linux/bttv
CONFIG_VIDEO_CPIA
 Documentation/video4linux/README.cplia
CONFIG_VIDEO_ZR36120
 Documentation/video4linux/zr36120.txt
CONFIG_RADIO_CADET
 http://linux.blackhawke.net/cadet.html
```

<http://roadrunner.swansea.uk.linux.org/v4l.shtml>

CONFIG\_RADIO\_RTRACK

<http://roadrunner.swansea.uk.linux.org/v4l.shtml>

/usr/src/linux/Documentation/video4linux/radiotrack.txt

CONFIG\_RADIO\_RTRACK2

<http://roadrunner.swansea.uk.linux.org/v4l.shtml>

/usr/src/linux/Documentation/video4linux/radiotrack.txt

CONFIG\_RADIO\_AZTECH

<http://roadrunner.swansea.uk.linux.org/v4l.shtml>

CONFIG\_RADIO\_GEMTEK

<http://roadrunner.swansea.uk.linux.org/v4l.shtml>

CONFIG\_RADIO\_MIROPCM20

<http://roadrunner.swansea.uk.linux.org/v4l.shtml>

CONFIG\_RADIO\_SF16FMI

<http://roadrunner.swansea.uk.linux.org/v4l.shtml>

CONFIG\_RADIO\_TRUST

<http://roadrunner.swansea.uk.linux.org/v4l.shtml>

CONFIG\_RADIO\_TYphoon

<http://roadrunner.swansea.uk.linux.org/v4l.shtml>

CONFIG\_RADIO\_ZOLTRIX

<http://roadrunner.swansea.uk.linux.org/v4l.shtml>

## 24. File systems

CONFIG\_AUTOFS\_FS

<ftp://ftp.kernel.org/pub/linux/daemons/autofs>

CONFIG\_AUTOFS4\_FS

<ftp://ftp.kernel.org/pub/linux/daemons/autofs/testing-v4>

CONFIG\_BFS\_FS

Documentation/filesystems/bfs.txt

CONFIG\_UMSDOS\_FS

Documentation/filesystems/umsdos.txt

CONFIG\_VFAT\_FS

Documentation/filesystems/vfat.txt

```
CONFIG_EFS_FS
 http://aeschi.ch.eu.org/efs
CONFIG_JFFS_FS
 http://www.developer.axis.com/software/jffs
CONFIG_CRAMFS
 Documentation/filesystems/cramfs.txt
 fs/cramfs/README
CONFIG_NTFS_FS
 /usr/src/linux/Documentation/filesystems/ntfs.txt
CONFIG_HPFS_FS
 Documentation/filesystems/hpfs.txt
CONFIG_DEVFS_FS
 Documentation/filesystems/devfs/README
CONFIG_DEVFS_DEBUG
 Documentation/filesystems/devfs/boot-options
CONFIG_ROMFS_FS
 /usr/src/linux/Documentation/filesystems/romfs.txt
CONFIG_SYSV_FS
 Documentation/filesystems/sysv-fs.txt
CONFIG_UDF_FS
 Documentation/filesystems/udf.txt
CONFIG_CODA_FS
 /usr/src/linux/Documentation/filesystems/coda.txt
 http://www.coda.cs.cmu.edu
CONFIG_ROOT_NFS
 Documentation/nfsroot.txt
CONFIG_SMB_FS
 Documentation/filesystems/smbfs.txt
CONFIG_SMB_NLS_DEFAULT
 http://www.hojdpunkten.ac.se/054/samba/index.html
CONFIG_NCP_FS
 Documentation/filesystems/ncpfs.txt
```

**25. Console drivers**

CONFIG\_VGA\_CONSOLE

ftp://metalab.unc.edu/pub/Linux/utils/console

CONFIG\_FB

/usr/src/linux/Documentation/fb/framebuffer.txt

<http://www.tahallah.demon.co.uk/programming/prog.html>

CONFIG\_FB\_CLGEN

Documentation/fb/clgenfb.txt

CONFIG\_FB\_VESA

/usr/src/linux/Documentation/fb/vesafb.txt

CONFIG\_FB\_MATROX

/usr/src/linux/drivers/video/matroxfb.c

/usr/src/linux/Documentation/fb/matroxfb.txt

CONFIG\_FB\_MATROX\_MAVEN

<ftp://platan.vc.cvut.cz/pub/linux/matrox-latest>

CONFIG\_FB\_ATY128

Documentation/fb/aty128fb.txt

**26. Sound**

CONFIG\_SOUND

Documentation/sound/README.modules

<ftp://ftp.infradead.org/pub/pcsp/>

CONFIG\_SOUND\_CMPCI

/Documentation/sound/CMI8338

CONFIG\_SOUND\_ES1370

/usr/src/linux/Documentation/sound/es1370

CONFIG\_SOUND\_ES1371

/usr/src/linux/Documentation/sound/es1371

CONFIG\_SOUND\_ESSSOLO1

/usr/src/linux/Documentation/sound/solo1

CONFIG\_SOUND\_MAESTRO

Documentation/sound/Maestro

CONFIG\_SOUND\_SONICVIBES

```
/usr/src/linux/Documentation/sound/sonicvibes
CONFIG_SOUND_MSNDCLAS
 Documentation/sound/MultiSound
CONFIG_MSNDCLAS_INIT_FILE
 Documentation/sound/MultiSound
CONFIG_MSNDCLAS_PERM_FILE
 Documentation/sound/MultiSound
CONFIG_SOUND_MSNDPIN
 Documentation/sound/MultiSound
CONFIG_MSNDPIN_INIT_FILE
 Documentation/sound/MultiSound
CONFIG_MSNDPIN_PERM_FILE
 Documentation/sound/MultiSound
CONFIG_SOUND_OSS
 http://www.opensound.com/oss.html
CONFIG_SOUND_AD1816
 /usr/src/linux/Documentation/sound/AD1816
CONFIG_SOUND_CS4232
 Documentation/sound/CS4232
CONFIG_SOUND_GUS
 Documentation/sound/ultrasound
CONFIG_SOUND_MPU401
 drivers/sound/Readme.cards
CONFIG_SOUND_NM256
 /usr/src/linux/Documentation/sound/NM256
CONFIG_SOUND_PAS
 Documentation/sound/PAS16
CONFIG_SOUND_PSS
 Documentation/sound/PSS
CONFIG_PSS_MIXER
 Documentation/sound/PSS
CONFIG_SOUND_SB
```

drivers/sound/Readme.cards  
CONFIG\_SOUND\_AWE32\_SYNTH  
    /usr/src/linux/Documentation/sound/README.awe  
    /usr/src/linux/Documentation/sound/AWE32  
CONFIG\_SOUND\_WAVEFRONT  
    Documentation/sound/Wavefront  
    Documentation/sound/Tropez+  
CONFIG\_SOUND\_YM3812  
    Documentation/sound/OPL3  
CONFIG\_SOUND\_OPL3SA1  
    Documentation/sound/OPL3-SA  
CONFIG\_SOUND\_OPL3SA2  
    Documentation/sound/OPL3-SA2  
CONFIG\_SOUND\_AEDSP16  
    /usr/src/linux/drivers/sound/lowlevel/README.aedsp16  
    /usr/src/linux/drivers/sound/lowlevel/aedsp16.c  
    /usr/src/linux/Documentation/sound/AudioExcelDSP16

## 27. USB support

CONFIG\_USB  
    Documentation/usb/  
        Documentation/usb/usb-help.txt  
CONFIG\_USB\_DEVICEFS  
    Documentation/usb/proc\_usb\_info.txt  
CONFIG\_USB\_OHCI  
    Documentation/usb/ohci.txt  
CONFIG\_USB\_SCANNER  
    Documentation/usb/scanner.txt  
    Documentation/usb/scanner-hp-sane.txt  
CONFIG\_USB\_MICROTEK  
    <http://fachschaft.cup.uni-muenchen.de/~neukum/scanner.html>  
CONFIG\_USB\_SERIAL  
    Documentation/usb/usb-serial.txt

CONFIG\_USB\_SERIAL\_GENERIC  
Documentation/usb/usb-serial.txt

CONFIG\_USB\_SERIAL\_VISOR  
<http://usbvisor.sourceforge.net>

CONFIG\_USB\_SERIAL\_FTDI\_SIO  
[http://reality.sgi.com/bryder\\_wellington/ftdi\\_sio](http://reality.sgi.com/bryder_wellington/ftdi_sio)

CONFIG\_USB\_SERIAL\_KEYSPAN  
<http://www.linuxcare.com.au/hugh/keyspan.html>

CONFIG\_USB\_IBMCAM  
Documentation/usb/ibmcam.txt  
<http://roadrunner.swansea.uk.linux.org/v4l.shtml>

CONFIG\_USB\_OV511  
Documentation/usb/ov511.txt

CONFIG\_USB\_DC2XX  
<http://www.gphoto.org>  
Documentation/usb/dc2xx.txt

CONFIG\_USB\_MDC800  
<http://www.gphoto.org>

CONFIG\_USB\_DABUSB  
<http://dab.in.tum.de>  
Documentation/usb/URB.txt

CONFIG\_USB\_PEGASUS  
Documentation/usb/rio.txt

CONFIG\_USB\_BLUETOOTH  
<http://developer.axis.com/software/index.shtml>

CONFIG\_USB\_NET1080  
drivers/usb/net1080.c

CONFIG\_USB\_HID  
Documentation/usb/input.txt

**28. Kernel hacking**

CONFIG\_MAGIC\_SYSRQ  
Documentation/sysrq.txt

## C • HTML-Sonderzeichen

Sonderzeichen	HTML	Sonderzeichen	HTML
>	&gt;	"	&quot;
<	&lt;	®	&reg;
©	&copy;	À	&Agrave;
Á	&Aacute;	Â	&Acirc;
Ä	&Auml;	Å	&Aring;
Æ	&Aelig;	Ç	&Ccedil;
È	&Egrave;	É	&Eacute;
Ê	&Ecirc;	Ì	&Igrave;
Í	&iacute; Í	&lcirc;	
Ї	&luml;	Ñ	&Ntilde;
Ò	&Ograve;	Ó	&Oacute;
Ӧ	&Otilde;	Ӧ	&Ouml;
Ø	&Oslash;	Ù	&Uacute;
Ú	&Ugrave;	Û	&Ucirc;
Ü	&Uuml;	Ԃ	&szlig;
á	&aacute; à	&agrave;	
â	&acirc;	ã	&atilde;
ä	&auml;	å	&aring;
æ	&aelig;	ç	&ccedil;
è	&egrave; é	&eacute;	
ê	&ecirc;	ë	&euml;
ѝ	&igrave; í	&iacute;	

• HTML-Sonderzeichen

3

---

î	&icirc;	ï	&iuml;
ñ	&ntilde;	ó	&oacute;
ò	&ograve;ô		&ocirc;
õ	&otilde;	ö	&ouml;
ø	&oslash;	ù	&ugrave;
ú	&uacute;û		&ucirc;
ü	&uuml;	ÿ	&yuml;
í	&iexcl;	¢	&cent;
£	&pound;¤		&curren;
¥	&yen;		&brvbar;
§	&sect;	«	&laquo;
»	&raquo;	°	deg;
±	&plusmn;	²	&sup2;
³	&sup3;	'	&acute;
µ	&micro;	•	&middot;
,	&cedil;	½	&frac12;
¼	&frac14;¾		&frac34;
¿			&quest;

## D • Die GPL

In diesem Anhang finden Sie einen vollständigen Abdruck der *GPL (General Public License)* – der lizenzierten Bestimmungen also, denen die Software der *FSF (Free Software Foundation)* unterliegt:

#### GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 675 Mass Ave, Cambridge, MA 02139, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

#### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

#### GNU GENERAL PUBLIC LICENSE

#### TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribu-

tion of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

**NO WARRANTY**

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. END OF TERMS AND CONDITIONS Appendix: How to Apply These Terms to Your New Programs. If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.> Copyright (C) 19yy <name of author>

This program 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 program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) 19yy name of author Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type 'show w'. This is free software, and you are welcome to redistribute it under certain conditions; type 'show c' for details.

The hypothetical commands 'show w' and 'show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than 'show w' and 'show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program 'Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989 Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

## **E • Glossar**

Dieses Kapitel bietet ein umfangreiches Glossar häufig in der UNIX-Welt verwendeter Begriffe mit Kurzdefinitionen zu diesen.

Querverweise sind durch das Zeichen „→“ symbolisiert.

## A

**ABI** – *Application Binary Interface*. Eine Spezifikation, die darauf hinzielt, Binärkompatibilität zwischen Anwendungen zu gewährleisten, die auf derselben Prozessorfamilie oder denselben Prozessoren unter UNIX System V Release 4 laufen.

**Account** – Das „Konto“, welches jeder →User eines Multiuser-→Betriebssystems oder einer Netzwerkumgebung benötigt, um mit diesem arbeiten zu können.

**Adresse** – Bei einer Adresse kann es sich handeln um:

- Den eindeutig lokalisierbaren Bereich im Speicher oder auf einem Datenträger
- Den Bestimmungsort einer →E-Mail-Nachricht
- Eine in der Datei /etc/hosts spezifizierte →IP-Adresse
- Eine →Internet-→Adresse

**AFS** – *Andrews file system*. Ein Programm für die Verwaltung nicht-lokalen Dateien innerhalb eines Systems, welches →TCP/IP verwendet.

**anonymes FTP** – Ein Verfahren, um mit →ftp auf einen →Internet-→Host zuzugreifen, ohne auf diesem einen →Account zu besitzen, um beispielsweise Dateien von diesem herunterzuladen.

**ANSI** – *American National Standard Institute*. US-Organisation zur Entwicklung von Standards.

**API** – *Application programming interface*. Funktionen, die von einem Betriebssystem zur Verfügung gestellt werden, damit andere Programme diese beispielsweise für die Dateiverwaltung nutzen können.

**Applikation** – Ein Programm, wie z. B. ein Textverarbeitungsprogramm oder eine Tabellenkalkulation; dt. Anwendung.

**Archie** – Ein im →Internet zur Verfügung gestelltes Suchsystem für die Lokalisierung von durch →anonymes FTP bereitgestellten Dateien.

**Archiv** – Eine Zusammenfassung von Dateien zur Dateisicherung oder vereinfachten Übertragung.

**Argument** – Zusätzliche Information, die einem Befehl in der Befehlszeile übergeben werden muss/kann, damit dieser dieses weiterbearbeitet.

