

Arnold Willemer

UNIX

Das umfassende Handbuch

Für alle
UNIX- und
Linux-Systeme
geeignet

- Installation, Konfiguration, Anwendung
- Systemadministration, Netzwerke, Programmierung
- Datensicherung, Optimierung, Einsatz als Server

Galileo Computing

Inhalt

Vorwort	23
---------------	----

TEIL I: KONZEPTE

1	Konzepte	31
1.1	Dateien	32
1.1.1	Dateitypen	32
1.1.2	Dateinamen	33
1.2	Datenstrom	34
1.3	Verzeichnisse	35
1.3.1	Umgang mit Verzeichnissen	35
1.3.2	Der UNIX-Verzeichnisbaum	36
1.3.3	Was ist wo?	36
1.3.4	Einbinden von Speichermedien	39
1.3.5	Ein Blick unter die Haube: i-nodes	42
1.4	Schichten und Shells	43
1.5	Das offene System	44
1.6	Mehrbenutzersystem	45
1.6.1	Eigentumsrechte von Dateien und Verzeichnissen	46
1.6.2	Der Administrator	47
1.7	Konsequenz: Sicherheit und Wartbarkeit	48

TEIL II: ANWENDUNG

2	Bedienung eines UNIX-Systems	51
2.1	Anmelden: Personenkontrolle	51
2.2	Fragen Sie Dr. UNIX	53
2.2.1	Referenzhandbuch man	53
2.2.2	info	56
2.2.3	Howto	58
2.2.4	Internet	58
2.3	So sage ich es meinem UNIX	60
2.4	Operationen mit Dateien	61
2.4.1	Eine kleine Beispielsitzung	62
2.4.2	Dateien auflisten: ls	64
2.4.3	Dateien kopieren: cp	70

2.4.4	Dateien verschieben oder umbenennen: mv	71
2.4.5	Dateien löschen: rm	72
2.5	Verzeichnisbefehle	73
2.5.1	Navigation	73
2.5.2	Verzeichnis anlegen: mkdir	75
2.5.3	Verzeichnis löschen: rmdir	76
2.6	Dateieigenschaften	76
2.6.1	Eigentümer wechseln: chown	77
2.6.2	Gruppenwechsel: chgrp	77
2.6.3	Berechtigungen: chmod	78
2.6.4	Neuer Zeitstempel: touch	83
2.6.5	Links: Zwei Namen, eine Datei	84
2.6.6	Besondere Dateien	89
2.6.7	Der Dateityp: file	89
2.7	Zugriff auf mehrere Objekte	90
2.7.1	Wildcards: *, ? und die eckigen Klammern	90
2.7.2	Sonderzeichen als Parameter	91
2.8	Editoren	92
2.8.1	vi	92
2.8.2	emacs	102
2.9	Suche nach der richtigen Datei	106
2.9.1	Suchen und Agieren im Verzeichnisbaum: find ..	107
2.9.2	Suchen und Agieren im Verzeichnisbaum: locate	113
2.9.3	Programmsuche: which und whereis	114
2.10	UNIX-Kommandos verknüpfen	115
2.10.1	Ein- und Ausgabe als Datenstrom	115
2.10.2	Umleitung	116
2.10.3	Piping	118
2.10.4	Quoting: Verschachtelte Befehle	119
2.11	Praktische Helfer	119
2.11.1	Ausgabe einer Datei: cat	120
2.11.2	Seitenweise: more	120
2.11.3	Durchsuchungsbefehl: grep	121
2.11.4	Wenn ich auf das Ende sehe: tail	123
2.11.5	Anfangsbetrachtungen: head	123
2.11.6	Ausschnitt: cut	124
2.11.7	Teilen: split	124
2.11.8	Zeilen umbrechen: fold	125
2.11.9	Zeichenumcodierung: tr	125
2.11.10	Unterschiede zwischen Textdateien: diff	127

2.11.11	Dateien aufs Byte geschaut	128
2.11.12	Wortzähler: wc	129
2.11.13	sort	129
2.11.14	sed	130
2.11.15	awk	134
2.12	Reguläre Ausdrücke	138
2.13	Pack deine Sachen und geh	141
2.13.1	Verschnüren: tar	141
2.13.2	Zusammenpressen: compress und gzip	144
2.13.3	Kombination aus Packen und Pressen	145
3	Prozesse	147
3.1	Parallele Prozesse starten	147
3.2	Prozesse im Gänsemarsch	149
3.3	Prioritäten: nice	150
3.4	Prozessliste anzeigen: ps	151
3.5	Stoppen eines Prozesses: kill	153
3.6	Programmabbruch	154
4	Umgebungsvariablen	157
5	Die Shell	161
5.1	Bourne-Shell (sh) und POSIX	161
5.2	Korn-Shell (ksh)	162
5.3	C-Shell (csh)	165
5.4	Bourne-Again-Shell (bash)	168
5.5	Arbeiten mit der Shell	172
5.5.1	Die for-Schleife	172
5.5.2	alias	175
5.5.3	Startupdateien der Shell	175
5.5.4	Shell aus der Shell starten	176
6	Ausgaben auf dem Drucker	177
6.1	BSD-Unix: lpr, lpq und lprm	177
6.1.1	Start des Druckauftrags	177
6.1.2	Druckkontrolle	178
6.2	AT&T: lp, lpstat und cancel	179

6.3	Die neue Generation: LPRng und CUPS	180
6.4	Druck formatieren: pr und a2ps	180
6.5	Zeitversetztes Arbeiten	181
6.6	Die aktuelle Zeit	182
6.7	Regelmäßige Arbeiten: crontab	183
6.8	Zeitversetzter Job: at	185

7 Mit UNIX produktiv werden 187

7.1	Büroanwendungen	187
7.1.1	OpenOffice.org	188
7.1.2	Andere Office-Pakete	190
7.2	Das Satzsystem T _E X	190
7.3	Bildbearbeitung: GIMP	196
7.4	Musik	196
7.4.1	Musik aufnehmen	197
7.4.2	MP3	198
7.5	Ogg Vorbis	200
7.6	CDs und DVDs	201
7.6.1	Hintergrund	201
7.6.2	Audio-CDs abspielen	202
7.6.3	Audio-CDs auslesen	202
7.6.4	Daten-CDs einbinden	203
7.6.5	CDs brennen mit K3b	203
7.6.6	Audio-CDs von der Konsole brennen	206
7.7	Video	206
7.7.1	Bewegte Scheiben	208
7.7.2	Abspielprogramme	209
7.7.3	UNIX im Satellitenreceiver	209

TEIL III: ADMINISTRATION

8 Der Administrator 213

8.1	Sonderrechte	213
8.2	Die Arbeitsumgebung des Administrators	215
8.2.1	Minimalsystem	215
8.2.2	Vorsätzliche Behinderung	216
8.3	Administrationstools	217
8.3.1	Sinn und Unsinn der Admintools	218
8.3.2	Start über X11	218

8.3.3	Webmin: Administration per Browser	221
8.3.4	Herstellerspezifische Administrationstools	228

9 Starten und Stoppen 241

9.1	Start des Systems	241
9.1.1	Bootprompt	242
9.1.2	Bootkonfiguration: lilo	243
9.1.3	Der Bootmanager GRUB	244
9.1.4	Bootprobleme	245
9.1.5	Durchlaufen der Runlevel (System V)	246
9.1.6	BSD: /etc/rc	249
9.1.7	System V: init.d	250
9.1.8	Konfigurationsdateien	253
9.2	Herunterfahren: shutdown	254
9.2.1	Alles bereit zum Untergang?	255
9.2.2	Wechsel in den Single-User-Modus	256

10 Benutzerverwaltung 257

10.1	Die Benutzerdatei /etc/passwd	257
10.2	Aufbau der /etc/passwd	259
10.3	Verborgene Passwörter: shadow	262
10.4	Benutzerpflege automatisieren	264
10.5	Benutzer-Konfigurationsdateien	265
10.6	Verzeichnisprototyp: /etc/skel	267
10.7	Gruppenverwaltung	268
10.8	Benutzerüberwachung	269
10.8.1	Accounting	269
10.8.2	who und finger	270
10.9	Kurzfristiger Benutzerwechsel: su	271
10.10	Administrationsaufgaben starten: sudo	272
10.11	Pseudobenzutzer zum Shutdown	275

11 Hardware 277

11.1	Hardwarezugriff unter UNIX: /dev	277
11.1.1	Aufgaben eines Treibers	277
11.1.2	Gerätedateien	278
11.1.3	Umgang mit Gerätedateien	280

11.1.4	Gerätenamen	281
11.2	Festplatten	282
11.2.1	SCSI-Festplatten	282
11.2.2	IDE-Festplatten	283
11.2.3	SATA-Festplatten	284
11.2.4	Inbetriebnahme	285
11.2.5	RAID-Systeme	286
11.2.6	Partitionieren	290
11.2.7	Dateisystem erstellen	292
11.2.8	Swapping	293
11.2.9	Einbinden eines Dateisystems: mount	295
11.2.10	Konsistenz der Dateisysteme	301
11.2.11	Journal-Dateisysteme	302
11.2.12	Belegungslisten: df und du	303
11.2.13	Zuteilung des Festplattenplatzes: quota	305
11.2.14	Maximalwerte	307
11.3	Diskettenlaufwerke	309
11.3.1	Formatieren und Beschreiben	309
11.3.2	mount und eject	309
11.3.3	tar und sync	310
11.3.4	MS-DOS-Disketten	310
11.4	CD-ROMs	311
11.5	CD-Brenner	312
11.5.1	Datensicherung	313
11.5.2	RW-Medien	316
11.5.3	Multisession	316
11.5.4	IDE-Brenner	317
11.5.5	Daten-DVDs brennen	319
11.6	USB	320
11.6.1	Den USB-Port beobachten	320
11.6.2	USB-Sticks und USB-Laufwerke	322
11.7	Notebooks	323
11.7.1	Touchpad und Maus	324
11.7.2	PCMCIA	324
11.7.3	Ruhezustand	325
11.7.4	Problematische Peripherie	326
11.7.5	ACPI	327
11.7.6	APM: Advanced Power Management	328
11.7.7	Strom sparen	329

12 Datensicherung 331

12.1	Vorüberlegungen	331
12.2	Das Bandlaufwerk	334
12.3	Dateisystem sichern: dump	335
12.4	tar (tape archiver)	339
12.5	cpio	343
12.6	Medien kopieren: dd	346
12.7	Das Sicherungstool AMANDA	347
12.8	Kommerzielle Datensicherungen	350
12.9	Beispiel für eine Sicherung auf CD-RW	350
12.10	Archivierung	353

13 Installationen 355

13.1	Software installieren	355
13.1.1	make als Installationswerkzeug	356
13.1.2	Solaris Packages	357
13.1.3	HP-UX: SD-UX	358
13.1.4	Red Hat Package Manager	359
13.1.5	Debian Pakete APT	360
13.2	Betriebssystem installieren	361
13.2.1	Linux-Installation von CD	363
13.2.2	Installation von FreeBSD	366
13.2.3	Installation von Red Hat Linux über das Netzwerk	368
13.2.4	Installation von Solaris/86	370
13.2.5	Neuinstallation HP-UX	372
13.3	Nationale Besonderheiten	374
13.3.1	Umgebungsvariablen LANG und LC_TYPE	374
13.3.2	Tastaturbelegung	375

14 Weitere Peripherie und Hardware 377

14.1	Druckeradministration	377
14.1.1	Übersicht	378
14.1.2	BSD-Unix: lpd, lpr, lpq und lprm	378
14.1.3	Linux-PC als Druckserver	382
14.1.4	System V: lpsched, lp, lpstat und cancel	385
14.1.5	LPRng	389
14.1.6	CUPS – Common UNIX Printing System	390

14.2	Terminals	395
14.2.1	Konfiguration der Terminals	396
14.2.2	Die Terminalvariable TERM	398
14.2.3	termcap	398
14.2.4	terminfo	399
14.2.5	Wenn das Terminal durcheinander ist	400
14.3	Anschluss eines Modems	401
14.4	Scannen	402
14.4.1	xsane als Fotokopierer	402
14.4.2	Schrifterkennung (OCR)	402
14.5	Anschluss eines PDAs oder Mobiltelefons	403
15	Tuning	407
15.1	Optimierung des Dateisystems	407
15.1.1	Überfüllung der Dateisysteme vermeiden	407
15.1.2	Defragmentierung	408
15.1.3	Blockgröße	409
15.1.4	Verteilung auf mehrere Festplatten	409
15.1.5	Ein eigenes Dateisystem für /tmp	410
15.1.6	Übervolle Verzeichnisse entsorgen	410
15.2	Ressourcen kennen	411
15.3	Wissen, wo der Schuh drückt	413
16	Informationen sammeln	419
16.1	Versionsinformationen: uname	419
16.2	Der syslog-Dämon und die messages-Datei	420
16.3	syslog-Dämon der neuen Generation syslog-ng	423
16.4	Umgang mit großen Protokolldateien	429
16.4.1	Protokolldateien beobachten	429
16.4.2	Dateien stutzen und rotieren	430
16.4.3	Automatisches Rotieren: logrotate	432
16.5	Briefe aus dem Nirwana	433
16.6	Bootzeitpunkt und Systemlast: uptime	433
16.7	Prozessbeobachter	434
16.8	Nicht immer mit Tötungsabsicht: kill	439
16.9	Offene Dateien	441
16.10	Das Verzeichnis /proc	443
16.11	Programmmzusammenbrüche (Coredump)	444
16.12	Systemabsturz (Kernel-Panic)	445

17 Die Dateien des Betriebssystems 447

17.1	Der Kernel	447
17.2	Module	449
17.3	Dynamische Bibliotheken	451

TEIL IV: NETZWERK

18 Netzwerk 455

18.1	Client-Server-Architekturen	456
18.1.1	Ethernet als Verkabelungsbeispiel	456
18.1.2	Die Pseudoschnittstelle loopback	457
18.1.3	Pakete in Paketen	458
18.2	TCP/IP, der Standard	458
18.2.1	Die IP-Adresse	458
18.2.2	Das Prüftool ping	467
18.3	Routing: Verbindung mehrerer Netzwerke	469
18.3.1	Gateways	469
18.3.2	Statische Festlegung einer Route	470
18.3.3	Statisches Routing: Ein Beispiel	472
18.3.4	Subnetze	477
18.3.5	Dynamisches Routen	480
18.3.6	CIDR – Classless Inter-Domain Routing	481
18.4	Ohne Kabel: WLAN	482
18.4.1	Access Point	482
18.4.2	Grundinformationen	483
18.4.3	Sicherheitsaspekte	483
18.4.4	Softwaresteuerung des WLAN-Adapters	485
18.4.5	Treiber für WLAN-Adapter	486
18.4.6	Funkgesteuerte Peripherie: Bluetooth	487
18.5	Namensauflösung	489
18.5.1	Der Host- und Domainname	490
18.5.2	Die Datei /etc/hosts	491
18.5.3	Die Datei /etc/services	492
18.5.4	Netzgruppen: /etc/netgroup	494
18.5.5	Domain Name Service: DNS	495
18.5.6	Network Information Service: NIS	505
18.5.7	Portable Verzeichnisse LDAP	509
18.6	Dynamische IP-Adressen (DHCP)	514
18.6.1	DHCP-Clients	515

18.6.2 DHCP-Server	516
18.7 Next Generation IPv6	518

19 Netzinformationen sammeln 523

19.1 ICMP und ping	523
19.2 Verbindung zwischen Prozessen: netstat	525
19.3 Anzeigen der Netzwerkadapter	526
19.4 Anzeigen der Routingtabelle	527
19.5 Routen verfolgen: traceroute	528
19.6 tcpdump	528
19.7 Wireshark	529
19.8 iftop	530
19.9 HP-UX: lanadmin	532