**Argumentliste** – Alle in einer Befehlszeile übergebenen →Argumente.

**ASCII** – *American Standard Code for Information Interchange*. Ein Codierungsschema, das die einzelnen für einen Rechner zur Verfügung stehenden Zeichen (Buchstaben, Ziffern, Interpunktionszeichen etc.) eindeutigen binären Werten zuordnet; populär in der PC-Welt.

**AT&T** – Die Muttergesellschaft der Bell Laboratories, den ursprünglichen Entwicklern von UNIX.

**Athena-Projekt** – Ein von MIT zwischen 1983 und 1991 laufendes Projekt, in dem u. a. die Grundlagen für die Entwicklung des X Window-Systems geschaffen wurden.

**Ausgabeumlenkung** – Verfahren, bei dem die Ausgabe von der Standardausgabe (meist ist hiermit der Bildschirm gemeint) in eine Datei oder auf ein anderes Gerät umgelenkt wird.

**ausloggen** – Das Beenden einer Arbeitssitzung in einem UNIX-System (→einloggen).

**Authentication** – Feststellen der Zugriffsrechte eines →Users auf ein Computersystem oder →Netzwerk.

**Avatar** – Ein anderer Name für den →Superuser.

**awk** – Eine UNIX-Programmiersprache für die Manipulation von Textdateien.

## B

**Background** – Eine →Prozess-Umgebung, in der Programme ohne Eingabe des →Users im Hintergrund laufen.

**Backspace** – Eine auf den meisten Tastaturen durch das Drücken einer bestimmten Taste ausführbares (Steuer-) Zeichen, das den →Cursor um eine Position nach links verschiebt, wobei das an dieser Position stehende Zeichen gelöscht wird.

**Backup** – Aktuelle Kopie der Dateien eines Computersystems, welche zu Sicherungszwecken angefertigt wird.

**Bandweite** – Die Übertragungskapazität eines Computers oder Kommunikationskanals, gemessen in Bit pro Sekunde (bps), Kilobit pro Sekunde (Kbps), Megabit pro Sekunde (Mbps) oder ein vielfaches.

**bash** – *Bourne-again shell*. Populärer →Befehlszeileninterpreter, der in vielen LINUX-Systemen als Standard-Interpreter verwendet wird.

**Befehlssatz** – Ein Satz von Instruktionen der Maschinensprache, die ein Prozessor erkennen und verarbeiten kann.

**Befehlszeileninterpreter** – Programm, das die Benutzereingaben auf syntaktische Korrektheit überprüft und in eine für das →Betriebssystem verständliche Form übersetzt.

**Benchmark** – Ein spezifizierter Standard, mit dem die Leistungsfähigkeit eines Computersystems gemessen werden kann.

**Benutzer** – →User

**Betriebssystem** – Die Software, die für die Verwaltung von Systemressourcen – wie Speicher, Festplatte, Prozessorzeit – und Peripheriegeräten wie Drucker, Monitore usw. verantwortlich ist. Es bietet somit die grundlegenden Funktionen, mit einem Computer überhaupt arbeiten zu können.

**binär** – Jedes Prinzip, das zwei verschiedene Zustände kennt.

**Binärdatei** – Eine Programm- oder Datendatei, die ihre Informationen in binärer, d. h. maschinenlesbarer, Form enthält.

**Bit** – *Binary digit*. Die kleinste Informationseinheit eines binären Zahlensystems mit den Zuständen 0 und 1 („aus“ und „an“).

**Blank character** – Ein Leer- oder Tabulatorzeichen (dt. Trennzeichen).

**Blind courtesy copy** – Abk. bcc. Eine Liste von Empfängern von →E-Mail-Nachrichten, wobei der „Original-Adressat“ nichts davon erfährt, dass Kopien der an ihn gesendeten Nachricht verschickt wurden.

**Block** – Zusammenhängender Bereich eines Datenträgers für die Aufnahme von Daten.

**Block device** – Ein Peripheriegerät, das dazu in der Lage ist, ein UNIX-Dateisystem zu speichern, z. B. Disketten- oder Bandlaufwerke.

**Block special file** – UNIX-Datei, welche die Schnittstelle zu einem →Block device bereitstellt.

**BNC-Connector** – Anschluss-Stecker für Koaxialkabel (→Thin-Ethernet).

**Bookmark** – „Lesezeichen“, das in einem Dokument platziert werden kann, um später schnell und einfach an diese Stelle zurückzukehren.

**Boot-Datenträger** – Ein Datenträger (meist Diskette oder Festplatte), von dem das →Betriebssystem geladen wird.

**Booting** – Der Vorgang des „Hochfahrens“ eines Computers, d. h. der →Hardware-Diagnose und des Ladens des →Betriebssystems.

**Bootstrap-Programm** – Ein Programm, welches das →Booting eines Computers einleitet.

**bounce** – Das Zurückschicken einer →E-Mail an den Absender, da beim Zustellen ein Fehler aufgetreten ist. Auch: der Vorgang, ein Computersystem vom →Multi-User- in den →Single-User-Modus zu stellen.

**Bourne-Shell** – Der älteste, auch heute noch in vielen UNIX-Systemen verwendete →Befehlszeileninterpreter.

**bpi** – Die Anzahl von →Bits, die ein Datenträger pro Zoll abspeichern kann (*bits per inch*).

**bps** – Die Anzahl von →Bits, die pro Sekunde während eines Datentransfers übertragen werden (*bits per second*).

**Browser** – Eine Anwendung zum Durchsuchen von →Internet-Ressourcen.

**BSD** – *Berkeley Software Distribution*. Eine der beiden UNIX-Hauptrichtungen (neben dem UNIX von AT&T).

**Buffer** – Ein hard- oder softwaremäßig reservierter Speicherbereich zum temporären Speichern von Daten; entsprechend wird für diesen Vorgang das Verb „buffern“ verwendet.

**Buffered I/O** – Gepufferte Ein-/Ausgabe, die in einem/von einem temporären Arbeitsbereich erfolgt.

**Bus** – Elektronische Leitung eines Computersystems.

**Byte** – *Binary eight digit*. Gruppierung von acht →Bits, die im Normalfall ein Zeichen repräsentiert.

## C

**C** – Mehrzweck-Programmiersprache, ohne welche die Entwicklung von UNIX (und damit LINUX) in dieser Form nicht möglich gewesen wäre.

**C-Bibliothek** – Engl. C library. Eine Sammlung von Standardfunktionen, die einem →C-Programmierer zur Verfügung steht.

**C-Compiler** – Ein Programm, das die Programmieranweisungen der Sprache →C in ein maschinenlesbares Programm überführt.

**C-Shell** – Populärer →Befehlszeileninterpreter, der im Rahmen der Entwicklung von →BSD-UNIX entwickelt wurde.

**Cache** – Eine Reihe von →Buffern, die dazu verwendet werden, Daten eines →Block device zwischenzuspeichern, so dass diese nicht von einem Programm neu vom Datenträger eingelesen werden müssen.

**Cache Memory** – Sehr schneller Zwischenspeicher, der für das temporäre Speichern von Daten oder Befehlen, die wahrscheinlich als Nächstes vom Prozessor benötigt werden, verwendet wird.

**Caddy** – Ein flacher Kunststoffeinschub, der zum Laden einer CD in ein →CD-ROM-Laufwerk dient.

**CAE** – *Common Application Environment*. Eine Reihe von Standards, die für die Anwendungsentwicklung entworfen wurden.

**Call** – In der Programmierung: ein Ausdruck, der sich an eine unabhängige Subroutine oder Funktion wendet (sie „aufruft“).

**Calling Environment** – Eine Liste der Variablen und ihrer Werte, die einem Programm zur Verfügung stehen.

**Carriage Return** – Das Steuerungszeichen (^M), das den Druckkopf oder die Zeilenmarke anweist, zur ersten Position der aktuellen Zeile zurückzukehren.

**CD-ROM** – *Compact disc, read-only memory*. Optisches Speichermedium zur Speicherung großer Mengen an Informationen (bis zu 700 MByte).

**CDE** – *Common Desktop Environment*. Eine Reihe von Spezifikationen, die eine →API für eine grafische Benutzerschnittstelle unter UNIX definiert.

**CERN** – *Conseil Européene pour la Recherche Nucléaire*. Europäisches Labor für Partikelphysik in Genf. Geburtsstätte des →World Wide Web.

**CFV** – *Call for Votes*. Eine Aufforderung an die →USENET-Newsgroup news.announce.newsgroups, eine Abstimmung über die Einrichtung einer neuen →Newsgroup vorzunehmen.

**Character Device** – Ein Gerät, das ein Zeichen zur Zeit anzeigt, verarbeitet oder speichert, wie z. B. ein Terminal, Drucker oder bestimmte Arten der Kommunikationsverbindung.

**Character I/O** – Character Input/Output. Jegliche Form der Kommunikation mit einem Peripheriegerät (Drucker, Terminal etc.), bei der die Informationen byteweise ohne Pufferung übertragen werden.

**Character Set** – Zeichensatz, der für die Verwendung auf einem Computer zur Verfügung steht (Buchstaben, Ziffern, Interpunktions- und Steuerzeichen etc.).

**Character Special File** – Eine UNIX-Datei, die eine Schnittstellendefinition zu einem →Character Device bietet.

**Character Terminal** – Ein Terminal, das nur Buchstaben, Ziffern und Interpunktionszeichen darstellen kann.

**Chat** – Interaktive Kommunikation mehrerer →User über ein →Netzwerk.

**Checksum** – Prüfsumme zur Gewährleistung einer fehlerfreien Datenübertragung.

**Child Process** – Ein →Prozess, der von einem anderem →Prozess (Elternprozess) aufgerufen wurde (dt. Kindprozess; auch Subprocess oder Unterprozess genannt).

**Chip** – Halbleiterschaltkreis, der eine Reihe von elektronischen Komponenten auf sich vereinigt.

**CISC** – *Complex instruction set computing*. Bestimmter Prozessortyp, der neben anderen Merkmalen z. B. über einen →Befehlssatz von über 100 Befehlen verfügt.

**Client** – Ein Gerät oder Programm, das Gebrauch von den von einem →Server zur Verfügung gestellten Diensten macht.

**Client/Server** – Eine Verarbeitungsarchitektur, die eine Aufgabenverteilung zwischen →Clients und →Servern eines →Netzwerks ermöglicht.

**Collision** – Der Versuch zweier →Nodes innerhalb eines →Netzwerks, gleichzeitig auf demselben Kanal zu senden.

**Core Dump** – Ein vom UNIX-Kernel unter bestimmten fehlerbezogenen Umständen generiertes Listing, das Auskünfte über den Speicherzustand während des Auftretens des Fehlers gibt.

**COSE** – *Common Open Software Environment*. Industriegruppe, die sich die Entwicklung einer standardisierten grafischen Benutzerschnittstelle (→GUI) zur Aufgabe gemacht hat.

**CPI** – Die Anzahl der Zeichen eines Schrifttyps, die auf einem Zoll (*characters per inch*) Platz finden.

**CPS** – Die Anzahl der Zeichen, die pro Sekunde (*characters per second*) während eines Datentransfers übertragen werden.

**CPU** – *Central processing unit*. Der Rechen- und Steuerbaustein eines Computers (Zentralprozessor).

**Crash** – Unerwarteter Systemhalt; nach diesem ist keine weitere Arbeit mit dem System mehr möglich.

**Cross Posting** – Einen Artikel an mehr als eine →Newsgroup innerhalb des →USENET verschicken.

**Cursor** – Schreibmarke

**Cut-and-Paste** – Das Ausschneiden eines markierten Bereichs eines Dokuments in einen temporären Speicher, aus dem er dann an einer anderen Stelle des Dokuments oder in ein anderes Dokument eingefügt werden kann.

## D

**Daemon** – Dt. Dämon. Ein UNIX-Programm, das im Hintergrund wichtige Systemdienste verrichtet, ohne dass weitere Eingaben notwendig sind.

**Dateisystem** – Verfahren zur Verwaltung von Informationen.

**Dateitypen** – Unterschiedliche Dateien innerhalb eines →Dateisystems; hierzu gehören:

- einfache Dateien – diese enthalten Daten und Programme des →Users
- Verzeichnisse – diese enthalten Dateien oder weitere →Verzeichnisse
- blockorientierte Dateien →Block special files, die Gerätetreiber für die Kommunikation mit Laufwerken bereitstellen
- zeichenorientierte Dateien →Character special files, die zeichenweise mit der →Hardware kommunizieren
- FIFO-Dateien – Dateien, die es Programmen, die nicht miteinander in Beziehung stehen, erlauben, Daten miteinander auszutauschen
- Sockets – Dateien, die es Programmen, die nicht in Beziehung zueinander stehen, erlauben, Informationen auszutauschen
- Symbolische Links – Dateien, die es erlauben, Dateien aus verschiedenen →Dateisystemen zu verbinden, so dass sie über einen Namen angesprochen werden können

**Dateizugriffsberechtigung** – Verfahren, mit dem festgelegt wird, welcher →User welche Rechte in der Handhabung von Dateien hat. Diese Rechte werden unterschieden nach:

- Leseberechtigung – Diese erlaubt es, eine Datei zum Lesen zu öffnen.
- Schreibberechtigung – Diese erlaubt es, eine Datei zu verändern; hierzu gehört auch das Löschen einer Datei.
- Ausführungsberechtigung – Diese erlaubt es, die Datei „auszuführen“; bei Programmdateien bedeutet dies, sie aufrufen zu können, bei →Verzeichnissen, in diese zu wechseln bzw. den Zugriff auf diese zu haben.

**Deadlock** – Ein Gerätezustand, der auftritt, wenn zwei Programme auf ein Signal vom jeweils anderen warten, um mit der Arbeit fortzufahren; dies kann im schlimmsten Falle zum →Crash führen.

**Deadly Embrace** →Deadlock

**Debugger** – Ein Software-Entwicklungs-Tool, das einen Programmierer dabei unterstützt, Fehler in einem Programm zu lokalisieren und zu korrigieren.

**Default** – Neuhochdeutsch für „Standardwert“.

**Delimiter** – Trennzeichen

**Demand Paging** – Verfahren, bei dem Programmsegmente nur dann in den Speicher gelesen werden, wenn sie benötigt werden.

**DES** – *Data encryption standard.* Standardmethode für die Datenverschlüsselung- und -entschlüsselung.

**Descriptor Table** – Eine Liste, die sämtliche Ein- und Ausgabefunktionen protokolliert, die von den einzelnen →Prozessen angefordert werden.