20 Grundlegende TCP/IP-Dienste 533

20.1 Super-Server inetd und xinetd	533
20.2 File Transfer Protocol (FTP)	536
20.2.1 Der Client	537
20.2.2 Konfiguration des FTP-Servers	542
20.3 Anonymer FTP-Server	543
20.4 TFTP, schnell und vertrauensvoll	544
20.5 Terminaldienst (telnet)	544
20.5.1 telnet-Client	544
20.5.2 Ausloggen bei laufendem Prozess	547
20.5.3 telnet-Dämon	548
20.6 Die r-Kommandos	548
20.7 Wenn Sicherheit vorgeht: ssh und scp	553

21 Internetanschluss 559

21.1 Zugang zum Internet	559
21.2 Firewall und Masquerading	564
21.2.1 Funktionsweise einer Firewall	565
21.2.2 Masquerading	570
21.3 Proxy	571
21.4 Einbrucherkennung: Intrusion Detection System	575
21.5 Gefahren und Sicherheit	576

TEIL V: UNIX ALS SERVER

22	Netzwerkdateisysteme	581
22.1	NFS – Network File System	581
22.1.1	Automatisches Mounten	587
22.1.2	Beispiel: Dynamisches Benutzerverzeichnis	588
22.2	SAMBA: UNIX im Windows-Netz	590
22.3	Novell-Zugriffe	608
22.4	Mac im Netz: netatalk	610
22.5	Festplatte im Netz	611
22.6	Zeitabgleich	612
23	Datenbanken	615
23.1	SQL-Spickzettel	615
23.1.1	Data Definition Language (DDL)	616
23.1.2	Data Manipulation Language (DML)	618
23.2	MySQL	620
23.2.1	Installation	620
23.2.2	Benutzerverwaltung	621
23.2.3	Administrationstools	622
23.2.4	Anlegen von Datenbanken	623
23.2.5	Datensicherung	623
23.2.6	Start und Stopp	624
23.3	PostgreSQL	624
23.3.1	Installation	625
23.3.2	Benutzer anlegen	626
23.3.3	Anlegen von Datenbanken	627
23.3.4	Datensicherung	627
23.3.5	Start und Herunterfahren	628
24	E-Mail	631
24.1	E-Mails lesen	631
24.1.1	Lokale Mail lesen	631
24.1.2	Mail von einem Mailserver lesen	632
24.1.3	Verschlüsseln und Signieren	635
24.2	Format einer E-Mail	640
24.3	UNIX und Mail	642
24.4	SMTP (Simple Mail Transport Protocol)	642

24.5	SMTP mit Autorisierung	643
24.6	Mailqueue	645
24.7	Verteilen der Post: sendmail -q	645
24.8	Weiterleiten der Post: aliases und forward	646
24.9	POP3	647
24.9.1	Kommunikation laut RFC 1939	648
24.9.2	Eine kleine Beispielsitzung	650
24.10	IMAP	651
24.11	Post sammeln: fetchmail	653
24.12	Mail-Server und Domain	654
24.13	Erstes Beispiel: Interne Firmenmail	655
24.14	Zweites Beispiel: Anbindung an das Internet	656
24.15	Postfix, die Alternative zu sendmail	658

25 Newsgroups 663

25.1	News lesen	664
25.1.1	Grundsätzliches Vorgehen	664
25.1.2	Der Offline-Reader Pan	666
25.1.3	Der Online-Reader KNode	667
25.1.4	Mozilla Thunderbird als Newsreader	667
25.2	Installation eines Newsservers	669
25.3	Beispiel: Newsserver zur Projektverwaltung	671
25.4	Gruppen anlegen	672
25.5	Verbindung nach außen	673
25.6	Newsgroups saugen	675
25.7	NNTP-Protokollbeschreibung	677

26 Webserver 681

26.1	Hypertext und HTML	681
26.2	Clients	686
26.3	Start des Apache-Servers	687
26.4	Die Konfigurationsdatei httpd.conf	687
26.5	Privatadministration per .htaccess	690
26.6	Kommunikation per HTTP	693
26.7	Virtuelles Hosting	696
26.8	CGI: Der Server schlägt zurück	697
26.9	Programmierte Websites mit PHP	700
26.10	Aktive Websites in Java: Tomcat	701
26.10.1	Installation	703

26.10.2 Entwicklungsumgebung	704
26.11 Der Client hilft mit: JavaScript	705

TEIL VI: DAS X WINDOW SYSTEM

27 Das X Window System	711
27.1 Grafische Oberfläche unter UNIX	711
27.2 Ein Überblick über die Architektur	713
27.2.1 Der X-Server	716
27.2.2 Der X-Client und seine Bibliotheken	717
27.2.3 Der Fenstermanager	720
27.3 X Window starten	721
27.3.1 Nacktstart mit xinit	722
27.3.2 Regulärer Start von X: startx	723
27.3.3 Grafisches Einloggen: Display Manager xdm	723
27.4 Umgang mit dem X Window System	725
27.4.1 Bedienungselemente des Athena Widget Set	725
27.4.2 Der Aufruf von X-Programmen	729
27.4.3 Cut and Paste	730
27.4.4 Das Terminalfenster xterm	731
27.4.5 Weitere praktische Helfer	734
27.5 Konfigurieren	734
27.5.1 Farbbeschreibung	734
27.5.2 Schriften	735
27.5.3 Bitmaps	738
27.5.4 Ressourcen	738
27.5.5 Konfiguration des Fenstermanagers	741
27.5.6 Fokus und Z-Anordnung	742
27.6 Desktops	743
27.6.1 CDE	743
27.6.2 KDE	747
27.6.3 GNOME	752
27.6.4 Der Wettstreit der freien Desktops	754
27.6.5 MacOS X	755
27.7 Das X Window System im Netz	756
27.7.1 X-Programme über das Netz starten	757
27.7.2 X-Zugang verriegelt	759
27.7.3 X-Anwendung per ssh starten	760
27.7.4 Autorisierter Fernstart (xauth)	761
27.7.5 X-Server-Software in Betrieb nehmen	762

27.7.6	Grafisches Einloggen über das Netz	763
27.7.7	Thin Client	767

TEIL VII: PROGRAMMIERUNG

28 Programmierung von Shellskripten 771

28.1	Erstellen und Starten eines Shellskripts	772
28.2	Variablen	772
28.2.1	Zugriff auf die Parameter	773
28.2.2	Prozessnummern	774
28.2.3	Weitere Standardvariablen	774
28.2.4	Zuweisungen	775
28.3	Ablaufsteuerung	777
28.3.1	Die Unterscheidung: if	777
28.3.2	Bedingungen	778
28.3.3	Rückgabewert von Programmen	780
28.3.4	Die Fallunterscheidung: case	781
28.3.5	Die while-Schleife	782
28.3.6	Die for-Schleife	784
28.3.7	Funktionen	786
28.4	Gruppieren von Anweisungen	787
28.5	Ein- und Ausgaben aus dem Skript	789
28.6	Start und Umgebung von Skripten	790

29 Perl 793

29.1	Interpreter und Skript	793
29.2	Variablen	794
29.2.1	Skalare	794
29.2.2	Variablenamen	796
29.2.3	Operationen auf Skalare	797
29.2.4	Arrays	799
29.2.5	Hash	801
29.2.6	Reguläre Ausdrücke	802
29.3	Interaktiv	803
29.3.1	Ein- und Ausgabe	803
29.3.2	Aufrufparameter	804
29.3.3	Umgebungsvariablen	805
29.4	Ablaufsteuerung	805
29.4.1	Bedingungen	805

29.4.2	if	806
29.4.3	for	808
29.4.4	foreach	810
29.4.5	Sonstige Schleifen: while und until	811
29.4.6	Funktionen	814
29.5	Dateien	815
29.5.1	Schreiben und Lesen	815
29.5.2	Umgang mit Dateien	817
29.6	Perl und UNIX	818
29.6.1	Aufruf von UNIX-Programmen	818
29.6.2	UNIX-Systemprogrammierung	818
29.7	Grafische Oberfläche: Tk	819
29.7.1	Widgets und Ressourcen	820
29.7.2	Kontrollelemente	821
29.7.3	Widget-Anordnung	828
29.8	Zugriff auf die Datenbank	831
29.9	Informationsquellen	832

30 Python 833

30.1	Interpreter und Skript	833
30.2	Ein- und Ausgabe und Variablen	833
30.3	Ein Fehler ist ein Ausnahmefall	834
30.4	Umgang mit Zahlen	835
30.4.1	Rechnen	835
30.4.2	Formatierte Ausgabe von Zahlen	836
30.5	Umgang mit Zeichenketten	838
30.5.1	Aneinanderhängen	838
30.5.2	String-Bibliothek	839
30.5.3	Konvertierung	840
30.6	Verzweigung	841
30.7	Bedingungen	842
30.8	Schleifen	843
30.8.1	for	843
30.8.2	while	844
30.9	Funktionen	845
30.10	Erweiterte Datentypen	846
30.10.1	Sequenzen	847
30.10.2	Listen	847
30.10.3	Tupel	848
30.10.4	Dictionaries	849

30.10.5 Klassen	850
30.10.6 Referenzen und Kopien	851
30.11 Dateien lesen und schreiben	852
30.12 Datenbankzugriffe	854
30.13 Netzwerkzugriffe	857
30.13.1 Auslesen einer Website	857
30.13.2 Zugriff auf einen POP3-Server	857
30.14 Tk, die grafische Oberfläche	858

31 Programmierwerkzeuge 865

31.1 C-Compiler	865
31.2 make	868
31.3 Debugger	873
31.3.1 dbx	874
31.3.2 adb (System V)	875
31.3.3 gdb GNU debug	876
31.4 Java	878
31.4.1 Portierbarkeit	878
31.4.2 Java-Entwicklung	879
31.4.3 jdb – der Java-Debugger	880
31.4.4 Applikation zusammenpacken: jar	881
31.5 Integrierte Entwicklungsumgebungen	881
31.6 Versionsverwaltung	884
31.6.1 SCCS (Source Code Control System)	885
31.6.2 RCS (Revision Control System)	886
31.6.3 CVS (Concurrent Versions System)	887
31.6.4 UNIX als CVS-Server	890
31.6.5 Versionsverwaltung Subversion	893
31.7 Analysewerkzeuge	900
31.7.1 Systemaufrufe verfolgen: strace und ltrace	900
31.7.2 Speicherlecks und -überläufe	901
31.8 Diverse Programmierhelfer	902
31.8.1 Kurzbetrachtung: lex und yacc	902
31.8.2 Verteilte Übersetzung: icecream	903

32 UNIX-Systemaufrufe 905

32.1 Die Funktion main	905
32.1.1 Aufrufparameter	906
32.1.2 Zugriff auf die Umgebungsvariablen	907

32.2	Fehlerbehandlung: errno	908
32.3	Dateizugriffe	909
32.3.1	Öffnen, Lesen und Schreiben	909
32.3.2	Positionieren: lseek	912
32.3.3	Dateihandle duplizieren: dup	913
32.3.4	Datei-Eigenschaften ermitteln	914
32.3.5	Datei-Eigenschaften ändern	916
32.3.6	Sperren	917
32.3.7	Link erzeugen: link, symlink	924
32.3.8	Löschen: unlink	924
32.3.9	Umbenennen: rename	925
32.3.10	Temporäre Dateien	925
32.4	Verzeichnisse	926
32.4.1	Auslesen: opendir, readdir, closedir	926
32.4.2	Ermitteln des Arbeitsverzeichnisses	927
32.4.3	Wechseln: chdir	927
32.4.4	Anlegen und Löschen: mkdir, rmdir	928
32.5	Prozesse	928
32.5.1	Multiprocessing contra Multithreading	929
32.5.2	Vervielfältigen von Prozessen: fork	930
32.5.3	exec und system	931
32.5.4	Synchronisation: wait	932
32.5.5	Prozessumgebung	933
32.5.6	Gemeinsamer Speicher: Shared Memory	935
32.5.7	Synchronisation mit Semaphoren	940
32.5.8	Message Queues	943
32.5.9	Leichtgewichtsprozesse: Threads	948
32.6	Signale	951
32.6.1	Signale senden: kill	953
32.6.2	Auf Signale warten: pause	953
32.6.3	Timeout setzen: alarm	953
32.6.4	Zombies vereiteln	954
32.7	Pipe	954
32.7.1	Prozesskommunikation per Pipe	954
32.7.2	Named Pipe oder FIFO	955
32.7.3	Drucken unter UNIX	955
32.8	Fehlerbehandlung mit syslog	956
32.9	Zeitfunktionen	958
32.10	Benutzer und Gruppen	959
32.10.1	Die Passwortdatei als Struktur	959
32.10.2	Auslesen der Passwortdatei	960

32.10.3 Gruppen	961
32.11 Grundlagen der Dämonisierung	963
32.12 Client-Server-Socketprogrammierung	963
32.12.1 Kommunikationsendpunkt: socket und close	965
32.12.2 Serveraufrufe: bind, listen und accept	966
32.12.3 Clientaufruf: connect	967
32.12.4 Datenaustausch: send und recv	967
32.12.5 Namensauflösung	968
32.12.6 Zahlendreher: ntoh und hton	969
32.12.7 Rahmenprogramm eines Client-Server-Paars	970
32.12.8 Mehrere Sockets parallel abfragen	974
32.12.9 IPv6 aus Programmiersicht	975
32.12.10 Client-Server aus Sicht der Performance	976
32.13 Verschlüsseln mit crypt	976
32.14 Reguläre Ausdrücke	978
32.15 Weitere Programmierschnittstellen	980
32.16 Systemkonformität	980
32.16.1 Polling	980
32.16.2 Rechte beachten	981
Anhang	983
A Die Entstehung und Entwicklung von UNIX	985
A.1 AT&T	985
A.2 UNIX an der Uni	986
A.3 UNIX wird kommerziell	987
A.4 Die Rache der Enterbten	988
A.5 Mac OS X	991
A.6 UNIX wird verkauft	992
B Glossar	995
C Literatur	1007
Index	1011

Die Zeit der zentralen Großrechner mit den passiven Terminals klingt aus. Gerade UNIX findet heute seinen Haupteinsatz als leistungsfähiges und zuverlässiges Serversystem in einer Netzwerkumgebung.

18 Netzwerk

Ein Netzwerk ist zunächst nichts anderes als die Verbindung mehrerer Computer, über die sie miteinander kommunizieren können. Das Verbindungsmedium war ursprünglich ein einfaches abgeschirmtes Kabel. Heute besteht ein Netzwerk aus Glasfasern, aktiven Switches oder Funkfrequenzen. In Kombination mit geeigneter Software können Sie von Ihrem Arbeitsplatz aus Ressourcen anderer Computer nutzen. Am bekanntesten dürfte die Möglichkeit sein, gemeinsam auf eine Festplatte oder einen Drucker zugreifen zu können. Das Motiv ist nicht allein Sparsamkeit. Netzwerke ermöglichen das Teilen von Daten und gewährleisten so höchste Aktualität.

Auf den ersten Blick scheint es, als würde ein Netzwerk Objekte wie Drucker oder Festplatten zur Verfügung stellen. Bei näherer Betrachtung handelt es sich aber um *Dienste* (engl. *services*). So werden die Druckdaten nicht an den Netzwerkdrucker selbst gesendet, sondern an einen Prozess. Dieser veranlasst dann als Dienstleister (engl. *server*) den Druck. Dieser Prozess muss nicht zwingend auf einem Computer laufen. Bei den heutigen Netzwerkdrucker läuft nicht nur eine Steuerelektronik, sondern ein kleiner Computer, der die Netzwerkkommunikation beherrscht. Da die Druckdaten nicht direkt in den Druck gehen, kann der Prozess vor dem Druck die Berechtigungen prüfen. Der teure Farblaserdrucker der Werbeabteilung soll beispielsweise nicht allen Angestellten zur Verfügung stehen, um ihre Urlaubsfotos auszudrucken.

Server

Das größte aller Netze ist das Internet. Hier gibt es Abertausende von Diensteanbietern. Neben dem World Wide Web bietet es eine schnelle, kostengünstige Kommunikation per E-Mail. Darum ist eine Firma in der Regel daran interessiert, vielen Arbeitsplätzen einen Zugang zu dieser Informationsquelle zu ermöglichen. Allerdings besteht auch das Risiko, dass das eigene Netz über das Internet ausgespäht oder angegriffen wird.

Internet

TCP/IP Jedes Netzwerk braucht Protokolle, die festlegen, welcher Teil der Nachricht die Adresse, der Absender, eine Kontrollinformation bzw. Daten ist. Inzwischen ist TCP/IP (Transmission Control Protocol/Internet Protocol) das unangefochten wichtigste Protokoll. TCP/IP ist für UNIX nicht nur das Zugangsprotokoll zum Internet, sondern auch die Basis für lokale Netzwerke. Dabei spielen die vom PC her bekannten Festplatten- und Druckserver keine so große Rolle wie das Verteilen von Anwendungen in Client-Server-Architekturen. In UNIX-Netzen geht es mehr um das Starten von Prozessen auf entfernten Maschinen oder um das Verteilen von Prozessen auf mehrere Maschinen.

Heterogene Netze In TCP/IP-Umgebungen finden Sie selten ausschließlich UNIX-Maschinen. So haben Sie immer wieder damit zu tun, auch MS-Windows- oder Mac-OS-Rechner zur Zusammenarbeit zu bewegen. Sie werden oft als Frontend benutzt, während das Backend unter UNIX läuft.

18.1 Client-Server-Architekturen

Clients und Server sind Prozesse Der Zweck eines Netzes ist es, einen Prozess auf einem Rechner mit Informationen zu versorgen, die auf einem anderen Computer vorhanden sind. Der Auslöser ist also immer ein Prozess, der eine Information anfragt und auf die Antwort wartet. Einen »Anfrager« nennt man Client, einen »Antworter« nennt man Server. Ein Server ist also in erster Linie ein Prozess und kein Computer. Derselbe Computer kann durchaus gleichzeitig als Client und als Server auftreten, indem er bestimmte Anfragen beantwortet und auf der anderen Seite Anfragen stellt.

Eine Software, die nach dem Client-Server-Prinzip arbeitet, ist auf zwei Seiten aufgeteilt. Sie ist quasi irgendwo »durchgesägt«. Das Frontend läuft auf dem Arbeitsplatzrechner, und ein anderer Teil arbeitet auf einem anderen, typischerweise zentralen Rechner, der dann auch meist als Server bezeichnet wird, weil auf ihm die Serverprozesse laufen.

18.1.1 Ethernet als Verkabelungsbeispiel

Bei der Verkabelung wird heutzutage meistens Ethernet eingesetzt. Diese Technik ist für lokale Netzwerke inzwischen nahezu konkurrenzlos. In seiner ursprünglichen Form bestand ein Ethernet aus einem Koaxialkabel¹ und je einem Widerstand an jedem Ende. Ein Computer wurde mit dem

¹ Ein Koaxialkabel besteht aus einem Draht, der von einer Abschirmung umgeben ist. Antennenkabel sind typischerweise auch Koaxialkabel.

Kabel durch einen Abgriff verbunden. Bei dem Koaxialkabel RG58 war dies ein auf dem BNC-Stecker basierendes T-Stück. Dieses T-Stück steckt direkt auf einem Transceiver am Ethernet-Controller.

Inzwischen verwendet man auch in kleinen Netzen längst eine Twisted-Pair-Verkabelung. Das Kabel besteht, wie der Name schon sagt, aus verdrehten Drähten. Es verbindet den Computer mit einem Hub oder einem Switch, der das Rückrat des Netzwerks darstellt. Moderne Netzwerkinterfaces können auch direkt miteinander verbunden werden. Sie schalten ihre Belegung um, wenn sie feststellen, dass sie mit einem anderen Netzwerkinterface statt mit einem Hub verbunden sind.

Twisted Pair

Jedes Netzwerkinterface hat seine eigene, weltweit eindeutige Nummer, die 48 Bit groß ist. Diese Nummer ist meist in das ROM des Controllers eingebrannt.² Der Ethernet-Controller lauscht die ganze Zeit am Kabel, und sobald ein Paket kommt, das die Nummer des Controllers als Adresse hat, holt er es in seinen Speicher und gibt es an das Betriebssystem weiter.

Will der Controller selbst Daten senden, packt er sie in Pakete zu je maximal 1500 Byte und setzt die Ethernet-Adresse des Zielcomputers, gefolgt von der eigenen Adresse, als Absender hinein. Zu guter Letzt enthält jedes Paket eine Prüfsumme, die CRC (Cyclic Redundancy Check), und schon ist das Paket sendebereit.

Senden

Wenn Sie sich im Detail dafür interessieren, was auf Controller-Ebene passiert, wie die Pakete genau aussehen und wie die Prüfsummen arbeiten, sollten Sie die Werke von Tanenbaum³ und Comer⁴ lesen.

18.1.2 Die Pseudoschnittstelle loopback

TCP/IP wird nicht nur verwendet, um mit anderen Rechnern Verbindung aufzunehmen. Manchmal führt der Rechner auch Selbstgespräche. So arbeitet die grafische Oberfläche von UNIX über TCP/IP. Dabei erfolgt die Kommunikation auf modernen Workstations vom eigenen Rechner zum eigenen Display. Bei Maschinen, die überhaupt keine Netzwerkanschlüsse haben, wird eine Schnittstelle gebraucht, über die der Rechner mit sich selbst Kontakt aufnehmen kann. Diese Schnittstelle nennt sich loopback, weil sie wie eine Schleife auf den Rechner selbst zurückführt.

² Bei Sun-Maschinen befindet sie sich in einem batteriegepufferten RAM. Darum empfiehlt es sich, sich diese Nummer zu notieren. Die Batterie könnte ja mal schlappmachen. Die Nummer wird beim Booten angezeigt.

³ Andrew S. Tanenbaum: Computer Networks. Prentice Hall, Englewood Cliffs, 1987.

⁴ Douglas E. Comer: Internetworking with TCP/IP. Prentice Hall, 2nd ed., 1991.

18.1.3 Pakete in Paketen

ARP Auf der Basis von Ethernet-Paketen werden IP-Pakete versendet. Als Adresse werden IP-Adressen und der Port verwendet (siehe unten). Dabei kennt ein Ethernet-Controller nur MAC-Adresse. Die IP-Adressen müssen also auf die MAC-Adresse abgebildet werden. Das Protokoll, das diese Abbildung steuert, nennt sich ARP (Address Resolution Protocol) und ist eigentlich unterhalb von TCP/IP anzusiedeln.

Bearbeiten der ARP-Tabelle Es kann ein Problem auftreten, wenn eine Netzwerkkarte zwischen Rechnern getauscht wurde. Da die IP-Adresse nun unter einer anderen MAC-Adresse zu finden ist, ist die ARP-Tabelle, die die Zuordnung von MAC-Adressen zu IP-Adressen enthält, nicht mehr gültig. Nach dem Tausch läuft diese IP-Adresse bei anderen Rechnern im Netz noch unter einer anderen MAC-Adresse oder umgekehrt. Die ARP-Tabelle kann unter UNIX mit dem Befehl `arp` bearbeitet werden. Mit der Option `-a` wird die Tabelle angezeigt, mit `-d` werden Einträge gelöscht, und mit `-s` kann ein Eintrag gesetzt werden. Näheres finden Sie in der Manpage von `arp`.

18.2 TCP/IP, der Standard

TCP/IP ist das Standardprotokoll für den Zugriff auf das Netzwerk. Es wurde von der Berkeley Universität seinerzeit für UNIX entwickelt. Diese enge Verbindung merkt man TCP/IP auch in anderen Umgebungen noch an.

18.2.1 Die IP-Adresse

Die Netzkennung in der IP-Adresse Die IP-Adresse ist eine 32-Bit-Zahl, die die Netzwerkschnittstelle eines Computers im Netz eindeutig bestimmt. Der erste Teil der Nummer bezeichnet das Netz, in dem sich der Computer befindet, der zweite Teil den Computer selbst. Alle Computer, die direkt miteinander verbunden sind, gehören zum gleichen Netz und haben die gleiche Netzkennung in ihrer IP-Adresse. Wie Sie mit einem Computer kommunizieren können, der eine fremde Netzkennung hat, wird beim Thema Routing (ab Seite 469) behandelt. Ohne besondere Maßnahmen reagiert der Computer auf den Versuch, auf eine fremde Netzwerkadresse zuzugreifen, mit der Fehlermeldung »no route to host«.

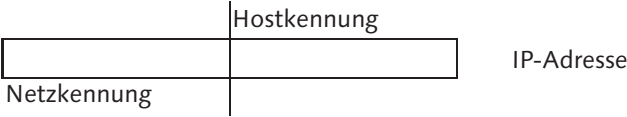


Abbildung 18.1 IP-Adresse

Der hintere Teil der IP-Adresse ist die Adresse des Computers im Netz. Um genau zu sein, ist sie an den Netzwerkkadapter des Computers gebunden. Da aber die meisten Computer nur einen Adapter haben, soll der einfacheren Lesbarkeit halber auch weiter von »Computern« die Rede sein. Jeder Adapter muss eine eigene Nummer haben. Mehrere Rechner mit gleicher Hostkennung führen in größeren Netzen zu schwer auffindbaren Fehlern.

Die Hostkennung
in der IP-Adresse

Die Grenze zwischen Netzkennung und Hostkennung liegt in lokalen Netzen oft auf einer Bytegrenze. Darum wird für IP-Adressen eine byteweise Darstellung gewählt. Da Dezimalzahlen am einfachsten zu lesen sind, schreibt man jedes der vier Byte einer IP-Adresse dezimal auf und trennt sie durch einen Punkt. Beispielsweise hat mein Arbeitsplatzrechner in meinem heimatlichen Netzwerk die IP-Adresse 192.168.109.144. Hexadezimal ergibt dies C0 A8 6D 90. Wieder als ganze Zahl geschrieben, lautet die Nummer 3232263568. Mein Macintosh hat die Adresse 192.168.109.25 oder hexadezimal C0 A8 6D 19 und damit 3232263449.

Darstellung

Darstellung	Arbeitsplatz	Macintosh
dezimal	3232263568	3232263449
hexadezimal	C0 A8 6D 90	C0 A8 6D 19
dotted decimal	192.168.109.144	192.168.109.25

Tabelle 18.1 Verschiedene Darstellungen von IP-Adressen

Netzwerkklasse und Netzwerkmaske

Die IP-Adresse bezeichnet also den Rechner bzw. dessen Netzadapter und das Netzwerk, in dem sich der Rechner befindet. Im Beispiel meines Netzes ist offensichtlich 192.168.109 bei beiden Rechnern gleich. Und tatsächlich benennen die ersten drei Bytes in diesem Fall das Netz und das letzte Byte den Rechner, weil in meinem Netzwerk die Netzwerkmaske auf 255.255.255.0 eingestellt ist. Man sollte also denken, ich könnte in mein privates Netz maximal 256 Rechner einbinden, da ein Byte die Werte von 0 bis 255 annehmen kann. Allerdings sind die 0 und die 255 für andere Zwecke reserviert, sodass ich mit 254 Rechnern in meinem Arbeitszimmer-Netz auskommen muss.

Netzwerk- maske und Netzwerkklasse

Welcher Teil der IP-Adresse zum Host und welche zum Netzwerk gehört, wird durch die Netzwerkmaske bestimmt. Die Maske ergab sich traditionell aus der Netzwerkklasse, zu der die IP-Adresse gehörte. Seit der Einführung von CIDR (siehe Seite 481) kann die Netzwerkmaske frei festgelegt werden. Zur optimalen Nutzung des IP-Adressraums sind die Netzwerkklassen im Internet also de facto nicht mehr vorhanden. In lokalen Netzwerken werden die Netzwerkmasken durchaus häufig noch an den Netzwerkklassen ausgelegt, auch wenn es technisch dafür keine Gründe gibt.

Die Netzwerkklasse wird durch die ersten Bits des ersten Bytes der IP-Adresse bestimmt.

Class A Ist das erste Bit 0, gehört die Adresse zu einem Class-A-Netz. Betrachtet man das erste Byte in binärer Darstellung, sieht sich also die einzelnen Bits an, so entscheidet das erste Bit darüber, ob die Zahl größer oder kleiner als 128 ist. Ist das erste Byte einer Netzwerkadresse kleiner oder gleich 127, gehört sie also zu einem Class-A-Netz. Class-A-Netze haben eine Netzwerkmaske von 255.0.0.0.

Warum heißt nun diese Zahl »Netzwerkmaske« und warum ist sie im ersten Byte 255? Mit der Netzwerkmaske und der UND-Verknüpfung kann man den Netzanteil einer IP-Adresse herausfiltern. UND ist ein logischer Ausdruck auf Binärebene, der genau dann 1 ergibt, wenn beide Operanden 1 sind. Um das an einem Beispiel zu demonstrieren, wird die Class-A-Netzwerkadresse 10.3.4.7 mit der Netzwerkmaske 255.0.0.0 gefiltert:

Adresse:				
dezimal	10	3	4	7
binär	00001010	00000011	00000100	00000111
Netzwerkmaske:				
dezimal	255	0	0	0
binär	11111111	00000000	00000000	00000000
Adresse	00001010	00000011	00000100	00000111
Netzwerkmaske	11111111	00000000	00000000	00000000
UND	-----			
Ergebnis	00001010	00000000	00000000	00000000
dezimal	10	0	0	0

Durch die UND-Verknüpfung von Adresse und Netzwerkmaske wird der Netzanteil der Adresse »herausmaskiert«. Der Netzanteil heißt hier

10.0.0.0. Das erste Byte ist die Netzwerkadresse. Der Rest ist die Adresse des Netzadapters.

Die 192 ist binär 11000000. Alle Werte, die kleiner als 192 sind, beginnen mit der Bitkombination 10 und gehören zu einem Class-B-Netz. Die Netzwerkmaske im Class-B-Netz ist 255.255.0.0, also sind die ersten zwei Bytes der Anteil der Netzwerkadresse. Der Rest ist die Rechneradresse. Class B

Die Nummern 224-255 im ersten Byte sind reserviert und dürfen nicht als Netzwerknummern verwendet werden. Dies sind die IP-Adressen, die mit drei Einsen beginnen. Das erste Byte in einem Class-C-Netz beginnt mit der Bitkombination 110 und liegt zwischen 192 und 223 (also auch mein Hausnetz). Es hat die Netzwerkmaske 255.255.255.0 und hat damit drei Bytes für die Netzkennung und ein Byte für die verschiedenen Computer. Class C

Bei einem größeren Unternehmen könnte ein Class-B-Netz mit den Nummern 128-191 schon sinnvoller sein, da hier ca. 16.000 Computer im Netz sein können. Richtig viele Computer passen in ein Class-A-Netz. Allerdings gibt es von diesen nur 128.

Der Bereich der Netze über 224 hat eine Sonderstellung. So dienen die Adressen des Class-D-Netzes für Multicast-Adressen. Durch das Ansprechen einer Adresse werden mehrere Rechner gleichzeitig adressiert. Anwendung findet diese Technik bei Routern und auch bei Rendezvous, einer Technologie von Apple zum Betrieb lokaler Netze, die keine Konfiguration erfordern. Class D

	Anzahl Bits für Netze	Anzahl der Netze	Anzahl Bits für Hosts	Anzahl der Hosts	Kennungen
Class A	7	128	24	16.777.214	1-127 (0xxxxxx)
Class B	14	16.384	16	65.534	128-191 (10xxxxxx)
Class C	21	2.097.152	8	254	192-223 (110xxxxx)
Class D					224-239 (1110xxxx)
Class E					240-255 (1111xxxx)

Tabelle 18.2 Anzahl der Hosts und Netze in den Klassen

Der Hostanteil einer Nummer darf weder nur 1 noch nur 0 sein. Wenn alle Bits in einer Host-ID auf 1 gesetzt sind, ist das die Broadcast-Adresse des Netzes. Ein Paket an diese Adresse wird von allen Rechnern des Netzes gelesen. Sind alle Bits 0, bezeichnet diese Adresse das Teilnetz. Dies wird beispielsweise beim Routing so verwendet.