**Device** – Jedes Gerät, das an ein UNIX-System angeschlossen ist, wie z. B. Disketten- und →CD-ROM-Laufwerke, Drucker etc.

**dezimal** – Das Zahlensystem betreffend, das auf der Zahl 10 basiert.

**Dhrystones** – Ein →Benchmark-Programm zur Quantifizierung der Leistungsfähigkeit von Computersystemen.

**Directory** – Dt. Verzeichnis. Eine Datei, welche die Dateinamen und →I-Nodes aller Dateien, die sie enthält, auflistet.

**Directory Tree** – →Verzeichnisbaum. Struktur, die in hierarchischer Form alle vorhandenen Dateien repräsentiert.

**disable** – Etwas ausschalten.

**Disabled Command** – Ein Befehl, der standardgemäß ausgeschaltet ist.

**Disk-Cache** – Speicherbereich, in dem Daten temporär zwischengespeichert werden, bevor sie von der Festplatte in den Speicher gelesen werden, oder umgekehrt.

**Diskless Workstation** – Ein →Netzwerk-Computer, der über keine eigenen lokalen Festspeicher (Diskettenlaufwerk, Festplatte) verfügt.

**Distribution** – Ein Software-Paket, das über das „nackte“ →Betriebssystem hinaus weitere →Applikationen beinhaltet.

**DMA** – *Direct Memory Access.* Ein Verfahren, bei dem Informationen direkt von z. B. der Festplatte in den Arbeitsspeicher geladen werden (oder umgekehrt), ohne dass der Prozessor daran beteiligt ist.

**DNS** – *Domain name service.* Ein Verfahren, bei dem ein →Domain-Name direkt in seine →IP-Adresse umgewandelt wird.

**Domain-Adresse** – Eine eindeutige Adresse zur Identifizierung eines Computers innerhalb des →Internets.

**Domain Name** – Ein eindeutiger Name zur Identifizierung eines →Internet→→Hosts.

**Doppelklicken** – Schnelles, zweimaliges Klicken einer Maustaste.

**DOS** – *Disk operating system.* Ein Möchtegern→Betriebssystem, mit dem sich viele Benutzer von INTEL-basierten PCs immer noch herumschlagen, welches Basis weiterer Microsoft→Betriebssystem(aufsätze) ist.

**dosemu** – Ein unter LINUX populäres Programm, das die Verwendung →DOS-basierter Anwendungen ermöglicht.

**Dot** – Englisch für „Punkt“. Dieser repräsentiert unter UNIX das aktuelle →Directory.

**Dot Dot** – Englisch für „Punkt Punkt“ (nicht Doppelpunkt!). Diese Zeichenkombination repräsentiert unter UNIX das dem aktuellen →Directory übergeordnete →Directory (Verzeichnis).

**Dot-Befehl** – Ein Befehl, der ausgeführt wird, ohne eine Sub-→Shell als →Child Process aufzurufen.

**Dot-Datei** – Datei, deren Name mit einem Punkt beginnt. Diese wird bei einem „normalen“ Listing der in einem →Directory vorhandenen Dateien nicht angezeigt und wird deshalb auch als „versteckte“ Datei bezeichnet.

**Dotted Decimal** – Verfahren, eine →IP-Adresse aus vier Dezimalzahlen zu bilden.

**Download** – Eine (Gruppe von) Datei(en) von einem anderen Computer auf den eigenen übertragen.

**DPI** – Die Auflösung in sog. Punkten, die ein Gerät pro Zoll (*dots per inch*) ausgeben kann.

**Drag-and-Drop** – Ein ausgewähltes Objekt (wie zum Beispiel ein Dateisymbol) auf ein anderes Objekt (wie zum Beispiel ein →Directory) bei gedrückter Maustaste „ziehen“, um einen →Prozess (zum Beispiel einen Kopievorgang) zu initiieren.

**Driver** – Jargon für „Device Driver“; eine Datei, die zur Ansteuerung eines Geräts (wie z. B. einem Drucker) dient.

**Dumb Terminal** – Ein „dummes Terminal“, bestehend aus Tastatur und Bildschirm, das der Ein-/Ausgabe von Informationen in/von einem →Server dient, ohne über eigene Prozessor- (→CPU) Kapazitäten zu verfügen.

## E

**E-Mail** – Ein →Netzwerk-Dienst, der es erlaubt, Nachrichten, Berichte, Memos etc. an andere →User zu verschicken.

**E-Mail-Adresse** – Die Adresse, die ein →User benötigt, um →E-Mails zu empfangen.

**EBCDIC** – *Extended Binary Coded Decimal Interchange Code*. Ein weiteres (→ASCII) Codierungsschema, das die einzelnen für einen Rechner zur Verfügung stehenden Zeichen (Buchstaben, Ziffern, Interpunktionszeichen, etc.) eindeutigen binären Werten zuordnet; vorwiegend in der Großrechnerwelt eingesetzt.

**Editieren** – Informationen in einer Datei ändern.

**Editor** – Eine →Applikation, die das →Editieren von Dateien ermöglicht.

**Einloggen** – Der Vorgang des Anmeldens bei einem Computersystem, indem der →User seinen User-Namen und ein →Passwort eingibt.

**Emotikons** – Zeichenkombinationen, die häufig in →E-Mails verwendet werden, um Gefühle auszudrücken. So bedeutet z. B. die Zeichenkombination :-> Freude, Amüsiertheit, etc. (drehen Sie Ihren Kopf um 90 nach links, um die Zeichen zu interpretieren).

**Emulator** – Ein (Hard- und/oder Software-implementiertes) Verfahren, etwas nachzubilden.

**enable** – Etwas einschalten.

**Error** – →Fehler

**erweiterter ASCII-Zeichensatz** – Ursprünglich wurden für die Codierung von →ASCII-Zeichen sieben Daten→Bits zur Bildung eines →Bytes verwendet (was  $2^8 = 128$  Zeichen ergab) – das achte war für Prüfzwecke reserviert; die Popularität des IBM-PCs machte jedoch bald die Integration internationaler Zeichen wie z. B. der deutschen Umlaute erforderlich, die durch Verwendung des achten →Bits im erweiterten ASCII-Zeichensatz Aufnahme fanden.

**Escape-Sequenz** – Folge von Zeichen, die durch das →dezimale →ASCII-Zeichen 27 eingeleitet wird. Sie kann beispielsweise zur Bildschirmsteuerung verwendet werden.

**Escape-Zeichen** – Ein Zeichen, das der →Shell mitteilt, dass ein Zeichen als solches interpretiert werden soll, ohne dass dessen besondere Bedeutung beachtet wird.

**Ethernet** – Populäres →Netzwerk-Protokoll und Verkabelungsschema mit einer Übertragungsrate von 10 Mbit/s (→Bandweite).

**Ethernet-Adresse** – Adresse, die einer →Netzwerk-Karte vom Hersteller zugewiesen wurde, z.B. 00:80:48:EE:27:24; da sich diese nun wirklich niemand merken kann, wird zur Adressierung eines Computers über ein →Netzwerk eine →IP-Adresse vergeben.

**Ethernet-Paket** – Einheit variabler Länge, die zur Übertragung von Informationen über das →Ethernet verwendet wird.

**Exit-Status** – Rückgabewert eines →Prozesses oder Befehls; er beträgt im Regelfall 0 bei erfolgreicher und 1 bei nicht erfolgreicher Ausführung.

## F

**FAQ** – *Frequently Asked Questions*. Eine Sammlung häufig gestellter Fragen mit den dazu gehörenden Antworten.

**Fast Ethernet** – Hochgeschwindigkeits→Ethernet mit einer Übertragungsrate von bis zu 100 Mbps (→Bandbreite).

**Fehler** – Die Diskrepanz zwischen dem erwarteten und dem tatsächlichen Ergebnis.

**Fenster** – Umgrenztes Rechteck, das innerhalb einer grafischen Benutzeroberfläche als Ansichtsbereich einer →Applikation dient.

**File Server** – →Netzwerk-Computer, der genutzt wird, um Dateien zu speichern.

**Filter** – Funktionalität eines UNIX-Befehls, Eingaben entgegenzunehmen, Operationen an diesen auszuführen und das Ergebnis weiterzuleiten.

**Firewall** – Ein Computer, der zwischen dem internen →Netzwerk und dem →Internet platziert ist, um nicht autorisierte Zugriffe zu vermeiden.

**Flame** – Eine beleidigende →E-Mail.

**Flame War** – Austausch von Standpunkten innerhalb einer →USENET→Newsgroup, der ursprünglich einmal sachlich begonnen haben mag, aber in einen Austausch von →Flames ausgeartet ist.

**Focus** – Das in einer →X Window-Umgebung aktive Fenster.

**fork** – Die Erzeugung eines neuen →Prozesses.

**Formfeed** – Das Steuerzeichen, das einen Seitenvorschub bei einem Drucker bewirkt.

**Fragmentierung** – Der →Block ist eine physikalische Größe bei der Unterteilung eines Datenträgers. Beträgt seine Größe beispielsweise 4.096 Byte, so wird eine Datei, die 7.000 Byte groß ist, auf mindestens zwei Blöcke „verteilt“. Strafverschärfend kommt hinzu, dass beim Schreibvorgang auf eine Festplatte jeweils freie

Blöcke beschrieben werden, ohne dass überprüft wird, ob diese zusammenhängend sind. Als Folge werden Dateien jeweils in „als frei“ erkannten Blöcken „stückweise“ abgespeichert, was zu eben dieser Fragmentierung führt – beim Einlesen muss das →Betriebssystem die einzelnen Portionen also wieder zu einem Ganzen zusammenführen.

**Free Software Foundation** – →FSF

**FreeBSD** – INTEL-basierte, frei kopierbare UNIX-Implementation.

**Freeware** – Software, deren Nutzung kostenlos ist, deren Copyright aber weiterhin bei dem/der AutorIn verbleibt.

**FSF** – *Free Software Foundation*. Organisation, die kostenlos erhältliche Software bereitstellt (Eigenentwicklungen oder Programme, die dieser Organisation zur Verfügung gestellt wurden).

**ftp** – Programm, das die Kommunikation mit einem fernen →Host ermöglicht; hierzu gehören die Übertragung und der Empfang von Dateien.

## G

**Gateway** – Gemeinsam genutzte Verbindung zwischen verschiedenen →Netzwerken, so dass diese miteinander kommunizieren können, ungeachtet der Tatsache, dass sie verschiedene Protokolle verwenden.

**Gemeinsamer Speicherzugriff** – →Shared Memory

**Gerätedatei** – Datei, die als Schnittstelle zu einem Gerät (→Device) dient.

**GID** – Gruppen-Identifizierungsnummer; auf diese wird zurückgegriffen, um die Rechte eines →Users festzulegen, wenn er/sie einer bestimmten Gruppe (→Group) angehört.

**globale Variable** – Eine Variable, die für die →Shell und für jedes andere Programm zur Verfügung steht.

**Globbing** – Der Vorgang, Dateinamen in der →Shell zu substituieren.

**GNU** – Projekt der →Free Software Foundation, das es sich zum Ziel gesetzt hat, ein frei erhältliches UNIX-System zu entwickeln („GNU is not UNIX“); LINUX, welches unter der GPL (Gnu public license, siehe Anhang D) steht, wird häufig auch als GNU/LINUX bezeichnet.

**Gopher** – →Client/Server-Anwendung, die →Internet-Ressourcen als eine Reihe von Menüs präsentiert, ohne dass der →User von den dahinter liegenden physikalischen Details Kenntnis haben muss.

**Gopherspace** – Alle →Internet-Ressourcen, auf die mittels →Gopher zugegriffen werden kann.

**Graphical User Interface** – →GUI

**Group** – Gruppe von →Usern; durch die Zuordnung eines Users zu einer bestimmten Gruppe können ihm spezifische Rechte zugewiesen werden, die er als →User selber oder als Mitglied einer anderen Gruppe nicht hätte.

**Groupware** – →Netzwerk-Software, die es einer Gruppe von Personen erlaubt, am selben Projekt zu arbeiten und gemeinsam Dateien zu nutzen.

**GUI** – *Graphical User Interface*. Grafische Schnittstelle, die es einem →User erlaubt, (Datei-) Operationen durch Anklicken mit einer Maus auszuführen, statt lange, kryptische Befehlsfolgen eintippen zu müssen.

## H

**Handshake** – Der Austausch von Steuercodes, um den Datenfluss zwischen zwei Geräten zu verwalten.

**Hard Link** – Ein →Verzeichnis-Eintrag, der den Dateinamen und den →I-Node einer Datei enthält.

**Hardware** – Alle physikalischen Komponenten eines Computersystems.

**hashen** – Der Vorgang des Erzeugens oder Reorganisierens einer →Hash-Tabelle.

**Hash-Tabelle** – Eine Hash-Tabelle weist jedem Datenbestandteil einen Index-Suchcode zu, der sein Auffinden mittels spezieller →Software gegenüber einer →sequenziellen Suche deutlich schneller gestaltet.

**Hauptgerätenummer** – Eine Nummer, die einem Gerät (→Device) zugeordnet ist.

**Header** – Informationen, die am Anfang einer Datei stehen.

**Header-Datei** – Eine Datei, die Definitionen von →Makros und →Variablen enthält.

**Heartbeat** – Eine Testfunktion für die Funktionsfähigkeit eines →Ethernet.

**hexadezimal** – Das Zahlensystem mit der Basis 16 betreffend.

**hidden character** – Ein nicht druckbares („verstecktes“) Zeichen, das (meist) eine bestimmte Funktion besitzt, wie zum Beispiel der →Backspace.

**hidden file** – →Dot-Datei.

**Hilfsgerätenummer** – Eine Zahl, die einem bestimmten Gerät einer Gerätekasse zugeordnet ist.

**History** – Ein Mechanismus, der die von Ihnen eingegebenen Befehle protokolliert, so dass auf diese wieder zurückgegriffen werden kann.

**History-Datei** – Die Protokolldatei der →History-Funktion.

**History-Substitution** – Ein Verfahren, mit dem man Befehle aus der →History-Datei →editieren kann, ohne dass die Befehlszeile vollständig neu eingegeben werden muss.

**Holy Wars** – Die Glaubenskriege innerhalb der UNIX-Gemeinde, die darüber ausgefochten werden, welches →Betriebssystem, welche Programmiersprache etc. das Gelbe vom Ei sei.