Verbindungsanforderung Wenn also zwei oder mehr Rechner in einem lokalen Netz direkt miteinander verbunden werden sollen, müssen folgende Dinge gewährleistet sein:

1. Der Netzwerkteil der IP-Adressen muss gleich sein.
2. Die Hostnummern der Rechner müssen sich unterscheiden.
3. Die Hostnummer darf weder die erste Adresse sein. Sie ist 0 und ist die Netzwerkadresse. Sie darf auch nicht die letzte Nummer des Netzwerks sein, also alle Bits auf 1 stehen haben, weil dies die Broadcast-Adresse ist.

Private IP-Adressen

Adressen für lokale Netze Für jede Klasse wurde in RFC⁵ 1597 ein Bereich von Adressen definiert, die im Internet nicht weitertransportiert werden. Eine solche Adresse sollte für lokale Netze verwendet werden, sofern keine eigenen internet-fähigen IP-Adressen reserviert worden sind.⁶ Anders ausgedrückt heißt das, dass diese Nummern beliebig oft in der Welt verwendet werden können. Da solche Pakete im Internet nicht weitergeleitet werden, macht es gar nichts, wenn noch jemand anderes genau die gleichen Netzwerkadressen für sein lokales Netz verwendet.

Klasse	Nummernkreis
Class A	10.0.0.0
Class B	172.16.0.0 bis 172.31.0.0
Class C	192.168.0.0 bis 192.168.255.0

Tabelle 18.3 Die privaten IP-Adressen

Freie Nummern gewährleisten Kollisionsfreiheit Wenn eine kleine Firma für ihr internes TCP/IP-Netz IP-Nummern vergeben will, sollte sie sich unbedingt Adressen aus einem dieser Bereiche aussuchen. Da keine dieser Nummern im Internet vorkommt, gibt es auch keinen Web-, FTP- oder E-Mail-Server mit einer solchen Adresse. Damit ist immer eindeutig, ob mit einer Adresse ein Rechner im lokalen Netz oder ein Server im Internet angesprochen wird. Es können auch durch eigene Adressen keine IP-Adressen ausgeblendet werden. Ein zweiter Grund hat mit der Sicherheit vor Angriffen aus dem Internet zu tun:

5 RFC ist die Abkürzung für *Request For Comment*, übersetzt etwa: Bitte um Kommentare. Die RFC stellen so etwas wie die Norm des Internets dar.

6 Internet-fähige IP-Bereiche können von einem Internet Provider für die eigene Organisation reserviert werden. In Deutschland ist dafür das Europäische RIPE in Amsterdam zuständig (<http://www.ripe.net>), das hierfür aber einen ausführlichen Antrag mit Begründung des Bedarfs für einen dreijährigen Zeitraum verlangt.

Da diese Nummern im Internet nicht geroutet werden, kann ein Angreifer die Rechner des Netzes kaum direkt erreichen. Er kann die Adresse des Rechners nicht durch das Internet schleusen, da diese nach RFC 1597 dort nicht transportiert wird.

Grundeinstellungen des Netzadapters: ifconfig

Mit dem Befehl `ifconfig` kann die IP-Adresse jeder Schnittstelle festgelegt werden. Das funktioniert auch im laufenden Betrieb. Für das Umstellen der IP-Nummer ist also unter UNIX kein Reboot erforderlich. Allerdings hängen an der IP-Nummer oft einige andere Konfigurationen. Der Aufruf von `ifconfig` hat folgende Struktur:

IP-Adressen im laufenden Betrieb ändern

Der Befehl ifconfig

ifconfig <Netzadapter> <IP-Nummer>

`ifconfig` erwartet als ersten Parameter die Bezeichnung des Netzadapter-Devices. Dabei wird allerdings der Pfadname `/dev` weggelassen. Die Namen dieser Gerädateien unterscheiden sich je nach System.

System	Interface-Name
SCO:	Je nach Hardware 3b0, wdn0 und so weiter
OS/2 ftp PCTCP:	nd0
Linux:	eth0
Solaris:	le0
FreeBSD:	ed0

Tabelle 18.4 Namen für den ersten Netzwerkadapter

Das einzige Device, das auf jeder Maschine den gleichen Namen hat, ist die Loopback-Einheit. Sie heißt `lo0`. Dieses Pseudo-Device verweist auf die eigene Maschine.

loopback

Soll mein Arbeitsplatzrechner auf die bereits genannte IP-Adresse umgestellt werden, lautet der Befehl unter Linux:

```
ifconfig eth0 192.168.109.144
```

Hier wird der ersten Ethernet-Karte die IP-Adresse 192.168.109.144 zugeordnet.

Der Befehl `ifconfig eth0` ohne weiteren Parameter zeigt die aktuelle Einstellung des Interfaces an. Beispiel:

Schnittstelleninfo

```

gaston:~ # ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:00:E8:59:88:0F
          inet addr:192.168.109.144  Bcast:192.168.109.255
          Mask:255.255.255.0
          inet6 addr: fe80::200:e8ff:fe59:880f/10 Scope:Link
          inet6 addr: fe80::e859:880f/10 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:12 errors:0 dropped:0 overruns:0
          carrier:0 collisions:0 txqueuelen:100
          Interrupt:5 Base address:0xef40

```

Wir wollen uns die wichtigsten Informationen ansehen, die dieser Befehl anzeigt. Aber Sie werden sicher schon einiges wiedererkennen. In der ersten Zeile ist unter `HWaddr` (Hardware Address) die MAC-Adresse zu sehen. Sie sehen, dass die Broadcast-Adresse der Netzwerkadresse entspricht und lediglich im Bereich der Host-Adresse alle Bits auf 1 stehen. Dadurch hat das letzte Byte den Wert 255. Die Netzwerkmaske ist erwartungsgemäß 255.255.255.0. Besonders interessant ist das Wörtchen `UP`. Es bedeutet, dass dieses Netzinterface aktiv ist. Sie können einen Netzadapter mit dem Befehl `ifconfig` abschalten. Sie müssen zum Abstellen den zusätzlichen Parameter `down` angeben. Zur Reaktivierung dient der Parameter `up`. Beispiel:

```
ifconfig eth0 down
```

Dies ist vor allem bei temporären Verbindungen von Bedeutung. Aber es kann auch zur Wartung ganz praktisch sein, eine Maschine schnell mal aus dem Netz nehmen zu können.

Wenn es Schwierigkeiten mit der Netzanbindung gibt, kann ein Aufruf von `ifconfig` mit der Netzschnittstelle als Parameter anzeigen, ob die Schnittstelle aktiv ist. Wenn die Netzwerkhardware nicht korrekt erkannt wird, meldet der Befehl einen IOCTL-Fehler (I/O-Control). Das ist eine Fehlermeldung des Hardwaretreibers. Wie schon im Abschnitt über die Gerätedateien (siehe Seite 278) erwähnt wurde, ruft UNIX die Treiber mit dem Systemaufruf `ioctl()` auf, wenn es Informationen über das Gerät braucht.

Dauerhaftes Einstellen der IP-Nummer

Die IP-Nummer wird normalerweise in einer der `rc-` oder `init-`Dateien festgelegt. Dazu wird wieder der Befehl `ifconfig` verwendet. Sie können die entsprechende Stelle leicht finden, indem Sie mit dem Befehl `grep` nach dem Wort »ifconfig« suchen. Allerdings wird die IP-Nummer unter-

schiedlich konfiguriert. Auf älteren UNIX-Systemen stand die IP-Nummer direkt hinter dem Aufruf von `ifconfig`, und man hat sie dort im `rc`-Skript einfach geändert, wenn der Rechner auf Dauer eine andere Adresse haben sollte. Finden Sie an dieser Stelle eine IP-Nummer, können Sie sie bedenkenlos ändern. Steht hinter dem `ifconfig` eine Umgebungsvariable oder finden sich dort Kommandos zum Auslesen einer Datei, sollten Sie die eigentliche Quelle suchen und die IP-Nummer dort ändern.⁷

Bei Solaris wird die IP-Adresse über den Hostnamen definiert. Für jeden Netzadapter gibt es eine Datei, die den Namen der Maschine (genauer: des Adapters) enthält. In der Datei `/etc/hostname.le0` steht beispielsweise der Name `sol`. Damit erhält der Netzwerkadapter `le0` den Namen `sol`. Zur Bestimmung der IP-Nummer wird `sol` in der Datei `/etc/hosts` gesucht. Eine Änderung der IP-Nummer würde also in der Datei `/etc/hosts` erfolgen.

Bei Solaris über
den Hostnamen

Unter Linux ist die Art, wie die IP-Adresse festgelegt wird, von der Distribution abhängig. SUSE verwendete dazu bis zur Version 8.0 seine Universalkonfigurationsdatei `/etc/rc.config`. Im Skript `/etc/init.d/network` finden Sie den Aufruf von `ifconfig`, der die Umgebungsvariable `IFCONFIG` benutzt, die in `rc.config` definiert ist.

`rc.config`

Ab der Version 8.0 verwendet SUSE das Skript `/sbin/ifup`, um die Netzadapter zu initialisieren. Es wird vom `rc`-Skript `/etc/init.d/network` gestartet. Die bisher in der Datei `rc.config` gesammelten Konfigurationen werden nun im Verzeichnis `/etc/sysconfig` in mehrere Dateien und Unterverzeichnisse verteilt. Darin befindet sich das Verzeichnis `network`, und darin wiederum finden Sie mehrere Dateien, die die Konfiguration der verschiedenen Netzadapter bestimmen. Alle haben das Präfix `ifcfg-`. Die Datei für den Ethernet-Adapter heißt `ifcfg-eth0` und hat beispielsweise folgenden Inhalt:

SUSE ab 8.0

```
BOOTPROTO=none
IPADDR=192.168.109.143
NETMASK=255.255.255.0
BROADCAST=192.168.109.255
NETWORK=192.168.0.0
STARTMODE=onboot
```

Die hier gesetzten Umgebungsvariablen werden in `/sbin/ifup` verwendet, um den Ethernet-Adapter zu initialisieren.

⁷ Wenn Sie an dieser Stelle eine IP-Nummer direkt angeben, funktioniert das natürlich auch. Falls aber jemand anders später versucht, die Nummer auf korrekte Art zu ändern, wird er sich wundern, warum nach jedem Reboot die alte IP-Nummer wieder aktiv ist.

Red Hat Gewisse Ähnlichkeiten zur Konfiguration unter der Linux-Distribution von Red Hat sind unverkennbar. Ebenso wie dort sind im Verzeichnis **/etc/sysconfig** die Konfigurationsdateien zu finden. Bei Red Hat heißt das Unterverzeichnis **network-scripts**, in dem Sie schließlich die Datei namens **ifcfg-eth0** finden, die einen vergleichbaren Inhalt wie bei SUSE hat.

Debian Unter Debian werden die Interfaces in **/etc/network/interfaces** eingetragen, ein Eintrag für eine Netzwerkarte könnte so aussehen:

```
auto eth0
iface eth0 inet static
    address 192.168.109.175
    netmask 255.255.255.0
    network 192.168.109.0
    broadcast 192.168.109.255
    gateway 192.168.109.10
    dns-nameservers 192.168.109.10
    dns-search mydomain
    name Ethernet LAN-Karte 0
```

Im gleichen Verzeichnis befinden sich Verzeichnisse, deren Inhalt im Zusammenhang mit dem Netzwerkstart ausgeführt werden. Der Inhalt von **if-pre-up.d** wird vor, der von **if-up.d** wird während des Hochfahrens gestartet. Zudem werden das Verzeichnis **run** ausgeführt, wenn das Netzwerk läuft. Die Skripte in **if-down.d** werden ausgeführt wenn das Netzwerk heruntergefahren wird und die in **if-post-down.d** danach.

FreeBSD FreeBSD verwendet die Datei **rc.config**, um eine Umgebungsvariable mit den Parametern des **ifconfig** zu setzen, die beim Booten in der Datei **rc.network** als Parameter für den Aufruf von **ifconfig** verwendet wird.

Auf einer unbekannten Maschine beginnt die Suche nach der Stelle, an der die IP-Nummer festgelegt wird, an dem Ort, an dem in den rc-Dateien der Befehl **ifconfig** abgesetzt wird. Dort müssen Sie ermitteln, woher **ifconfig** seine Informationen bekommt.

Die etwas einfachere Variante dürfte die Verwendung des jeweiligen Administrationstools sein, das auf allen Plattformen eine Möglichkeit zur Einstellung der IP-Nummern anbietet. Eine kurze Übersicht über diese Programme finden Sie ab Seite 217. Die Tools haben den Vorteil, dass sie eine systemkonforme Einstellung der IP-Nummer gewährleisten.

18.2.2 Das Prüftool ping

Wenn es einen heimlichen Superstar unter den einfachen TCP/IP-Werkzeugen für die Konfiguration von Netzen gibt, so ist es `ping`. Das Programm sendet kleine Pakete an den angegebenen Zielrechner. Da dieser ein Ping-Paket sofort zurücksendet, kann Ping feststellen, ob der Rechner gestartet ist, ob die Verkabelung korrekt ist und wie schnell und zuverlässig die Verbindung zu diesem Rechner ist.

ping prüft
Verbindungen

Der Befehl ping

```
ping [<Optionen>] <IP-Nummer>
```

Bevor Sie fremde Rechner anpingen, sollten Sie zunächst die Nummer 127.0.0.1 aufrufen. Das ist die logische localhost-Adresse. Wenn sie versagt, funktioniert das TCP/IP auf Ihrem Rechner nicht. Anschließend sollten Sie die eigene Nummer einmal per `ping` aufrufen. Damit testen Sie, ob das eigene Netzwerkinterface richtig konfiguriert ist. Wenn das nicht klappt, werden Sie auch keine Verbindung nach außen bekommen. Schließlich eignet er sich auch zum Test der Namensauflösung (siehe Seite 490), indem Sie in mit dem Namen des Hosts aufrufen. Hat alles andere geklappt, aber der Ping erreicht nicht den Namen, stimmt die Namensauflösung nicht.

Lokales TCP/IP
prüfen

Bei einigen Implementationen wird nur die lapidare Meldung »xxx is alive« (beispielsweise SunOS) ausgegeben. Andere Implementierungen (so beispielsweise MS Windows) senden fünf Pakete und enden dann bereits mit einer Zusammenfassung. Beides ist unzureichend, wenn Sie beispielsweise sehen wollen, ob ein Wackelkontakt im Kabel ist. Durch entsprechende Optionen (`-t`) können Sie aber die typische Ausgabe von `ping` erhalten. Diese läuft durch, bis Sie **ctrl+C** drücken, und zeigt dann eine Statistik an:

ping zum
Dauerläufer
machen

```
PING gaston.willemer.edu (192.168.109.144): 56 data bytes
64 bytes from 192.168.109.144: icmp_seq=0 ttl=255 time=0.172 ms
64 bytes from 192.168.109.144: icmp_seq=1 ttl=255 time=0.099 ms
64 bytes from 192.168.109.144: icmp_seq=2 ttl=255 time=0.095 ms
64 bytes from 192.168.109.144: icmp_seq=3 ttl=255 time=0.093 ms
64 bytes from 192.168.109.144: icmp_seq=4 ttl=255 time=0.094 ms
64 bytes from 192.168.109.144: icmp_seq=5 ttl=255 time=0.093 ms
64 bytes from 192.168.109.144: icmp_seq=6 ttl=255 time=0.093 ms
64 bytes from 192.168.109.144: icmp_seq=7 ttl=255 time=0.098 ms
--- gaston.willemer.edu ping statistics ---
8 packets transmitted, 8 packets received, 0% packet loss
round-trip min/avg/max = 0.093/0.104/0.172 ms
```


Das Beispiel zeigt die Statistik einer gut laufenden Verbindung. Jedes zurückkommende Paket wird gemeldet. Die `icmp_seq` ist lückenlos. Bei dieser Nummer handelt es sich um Paketnummern. Da das Protokoll ICMP, unter dem `ping` läuft, verlorene Pakete nicht wiederholt, deutet eine durchgängige `icmp_seq` darauf hin, dass die Verbindung zuverlässig ist. Die Laufzeiten sind minimal. Dass das erste Paket etwas länger braucht, ist völlig normal.

ttl Die Abkürzung `ttl` bedeutet *time to live*, also übersetzt etwa Lebensdauer. Ein Ping-Paket versucht sein Ziel auch durch Wechsel des Netzes zu erreichen. Dabei passiert es jedes Mal einen Router. Das ist ein Rechner, der die Verbindung zwischen zwei Netzwerken herstellt.

Bei jedem Wechsel wird der `ttl`-Wert des Pakets um eins heruntergezählt. Sobald der Wert 0 ist, wird das Paket nicht mehr weitergeleitet. Damit wird verhindert, dass Pakete, die ihr Ziel nicht finden, endlos im Internet umherschwirren. Mit der Option `-t` kann der Startwert für `ttl` verändert werden.

**ping kann auch
große Pakete
senden**

Sie können die Paketgröße von `ping` beliebig festlegen. Dazu gibt es je nach System den Parameter `-s`. In manchen Fällen wird die Größe in Byte als weiterer Parameter hinter dem Ziel angegeben. Mit dieser Option können Sie große Datenpakete und damit eine große Netzlast erzeugen.

Das folgende Beispiel zeigt die Ausgabe von `ping` mit einer Paketgröße von 40.000 Byte über eine nicht sehr verlässliche Verbindung:

```
gaston> ping -s 40000 192.168.109.137
PING 192.168.109.137 (192.168.109.137) from 192.168.109.144
: 40000(40028) bytes of data.
40008 bytes from 192.168.109.137: icmp_seq=6 ttl=255 time=73.689 msec
40008 bytes from 192.168.109.137: icmp_seq=13 ttl=255 time=73.742 msec
40008 bytes from 192.168.109.137: icmp_seq=21 ttl=255 time=73.624 msec
40008 bytes from 192.168.109.137: icmp_seq=23 ttl=255 time=72.995 msec
40008 bytes from 192.168.109.137: icmp_seq=38 ttl=255 time=73.151 msec
40008 bytes from 192.168.109.137: icmp_seq=39 ttl=255 time=72.456 msec
40008 bytes from 192.168.109.137: icmp_seq=40 ttl=255 time=72.279 msec

--- 192.168.109.137 ping statistics ---
46 packets transmitted, 7 received, 84% loss, time 45090ms
rtt min/avg/max/mdev = 72.279/73.133/73.742/0.639 ms
gaston>
```

Der Wert für `time` ist recht konstant, aber natürlich höher als beim vorigen Versuch. Das liegt daran, dass die größeren Pakete länger unterwegs sind. An den großen Lücken bei `icmp_seq` ist aber zu erkennen, dass immer wieder Pakete zerstört werden. Bei einem lokalen Netzwerk kann dies ein Hinweis auf einen Wackelkontakt, einen fehlenden Abschlusswiderstand oder einen defekten Hub sein.

18.3 Routing: Verbindung mehrerer Netzwerke

Aus den verschiedensten Gründen werden Netzwerke in mehrere kleinere Netzwerke aufgeteilt. Aufgrund technischer Gegebenheiten ist dies notwendig, wenn beispielsweise unterschiedliche physikalische Netzträger verwendet werden, etwa Ethernet und Token-Ring.

Aus geografischen Gründen ist es notwendig, wenn eine Filiale über Modem oder ISDN angekoppelt werden soll. Ein weiterer Grund kann das Ziel sein, die Netzlast in den einzelnen Netzen zu reduzieren.

Jeder Host kann von jedem Host im gleichen Netzwerk über seine IP-Adresse erreicht werden. Dagegen ist eine Internet-Adresse eines fremden Netzes oder Subnetzes nur dann erreichbar, wenn die Pakete weitervermittelt werden. Dazu dient ein Gateway.

Gateways
verbinden
Netzwerke

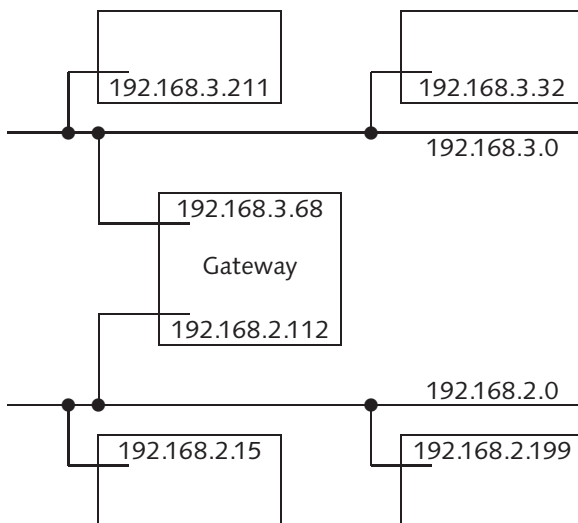


Abbildung 18.2 Ein Gateway verbindet zwei Netze

18.3.1 Gateways

Ein Gateway ist ein Computer mit zwei Netzwerkanschlüssen, die jeweils an ein anderes Netz angeschlossen sind. Jede Schnittstelle hat eine eigene IP-Adresse, die dem entsprechenden Netzwerk zugeordnet ist. Kommt ein Paket auf der einen Netzwerkkarte für das jeweils andere Netz an, wird es vom Gateway in das andere Netz eingespeist.

18.3.2 Statische Festlegung einer Route

Jeder Rechner verwaltet eine Routing-Tabelle, in der er speichert, auf welchen Wegen er welche Netze oder sogar einzelne Rechner erreichen kann. Um den Weg zu einem Rechner in einem fremden Netz zu definieren, wird der Befehl `route` verwendet. Abhängig vom Ziel gibt es drei Varianten, eine Route anzulegen:

route: Setzen einer Route

```
route add host <Host> gateway <Gateway> metric <Metric>
route add net <Netz> gateway <Gateway> metric <Metric>
route add default gateway Gateway metric Metric
```

Der erste Parameter nach `route add` ist das Ziel der Route. Mit *Gateway* wird der weiterleitende Rechner angegeben, und mit *Metric* geben Sie die Priorität der Route an.

metric Der Parameter `metric` wird benötigt, wenn es zu einem Ziel mehrere Routen mit unterschiedlicher Geschwindigkeit gibt. Der schnellsten Route wird die höchste Priorität (beispielsweise 1) gegeben. Fällt diese aus, kann auf die Route mit der niedrigeren Priorität 2 oder 3 ausgewichen werden. Die `metric`-Information wird nur beim dynamischen Routing verwendet, um eine Bewertung der Qualität der Strecke vorzunehmen. Bei statischem Routing ist der Parameter irrelevant.⁸

Aufrufparameter von route Leider sind die Aufrufparameter des Kommandos `route` nicht ganz einheitlich. So kann bei manchen Versionen das Schlüsselwort `gateway` auch als `gw` abgekürzt werden. Andere Implementationen benötigen die Schlüsselwörter `gateway`, `metric`, `net` und `host` gar nicht, da sie sich aus der Reihenfolge bzw. aus der Art der Parameter von selbst ergeben. Der erste Parameter ist immer das Ziel. Ob das Ziel ein Host oder ein Netz ist, lässt sich direkt an der IP-Nummer ablesen. An zweiter Stelle steht immer das Gateway, und der letzte Parameter ist immer die Priorität der Route.

route to host Der Rechner 192.168.2.15 in Abbildung 18.2 soll als Ausgangspunkt dienen, um die Einrichtung der Routingtabelle zu demonstrieren. Wollen Sie von dort eine Route auf den Rechner 192.168.3.32 legen, muss das Paket an das Gateway gesendet werden. Die vom Rechner aus erreichbare Adresse lautet 192.168.2.112. Der Befehl dazu lautet also:

⁸ vgl. Hunt, Craig: TCP/IP Network Administration. O'Reilly, Sebastopol, 1994. p. 138.

```
route add host 192.168.3.32 gateway 192.168.2.112 metric 1
```

Sie brauchen nicht jeden einzelnen Rechner, sondern können mit einem Mal die Route für das gesamte Netz 192.168.3.0 angeben. Gateway und Metric bleiben gleich. Der Befehl lautet dann:

```
route add net 192.168.3.0 gateway 192.168.2.112 metric 1
```

Schließlich können Sie den Rechner anweisen, alle unbekannten Netzwerkadressen über ein bestimmtes Gateway hinauszuschleusen:

```
route add default gateway 192.168.2.212 metric 1
```

Der Eintrag `default` entspricht der IP-Nummer 0.0.0.0. Im Beispiel können Sie dann eine Default-Route setzen, wenn es zu anderen Netzen kein weiteres Gateway gibt. In Netzen mit Internetzugang werden normalerweise alle direkt erreichbaren Netze mit expliziten Routen bestimmt und das Gateway zum Internet auf `default` gesetzt.

Beim Anlegen einer Route können Sie auch die Netzwerkmaske angeben. Das ist erforderlich, wenn im lokalen Netz Subnetze (siehe Seite 477) verwendet werden. Im Internet wird durch den Einsatz von CIDR (siehe Seite 481) zu jeder Route die Netzwerkmaske angegeben. Sie geben die Netzwerkmaske durch einen zusätzlichen Parameter an, dem Sie das Schlüsselwort `netmask` voranstellen:

```
route add net 192.168.3.0 netmask 255.255.255.128 \
gateway 192.168.2.112 metric 1
```

In diesem Fall würden die Adressen 192.168.3.1 bis 126 über das Gateway 192.168.2.112 geleitet. Der Backslash am Ende der Zeile bewirkt, dass die Befehlseingabe in der nächsten Zeile weitergeht.

Einträge in der Routing-Tabelle können wieder gelöscht werden:

route: Entfernen einer Route

```
route delete net <Zieladresse> <Gateway>
route delete host <Zieladresse> <Gateway>
```

Der Befehl `netstat -r` zeigt die aktuellen Routingtabellen an:

```
gaston> netstat -r
Kernel IP routing table
Destination    Gateway Genmask         Iface
192.168.109.0  *      255.255.255.0    eth0
loopback       *      255.0.0.0        lo
```

In der ersten Spalte stehen die Ziele, und unter `Gateway` findet sich der Router, der aufgesucht wird, um das Ziel zu erreichen. Die Spalte `Genmask` beschreibt die Netzwerkmaske, die bei dem angegebenen Ziel vorausgesetzt wird, und zu guter Letzt folgt das Interface, über das die Pakete abgesetzt werden.

18.3.3 Statisches Routing: Ein Beispiel

[zB] Das Beispiel geht von einer Firma mit zwei Filialen und zwei Hauptgeschäftsstellen aus. Die eine Zentrale liegt in Gintoft, die andere in Norgaardholz. Jede der Zentralen hat 100 PCs. Die Filialen liegen in Hamburg und Frankfurt/Oder und haben je 2 PCs.

Die IP-Nummern für die Netze werden wie folgt festgelegt:

IP-Nummer	Bereich
192.168.108.0	Gintofter Netz
192.168.109.0	Norgaardholzer Netz
192.168.110.0	Hamburger Netz
192.168.111.0	Frankfurter Netz

Tabelle 18.5 Übersicht über die IP-Nummern im Beispiel

Die Arbeitsplätze sollen von 1 bis 190 durchnummeriert werden. Die Server sollen als 201, weitere als 202, 203 und Router als 254 angesprochen werden.

`/etc/hosts` Da Nummern sehr abstrakt sind, sollen die Rechner Namen erhalten. Die Arbeitsplätze in Gintoft erhalten das Präfix **gin**, die von Norgaardholz **nor**. In der Datei `/etc/hosts` (siehe Seite 491) hält UNIX die Liste, die die Namen auf die IP-Nummern abbildet. Die `/etc/hosts`-Datei sieht etwa so aus:

```
192.168.108.1    gin1
192.168.108.2    gin2
192.168.108.3    gin3
192.168.108.254 ginrout1
192.168.108.201 ginsrv1

192.168.109.1    nor1
192.168.109.2    nor2
192.168.109.3    nor3
192.168.109.254 norrout1
192.168.109.201 norsrv1
```

```

192.168.110.1    hh1
192.168.110.2    hh2
192.168.110.254  hhrou1

192.168.111.1    ffo1
192.168.111.2    ffo2
192.168.111.254  fforou1
    
```

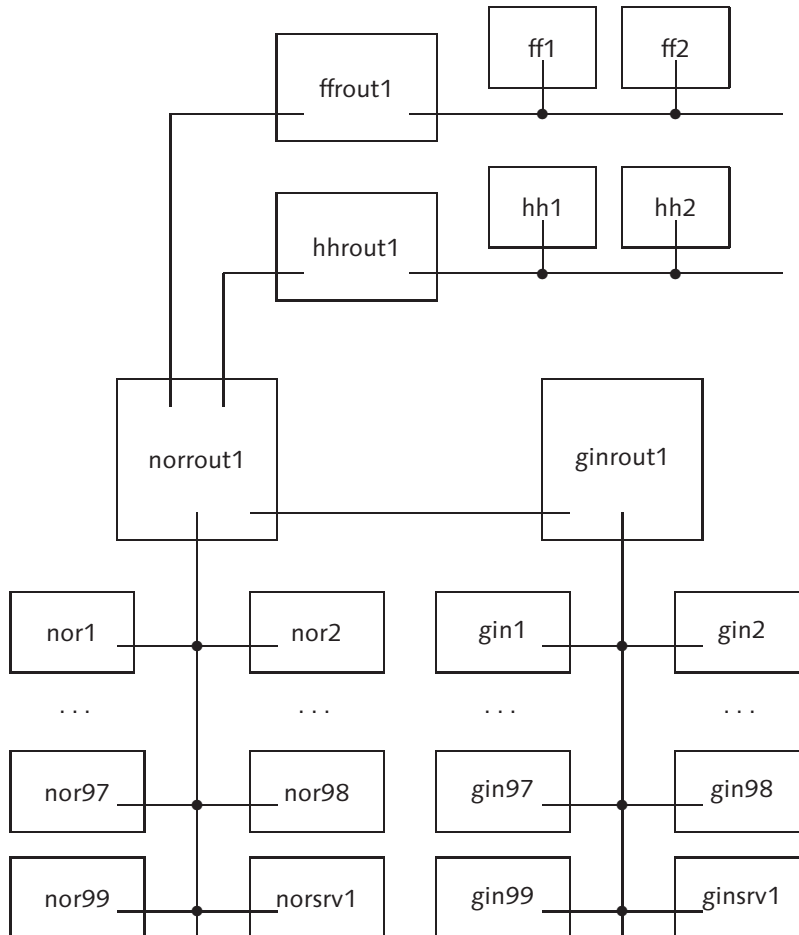


Abbildung 18.3 Netzskizze

Da es im Beispiel nur ein Gateway in jedem Netz gibt, ist die Konfiguration der Arbeitsplätze recht einfach: Sie erhalten jeweils einen Default-Eintrag auf ihren Router. Für die Norgaardholzer Arbeitsplätze und den dortigen Server lautet er:

Nur ein Gateway:
Default-Route

```
route add default norrout1 1
```

route unter
MS Windows

Bei Arbeitsplätzen mit MS Windows wird der Eintrag unter »Systemkonfiguration - Netzwerk - Protokolle - TCP/IP - Gateway« eingetragen. Hier ist Platz für eine oder mehrere Default-Routen. Wenn es notwendig wird, einem Windows-Rechner Routen zu verschiedenen Netzwerken über unterschiedliche Gateways zuzuweisen, erreichen Sie das nicht mehr über grafische Dialoge. Hier müssen Sie den `route`-Befehl in die **AUTOEXEC.BAT** schreiben. Wenn Sie im Beispiel davon ausgehen, dass in Norgaardholz ein Router `norrout2` den Zugang zum Internet realisiert, können Sie diesen in den Eintrag »Gateway« in der Systemkonfiguration eintragen. Für die Routen zu den Netzen Gintoft, Hamburg und Frankfurt müssten Sie folgende Befehle in der **AUTOEXEC.BAT** eintragen:

```
route add 192.168.108.0 mask 255.255.255.0 norrout1
route add 192.168.110.0 mask 255.255.255.0 norrout1
route add 192.168.111.0 mask 255.255.255.0 norrout1
```