**Home Directory** – Ein →Verzeichnis, das die Dateien für eine bestimmte User-ID (→UID) enthält.

**Homepage** – Eine einleitende Startseite im →World Wide Web.

**Hop** – Eine einfache Verbindung zwischen zwei Computersystemen. Läuft beispielsweise das Versenden einer →E-Mail vom Absender zum Adressaten über sieben Systeme, spricht man von sechs Hops.

**Host** – Ein Computersystem, das in einem Netzwerk erreichbar ist.

**HP-UX** – Die UNIX-Version von Hewlett-Packard, die speziell auf deren Computer-Hardware angepasst ist.

**HTML** – *Hypertext Markup Language*. Seitenbeschreibungssprache für das Erstellen von Seiten im →World Wide Web.

**HTTP** – *Hypertext Transfer Protocol*. Das Protokoll, das verwendet wird, um die Verbindungen zwischen mehreren →Hypertext-Seiten zu handhaben.

**Hypermedia** – Ein Ausdruck, der nicht →sequenzielle Anwendungen beschreibt, die interaktive Verknüpfungen verschiedener Multimedia-Elemente besitzen.

**Hypertext** – Eine Methode, Informationen in nicht →sequenzieller Form zu präsentieren, indem durch Auswählen spezieller Textelemente auf neue Abschnitte zugegriffen wird.

# I

**I-Node** – *Information Node*. Eine Datenstruktur, die eine Datei beschreibt.

**I-Node-Tabelle** – Liste aller →I-Nodes eines →Dateisystems.

**I-Nummer** – Eine Nummer, die benutzt wird, um einen →I-Node zu identifizieren.

**IAB** – *Internet Architecture Board*. Koordinationskomitee für die Verwaltung des →Internets.

**Icon** – Ein Bildschirmelement innerhalb eines →GUI, das ein Benutzer durch Anklicken handhaben kann.

**Include-Datei** – →Header-Datei

**Index-Node** – →I-Node

**Inheritance** – Ein Verfahren, bei dem in der objektorientierten Programmierung die Eigenschaften einer Objektklasse an untergeordnete Klassen vererbt werden.

**Init** – Der erste →Prozess, der vom UNIX-→Kernel erzeugt wird, wenn das System hochfährt.

**Initialisierungsstring** – Eine Zeichenkette, die nach dem →Einloggen zum Terminal geschickt wird, um dieses zu konfigurieren.

**Initialisierungsdatei** – Eine Datei, deren Befehle ausgeführt werden, während das System startet.

**inkrementelles Backup** – Das Sichern von ausschließlich den Dateien, die seit dem letzten →Backup verändert wurden oder neu hinzugekommen sind.

**Input** – Das Eingeben von Informationen.

**Input Redirection** – Der Vorgang, bei dem der →Input nicht über die Standardeingabe (Tastatur) vorgenommen wird, sondern aus z. B. einer regulären Datei gelesen wird.

**Input/Output** – Der Datentransfer zwischen einem Computer und seiner →Peripherie.

**integrierter Befehl** – Befehl, der tatsächlicher Bestandteil eines →Befehlszeileninterpreters ist und der somit ausgeführt werden kann, ohne dass hierzu ein neuer →Prozess gestartet werden muss.

**interaktiv** – Eine Funktionalität, die einen Dialog zwischen einem Benutzer und einem Programm ermöglicht.

**Interface** – Die Schnittstelle zwischen verschiedenen Teilen eines Systems.

**Internet** – Eine weltweite Verbindung von Computersystemen, welche die Nutzung von Diensten wie →E-Mail, →Chat, →World Wide Web, →ftp und so weiter ermöglicht, die unter Verwendung von →TCP/IP miteinander kommunizieren.

**Internet-Adresse** – Die absolute →Adresse, über die ein →User oder ein Computer innerhalb des →Internet angesprochen werden kann.

**Interpreter** – Ein Programmiersprachenübersetzer, der bei jeder Ausführung eines Programms eine zeilenweise (→sequentielle) Übersetzung des Programmcodes in eine für die Maschine verständliche Sprache vornimmt.

**Interprozesskommunikation** – Alle Methoden, die angewandt werden, um Informationen zwischen zwei Programmen auszutauschen, unabhängig davon, ob diese auf dem gleichen Computersystem ablaufen oder über ein →Netzwerk miteinander kommunizieren.

**Interrupt** – Ein Signal, das an den Prozessor geschickt wird, um die normale Verarbeitung zu unterbrechen.

**Interrupt-Handler** – Software im UNIX→Kernel, welche die System→Interrupts verwaltet.

**IP** – *Internet Protokoll*. Das Verbindungsprotokoll, welches das Senden von Daten mittels Überwachung von →Internet-Adressen reguliert.

**IP-Adresse** – Eine eindeutige 32-Bit-Zahl, die einen →Internet→Host identifiziert.

**IPC** - *Interprocess communication*. Zusammenfassende Beschreibung der Methoden, die dazu verwendet werden, um Informationen zwischen zwei Programmen auszutauschen, die auf demselben Computer oder auf unterschiedlichen Computern innerhalb eines Netzwerks laufen. Hierzu gehören →Pipes, →Sockets, →Semaphoren, gemeinsam genutzter Arbeitsspeicher usw. (→Interprozesskommunikation)

**IRC** – Internet Relay Chat →Chat.

**ISO** – Internationale Organisation zur Standardisierung des Kommunikations- und Informationsaustausches.

**ISO/OSI-Referenzmodell** – Ein von der →ISO definiertes →Netzwerk-Referenzmodell, das die Kommunikation von Computern in sieben miteinander verbundene Schichten aufteilt.

## J

**Jabber** – Eine kontinuierliche, bedeutungslose Übertragung, die von einem Netzwerkgerät erzeugt wird; diese kann durch Fehlbedienung oder Fehlfunktion einer Netzwerkkomponente entstehen.

**Java** – Eine von Sun Microsystems entworfene Programmiersprache, um Anwendungen im →World Wide Web →interaktiv nutzen zu können.

**Job** – Ein Befehl oder eine Gruppe von Befehlen, die als Einheit fungieren (→Prozess).

**Job-Kontrolle** – Ein UNIX-Mechanismus, welcher der Verwaltung von →Jobs dient. Hierzu gehören das Anhalten, das Stellen in den Vordergrund oder in den Hintergrund (→Background) usw.

**Job-Nummer** – Nummer, über die →Jobs zur weiteren Manipulation angesprochen werden können.

**JPEG** – Ein Bildkompressionsstandard, der entworfen wurde, um weniger Haupt- oder Plattspeicher in Anspruch zu nehmen.

**Jukebox** – Ein →CD-Wechsler, der automatisch die verschiedenen CDs →mountet.

## K

**Kassettenlaufwerk** – →Peripherie-Gerät, das Magnetbandkassetten liest und beschreibt.

**Kerberos** – Ein →Netzwerk-Sicherheitssystem, das die Gültigkeit eines →Users überprüft, der einen bestimmten Netzwerkdienst anfordert.

**Kermit** – Ein Datei-Übertragungsprotokoll für die gesicherte Übertragung von Daten über normale Telefonleitungen.

**Kernel** – Der zentrale, speicherresidente Teil des UNIX→Betriebssystems, der Ressourcen zuordnet, Speicher verwaltet und →Prozesse überwacht.

**Kernel-Adressraum** – Der Teil des Hauptspeichers, auf den nur der →Kernel zurückgreifen kann.

**Keyword** – Reserviertes Schlüsselwort in einer Programmiersprache mit einer speziellen Bedeutung.

**Kill-Signal** – Das Signal, das an einen →Prozess gesendet wird, damit dieser sofort beendet wird (Signalnummer 9).

**Kilobit** – KB oder Kbit. 1.024 →Bits.

**kilobits per second** – Kbps. Die Anzahl der →Kilobits, die pro Sekunde übertragen werden.

**Kommunikationssteuerschicht** – Die fünfte der sieben Schichten des →ISO/OSI-Referenzmodells, welche die Datenübertragung koordiniert und Verbindungen aufrechterhält.

**Korn-Shell** – Weiterentwicklung der →Bourne-Shell.

## L

**LAN** – *Local area network*. Lokales →Netzwerk, bestehend aus einer über einen Kommunikationskanal miteinander verbundenen Gruppe von vernetzten Computern und →Peripheriegeräten.

**LATEX** – Funktionserweiterungen zu →TEX, einem populären Satzsystem.

**Laufzeitbibliothek** – Eine Reihe von Bibliotheks Routinen (→C-Bibliothek), mit denen häufig verwendete Funktionen ausgeführt werden können, die von einem Programmierer in sein Programm integriert werden können.

**LILO** – LINUX Loader. Ein →Boot-Manager-Programm, das es ermöglicht, zwischen verschiedenen →Betriebssystemen zum Hochfahren eines Rechners zu wählen.

**limited regular expressions** – →reguläre Ausdrücke.

**Link** – Eine Verbindung zwischen einem Dateinamen und dem entsprechenden →I-Node. Auch die Verbindung in einem Hypertext-Dokument zu einem anderen.

**Link-Editor** – Ein Tool zur Software-Entwicklung, das getrennt voneinander kompilierte Programme zu einem einzelnen ausführbaren Programm verbindet (→linken).

**linken** – Der letzte Arbeitsschritt bei der Programmerstellung, bei dem die verschiedenen Objektdateien miteinander verbunden werden.

**Linker** – →Link-Editor

**LINUX** – Beachten Sie zu diesem Thema den Rest des Buches ...

**Listserver** – Automatisches Mailsystem im →Internet.

**Login** – Der Verbindungsauflauf mit einem UNIX-System (→einloggen).

**Loginname** – Der Name, mit dem sich ein →User in ein UNIX-System →einloggen kann.

**Login-Shell** – Die →Shell, mit der Sie standardgemäß nach dem →Einloggen in ein UNIX-System kommunizieren.

**Logout** – Das Beenden einer Arbeitssitzung (→ausloggen).

**lokale Variable** – Eine →Variable eines Programms oder einer →Shell, die für andere Programm(teile) oder →Shells nicht verfügbar ist.

## M

**Mach** – Ein →Betriebssystem, das den Einsatz von Multi-Prozessorsystemen und →Multitasking ermöglicht.

**Mail** – →E-Mail

**Mail Reflector** – Ein Programm zur Verwaltung von →E-Mail-Listen im →Internet.

**Mail Transport Agent** – Ein Programm zur Übertragung und Verteilung von →E-Mails.

**Mail User Agent** – Ein Programm, das neben dem Versenden und Empfangen von →E-Mails →Editor-Funktionen und Funktionen zum Weiterleiten und Löschen von →Mails zur Verfügung stellt.

**Mailbox** – Der für →E-Mails reservierte Speicherbereich einer Festplatte.

**Mailer** – Ein Programm zum Senden und Empfangen von →E-Mails.

**Mailing-Liste** – Eine Gruppe von →Usern des →Internet, die automatisch die an einen →Listserver gesendeten Nachrichten empfangen.

**make** – Ein Werkzeug, das bei der Software-Entwicklung zur Verwaltung von Abhängigkeiten unter Quellcodedateien dient.

**makefile** – Eine mit →make erstellte Datei, in der Quellcode dateien, Objektdateien sowie die jeweiligen Abhängigkeiten aufgelistet werden.

**Makro** – Eine Folge von Anweisungen, die gespeichert wurden, um komplexe oder häufig wiederholte Handlungsabläufe zu automatisieren.

**man-Seiten** – Die einzelnen Einträge des UNIX-Systemhandbuchs, die auf dem Bildschirm dargestellt werden können.

**Maschinensprache** – Die Sprache, die von einem Computer intern verwendet wird (→Befehlssatz).

**Maus** – Eingabegerät mit einer oder mehr Tasten, das zum Markieren und Manipulieren von Objekten innerhalb eines →GUI verwendet werden kann.

**MBONE** – Multicast backbone. Methode zur Echtzeitübertragung von digitalen Videodaten über das Internet.

**Megabit** – Mbit. 1.048.576 →Bits.

**Megabits pro Sekunde** – Mbps. Die Anzahl der über ein →Netzwerk übertragenen →Bits pro Sekunde.

**Megabyte** – MB oder MByte. 1.048.576 →Bytes.

**Megahertz** – MHz. Anzahl der Schwingungen pro Sekunde.

**Menü** – Eine Reihe von Befehlen oder Optionen, die ein Programm auf dem Bildschirm zur Verfügung stellt.

**message channel** – Form der →Interprozesskommunikation, bei dem zwei Programme zum Informationsaustausch auf dem gleichen Computersystem betrieben werden.

**Microcode** – Anweisungen auf Prozessorebene, die festlegen, wie der Prozessor arbeitet.

**Middleware** – Software, mit der →Applikationen von darunter liegenden →Netzwerk-Prozessen abgeschirmt wird.

**Mikrokernell** – Eine alternative →Kernel-Architektur, die im →Betriebssystem →Mach integriert wurde.

**MIME** – *Multipurpose Internet Mail Extensions*. Erweitertes →E-Mail-Übertragungsprotokoll, mit dem neben reinen Textinformationen auch binäre Daten, wie z. B. Grafiken und Programmdateien, übertragen werden können.

**Moderator** – Eine Person oder eine Gruppe von Personen, die den Inhalt der an eine →USENET-→Newsgroup gesendeten Nachrichten daraufhin überprüft, dass diese den selbst gesetzten Standards der Gruppe entsprechen.

**Moderierte Newsgroup** – →Newsgroup, deren Nachrichten von →Moderatoren verwaltet werden.

**Modulo** – Teilungsrest bei einer ganzzahligen Division.

**Mosaic** – Ein Programm für die Nutzung des →World Wide Web.

**Motif** – Eine weit verbreitete X Window-basierte grafische Benutzeroberfläche (→GUI).

**Mount-Punkt** – →Verzeichnis, auf das ein →Dateisystem ge→mountet wird.

**mounten, Mounting** – Einbinden von →Dateisystemen in ein UNIX-System.

**MUD** – *Multiuser Dungeon*. Ein textorientiertes Fantasy-Rollenspiel im →Internet, an dem mehrere →User gleichzeitig teilnehmen können.