Gateway-Konfiguration zwischen Gintoft und Norgaardholz

Zwischennetz
zwischen den
Routern

Die Konfiguration der Router ist interessanter. `norrout1` hat zwei Interfaces. Das eine mit der Nummer 192.168.109.254 ist eine Ethernet-Karte. Das andere Interface führt zur Telefonleitung und verbindet sich mit `ginrout1`. Zwischen `ginrout1` und `norrout1` gibt es also wiederum ein Netzwerk. Da dieses Netz nur den beiden Routern bekannt ist, verwenden Sie dort einfach irgendeine freie Nummer, beispielsweise 192.168.1.109 für `norrout1` und 192.168.1.108 für `ginrout1`. Da dieses »Zwischennetz« nur zwischen den Routern bekannt ist, ist es nicht unbedingt erforderlich und wird von einigen Routern gar nicht benötigt. Allerdings wird das Verständnis des Routings durch den Wegfall auch nicht leichter. Die Routingtabelle des `norrout1` lautet also:

```
# die Route auf das eigene, interne Netz
route add 192.168.109.0 192.168.109.254 1
# Gintoft über den Gintofter Router
route add 192.168.108.0 192.168.1.108 1
# Gintofter Router über WAN
route add 192.168.1.108 192.168.1.109 1
```

Damit kann `norrout1` nur Nachrichten nach Gintoft bearbeiten. Andere Pakete bearbeitet er nicht. Ein Paket von Norgaardholz nach Gintoft würde nun seinen Weg finden, aber nicht mehr zurück, da in Gintoft der Router nicht weiß, wie das Paket nach Norgaardholz gesendet werden soll. Auf `ginrout1` müssten die Routen also spiegelbildlich zum `norrout1` eingerichtet werden. Noch einfacher wird es, wenn man davon ausgehen

kann, dass `norrou1` alle Verbindungen zu den Filialen übernimmt. Dann braucht `ginrou1` nur folgenden Eintrag:

```
route add default 192.168.1.109 1
```

Anbindung der Filialen

Nun werden in Norgaardholz noch die Routen zu den Filialen in Hamburg und Frankfurt/Oder gebraucht. Wir geben der WAN-Schnittstelle in Hamburg die Nummer 192.168.1.110 und der in Frankfurt die Nummer 192.168.1.111. Durch die Routen

```
route add 192.168.110.0 192.168.1.110 1 # HH über HH-Router
route add 192.168.111.0 192.168.1.111 1 # Ffo über Ffo-Router
route add 192.168.1.110 192.168.1.109 1 # HH-Router über WAN
route add 192.168.1.111 192.168.1.109 1 # Ffo-Router über WAN
```

wird eine Verbindung von Norgaardholz nach Hamburg oder Frankfurt auf gleiche Weise hergestellt wie nach Gintoft. Natürlich müssen auch `hhrou1` und `fforou1` mit den entsprechenden Routing-Einträgen versehen werden.

Was passiert aber, wenn Hamburg und Gintoft gleichzeitig arbeiten wollten? Damit jede Außenstelle jederzeit eine freie Leitung vorfindet, werden drei Modems oder zwei ISDN-Karten besorgt. Da jede ISDN-Karte zwei B-Kanäle betreuen kann, reichen zwei ISDN-Anschlüsse, da jeder zwei B-Kanäle besitzt. Damit die Zuordnung klar ist, werden für den `norrou1` mehrere WAN-Adapter eingerichtet:

```
route add 192.168.1.110 192.168.1.2 1 # HH-Router über WAN 2
route add 192.168.1.111 192.168.1.3 1 # Ffo-Router über WAN 3
```

Würde nun Gintoft deutlich häufiger mit Hamburg zu tun haben als Norgaardholz, würden Sie die WAN-Anbindung dorthin natürlich nicht über Norgaardholz, sondern über Gintoft legen. Dabei würden Sie sogar einen ISDN-Anschluss sparen, da Gintoft und Norgaardholz je zwei Kanäle brauchten. Allerdings würde das Routing geringfügig komplizierter, da der Router `norrou1` für eine Verbindung nach Hamburg auf `ginrou1` verweisen müsste.

Alternativen

Insbesondere bei Firmen mit vielen Filialen stellt sich die Frage, ob man nicht abwechselnd alle Filialen über eine Telefonleitung ansprechen kann. Das hängt natürlich davon ab, wie die Filialen mit den Rechnern in den Zentralen arbeiten. Findet nur nachts ein Datenabgleich statt, kann jede Filiale einzeln nacheinander aktualisiert werden. Dazu müssen die verschiedenen Netzwerkadressen auf verschiedene Telefonnummern ab-

gebildet werden. In dem Moment, wo aber die Filialen in Konkurrenz und zeitlich nicht vorhersehbar auf die Zentralen zugreifen und umgekehrt, empfiehlt sich eine Trennung nach Kanälen. Ein ISDN-Anschluss ist nicht sehr teuer, und die Möglichkeit, dass eine Filiale die Zentrale nicht erreicht, weil die Leitung durch eine andere belegt ist, ist ausgeschlossen. Beim Thema Kosten sollte auch geprüft werden, ob es nicht sogar günstiger ist, eine Standleitung zu verwenden.

Von der IP-Nummer zur Telefonnummer

Bisher wurde völlig übergangen, wie Sie eine TCP/IP-Verbindung über ISDN legen können. Tatsächlich ist das recht einfach möglich. Leider ist die Art der Konfiguration von dem verwendeten Router abhängig.

Als Beispiel soll ein Router unter Linux verwendet werden. Dort gibt es bei jeder Distribution das Paket `i4l` (ISDN for Linux). Die einzelnen B-Kanäle der ISDN-Karten werden auf ISDN-Devices abgebildet. Bei einer einfachen Karte lauten sie **isdn0** und **isdn1**. Für jedes Device wird mit dem Befehl `isdnctrl` die anzurufende Telefonnummer angegeben. Sobald das Interface angesprochen wird, wird die Verbindung gewählt. Werden während einer gewissen Zeit keine Daten mehr transferiert, »legt« die Software »auf«, um Verbindungskosten zu sparen.

Sicherung des ISDN-Zugangs

Einen anrufbaren ISDN-Zugang wird jede Firma als Sicherheitsrisiko empfinden, solange nicht sichergestellt ist, dass er nur von der Filiale benutzt werden kann. Sie können für das Interface die Nummer festlegen, die der Anrufer haben muss. Alle anderen Telefonnummern werden dann durch das Interface abgewiesen. Es ist für einen Angreifer leicht, seine Telefonnummer zu unterdrücken. Aber das nützt ihm nichts, da er die Nummer der Filiale vorweisen muss, und das ist nicht so einfach.

Callback

Eine weitere Sicherungsmöglichkeit besteht darin, einen Callback zu installieren. Die Filiale ruft an und signalisiert damit, dass sie eine Verbindung haben möchte. Die Zentrale nimmt den Anruf entgegen und stellt fest, welche IP-Nummer ihr Gegenüber hat. Sie weist den Anruf jedoch zunächst zurück. Nun wählt sie die hinterlegte Telefonnummer der empfangenen IP-Nummer und ruft so die Filiale sofort zurück. Auf diese Weise ist sicher, dass nur die Filiale einen Zugang hat. Selbst wenn es einem Angreifer gelingen würde, die Telefonnummer der Filiale zu fälschen, würde die Zentrale ihn anschließend nicht zurückrufen, sondern die richtige Filiale.

Paket auf Reisen

Zur Veranschaulichung des Routings soll ein Paket von gin3 nach norsrv1 und zurück gesendet werden. Es würde also auf gin3 der Befehl

```
ping norsrv1
```

abgesetzt. Zunächst würde über die Datei **/etc/hosts** von gin3 festgestellt, dass norsrv1 die IP-Nummer 192.168.109.201 hat. Das Paket erhält diese Nummer als Zieladresse, und die eigene Adresse wird in den Absender gesteckt. Bereits dem Rechner gin3 ist klar, dass das Paket nicht zum eigenen Netz gehört. Gäbe es auf gin3 keinen Routing-Eintrag, erhielten Sie die Meldung:

```
no route to host
```

Durch die Default-Route wird das Paket erst einmal zu ginrout1 geleitet. Dieser liest wiederum die Adresse. Wäre ihm keine Route bekannt, würde er das Paket mit der Fehlermeldung »no route to host« an gin3 zurückschicken. Da er aber eine Route hat, die an die ISDN-Schnittstelle des norrout1 gerichtet ist, wird das Paket auf die Reise geschickt. norrout1 erkennt in der Adresse eine gültige Adresse für das Netz an seiner Ethernet-Karte. Also reicht er das Paket an diese Karte weiter. Auf dem Ethernet findet es nun norsrv1.

Gemäß dem ping-Protokoll tauscht norsrv1 nun die Sender- und die Empfängeradresse und schickt das Paket wieder los. Würde hier nicht die Default-Route greifen, würde norsrv1 auf einem Paket sitzen, das nicht ins eigene Netz gehört und für das es keine Route gibt. Da eben gerade die Adresse von gin3 unbekannt ist, kann gin3 von diesem Problem nichts mitbekommen. Bleiben Pakete also ohne Fehlermeldung einfach aus, spricht alles dafür, dass sie korrekt das eigene Netz verlassen konnten, aber im fremden Netz nicht mehr den Weg zurück fanden. In diesem Fall können Sie sicher sein, dass das Problem auf norsrv1 oder norrout1 liegt.

Da es aber die Default-Route gibt, geht das Paket an norrout1. norrout1 erkennt die Nummer als Netznummer von Gintoft und besitzt eine Route dorthin über ginrout1. Dieser gibt das Paket auf das Ethernet, sodass es zu gin3 zurückfindet.

18.3.4 Subnetze

Anstatt für jedes Teilnetz eine eigene Netzkennung zu verwenden, können Sie auch ein großes Netz in Unternetze zerteilen. Von außen ist diese

Subnetting ist nach außen unsichtbar

Unterteilung nicht zu sehen, und das Netz wird von fremden Routern als ein einheitliches Netz betrachtet. Erst wenn Pakete ins Innere des Netzes gelangen, werden sie durch die internen Router weitergeleitet.

Netzwerkmaske Um ein Netzwerk derart zu unterteilen, müssen Sie die Netzwerkmaske verändern. Die Netzwerkmaske gibt an, welcher Teil der IP-Nummer das Teilnetz bestimmt und welcher Teil der Nummer einen einzelnen Rechner festlegt. Die Maske enthält für jedes Bit, das zur Netzwerkadresse gehören soll, eine 1. Für die drei Klassen von Netzen sind folgende Netzwerkmasken Standard:

Klasse	hexadezimal	dezimal
Class A	0xFF000000	255.0.0.0
Class B	0xFFFF0000	255.255.0.0
Class C	0xFFFFF000	255.255.255.0

Tabelle 18.6 Standardnetzwerkmaske

In Abbildung 18.4 werden drei IP-Nummern der verschiedenen Netzwerkklassen anhand der Standardnetzwerkmaske in einen Netz- und einen Hostanteil aufgegliedert.

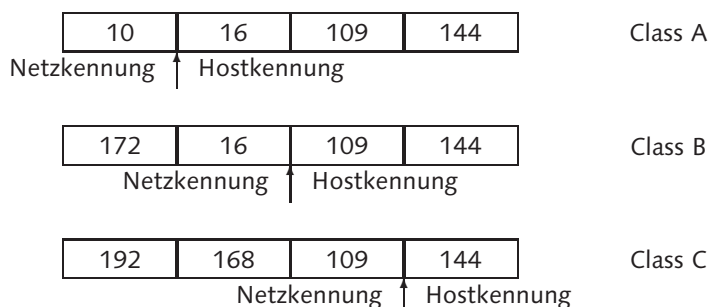


Abbildung 18.4 Netzwerkklassen und Netzwerkmaske

Die Netzwerkmaske ist bitweise konfigurierbar, und zwar kann der Anteil der Netzwerkkennung vergrößert werden. Wird eine Netzwerkmaske um ein Bit erhöht, wird damit das Netz in zwei Teilnetze zerlegt. Bei einem Class-C-Netz würde dann die Netzwerkmaske 255.255.255.128 lauten. Die 128 mag etwas überraschen. Aber von dem rechten Byte wird das am weitesten links stehende Bit verwendet. Die Dualdarstellung von 128 ist 1000000. Damit gehören alle Hostnummern, die größer als 128 sind, zu dem einen Teilnetz, und alle, die kleiner sind, gehören zu dem anderen Teilnetz.

Da die Netzwerkmaske immer so aufgebaut ist, dass in der Dualdarstellung, von links beginnend, nur Einsen und ab einer gewissen Grenze dann nur noch Nullen folgen, gibt es eine alternative Schreibweise, die hinter einem Schrägstrich nur angibt, wie viele Einsen die Netzwerkmaske enthält. Für ein einfaches Class-C-Netzwerk sieht das so aus: 192.168.109.144/24. Bei der obigen Netzwerkmaske 255.255.255.128 käme noch ein Bit hinzu, also 192.168.109.144/25. Man spricht auch von der CIDR-Schreibweise (siehe Seite 481).

Alternative
Schreibweise

Bei der Teilung in Subnetze gehen auch IP-Nummern verloren, die nicht mehr verwendet werden können. Denn für beide Teilnetze gilt die Regel, dass Adressen, bei denen alle Hostbits 0 oder 1 sind, nicht für die Adressierung von Rechnern verwendet werden dürfen. Das wären im Beispiel die Nummern 0, 127, 128 und 255.

In Abbildung 18.5 ist eine Class-B-Adresse mit Subnetzwerkmaske dargestellt. Die Standardnetzwerkmaske ist bei Class B 255.255.0.0. In diesem Fall soll das erste Halbbyte der Hostkennung noch mit zur Netzwerkadresse genommen werden. Von links beginnend haben die Bits den Wert 128, 64, 32 und 16. Die andere Hälfte des Bytes bleibt null, da es nicht in die Netzwerkmaske einfließen soll. Die Summe aus 128, 64, 32 und 16 ist 240. Das ist damit die Netzwerkmaske des dritten Bytes.

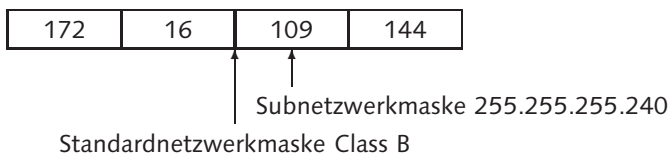


Abbildung 18.5 Subnetzwerkmaske

Eine Aufteilung des Netzes kann die Netzbelastung reduzieren. Gehen wir beispielsweise von einer Softwarefirma aus, die das Class-C-Netz mit der Nummer 192.168.2.x hat. In dem unteren Stockwerk des Firmengebäudes befindet sich die Verwaltung, deren Mitarbeiter auf einer Netzwerklplatte in unregelmäßigen Abständen Dokumente ablegen und in der Kundendatei suchen. Im oberen Stockwerk sitzen die Programmierer, die Netzwerkprogramme schreiben und von Zeit zu Zeit Belastungstests machen. Immer wenn solche Tests anlaufen, kommt die Sekretärin nicht mehr an ihre Kundendaten heran, weil das Netz überlastet ist. Schafft sie es trotzdem, werden durch ihre zusätzlichen Zugriffe die Testergebnisse verfälscht.