**Multiprocessing** – Die Fähigkeit eines →Betriebssystems, mehr als einen Prozessor anzusprechen.

**Multitasking** – Die gleichzeitige Ausführung mehrerer Programme.

**Multithreading** – Die gleichzeitige Verarbeitung mehrerer →Tasks oder →Threads innerhalb eines Programms.

**Multiuser-System** – Computersystem, bei dem sich mehrere →User gleichzeitig →einloggen können.

## N

**Named Pipe** – Eine temporäre Datei, die prozessunabhängig ist und die es somit erlaubt, dass nicht aufeinander bezogene →Prozesse Informationen austauschen können.

**NetBSD** – Kostenlose Implementierung von UNIX, die auf →BSD basiert.

**Netiquette** – Eine Sammlung von ungeschriebenen Gesetzen, die z. B. beim Versenden von →E-Mails beachtet werden sollten; werden diese nicht beachtet, kann es schnell zu einem →Flame War kommen.

**Netzwerk** – Eine Gruppe von Computersystemen und →Peripheriegeräten, die über einen Kommunikationskanal miteinander verbunden sind und so mehreren →Usern den gemeinsamen Zugriff auf Daten und andere Ressourcen ermöglichen.

**Netzwerkarchitektur** – Der Aufbau eines →Netzwerks samt →Hardware, →Software, Zugriffsmethoden und verwendeten →Protokollen.

**Netzwerkdateisystem** – →NFS

**Netzwerkschicht** – Die dritte der sieben Schichten des →ISO/OSI-Referenzmodells, welche die →Protokolle zur Datenübermittlung definiert und somit sicherstellt, dass die Informationen an den korrekten Bestimmungsknoten gelangen.

**Netzwerkschnittstellenkarte** – →NIC

**Netzwerkverwaltungsbefehle** – Die einem →Systemadministrator zur Verfügung stehenden Befehle zur →User-Verwaltung, Systemsicherheit sowie zur Ausführung von allgemein netzwerkbezogenen Operationen.

**Newbie** – Neuling in der →Internet-Welt.

**Newsgroup** – Eine →E-Mail-Gruppe im →USENET, die sich der Diskussion bestimmter Sachgebiete widmet.

**Newsgroup-Kategorien** – Formale Struktur der einzelnen →USENET→Newsgroups.

**Newsreader** – Ein Programm zum Lesen von Artikeln, die in →USENET→Newsgroups veröffentlicht werden.

**NFS** – *Network File System*. Ein →Dateisystem, das die gemeinsame Nutzung von Dateien innerhalb eines →Netzwerks ermöglicht, als handele es sich bei diesem um ein lokales →Dateisystem.

**NIC** – *Network Interface Card*. Die →Hardware-Komponente, die zusammen mit der →Netzwerk→Software den Informationsaustausch im →Netzwerk kontrolliert.

**NIS** – *Network Information Service*. Ein Programm zur Verwaltung von Passwörtern, →Host-Adressinformationen und Zugriffsberechtigungen sowie für die Verwaltung von Daten für den Betrieb von →NFS.

**No News Is Good News** – UNIX-Philosophie, die aussagt, dass nur dann Meldungen vom →Betriebssystem oder anderen Programmen ausgegeben werden sollten, wenn ein Fehler oder Problem bei der Verarbeitung aufgetreten ist. Ein problemloser Programmablauf verursacht somit keine Meldungen, was eine gute Meldung ist.

**Node** – Die Funktion eines →Verzeichnisses als Knoten innerhalb des hierarchischen →Dateisystems.

**Non-Fatal-Error** – Ein nicht schwerwiegender Fehler, der die weitere Funktionsfähigkeit des Gesamtsystems nicht beeinflusst.

**Null-Argument** – Ein Befehlszeilen→Argument, das keinen Wert besitzt, aber unbedingt angegeben werden muss, damit die einzelnen →Argumente in ihrer korrekten syntaktischen Position stehen.

**Null-Gerät** – Die Datei /dev/null. Werden Daten an diese gesendet, landen sie im Daten-Nirvana.

**Null-Länge-Datei** – Eine Datei ohne Inhalt, die also null Bytes enthält.

**Null-String** – Eine leere Zeichenkette.

**Null-Zeichen** – Auch: binäre Null. Ein Wert, der das Ende einer Zeichenkette, einer Datei oder einen leeren Zeiger signalisiert.

## O

**Objekt** – Eine distinkte Einheit.

**Objektdatei** – Eine Datei, die aus der Kompilierung von Quellcode entsteht.

**objektorientiert** – Computersysteme, →Betriebssysteme, Programmiersprachen und →Applikationen, welche die Verwendung von →Objekten unterstützen.

**objektorientierte Programmierung** – Kurz: OOP. Programmiermodell, bei dem ein Programm als eine Reihe von selbständigen Objekten betrachtet wird, die mit anderen Objekten durch die Übermittlung von Nachrichten in Verbindung tritt.

**obligatorisches Argument** – →Argument, das einem Befehl übergeben werden muss, wie z. B. die Angabe mindestens eines Dateinamens bei der Verwendung des UNIX-Dateilöschenbefehls.

**offene Datei** – Die aktuell verwendete Datei.

**offenes System** – →Hard- oder →Software, die nicht →proprietär gebunden ist.

**Offline-Reader** – Programm zum Lesen von →USENET→Newsgroups, ohne mit dem →Internet verbunden zu sein.

**oktal** – Das Zahlensystem mit der Basis 8 betreffend.

**Oktett** – Anderes Wort für →Byte (acht →Bits) im →Internet.

**Open Desktop** – →GUI des UNIX-Herstellers SCO.

**Open Look** – →GUI, die auf →X Window basiert.

**OpenGL** – *Open Graphics Library*. Eine Reihe von Grafikbibliotheken aus dem Hause SGI (Silicon Graphics) für die Entwicklung von 3D-Grafikanwendungen.

**OpenServer** – Eine Reihe UNIX-basierter Produkte des Herstellers SCO.

**OpenWindows** – Eine grafische Benutzeroberfläche (→GUI) für das UNIX-System Solaris der Firma SUN.

**Operand** – Daten, mit denen mathematische Funktionen durchgeführt werden.

**Option** – Ein Befehlszeilen→Argument, das die Funktion eines Befehls modifiziert.

**optionales Argument** – Ein Befehlszeilen→Argument, das für die Ausführung eines Befehls nicht zwingend notwendig ist, im Gegensatz zum →obligatorischen Argument.

**OSF** – *Open Software Foundation*. Eine nicht kommerzielle Vereinigung, die sich die Entwicklung von Standards für Computer- und →Netzwerk-Systeme auf ihre Fahnen geschrieben hat.

**OSF/1** – Von der →OSF entwickeltes →Betriebssystem, das auf dem →Mach→Kernel basiert.

**Owner** – UNIX-Benutzer, der eine Datei erstellt und den Zugriffsmodus und die Zugriffsberechtigungen für diese festlegen kann.

## P

**Pager** – Ein Programm, das die bildschirmweise Anzeige einer Datei auf einem Terminal ermöglicht.

**Paging** – Verfahren, bei dem Bereiche von Programmen in einen Speicherpuffer verlagert werden. Erst wenn dieser Puffer voll ist, werden sie in einen Puffer auf die Festplatte verschoben. Da das →Betriebssystem bei einer Anforderung zunächst im Speicherpuffer und erst dann auf der Festplatte nach diesen Informationen sucht, ist dieses Verfahren dem →Swapping, bei dem ganze →Prozesse ausgelagert werden, geschwindigkeitsmäßig überlegen.

**Paket** – Datenblock, der innerhalb eines →Netzwerks versendet wird.

**Panic** – Eine Situation, in der das UNIX→Betriebssystem feststellt, dass der →Kernel nicht weiter betrieben werden kann; als Folge wird das System heruntergefahren, um weitere Schäden zu verhindern.

**Parallelverarbeitung** – Gleichzeitige Verarbeitung eines →Prozesses durch mehrere Prozessoren.

**Parameter** – Ein Wert, der an ein Programm oder eine Funktion weitergegeben wird.

**Parität** – Verfahren, das dazu dient, Fehler bei der Übertragung von Daten auszuschließen.

**Passwort** – Eine Zeichenkette, mit der ein →User sich beim →Login legitimiert.

**Pattern Matching** – Übereinstimmung zwischen gesuchten und gefundenen →Such-Strings.

**Peripherie** – Jede Art von externem Gerät, d. h. Gerät, das nicht direkt auf dem Motherboard eines Computers sitzt, wie z. B. Drucker, Festplatten usw.

**Perl** – *Practical Extraction and Report Language*. Programmiersprache zur Manipulation von Text, Dateien und →Prozessen (siehe Kapitel 17).

**Pfadname** – Eine Liste mit →Verzeichnisnamen, die durch einen Schrägstrich (/) voneinander getrennt sind.

**PGP** – →Pretty Good Privacy

**physikalisches Gerät** – →Hardware

**PID** – *Process ID*. Eine Zahl, die einem →Prozess bei seinem Aufruf zugewiesen wird, so dass er, bezugnehmend auf diese Zahl, weitergehend manipuliert werden kann (→Job-Kontrolle).

**Pine** – Populäres, menügesteuertes →E-Mail-Programm.

**Pinguin** – Das offizielle LINUX-Logo (→Tux):

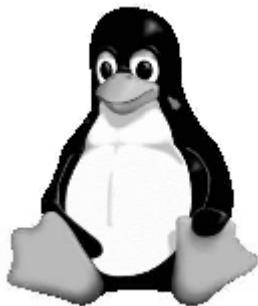


Abbildung E.1: Der LINUX-Pingu Tux

**Pipe** – Verfahren, bei dem ein Befehl Informationen zur weiteren Verarbeitung an einen anderen Befehl weiterleitet.

**Pipe, benannte** – →Named Pipe

**Pipe-Zeichen** – Das Zeichen |.

**Pipeline** – Eine Gruppe von durch eine →Pipe miteinander verbundenen Befehlen.

**Pipelining** – Verfahren zur Dekodierung von Anweisungen in der →Mikroprozessor-Architektur.

**Plattform** – Eine →Betriebssystem-Umgebung.

**Port** – Die Schnittstelle, an welcher der Kommunikationskreislauf eines →Netzwerks endet.

**Portabilität** – Möglichkeit, ein →Betriebssystem oder eine →Applikation von einem →Hardware-Typen auf einen anderen zu übertragen.

**portierbar** – Die Möglichkeit, →Applikationen auf mehr als einer (UNIX)- →Plattform laufen zu lassen.

**POSIX** – *Portable Operating System Interface*. Eine Reihe von Bedingungen für →portierbare →Betriebssysteme.

**posten** – Das Versenden von einzelnen Artikeln an eine →USENET–→Newsgroup oder an die Mitglieder einer →Mailing-Liste.

**PostScript** – Eine von Adobe Systems entwickelte Seitenbeschreibungssprache, die →Hardware-unabhängigen Einsatz findet.

**Poststempel** – Der Bereich in der Kopfzeile einer →E-Mail, der den Adressaten und das Empfangsdatum enthält.

**PPID** – →Umgebungsvariable, mit der die →PID eines →Prozesses an die übergeordnete →Shell übergeben wird.

**PPP** – *Point-to-Point Protokoll*. →TCP/IP–→Protokoll, das zur Übertragung von →Paketen über Modem/Telefonverbindungen dient.

**präemptives Multitasking** —>Multitasking, bei dem das →Betriebssystem über eine bestimmte Zeitdauer (abhängig von ihrer Priorität) eine Anwendung ausführt, wobei ein gleichzeitiger Zugriff auf die →CPU durch einen anderen →Prozess möglich ist.

**Präprozessor** — →Software-Entwicklungs-Tool zur Vorverarbeitung eines →Quellcodes vor Weitergabe an den →Compiler oder →Assembler.

**Pretty Good Privacy** – PGP. Ein populäres, von Phil Zimmermann entwickeltes Programm zur Verschlüsselung von Dateien. Da das Knacken solch codierter Nachrichten astronomische Rechenkapazitäten erfordern würde, kann die Umschreibung „pretty good“ (dt. ziemlich gut) nur ein Euphemismus sein.

**Primär-Mailbox** —>Mailbox, in die →E-Mail→Programme zunächst alle empfangenen →E-Mails ablegen.

**Privacy Enhanced Mail** —>PEM

**Privileged Account** – Konto des →Superusers.

**Programm** – Sammlung von maschinenlesbaren, ausführbaren Anweisungen, die in einer →Datei enthalten sind.

**Prompt** – Bereitschaftszeichen

**propriétär** – herstellerspezifisch

**Protokoll** – Alle Angaben zur Vorgehensweise bei der Datenübertragung von →Netzwerk-Anwendungen.

**Protokoll-Stack** – →Software-Schichten, welche die Computer/Computer- oder Computer/Netzwerk-→Protokolle definieren.

**Prozedurorientierte Sprache** – Programmiersprache, dessen fundamentales Element die Prozedur, Subroutine oder Funktion ist.

**Prozess** – Die Ausführung eines →Programms; wenn z. B. mehrere →User gleichzeitig mit einem →Editor arbeiten, so wird dieser mehrmals in den Speicher geladen und liegt als jeweils eigener Prozess vor.

**Prozess ID** – →PID

**Prozess-Status** – Der aktuelle Zustand eines →Prozesses. UNIX kennt acht verschiedene Zustände:

- running in user mode – Benutzermodus
- running in kernel mode – →Kernel-Modus
- waiting – warten
- sleeping – pausieren
- idle – Leerlauf
- swapping – auslagern
- stopped – unterbrochen
- dead – tot; siehe auch →Zombie

**Prozessverwaltung** – Handhabung einzelner →Prozesse über →PID, →UID, →GID und verschiedene →Umgebungsvariablen.

**Punkt** – Das Zeichen „.“, das entweder zur Symbolisierung des aktuellen →Verzeichnisses dient, eine bestimmte Form des →Programm-Aufrufs (→Dot-Befehl) ermöglicht oder Einzelzeichen in →regulären Ausdrücken darstellt.

## Q

**QNX** – Von Quantum Software Systems entwickeltes, UNIX-ähnliches →Betriebssystem, das auf →Mikrokernel-Technologie basiert.

**Quellcode** – In einer Programmiersprache geschriebene, ursprüngliche Version eines →Programms, die noch in eine für den Prozessor verständliche Sprache übersetzt werden muss.

**Quellcode-Lizenz** – →Software-Lizenz, bei welcher der Käufer neben dem ablauffähigen Programm auch den zugehörigen →Quellcode erwirbt.

**Quelldatei** – Datei, die einen →Quellcode enthält. Auch: Datei, die Daten an eine andere Datei übergibt.

**quoten** – Der Vorgang des Außer-Funktion-Setzens von Sonderzeichen. Auch: Bestandteile anderer →E-Mails zitieren.

## R

**RAID** – *Redundant Array of Inexpensive Disks*. Mehrstufiges Verfahren zur Erhöhung der Datensicherheit durch die Verwendung verschiedener Laufwerks-/Controller-Architekturen.

**Rangfolge** – Die Reihenfolge, in der arithmetische Operationen durchgeführt werden („Punktrechnung geht vor Strichrechnung“ usw.).

**Raw Device** – →Gerät, das ohne Zwischenpuffer direkt gelesen oder beschrieben werden kann.

**Raw Mode** – Modus, bei dem unter Umgehung des UNIX-Ein-/Ausgabesystems direkt auf ein Gerät geschrieben wird.

**rc** – Eine als Ersatz für die →Bourne Shell gedachte →Shell, die jedoch bezüglich ihrer Funktionsvielfalt gegenüber beispielsweise der →Bash stark zurückfällt.

**RCS** – *Revision Control System*. Eine Reihe von Programmen zur Rückverfolgung umfangreicher Änderungen im →Dateisystem.

**read-only** – Zustand, bei dem Daten (z. B. in Dateien) nur gelesen, aber nicht manipuliert (verändert, gelöscht) werden können.

**Read-Only-Dateisystem** – →Dateisystem, das auf →read-only gesetzt wurde, so dass die in ihm enthaltenen Dateien nicht manipuliert werden können.

**Reduced Instruction Set Computing** – →RISC

**regulärer Ausdruck** – Zeichenfolge zur Repräsentierung einer zu suchenden Zeichenfolge, in der bestimmte →Metazeichen verwendet werden können, die z. B. Zeichengruppen ersetzen, Zeilenpositionen bestimmen usw.

**reguläres Zeichen** – Zeichen, das in →regulären Ausdrücken nur sich selber repräsentiert, ohne eine Sonderbedeutung zu besitzen.

**rekursiv** – In der Programmierung: die Fähigkeit eines Programm(teil)s, sich selber aufzurufen. Bei UNIX-Befehlen: die Fähigkeit, ganze Verzeichnisstrukturen zu manipulieren.

**relationale Operatoren** – Operator zum Vergleich zweier oder mehrerer Ausdrücke auf: Gleichheit, Ungleichheit, Größer-Gleichheit, Kleiner-Gleichheit, kleiner, größer.

**relativer Pfadname** – →Pfadname, der mit dem aktuellen →Verzeichnis beginnt.

**Remote File System** – →RFS

**Remote Procedure Call** – →RPC

**Remote Login** – →Einloggen auf einem entfernten Computer.

**Remote Mapping** - →Sicherheitsverfahren unter RFS zur Verwaltung der Zugriffsrechte von →Usern.

**Reset String** – Zeichenkette, die an ein Terminal gesendet wird, um es neu zu starten.

**Resolver** – Ein →Programm zur Umwandlung von →Hostnamen in ihre →IP-Adressen.

**Ressource** – Quelle; eine →Software- oder →Hardware-Komponente.

**Ressourcen-Sharing** – Fähigkeit eines →Betriebssystems, einzelne →Ressourcen verschiedenen →Prozessen gleichzeitig zur Verfügung zu stellen.

**Restricted Account** – →Account, der beschränkte Zugriffsrechte besitzt, um die Systemsicherheit zu erhöhen.

**Restricted Shell** – In ihrer Funktion eingeschränkte →Shell, die es einem →User beispielsweise nicht erlaubt, das aktuelle →Verzeichnis zu verlassen, bestimmte →Umgebungsvariablen zu ändern usw.

**Return Code** – →Exit-Status

**Revision Control System** – →RCS

**RFC** – *Request for comments*. Dokumente, mit denen versucht wird, →Internet-Standards zu beschreiben.

**RFD** – *Request for discussion*. Nachricht, die an die Gruppe news.announce.newsgroups im →USENET gesendet wird, um die Bildung einer neuen Newsgroup vorzuschlagen.

**RFS** – *Remote File System*. Ein Übertragungsprotokoll, mit dem im →Netzwerk →Ressourcen auf einem lokalen Computer betrieben werden können, als seien diese ebenfalls lokal.

**RISC** – *Reduced Instruction Set Computer*. Ein Prozessortyp, der nur über eine begrenzte Anzahl von Befehlen in seinem →Befehlssatz verfügt.

**rksh** – →Restricted Shell-Version der →Korn-Shell, in der ein →User sein →Home-Verzeichnis nicht verlassen, die →Umgebungsvariable PATH nicht verändern und auch keine →Ausgabeumlenkung vornehmen darf.

**rn** – Populärer →Newsreader zum Abonnieren von →USENET→Newsgroups.

**Rohdaten** – Noch nicht verarbeitete Daten.

**Root** – Das Haupt- oder Stamm→Verzeichnis, das sich an oberster Stelle in der Hierarchie eines →Dateisystems befindet, von dem also alle anderen →Verzeichnisse als Unterverzeichnisse abzweigen. Auch: Name des →Superusers.

**Root-Dateisystem** – Das im →Single-User-Modus zur Verfügung stehende →Dateisystem.

**Root-Login** – Benutzername des →Superusers, der beim →Login verwendet wird.

**rot13** – Einfaches Verschlüsselungsverfahren, das sich einer gewissen Popularität in →USENET→Newsgroups erfreut, das häufig zur Verschlüsselung anstößigen Materials oder sogenannter Spoiler (dt. Spielverderber; dies sind Informationen z. B. über den Ausgang eines Buchs oder Films oder Lösungsansätze von Spielen) dient. Da die Entschlüsselung sehr einfach ist (alle Buchstaben werden mit dem ihnen jeweils dreizehnten folgenden Buchstaben des Alphabets vertauscht, Zahlen und Interpunktionszeichen bleiben hiervon unbeeinflusst), kann es wirklich nur dazu verwendet werden, Nachrichten dahingehend zu verschlüsseln, dass diese „irrtümlich“ von jemandem gelesen werden, der es nicht explizit darauf anlegt.

**RPC** – *Remote Procedure Call*. Eine Reihe von Prozeduren zur Implementierung von →Client/Server-Architekturen in der Programmierung.

**Rsh** – →Restricted Shell der →Bourne-Shell.

**run** – Ein Programm ausführen.

**Runnable Process** – Ein Prozess, der sich so lange im Wartezustand befindet, bis die zu seiner Ausführung notwendigen →Ressourcen wieder verfügbar sind.

## S

**SAGE** – *System Administrator's Guild*. Interessengruppe, deren Hauptziel es ist, Richtlinien für die Ausbildung und Qualifizierung von Systemadministratoren aufzustellen.

**SATAN** – *Security Administrator Tool for Analyzing Networks*. Softwarepaket zur Überprüfung von Sicherheitslücken in einem →Netzwerk. Erhältlich z. B. unter der Adresse [ftp.demon.co.uk](http://ftp.demon.co.uk).

**scs** – *Source Code Control System*. Ein Programmpaket, das der Rückverfolgung von Dateiänderungen während der →Software-Entwicklung dient.

**Scheduler** – Komponente des UNIX-→Kernels zur Festlegung der Ausführung von Zeitpunkt, Zeitdauer und Reihenfolge von →Prozessen.

**Schneckenpost** – →Snailmail. Synonym für die Deutsche Post AG.

**Schnittstellenstandard** – Standardisiertes Verfahren für die Verbindung von →Geräten mit unterschiedlichen Funktionen.

**Schreibberechtigung** – Berechtigung, Daten in eine neue Datei zu schreiben oder bereits existierende Dateien zu verändern oder zu löschen.

**SCO** – *Santa Cruz Operation*. Größter kommerzieller Anbieter von UNIX-→Betriebssystemderivaten; in der Zwischenzeit vom LINUX-Spezialisten Caldera übernommen

**SCO OpenServer** – Reihe von UNIX-Produkten der Firma →SCO. Hierzu gehören →OpenServer Desktop, OpenServer Host und OpenServer Enterprise.

**SCO UNIX** – Populäre UNIX-Version von →SCO, die auf →System V Version 3.2 mit vielen SVR4-Erweiterungen basiert.

**scrollen** – Verschieben („rollen“) des aktuellen Bildschirmbereichs.

**SCSI** – *Small Computer System Interface*. Hochgeschwindigkeitsschnittstelle für den Anschluss von →Peripheriegeräten.

**SCSI-Bus** – Anderer Name für →SCSI-Schnittstellen und →SCSI-Übertragungsprotokolle.

**SCSI-Terminator** – Endwiderstand zur Vermeidung von Signalechoeffekten, der auf beiden Enden eines →SCSI-Busses vorhanden sein muss.

**SDK** – *Software Development Kit*. Paket mit Werkzeugen für die →Programm-Entwicklung, wie →Bibliotheken, technische Dokumentationen usw.

**sekundäre Mailbox** – →Mailbox, in die bereits gelesene →E-Mails abgelegt werden.

**Sekundärprompt** – Eingabe- →Prompt, der von der →Shell angezeigt wird, um anzugeben, dass der eingegebene Befehl unvollständig ist und somit weitere Eingaben erforderlich sind.

**Semaphore** – Ein →Interprozess-Kommunikationssignal, das den Status von gemeinsam verwendeten System-→Ressourcen anzeigt.

**Semikolon** – Das Zeichen „;“, das zum Trennen von mehreren Befehlen in einer Befehlszeile oder einem →Skript dient.

**sendmail** – →Programm, das in →E-Mail-Systemen die von einem →Mail User Agent übergebenen Nachrichten an ihren Bestimmungsort weiterleitet.

**sequenziell** – Schrittweise hintereinander.

**Server** – →Programm oder Computersystem, das Dienste oder →Ressourcen über ein →Netzwerk anbietet.

**SGML** – *Standard Generalized Markup Language*. →ISO-Standard, mit dem die Struktur und die Techniken für die Verwaltung digitaler Dokumente definiert wird.

**Shared Memory** – Kommunikationstechnik, bei der derselbe Speicherbereich von mehreren Programmen genutzt werden kann, ohne dass diese kollidieren.

**Shareware** – →Software-Vertriebsform, bei welcher der Nutzer die Software über einen begrenzten Zeitraum kostenfrei verwenden kann, um sie zu evaluieren. Gefällt das Produkt, wird erwartet, dass Sie sich gegen eine (meist, verglichen mit üblichen kommerziellen Produkten, geringe) Lizenzgebühr registrieren lassen, wonach Sie in der Regel eine gedruckte Dokumentation, technische Unterstützung und (regelmäßige) Informationen über neue →Programm-Versionen erhalten.

**Shell** – Ein →Befehlszeileninterpreter von UNIX. Die Shell analysiert die vom →User eingegebenen Befehle und gibt sie zur Verarbeitung an das eigentliche →Betriebssystem weiter.

**Shell-Datei** – →Skript

**Shell-Escape** – Das Zeichen „!“. Es wird innerhalb eines →Programms verwendet, um zu signalisieren, dass der folgende Befehl zur Verarbeitung an die →Shell weitergegeben werden soll und nicht vom Programm selber verarbeitet wird.

**Shell-Funktion** – Eine Reihe von Befehlen, die von der →Shell zur späteren Ausführung gespeichert werden.

**Shell-Out** – Das Aufrufen einer →Subshell aus einer →Applikation heraus.

**Shell-Programm** – →Skript

**Shell-Programmiersprache** – Integrierte Programmiersprache einer →Shell für die Erstellung von →Skripten

**Shell-Prompt** – →Prompt

**Shell-Skript** – →Skript

**Shutdown** – →Shell-Skript, mit dem der Systemadministrator das System geregelt herunterfährt.

**Signal** – Kurze Nachricht, die vom UNIX-→Kernel oder von einem →User an einen →Prozess gesendet werden kann.

**Signatur** – Textdatei, die automatisch an das Ende einer →E-Mail gehängt werden kann; sie enthält meist Ihren Namen, die →E-Mail-Adresse und häufig noch einen (manchmal mehr, manchmal weniger) geistreichen Spruch.

**Single-User-Status** – UNIX-Status, in dem nur der Superuser Zugriff über die Konsole auf das System hat; dient dazu, Wartungsarbeiten am System vornehmen zu können.

**Single-User-System** – →Betriebssystem, das nur von einer Person zur Zeit verwendet werden kann. Klassische Single-User-Systeme sind Win/DOS, OS/2 und das Macintosh System 7.

**Sitzung** – Die Zeitdauer zwischen →Login und →Logout, während der ein →User das System nutzt.

**Skalierbarkeit** – Die Möglichkeit, ein System beliebig erweitern zu können, um es in seiner Größe und Verarbeitungs-/Übertragungsgeschwindigkeit gestiegenen Anforderungen entsprechend anzupassen.

**Skript** – Liste mit →Befehlen einer →Shell-Programmiersprache, die bei Aufruf interpretiert von der →Shell abgearbeitet wird. Hierbei kann es sich im einfachsten Fall um eine →sequenzielle Folge von Befehlen handeln, die aber bei komplexeren Skripten durch Kontrollstrukturen in ihrer Abarbeitung gesteuert werden.

**SLED** – *Single Large Expensive Disk*. Alternative zu →RAID, bei der eine große Festplatte zur Datensicherung verwendet wird.

**Sleep** – Die zwischenzeitliche Unterbrechung bei der Ausführung eines →Prozesses, wie sie z. B. beim Warten auf zusätzliche Benutzereingaben erfolgt.

**SLIP** – *Serial Line Internet Protocol*. Übertragungsprotokoll, das für →TCP/IP über serielle Direktverbindungen zwischen Computersystemen oder einen Verbindungsaufbau über Fernsprechleitungen in Verbindung mit einem Modem ermöglicht.

**Smart Terminal** – Terminal, das unabhängig vom →Host Funktionen ausführen kann.

**Smiley** – Zeichenkombination, die dazu verwendet wird, Emotionen zu repräsentieren (→Emotikon).

**SMP** – →Symmetrisches Multiprocessing

**SMTP** – *Single Mail Transport Protocol*. →TCP/IP→Protokoll zur Steuerung von →E-Mail-Übertragungen.

**Snailmail** – Ironische Bezeichnung für Briefpost.

**Sneakernet** – Etwas gehässige Bezeichnung für Umgebungen, in denen Daten auf eine Diskette kopiert werden müssen, damit sie auf einen anderen Computer übertragen werden können, weil kein →Netzwerk vorhanden ist. Auch als „Turnschuh-Netzwerk“ bekannt.