Netzbelastung
durch Subnetting
reduzieren

Man teilt das Netz daher logisch durch die Netzwerkmaske. Physisch werden die Kabel getrennt und durch ein Gateway verbunden, das zwei Netzwerkkarten besitzt. So kann ein Programmierer seine Dokumentation zwecks Rechtschreibprüfung immer noch an die Sekretärin senden. Das Gateway wird die Pakete von dem einen Netz in das andere übertragen. Der restliche Netzverkehr bleibt im jeweiligen Teilnetz.

Konsistenz der
Subnetzwerk-
maske

Die Netzaufteilung durch Subnetting ist nur für Rechner im Netz sichtbar, da die Maske auf den lokalen Rechnern im Netz-Device und in den Routingtabellen festgelegt wird. Es ist also wichtig, dass alle Rechner die gleiche Netzwerkmaske bekommen. Für außenstehende Rechner erscheint das Netz homogen, da sie eine Subnetzwerkmaske für ein fremdes Netz nicht kennen und so von einer Standardnetzwerkmaske ausgehen müssen.⁹

Problem bei
HP-UX 10.20

In der Dokumentation von CUPS (siehe Seite 390) wird darauf hingewiesen, dass eine Subnetzdefinition, die nicht auf den Bytegrenzen liegt, bei HP-UX 10.20 Schwierigkeiten beim Broadcast verursachen könnte. In der CIDR-Schreibweise bedeutet das, dass nur 8, 16 oder 24 hinter dem Schrägstrich stehen sollte. Mit HP-UX 11 sei das Problem beseitigt.

18.3.5 Dynamisches Routen

Insbesondere das Internet mit seinen vielen Teilnetzen und der ständigen Veränderung erfordert ein Routingverfahren, das sich dynamisch anpassen kann. Die Routingtabellen werden dabei nicht vom Administrator festgelegt, sondern von Dämonen verwaltet, die mit den Dämonen der Nachbarnetze Informationen über die Qualität der Verbindung und über die Erreichbarkeit anderer Netze austauschen. Auf diese Weise werden nicht nur Engpässe oder gar Ausfälle von Leitungen erkannt. Es ist sogar möglich, die Pakete über die nächstbeste Verbindung zu schicken. In den Routingtabellen erhält der Parameter `metric` eine zentrale Bedeutung, weil er bei schlechter Verbindung erhöht wird.

Nichts für das LAN

In einem Firmennetz werden diese dynamischen Verfahren normalerweise nicht eingesetzt. Dort werden üblicherweise keine Ausfallleitungen gelegt, auf die ein dynamisches Verfahren ausweichen könnte. Dort, wo kritische Verbindungen durch Ersatzleitungen abgesichert werden, handelt es sich um ein einfaches Backup, das Sie schnell durch das manu-

⁹ Unter CIDR sind die Netzwerkmasken auch außerhalb des lokalen Netzes bekannt. Hier werden die Netzwerkmasken von den Routern verwaltet und weitergegeben. Aber auch hier gilt, dass ein lokales Subnetting nicht unbedingt mit der nach außen bekannten Netzwerkmaske übereinstimmen muss.

elle Ändern von zwei Routingeinträgen in Betrieb nehmen können. Die Komplexität ist überschaubar, und die Veränderungen in den Netzen sind vorhersehbar und meist gut geplant.

Das dynamische Routen wird beispielsweise durch den Dämon `routed` realisiert, der das RIP (Routing Information Protocol) implementiert. Der Dämon `gated` beherrscht neben RIP auch das externe Routing EGP (Exterior Gateway Protocol). Das dynamische Routen ist das Rückgrat des Internets. Da die Router ständig Informationen über die Qualität der Leitungen austauschen, können defekte Leitungen durch eine Anpassung der Routingtabellen automatisch umgangen werden.

Das EGP informiert über die Erreichbarkeit autonomer Systeme. Ein autonomes System kann ein komplexes Netzwerk mit diversen internen Routern sein. Es muss nur nach außen abgeschlossen sein.¹⁰

18.3.6 CIDR – Classless Inter-Domain Routing

Mit zunehmender Beliebtheit des Internets wurden dessen Engpässe immer sichtbarer. So war die Idee, 32 Bits für die IP-Nummer zu verwenden, nicht weitreichend genug, wie sich beim Ausbau des Netzes herausstellte. Zwar kann man mit 4 Byte etwa vier Milliarden Rechner durchnummerieren (und das war immerhin die damalige Größe der Weltbevölkerung), aber bei genauerem Hinsehen war die Zahl doch nicht so großzügig angesetzt. So gehen bei jedem Netz zwei Adressen für die 0 und die Broadcast-Adresse verloren. Hinzu kommt, dass eine Firma, die ein Class-C-Netzwerk betreibt, im Normalfall nicht alle 254 Adressen auch wirklich einsetzt.

So wurde um 1993 das CIDR eingeführt.¹¹ Die Idee war, dass man jeder Netzwerkadresse eine Netzwerkmaske mitgab und dass erst durch diese bestimmt wird, wie viele Rechner in ein Teilnetz gehören. Wenn es nun noch gelingt, die benachbarten IP-Nummern lokal zu bündeln, lässt sich sogar die Anzahl der Routingeinträge reduzieren. CIDR war eigentlich als Übergangslösung bis zum Einsatz der neuen 128-Bit-IP-Adressen des IPv6 (siehe Seite 518) gedacht.

Die Änderungen durch die neue Norm konnten recht problemlos und schnell im Internet umgesetzt werden, da sie nur die Routingtabellen be-

¹⁰ vgl. Hunt, Craig: TCP/IP Network Administration. O'Reilly, Sebastopol, 1994. pp. 142.

¹¹ vgl. Nemeth, Evi/Snyder, Garth/Seebass, Scott/Hein, Trent R.: UNIX Systemverwaltung. Markt+Technik – Prentice Hall, 2001. S. 357–360.

treffen. Oft müssen die Einstellungen nur in den Routern gesetzt werden, da die Arbeitsplätze in den meisten Fällen ohnehin nur die Defaultroute zum Gateway verwenden. Da Router ihre Routingtabellen dynamisch austauschen, ist die Konsistenz leicht zu gewährleisten. CIDR war neben dem Einsatz des Masquerading (siehe Seite 570) die entscheidende Technik, um das Problem der ausgehenden IP-Nummern zu umgehen.

In lokalen Netzwerken wird oft noch nach den alten Klassen konfiguriert, obwohl inzwischen jedes Netzwerkinterface mit einer Netzwerkmaske frei zu konfigurieren sein sollte. Im Allgemeinen wird der Administrator nicht so knappe Adressräume haben wie im Internet. Der Vorteil der alten Klassen ist, dass man die Netzwerkmasken nicht explizit dokumentieren muss, sondern sie sich von selbst verstehen.

18.4 Ohne Kabel: WLAN

WLAN ist eine beliebte Alternative zur Anbindung an ein Netzwerk über ein Ethernetkabel. Die meisten Privathäuser besitzen leider keine Netzwerkverkabelung und so ist WLAN eine praktische Möglichkeit, Computer in das Netzwerk zu integrieren, die ein paar Wände oder Stockwerke entfernt stehen.

Treiber-
unterstützung

Nicht alle WLAN-Adapter werden von UNIX unterstützt. Relativ problemlos ist die Installation eines Centrino-Notebooks unter Linux. Die Firmware für die WLAN-Bausteine sind im Modul `ipw2100` enthalten. Für andere Hersteller finden sich die Treiber oft im Internet. Allerdings muss man dazu erst einmal herausfinden, welcher Chip in den Geräten eingebaut ist. Bei Notebooks ist dies noch vergleichsweise einfach. Bei USB-Adaptern sind diese Informationen insbesondere vor dem Kauf kaum zu ermitteln.

18.4.1 Access Point

Ein Access Point ist ein Gerät mit einem Netzwerkanschluss und einer WLAN-Antenne. Er transportiert seine Daten von einem meist ethernet-basierten LAN per Funk weiter an die mit WLAN ausgestatteten Rechner. Die WLAN-Adapter gehören in solchen Fällen zum gleichen Netzwerk wie die LAN-Adapter. Bei Netzen mit größerer Netzlast ist es sinnvoll, vor den Access Point einen Router zu setzen, so dass das WLAN ein separates Netzwerk darstellt, in dem nur die Pakete unterwegs sind, die für die per Funk angeschlossenen Teilnehmer interessant sind.

Ein Access Point kann auch im Ad-Hoc-Modus betrieben werden. In diesem Fall ist sein Zweck nicht die Weiterleitung der Pakete an ein LAN, sondern bildet den Vermittlungspunkt zwischen mehreren Computern mit WLAN-Anschluss.

Ad Hoc

Ein Access Point ist häufig in den Internet-Routern enthalten, die von den DSL-Providern als Anschlussmöglichkeit angeboten werden. Diese Geräte bestehen meist aus einem Router ins Internet, einem kleinem Switch zum Anschluss mehrerer Ethernetkabel und einem Access Point. Meist enthalten diese Geräte auch gleich noch eine Firewall und einen DHCP-Server. Auch wenn es von außen nicht immer zu erkennen ist, laufen diese Geräte oft unter Linux, das in ihrem Flashspeicher abgelegt ist.

18.4.2 Grundinformationen

Ein Netzwerk nach IEEE 802.11 wird durch eine Namenskennung, die sogenannte ESSID gekennzeichnet. Netzwerk-Adapter nach IEEE 802.11b können bis zu einer Geschwindigkeit von 11 MBit/s übertragen. Die IEEE 802.11g schafft bis zu 54 MBit/s. Allerdings muss man berücksichtigen, dass jede Wand und fast jeder metallische Gegenstand die Geschwindigkeit herabsetzt.

18.4.3 Sicherheitsaspekte

Die Signale eines WLAN werden standardmäßig unverschlüsselt versandt. Da viele Access Points auch einen DHCP-Service liefern, ist es leicht möglich, mit einem unkonfigurierten WLAN-fähigen Rechner direkt auf fremde Kosten ins Internet zu gelangen. Vielen Besitzern eines solchen Zugangs sind die Kosten durch Fremdnutzer gleich. Sie haben ihr WLAN oft im Zusammenhang mit einer Flat Rate bekommen.

Ein offener Zugang ins Internet ermöglicht es einem Angreifer aber auch, unter der Anschlusskennung des WLAN-Besitzers unerlaubte Dinge zu tun. So bietet es sich förmlich an, illegale Raubkopien oder pornografische Darstellungen über den Zugang des Nachbarn zu laden. Im Falle einer strafrechtlichen Verfolgung wird die Polizei an dessen Wohnungstür klingeln. Nach aktueller deutscher Rechtsprechung wird der Betrieb eines unverschlüsselten WLAN als Teilschuld bewertet, wenn ein Dritter darüber Rechtsverletzungen begeht. Im juristischen Slang wird ein solcher Betreiber als Störer bezeichnet, obwohl er die Kommunikation in

Rechtliche
Konsequenzen

diesem Falle ja nicht stört, sondern erst ermöglicht.¹² Es ist aber leicht nachzuvollziehen, dass der Nachbar beim illegalen Herunterladen raubkopierter Musik lieber Ihren offenen WLAN-Zugang nutzen wird als seinen eigenen. Falls eines Tages die Musikindustrie an die Tür klopft, wird es aufgrund der IP-Adressenrückverfolgung Ihre Tür sein und nicht seine.

Ein weiteres Problem ist, dass mit dem WLAN ein offener Zugang zum Netzwerk vorliegt, der von außen leicht erreichbar ist. Wer zwischen Windows-Rechnern ein paar Freigaben veröffentlicht, denkt oft nicht daran, dass der WLAN-Zugang zum Internet auch das lokale Netzwerk veröffentlicht. Hinzu kommt, dass auf Windows-Rechnern standardmäßig ein paar Freigaben eingerichtet sind, von denen der normale Anwender gar nichts weiß.

Verschlüsselung per WEP

Die Daten zwischen einem Access Point und einer WLAN-Karte können verschlüsselt werden. Dazu gibt es diverse Verfahren, die alle mit einem Schlüssel arbeiten. Das WEP ermöglicht eine Verschlüsselung mit einem Schlüssel von 128 und mit 64 Bit. Dementsprechend heißen sie WEP128 (manchmal auch WEP104) oder WEP64 (alias WEP40). Die Eingabe des Schlüssels kann je nach Software als Zeichenkette oder in hexadezimaler Fassung erfolgen. Falls der Kontakt gar nicht zustande kommen will, kann der Grund darin liegen, dass der Hex-Code eingegeben wird, obwohl das Programm den Schlüssel als Zeichenkette erwartet. Theoretisch dauert das Knacken des WEP128 zwar ungeheuer lange. Man hat bei diesen Berechnungen allerdings übersehen, dass sich bestimmte Kombinationen in der Kommunikation vorhersehbar wiederholen. Dadurch ist ein Einbruch durchaus möglich, sofern der Angreifer genügend kriminelle Energie aufbringt. Die Werkzeuge dafür stehen im Internet bereit. Nach neuester Rechtsprechung sind sie natürlich verboten.¹³

WPA Mit dem neuen Standard IEEE 802.11g wurde das Verschlüsselungsverfahren WPA eingeführt. Der Hauptunterschied zu WEP liegt darin, dass der Schlüssel alle 10 KByte geändert wird. Damit sollte dann ein Angreifer ausreichend verwirrt werden, um einen Einstieg zu vermeiden. Allerdings müssen für den Einsatz von WPA alle Geräte mit einem WLAN-Adapter nach IEEE 802.11g ausgestattet sein. Außerdem muss darauf geachtet werden, dass auch die Treibersoftware der Karten WPA beherrscht. Das Problem betrifft aber weniger Linux, weil dort die Verfügbarkeit der

¹² Diese Ausführungen sind mit aller Vorsicht zu genießen. Der Autor ist kein Jurist.

¹³ Und da niemand verbotene Software einsetzen wird, ist es nach Gesetz auch nicht notwendig und sogar verboten, dass ich Ihnen verrate, wovon Sie sich wappnen müssen.

WPA-Verschlüsselung nicht von der Großzügigkeit des Treiberherstellers des WLAN-Adapters abhängig ist.

Die meisten Access Points ermöglichen es, nur bestimmte Rechner zu akzeptieren. Dabei wird geprüft, ob die MAC der Netzwerkkarte mit der hinterlegten übereinstimmt. Die MAC könnte man als eine Art Seriennummer für Netzwerkkarte bezeichnen. Fremde Rechner werden abgewiesen. Dieses Verfahren ist nur in Verbindung mit einer Verschlüsselung sinnvoll, weil ein Angreifer andernfalls eine zugelassene MAC-Adresse aus dem Datenverkehr abhören kann.

MAC-Adresse

Grundsätzlich gilt, dass auch eine noch so schlechte Absicherung immer noch besser ist als gar keine. Es ist wie bei der Unterhaltung zweier Gladiatoren. »Glaubst Du wirklich, dass Du dem Löwen entkommen kannst, wenn Du Laufschuhe anziehst?« – »Nein, ich muss aber auch nicht schneller als der Löwe sein. Es reicht, wenn ich schneller als Du bin.« Sucht also jemand einen WLAN-Zugangspunkt zum Internet, wird er den unverschlüsselten nehmen, egal, wie schwach die Verschlüsselung des anderen ist.

Der Turnschuh

18.4.4 Softwaresteuerung des WLAN-Adapters

Um eine schnelle Anbindung an WLANs zu erreichen, gibt es auch unter UNIX immer mehr Werkzeuge. Da der schnelle Wechsel vor allem von Anwendern durchgeführt werden wird, sind die passenden Tools oft mit einer grafischen Oberfläche ausgestattet. Ein typischer Vertreter ist der KNetworkManager des KDE. Er zeigt alle Access Points der Umgebung mit Ihren ESSIDs an.