**Sniffer** – Programm zum Ausspionieren des →Netzwerk-Verkehrs eines Systems, um an →Login-IDs, →Passwörter etc. heranzukommen.

**Socket** – Verfahren zur →Interprozess-Kommunikation zwischen →Prozessen, die nicht zur gleichen Zeit auf demselben System ausgeführt werden.

**Soft Link** – Zugriff von einer Datei auf eine andere mittels ihres Dateinamens (nicht ihres →I-Nodes); →symbolischer Link.

**Software** – Nicht physikalische Komponenten eines Computersystems, wie das →Betriebssystem, →Applikationen, Treiber usw. Andere Bezeichnung für →Programme.

**Solaris** – Von SunSoft entwickelte, auf →System V Release 4 basierende UNIX-Version für INTEL-basierte PCs und Sun →Workstations.

**Sonderzeichen** – Zeichen, das für das →Betriebssystem eine spezielle Bedeutung hat.

**sortieren** – Eine Gruppe von Elementen in eine logische Reihenfolge bringen, wobei hierfür unterschiedliche Kriterien verwendet werden können, wie z. B. eine alphabetische, eine nummerische, eine Größen- oder Datums-orientierte Sortierung.

**Spamming** – Die →Mailbox eines →Users mit (unerwünschter) →E-Mail überfluten.

**spawn** – →fork

**SPEC-Benchmarks** – *System Performance Evaluation Cooperation Benchmark*. Eine Reihe von standardisierten Tests, die zur Messung der Leistung von →Workstations dienen (sechs dieser Tests überprüfen die Leistungsfähigkeit von Fließkomma-Operationen, vier die von Integer-Operationen).

**speichern** – Eine neue Datei erstellen oder eine bereits existierende mit (geänderten) Informationen überschreiben.

**Spooft** – →Sniffer

**spool** – *Simultaneous peripheral operation on-line*. Verfahren, bei dem Informationen vor der Übertragung auf ein →Peripheriegerät (auf der Festplatte) zwischengespeichert werden, bis das entsprechende →Gerät zur Verfügung steht.

**Spooler** – →Software für die Verwaltung von Druckaufträgen; sie →spools jeden Druckauftrag in einer eigenen Datei, so dass diese nacheinander geordnet ausgedruckt werden können.

**Stammverzeichnis** – Das Verzeichnis, vom dem alle anderen Verzeichnisse abzweigen (→Root).

**Standard-Fehlerausgabe** – Datei, an die ein →Programm Fehlermeldungen ausgibt. Wird keine →Ausgabeumlenkung vorgenommen, ist dies der Bildschirm.

**Standardausgabe** – Datei, an die ein →Programm Meldungen ausgibt. Wird keine →Ausgabeumlenkung vorgenommen, ist dies der Bildschirm.

**Standardeingabe** – Datei, von der ein →Programm Eingaben erhält. Wird keine →Eingabeumlenkung vorgenommen, ist dies die Tastatur.

**Standardverzeichnisse** – Das →Dateisystem, das in seiner Grundstruktur auf allen UNIX-Systemen zu finden ist.

**Stapelspeicher** – Reservierter Speicherbereich, in dem →Programm-interne Operationen protokolliert werden.

**Startbit** – Das →Bit, das den Beginn eines neuen Datenworts (→Byte) in der asynchronen Kommunikation kennzeichnet.

**Startup-Datei** – Datei, die von der →Login-Shell beim →Einloggen in ein System standardgemäß ausgeführt wird, um individuelle Anpassungen vorzunehmen.

**Status** – Zugriffsberechtigung, die den einzelnen →Usern zugeordnet wird.

**Statuszeile** – Die letzte Zeile in der Bildschirmsdarstellung einer →Applikation, in der Informationen zum aktuellen Betriebsablauf enthalten sind.

**Sticky Bit** – →Bit, das mit einer ausführbaren Datei verknüpft ist; es bewirkt, dass ein →Programm nach seiner Ausführung im Speicher verbleibt, so dass es von anderen →Usern schneller geladen werden kann.

**Stopbit** – Das →Bit, welches das Ende eines Datenworts (→Byte) in der asynchronen Kommunikation kennzeichnet.

**store-and-forward** – Verfahren, bei dem Nachrichten vor der Weiterleitung an ihre ursprüngliche Zieladresse zwischengespeichert werden, um beispielsweise Spitzenzeiten im →Netzwerk-Verkehr zu umgehen.

**Stream** – Eine Abfolge ähnlicher Eingaben.

**Streamerkassette** – Kassettensystem mit sehr hoher Datenübertragungsgeschwindigkeit zum Speichern/Wiederherstellen von Daten.

**String** – Abfolge von null oder mehr Zeichen; eine Zeichenkette.

**Subshell** – Duplikat der übergeordneten →Shell, die einen Befehl oder eine Liste von Befehlen ausführt.

**subscribe** – Sich bei einer →USENET-Newsgroup oder einer →Mailing-Liste anmelden.

**Such-String** – Zeichenkettenfolge, nach der gesucht werden soll.

**Suchmuster** →Such-String

**Suchmusterübereinstimmung** →Pattern Matching

**Suchpfad** – Eine Liste mit →Verzeichnissen, welche die →Shell nach ausführbaren →Programmen durchsuchen soll.

**Sun Microsystems** – Hersteller von Hochleistungs→Workstations, die mit →SunOS/→Solaris betrieben werden.

**SunOS** – UNIX- →Betriebssystem von SunSoft. →Solaris.

**Superblock** – →Block, der die Steuerinformationen (Größe, Anzahl der →I-Nodes, Position freier Blöcke usw.) für ein →Dateisystem enthält.

**Supercomputer** – Computer von extremer Leistungsfähigkeit, der bis zu mehreren Billionen Operationen in der Sekunde ausführen kann.

**Superserver** – UNIX-basiertes Multiprozessorsystem, das als Datei→Server in Netzwerken dient.

**Superskalar** – Mikroprozessor-Architektur, bei der mehr als eine Anweisung pro Taktzyklus ausgeführt werden kann, da sie über mehr als eine Ausführungseinheit verfügt.

**Superuser** – Ein privilegierter →User, der über mehr Zugriffsrechte verfügt, um Verwaltungsaufgaben durchführen zu können, die anderen →Usern aus Sicherheitsgründen untersagt sind.

**surfen** – Sich den Weg durch den multimedialen Teil des Internets, des →World Wide Web, mit seinen Angeboten bahnen.

**suspendieren** – Einen →Prozess zeitweilig stoppen, um ihn dann später im Vorder- oder Hintergrund weiterlaufen zu lassen.

**Swap-Bereich** – Ein Bereich der Festplatte, auf dem Teile von laufenden →Prozessen aus dem Speicher ausgelagert werden.

**swappen** – Der Vorgang des Auslagerns von →Prozessen aus dem Arbeitsspeicher auf die Festplatte, um diesen anderen →Prozessen zur Verfügung stellen zu können, so dass mehr →Prozesse geladen und ausgeführt werden können, als der tatsächliche physikalische Arbeitsspeicher es sonst zulassen würde.

**Swapping** →swappen

**Symbolischer Link** →Link im →Dateisystem, das den Pfadnamen einer anderen Datei referenziert.

**Symlink** →Symbolischer Link

**Symbolischer Debugger** →Software-Entwicklungsprogramm zur Fehlersuche in anderen →Programmen.

**symmetisches Multiprocessing** – Kurz: SMP. Verfahren, bei dem die gesamte Arbeitslast eines Systems auf mehrere Prozessoren verteilt wird.

**Syntax** – Die formalen, grammatischen Regeln, die bei der Ausführung von Befehlen oder bei der Programmierung beachtet werden müssen.

**Syntaxfehler** – Fehler in der Handhabung der →Syntax einer Programmiersprache oder eines Befehls.

**System III** – →AT&T-Version von UNIX, das als erstes System auch auf INTEL-basierten PCs ablauffähig war. Grundlage für die SCO-UNIX-Version XENIX.

**System V** – Aktuelle UNIX-Version mit dem „vollen“ Namen System V Release 4.2 (auch SVR4 oder SVR4.2 abgekürzt).

**System herunterfahren** – Ordnungsgemäßes Beenden des →Betriebssystems, das durchgeführt werden muss, um Datenverluste zu vermeiden.

**Systemadministration** – Verwaltungsaufgaben, die vom →Systemadministrator durchgeführt werden müssen; hierzu gehören:

- Das Starten und ggfs. ordnungsgemäße Wieder-Herunterfahren des Systems
- Einstellungen von Systemdatum und Zeit
- Die Zuweisung von →User- und →Gruppen→Accounts mit den dazugehörigen →Passwörtern
- Die Installation, Aktualisierung und das Löschen von →Applikationen und neueren →Betriebssystemsversionen
- Die Erstellung von →Backups
- Das Einrichten und Entfernen von →Dateisystemen
- Die Überwachung der Systemsicherheit
- Das Hinzufügen und Entfernen von →Hardware-Komponenten usw.

**Systemadministrationsbefehle** – Befehle, die ausschließlich dem →Systemadministrator zur Verwaltung eines UNIX-Systems zur Verfügung stehen.

**Systemadministrator** – Person, die für die Verwaltung eines UNIX-Systems verantwortlich ist. Dieser hat →Superuser-Privilegien. Bei sehr großen Systemen kann es erforderlich sein, mehrere Systemadministratoren zur Bewältigung aller anfallenden Aufgaben zu beschäftigen.

**Systemaufruf** – Der Vorgang, dem →Kernel mitzuteilen, dass eine bestimmte Operation (die Steuerung eines →Prozesses, die Verwaltung von Ein-/Ausgaben, das Ansprechen eines →Peripherie-Geräts) auszuführen ist.

## T

**tar-Archiv** – →Archivdatei, die mit dem UNIX-Befehl tar erstellt wurde; auch Tarball genannt.

**Task** – Einzelement eines laufenden →Programms, wobei nicht unterschieden wird, ob es sich hierbei um einen →Prozess des →Betriebssystems oder einer →Applikation handelt.

**Task-Switching** – Wechseln zwischen einzelnen →Tasks, um eine bessere Ausnutzung von System→Ressourcen zu erreichen.

**Tastaturpuffer** – Systemspeicherplatz, der zum Speichern von unmittelbar vorausgegangenen Tastatureingaben verwendet wird.

**Tastenpuffer** – Verfahren, bei dem Tastatureingaben, die nicht unmittelbar verarbeitet werden können, zwischengespeichert und dann zum nächstmöglichen Zeitpunkt verarbeitet werden.

**TB** – →Terabyte

**TCL** – *Tool Command Language*. Erweiterbare →Skript-Sprache, die als →C-Bibliothek zur Verfügung gestellt wird.

**TCP** – *Transmission Control Protocol*. Für den Verbindungsaufbau zuständiges →Protokoll auf der Ebene der →Transportschicht.

**TCP/IP** – *Transmission Control Protocol/Internet Protocol*. Eine Reihe von Kommunikations-→Protokollen zur Steuerung des Medienzugriffs, der Dateiübertragung, der →E-Mail-Funktionen usw. Hierzu gehören:

- **ARP** – *Address Resolution Protocol*. Konvertiert →Internet- und →Ethernet-Adressen.
- **ICMP** – *Internet Control Message Protocol*. Dient zur Fehlerprüfung.
- **PPP** – *Point-to-Point Protocol*. Stellt synchrone und asynchrone Netzwerkverbindungen zur Verfügung.
- **RARP** – *Reverse Address Resolution Protocol*. Konvertiert →Ethernet- und →Internet-Adressen.
- **SLIP** – *Serial Line Internet Protocol*. Dient der IP-Implementierung über serielle Leitungen.
- **SNMP** – *Simple Network Management Protocol*. Dient zur Durchführung verteilter Netzwerkverwaltungsaufgaben.
- **UDP** – *User Data Protocol*. Dient dem Verbindungsaufbau.

**TCP/IP-Architektur** – Die vier Schichten des TCP/IP-Protokollpaket; dies sind:

- Die Anwendungsschicht
- Die Host-zu-Host-Transportschicht
- Die Internet-Schicht
- Die Netzwerkzugriffs-Schicht

**Tcsh** – Weiterentwicklung der →C-Shell.

**Telarbeit** – Beschäftigungsform, bei welcher der Angestellte vornehmlich zu Hause an seinem Computer arbeitet, der über Modem/Telefonleitung mit dem →Netzwerk seiner Firma verbunden ist, statt ins Büro fahren zu müssen.

**Telekonferenz** – Fernkonferenz, die mit Hilfe von Audio- und Videoverbindungen über Computersysteme abgehalten wird.

**Telnet** – Protokoll, das einen Verbindungsaufbau zu einem entfernten →Host ermöglicht, so dass an diesem gearbeitet werden kann, als sei man direkt mit einem →Terminal an diesen angeschlossen.

**Temp-Verzeichnis** – →Verzeichnis, das übergangsweise (temporär) Dateien speichert.

**10/100** – Eine Bezeichnung für Geräte, die sowohl →Ethernet mit 10 →Mbps als auch →Fast Ethernet mit 100 →Mbps unterstützt.

**Terabyte** – TByte.  $2^{40} = 1.099.511.627.776$  →Byte.

**Terminal** – Kombination aus Tastatur und Bildschirm.

**Terminal-Emulation** – →Programm, mit dem ein PC oder eine →Workstation betrieben werden kann, als ob er oder sie ein →Terminal wäre.

**T<sub>E</sub>X** – Professionelles Satzsystem, das als Grundlage einfache Textdateien mit Steuerinformationen zu hochqualitativen DTP-Dokumenten verarbeitet; speziell für wissenschaftliche Textverarbeitung geeignet.

**Texteditor** – →Programm zum →Editieren von →Textdateien.

**Textdatei** – Datei, die ausschließlich aus reinen Text- (→ASCII-) Zeichen besteht, ohne Steuerinformationen wie z. B. Formatierungselemente zu enthalten.

**Textverarbeitung** – →Programm für die Erstellung von Textdokumenten, die Formatierungselemente (Zeichenformatierungen wie Fettdruck, Kursivdruck, etc., Absatzformatierungen wie Blocksatz, Zentrierungen etc. und weitere Formatierungen) enthalten kann.

**Thick-Ethernet-Kabel** – Koaxial-Verbindungsleitung, die Verwendung in →Ethernet-basierten →Netzwerken mit Entfernung bis zu 1.000 Metern zwischen den einzelnen Knoten findet.

**Thin-Ethernet-Kabel** – Koaxial-Verbindungsleitung, die Verwendung in →Ethernet-basierten →Netzwerken mit Entfernung bis zu 165 Metern zwischen den einzelnen Knoten findet.

**Thread** – Zur gleichen Zeit ablaufender →Prozess als Teil eines übergeordneten →Prozesses, wie er in →Multitasking→Betriebssystemen bei mehreren gleichzeitig ablaufenden →Programmen auftreten kann. Auch: thematisch zusammenhängende Reihe von Artikeln in einer →USENET→Newsgroup.

**Thread-Newsreader** – →Newsreader, der thematisch zusammenhängende Artikel gruppieren, um so den Diskussionsfortgang besser verfolgen zu können.

**Ticket** – Markierung, die von →Kerberos-Systemen verwendet wird und den Benutzernamen, die Adresse und die vom Benutzer angeforderten Dienste aufführt.

**Tilde** – Das Zeichen „~“. Bezeichnet u. a. in der →Korn- und der →C-Shell, der →bash, der →tcsh und der →Zsh das →Home-Directory.

**Timestamp** – Uhrzeit, zu der eine Datei das letzte Mal aufgerufen oder zu der sie oder der zu ihr gehörende I-Node verändert wurde.

**tn3270** – Variante des →Telnet-Protokolls, um auf IBM-Großrechner zugreifen zu können.

**Toggle** – Das Hin- und Herschalten zwischen zwei Befehlsoptionen.

**Toolkit** – Sammlung von Hilfsprogrammen zur →Software-Entwicklung.

**Trackball** – Eingabegerät, das alternativ zu einer →Maus verwendet werden kann, wobei bei diesem nicht das gesamte Gehäuse zur Positionierung des Zeigers verschoben wird, sondern die sich in ihm befindende Kugel.

**Transportschicht** – Die vierte der sieben Schichten des →ISO/OSI-Referenzmodells, welche die →Protokolle für Nachrichtenstrukturen definiert und eine korrekte Übermittlung/Datenübertragung koordiniert und Verbindungen aufrechterhält.

**Trap** – Anweisung, die als Schnittstelle zwischen einer →Applikation und dem →Kernel verwendet wird und bewirkt, dass der →Kernel den erforderlichen Dienst oder Systemaufruf ausführt.

**Trojanisches Pferd** – Variation eines Computervirus, die vorgibt, ein nützliches →Programm zu sein, so dass neugierige Benutzer es ausprobieren, wodurch seine (zer)störende Funktion aktiviert wird.

**Tux** – Der LINUX→Pinguin

**Twinaxial-Kabel** – Kabel mit zwei Koaxial-Adern, die von einer Isolierabschirmung ummantelt sind.

**Twisted-Pair-Kabel** – Abgeschirmtes, paarig verdrilltes Kupferdrahtkabel mit Ummantelung, bei dem jede Ader separat abgeschirmt ist.

**Typeahead** – Eingaben, die vor Erscheinen des →Prompts vorgenommen werden.

## U

**Übergeordnetes PID** – →PPID.

**übergeordneter Prozess** – →Prozess, der Verzweigungen zu anderen →Prozessen besitzt. Dieser kann über sein →PPID identifiziert werden.

**übergeordnetes Verzeichnis** – Das →Verzeichnis, das innerhalb eines hierarchischen →Dateisystems oberhalb des →aktuellen Verzeichnisses steht.

**Überlastung** – Situation, in der ein virtuelles Speichersystem nur noch mit dem Ein- und Auslagern von →Programmen beschäftigt ist, ohne dass Zeit verbleibt, Anwendungen weiter auszuführen.

**UID** – *User identification number*. Ein Zahl zur Identifizierung eines →Users, auf die beispielsweise bei der Verwaltung der →Dateizugriffsberechtigungen zurückgegriffen wird.

**ULTRIX** – UNIX-Version der Firma Digital Equipment, die in ihrem äußeren Erscheinungsbild und ihrer Funktionalität →BSD-UNIX entspricht.

**umask-Wert** – →Oktaler Wert, der bei der Erstellung neuer Dateien Zugriffsberichtigungen festlegt.

**Umgebungsvariable** – →Variable, die der →Shell zur weiteren Manipulation zur Verfügung gestellt wird.

**Umlenkung** – Verfahren, bei dem die Standardein- oder Standard(fehler)ausgabe aus/in einer Datei erfolgt, statt durch ein →Terminal.

**Unbuffered Device** – →Raw Device

**ungeschirmtes Kabel** – Kabel, das keine Ummantelung zur Abschirmung elektromagnetischer Einflüsse oder Hochfrequenzen besitzt.

**ungeschirmtes Twisted-Pair-Kabel** – UTP. Kabel mit zwei oder mehr verdrillten Kupferdrahtkabeln, wobei der Schutz vor Interferenzen desto größer ist, je mehr Verdrillungen es besitzt.

**Unicode** – 16→Bit-Code, der somit 65.536 verschiedene Zeichen darstellen kann.

**Uniform Resource Loader** – →URL

**Uninterruptible Power Supply** – →UPS

**UNIX-Lookalike** – →Betriebssystem, das so aussieht wie UNIX und seine Funktionalität besitzt, aber keinen →AT&T-Code enthält.

**UnixWare** – Auf →System V Release 4.2 basiertes UNIX, das von →SCO vertrieben wird.

**Unmoderierte Newsgroup** – →USENET→Newsgroup, die nicht von einem →Moderator betreut wird, so dass die in ihr erscheinenden Artikel von niemandem auf ihre Inhalte und Form überprüft werden.

**Unnamed Pipe** – Datei, welche die Standardausgabe eines →Programms an die Standardeingabe eines weiteren →Programms weiterleitet.

**unsichtbare Datei** – →Dot-Datei

**unsichtbares Zeichen** – Jedes nicht darstellbare Zeichen, wie das Leerzeichen, →Backspace oder die Steuerzeichen.

**untergeordneter Prozess ohne Zuordnung** – →Prozess, der keine Zuordnung mehr zu seinem übergeordneten →Prozess besitzt.

**Unterverzeichnis** – →Verzeichnis, das in einem anderen →Verzeichnis enthalten ist. Da jedes →Verzeichnis von der →Root abzweigt, ist somit genaugenommen jedes →Verzeichnis außer der →Root ein Unterverzeichnis.

**unsubscribe** – Das Abonnement einer →USENET-Newsgroup abmelden.

**Update** – Version einer →Software, die Fehlerkorrekturen enthält. Sie ist an der Veränderung der →Versionsnummer hinter dem Dezimalpunkt zu erkennen (Version 1.3 ist ein Update zur Version 1.2).

**Upgrade** – Version einer →Software, die zusätzliche Leistungsmerkmale enthält. Sie ist an der Veränderung der →Versionsnummer vor dem Dezimalpunkt zu erkennen (Version 2.0 ist ein Upgrade zur Version 1.3).

**UPS** – *Uninterruptible Power Supply - Unterbrechungsfreie Stromversorgung.* Stromquelle zur Unterstützung von Computersystemen bei Stromausfall oder Leistungsabfall.

**URL** – *Uniform Resource Loader.* Verfahren für den Zugriff auf →Internet→Ressourcen, wobei sowohl die Zugriffsmethode als auch Art und Inhalt der Ressource selber in dieser enthalten sind.

**USENET** – *User Network.* Internationales, nicht kommerzielles →Netzwerk, das in Form von →Newsgroups Kommunikationsforen für die unterschiedlichsten Themen (es gibt Tausende verschiedener Newsgroups ...) zur Verfügung stellt.

**USENET-Artikel** – Individuelle Nachricht, die an eine der →USENET→Newsgroups gesendet wird.

**User** – Dt. Benutzer. Person, die einen →Account auf einem System besitzt und somit mit diesem arbeiten darf.

**Utility** – Dt. Werkzeug. Ein Hilfs→Programm zur Erledigung bestimmter begrenzter Aufgaben.

**UUCP** – Verfahren, das zur seriellen Kommunikation zwischen UNIX-Systemen verwendet wird.

## V

**Variable** – Ein Datenelement, dem unterschiedliche (variable) Werte zugewiesen werden können.

**Versionsnummer** – Von einem Entwickler verwendetes Verfahren, um bestimmte Versionen einer →Soft- oder →Hardware zu kennzeichnen.

**vertikale Anwendung** – Spezialisierte →Applikation, die für einen begrenzten Markt entwickelt wurde.

**Verzeichnisbefehle** – UNIX-Befehle, die ausschließlich für die Manipulation von →Verzeichnissen geeignet sind.

**Verzeichnis** – →Directory

**VG-AnyLan 100** – Modifikation des →Ethernet-Standards, die Geschwindigkeiten zwischen 10 und 100 →Mbps ermöglicht.

**virtuelle Konsole** – Auf vielen UNIX-Systemen zur Verfügung stehende Funktionalität, die es ermöglicht, durch das Drücken bestimmter Tastenkombinationen zwischen verschiedenen Bildschirmen (die physisch alle auf dem gleichen Monitor angezeigt werden) hin- und herzuschalten, um sich beispielsweise unter mehreren →User-Namen gleichzeitig →einloggen zu können.

**virtueller Speicher** – Verfahren bei der Speicherverwaltung, bei der Informationen aus dem Arbeitsspeicher auf die Festplatte ausgelagert werden.

**Vordergrund** – Verarbeitungsumgebung, in der ein →Programm direkt von der Eingabe des →Terminals gesteuert wird.

**Vordergrundverarbeitung** – Verfahren, bei dem ein →Programm in den →Vordergrund gestellt wird, damit direkte Eingaben vom →Terminal aus möglich sind.

## W

**WABI** – *Windows Application Binary Interface*. →Programm, das es ermöglicht, 16-Bit-Windows-→Applikationen unter UNIX laufen zu lassen.

**WAIS** – Dienst, mit dem auf Textdatenbanken im →Internet zugegriffen werden kann.

**Wakeup** – Eine Ausführung nach einem →Sleep wieder aufnehmen.

**Warteschlange** – Eine temporäre Liste mit Elementen, die der Reihe nach abgearbeitet werden.

**Web-Browser** – →Client→Applikation, um im →World Wide Web angebotene →Hypertext-Dokumente anzuzeigen.

**Whetstones** – →Benchmark-Verfahren, bei dem die Leistungsfähigkeit unterschiedlicher Computersysteme miteinander verglichen wird.

**Whitespace character** – →Blank character

**Widget** – Element, das Programmierern die Gestaltung von grafischen Benutzeroberflächen (→GUIs) erleichtert.

**Wiederherstellen** – Laden von Dateien aus einem →Backup, um defekte oder manipulierte Daten zu korrigieren.

**Wildcard** – „Jokerzeichen“ in einem Dateinamen, das eine Reihe möglicher Zeichen repräsentiert, um beispielsweise Dateigruppen zu referenzieren.

**Window-Manager** – →X Window→Programm, über das festgelegt werden kann, wie Fenster in einer →GUI dargestellt und wie die Interaktionstechniken zwischen →User und System zur Verfügung gestellt werden.

**Workgroup** – Eine Arbeitsgruppe von Personen, die gemeinsam zur Bearbeitung von Dateien über ein →Netzwerk auf diese zugreifen können.

**Workstation** – Kleiner Hochleistungscomputer.

**World Wide Web** – Kurz: WWW, W3 oder einfach „Das Web“. Multimедialer Teil des →Internet, in dem die Informationen auf →Hypertext-Seiten dargestellt werden.

## X

**X Consortium** – Herstellerkonsortium, das Produkte entwickelt, die auf der →X Window-Spezifikation basieren.

**X Terminal** – Grafik→Terminal für den Gebrauch des →X Window-Systems mit den für dieses entwickelten →Applikationen.

**X Window** – →GUI für die Verwendung unter UNIX-Systemen.

**X Client** – →X Window→Applikation.

**X Server** – →Server, der →X Clients →Ressourcen zur Verfügung stellt.

**X.400** – Standard für öffentliche oder private →E-Mail-Verteilungssysteme, der festlegt, wie Nachrichten über ein →Netzwerk oder zwischen zwei oder mehreren miteinander verbundenen →Netzwerken übertragen werden.

**X.500** – Standard für ein globales Verzeichnissystem, das einem unter →X.400 laufenden →E-Mail→Programm zum Lokalisieren von →E-Mail-Anwendern dient.

**X/Open** – Europäische Organisation, deren Ziel es ist, Standards für →offene Systeme zu definieren.

**XENIX** – Eine ursprünglich von Microsoft und →SCO gemeinsam entwickelte UNIX-Version, die seit 1987 von SCO alleine vertrieben wurde.

**X11R3, X11R4, X11R5, X11R6** – Abkürzungen der verschiedenen Versionen der X11-Spezifikation für →X Window.

**XFree86** – Version des X11R6-X Window-Systems, das kostenlos für INTEL-basierte UNIX-Systeme erhältlich ist.

**Xmodem** – Populäres Datenübertragungs→Protokoll, mit dem jeweils mit einer →Checksum versehene →Pakete von 128 Byte Größe übertragen werden.

## Y

**Yellow Pages** – Der ursprüngliche Name des *Network Information Service* (→NIS).

**Ymodem** – Populäres Datenübertragungs→Protokoll, mit dem jeweils mit einer →Checksum versehene →Pakete von 1 KByte Größe übertragen werden.

## Z

**Zmodem** – Populäres Datenübertragungs→Protokoll, mit dem →Pakete unterschiedlicher (von der Leitungsqualität abhängiger) Größe übertragen werden. Bei einem Abbruch der Verbindung kann mit diesem Protokoll (im Gegensatz zu →Xmodem und →Ymodem) nach erneutem Verbindungsaufbau an der gleichen Stelle mit der Übertragung fortgefahren werden, ohne die bereits übertragenen Datenpakete neu senden zu müssen.

**Zombie** – Ein toter →Prozess, der noch nicht aus der Prozesstabellen gelöscht wurde. Meist harmlos, da er – außer seiner Position in der Prozesstabellen – keinen Speicherplatz verbraucht und meist auch sofort verschwindet. Hartnäckigere Zombies lassen sich manchmal nur durch erneutes →Booten des Systems entfernen.

**Zsh** – Für die →Bourne- und die →C-Shell als Ersatz gedachter Befehlszeileninterpreter mit erweiterten Funktionen.