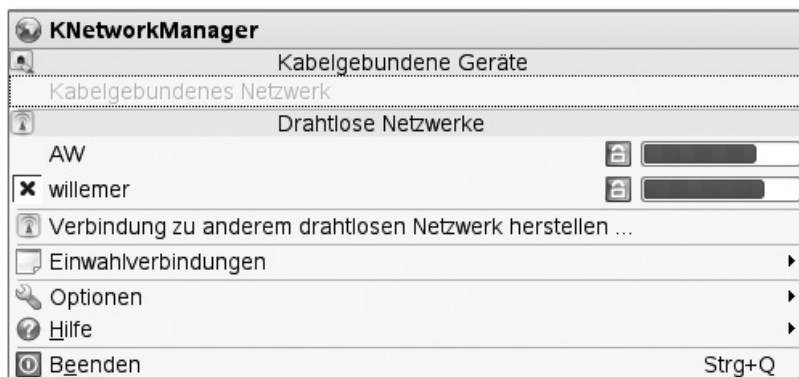


Abbildung 18.6 KNetworkManager

Die Abbildung 18.6 zeigt den KNetworkManager. In diesem Fall stehen die Netzwerke »AW« und »willemer« zur Verfügung. Mit dem letzteren ist der Rechner aktuell verbunden. Das zeigt das Kreuz. Durch Mausklick wählen Sie das Netz, in das Sie sich einwählen wollen. Bei einem bisher unbekannten Netz erscheint ein Dialog. Dort wird die Verschlüsselungsart angezeigt und es gibt die Möglichkeit, den Schlüssel anzugeben.

Konsolenzugriff Parallel zum bekannten Befehl `ifconfig` gibt es das Programm `iwconfig`, das die Details der WLAN-Adapter anzeigt. Beispiel:¹⁴

```
libo:~ # iwconfig
lo          no wireless extensions.

eth0        IEEE 802.11b  ESSID:"willemer"  Nickname:"ipw2100"
           Mode:Managed  Frequency:2.462 GHz  Access Point: 00:50:FC:D4:B6:FB
           Bit Rate=11 Mb/s   Tx-Power:16 dBm
           Retry min limit:7   RTS thr:off   Fragment thr:off
           Encryption key:0010-0200-3004-0050-0600-7008-00   Security mode:open
           Power Management:off
           Link Quality=100/100  Signal level=-35 dBm
           Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
           Tx excessive retries:0  Invalid misc:0  Missed beacon:0

eth1        no wireless extensions.

sit0        no wireless extensions.

libo:~ #
```

Der aufmerksame Leser wird am Encryption key ablesen können, dass dieser WLAN-Adapter nicht WPA, sondern WEP verwendet. Das hängt damit zusammen, dass es sich um einen Adapter nach IEEE 802.11b handelt. Diese Geräte beherrschen kein WPA.

18.4.5 Treiber für WLAN-Adapter

Das Schweigen der Hersteller

WLAN-Adapter enthalten zwar einen Windows-Treiber, aber die Anwender von UNIX sind für die Hersteller offenbar noch kein Markt. Darüber hinaus geben sie auch keine Informationen an die Open-Source-Entwickler weiter, so dass es mit der Treibersituation nicht zum Besten steht. Häufig auftretende Bausteine wie der Intel-Chip im Centrino-Notebook können von Linux betrieben werden.¹⁵ Ferner finden sich mit der Zeit immer mehr Treiber für bestimmte Chips im Internet. Die Verschwiegenheit der Anbieter geht allerdings so weit, dass der Kunden oft nicht weiß, welcher Chip in welcher Karte eingebaut ist. Es kommt sogar vor, dass

¹⁴ Den Schlüssel habe ich übrigens aus nahe liegenden Gründen geändert. Bei aller Gastfreundschaft möchte ich ungern alle Leser des Buchs in meinem Netzwerk begrüßen können.

¹⁵ Auch wenn anzumerken ist, dass dieser Treiber nicht frei ist.

in mehreren Notebooks der gleichen Typenbezeichnung unterschiedliche Chips verbaut wurden. Bei USB-Adaptern ist die Situation zur Zeit der Entstehung dieser Zeilen so, dass es überhaupt keine Treiber gibt.

Diese Situation führt zu Verzweiflungstaten. Wenn kein Linux-Treiber verfügbar ist, wäre es immerhin schon ein Fortschritt, wenn man den Windows-Treiber verwenden kann. Genau diesen Weg beschreitet das Programm `ndiswrapper`. Es wird ausgenutzt, dass jeder Netzwerktreiber eine NDIS-Schnittstelle haben muss. Das Programm entnimmt dem Windows-Treiber diesen NDIS-Anteil und bettet ihn so ein, dass Linux diesen Treiber für das Ansprechen des Adapters verwendet.

Windows-Treiber

Die Bedienung erfolgt in zwei Schritten. Zunächst wird der Windows-Treiber in den `ndiswrapper` integriert. Im zweiten Schritt wird das Modul `ndiswrapper` geladen.

Im Beispiel wird der Siemens USB-Adapter Gigaset 108 verwendet. Auf der beiliegenden CD findet sich der Treiber in dem Verzeichnis:

/Installation/Gigaset USB Adapter 108/Driver/Windows XP & 2000

Dessen Inhalt besteht aus:

```
-r-xr-xr-x 1 arnold root 360256 27. Jul 2005 ar5523.sys
-r-xr-xr-x 1 arnold root 149392 27. Jul 2005 ar5523.bin
-r-xr-xr-x 1 arnold root 12705 22. Aug 2005 net5523.inf
-r-xr-xr-x 1 arnold root 8263 29. Aug 2005 net5523.cat
```

Diese Dateien werden kopiert und der Treiber über die INF-Datei in den `ndiswrapper` installiert.

```
libo # ndiswrapper -i net5523.inf
installing net5523 ...
libo # modprobe ndiswrapper
libo #
```

Beim nächsten Start des KNetworkManager kann bereits auf den WLAN-Adapter zugegriffen werden.

18.4.6 Funkgesteuerte Peripherie: Bluetooth

Im Gegensatz zum WLAN dient Bluetooth zur Kopplung von Geräten über kurze Distanz. Insbesondere im Mobiltelefonbereich wird diese Technik zum Anschluss von Freisprechanlagen oder zum Austausch von Adressen oder Bildern zwischen den Geräten eingesetzt.

Für PCs und Notebooks werden Bluetooth-Adapter in erster Linie zur Kommunikation mit Mobiltelefonen benutzt. Auf diese Weise können Fotos der eingebauten Kamera geladen, der Kalender synchronisiert oder als Modem für den Internetzugang von unterwegs verwendet werden.

Ein Bluetooth-Gerät hat immer eine Nummer, die aus sechs Byte besteht. Diese Nummer wird im Allgemeinen hexadezimal, durch Doppelpunkte getrennt, dargestellt.

Ist ein UNIX-Gerät mit einem Bluetooth-Adapter ausgestattet, wird der HCI-Dämon im Hintergrund gestartet, der die Kommunikationsanfragen beantwortet. Die Konfigurationsdatei **hcid.conf** finden Sie im Verzeichnis **/etc/bluetooth**. Unter den Optionen können Sie einstellen, eine Verbindungs-PIN zu hinterlegen und ob der Rechner automatisch verbinden oder den Benutzer fragen soll. Unter Device kann der Name eingestellt werden, unter dem der Computer von anderen Bluetooth-Geräten erkannt wird.

Unter KDE können Sie mit Hilfe des Konquerors eine Übersicht über alle Bluetooth-Geräte der Umgebung bekommen. Um einen Dienst zu nutzen, klicken Sie einfach das entsprechende Gerät an.

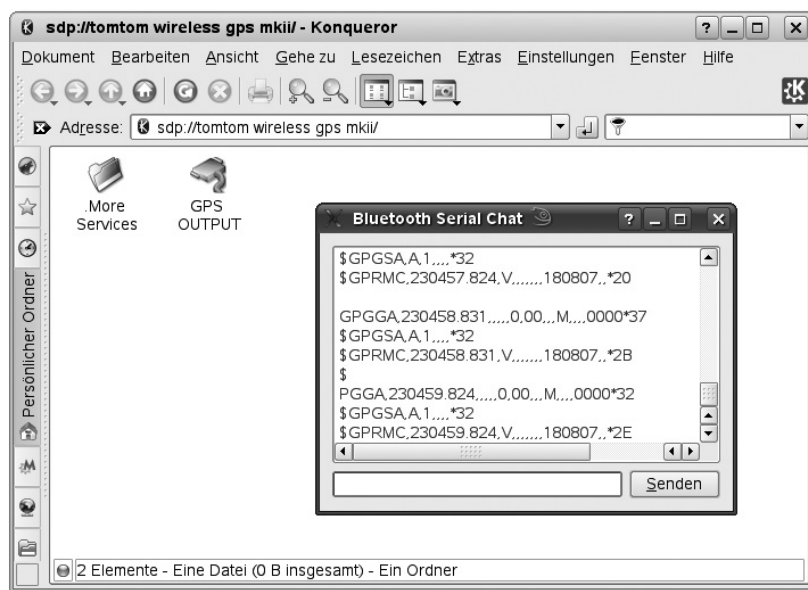


Abbildung 18.7 Bluetooth GPS Maus

Sie können sehen, welche Dienste die angeklickte GPS-Maus hat. Hier wurde der GPS OUTPUT angeklickt, und in einem Terminal werden die vom GPS-Gerät gelieferten Daten angezeigt.

In der Abbildung 18.8 sehen Sie ein Mobiltelefon, das seine Dienste anzeigt. Sie sehen auch die interne Datenstruktur, über die Sie einfach auf Dateien auf dem Telefon zugreifen können.

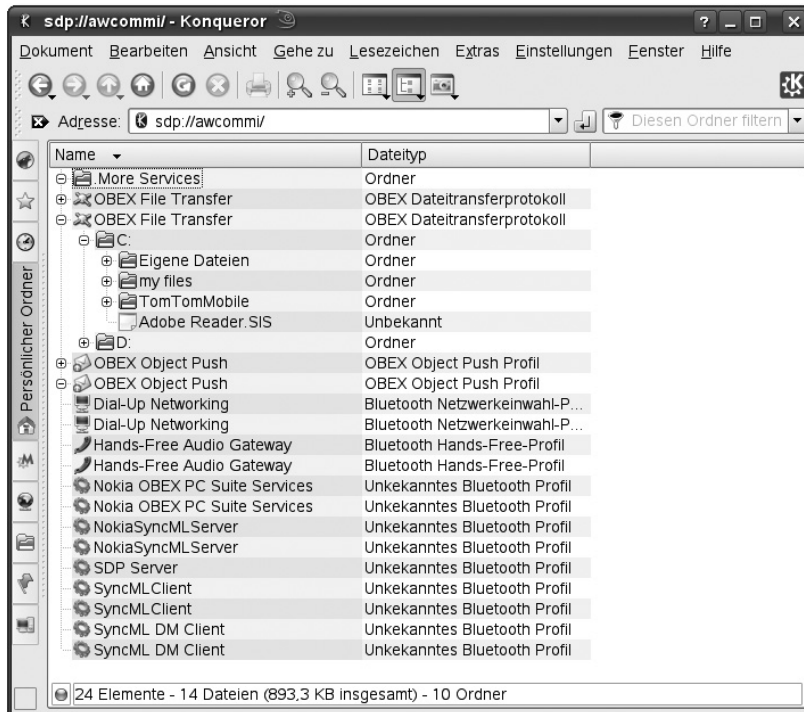


Abbildung 18.8 Bluetooth GPS Maus

18.5 Namensauflösung

Während Computer mit Zahlen wunderbar zurechtkommen, verwenden Menschen lieber Namen. Das Auflösen der Namen in Zahlen und umgekehrt kann aber wieder dem Computer übertragen werden, denn im Netzwerk ist es wichtig, dass Namen konsistent bleiben.

18.5.1 Der Host- und Domainname

Jede UNIX-Maschine hat standardmäßig einen Namen, den sogenannten Hostnamen. Dieser erhält aber erst seine besondere Bedeutung, wenn die Maschine ins Netz geht, da sie dann unter diesem Namen angesprochen werden kann. Die Zuordnung des Hostnamens zu der Maschine erfolgt durch den Befehl

```
hostname: Festlegen eines Rechnernamens
```

```
hostname <Name>
```

Namensgebung
bei Startup

Dieser Befehl wird im Allgemeinen in einer der rc-Dateien beim Hochfahren des Systems ausgeführt. Seinen Parameter entnimmt der Befehl `hostname` oft einer besonderen Datei, beispielsweise bei Linux der Datei **/etc/HOSTNAME** oder **/etc/hostname**. Solaris besitzt sogar für jeden Netzadapter eine eigene Namensdatei, beispielsweise **/etc/hostname.le0**.

Im Internet wird der einzelne Computer vollständig mit seinem Namen und seiner Domäne identifiziert. Normalerweise wird die Domäne, durch einen Punkt abgetrennt, an den Hostnamen angehängt. Beispielsweise besagt der Name `gaston.willemer.edu`, dass der Hostname `gaston` lautet und der Rechner zur Domäne `willemer.edu` gehört. Innerhalb seiner Domäne braucht der Rechner nicht mit seinem Domänennamen angesprochen zu werden. Diese dreigeteilten Namen kennen Sie aus dem Internet. So ist beispielsweise bei `www.willemer.de` der Anteil `www` der Name des Rechners in der Domäne `willemer.de`.

Um genau zu sein, baut sich ein Domänenname von hinten nach vorn auf. So bezeichnet man das »de« als Toplevel Domain. Neben »de« für Deutschland gibt es unter anderem »nl« für Niederlande oder »dk« für Dänemark. Die Domänen in den USA wurden nicht in einer nationalen Toplevel Domain zusammengefasst, sondern werden in »com« für kommerzielle Organisationen, »gov« für Regierungsstellen und »edu« für Universitäten, Bildungs- und Forschungseinrichtungen eingeteilt. Von der Toplevel Domain wird mit dem Punkt die eigentliche Domäne abgeteilt. Diese Domänen können jeweils durch einen Punkt noch einmal in beliebig viele Subdomänen unterteilt werden. Wird mit dem Namen ein Computer bezeichnet, ist nur der erste Begriff bis zum ersten Punkt der Hostname. Beispielsweise könnte der Praktikumsrechner des Fachbereichs Informatik an der Universität Gintoft so heißen:

```
praktikum.informatik.universitaet.gintoft.de
```

18.5.2 Die Datei `/etc/hosts`

Um einen fremden Rechner nicht immer über seine IP-Adresse ansprechen zu müssen, gibt es die Datei `/etc/hosts`, die jeder IP-Adresse einen oder mehrere Namen zuordnet. Die Struktur eines Eintrags in der Datei `/etc/hosts` sieht folgendermaßen aus:

Struktur einer Zeile in `/etc/hosts`

```
<ipadresse>          <name> <nickname> ... # <Kommentar>
```

Links steht immer die IP-Nummer des Rechners. Das ist die bereits bekannte »dotted decimal«-Schreibweise, also vier Dezimalzahlen, durch Punkte getrennt. Es folgen ein oder mehrere Namen. Der erste Name ist der wichtigste. Wird eine Verbindung von außen aufgebaut, erfährt die Maschine nur die IP-Adresse. Um diesem Rechner einen Namen zu geben, wird der erste Name aus der `/etc/hosts`-Datei verwendet. Auch bei der Zuordnung von Rechten an bestimmte Rechner wird meist ein Name in die jeweilige Konfigurationsdatei eingetragen. Will ein Rechner seine Rechte geltend machen, erfährt das System seine IP-Nummer. Diese wird dann durch den ersten Namen in der `hosts`-Datei ersetzt und mit dem eingetragenen Rechnernamen verglichen. Der erste Name ist also der »offizielle« Name, die anderen Namen bezeichnet man als »nickname« (Spitzname). Nach dem Hashzeichen (#) können Sie Kommentare einfügen. Auch der Hostname der eigenen Maschine sollte in der Datei eingetragen werden, da sonst Dienstanfragen an den eigenen Host über das TCP/IP nicht erkannt werden. Bei Solaris ist es sogar unverzichtbar, da sonst die Maschine ihre IP-Nummer nicht findet.

Die Reihenfolge
der Namen ist
wichtig

Ein Beispiel für eine `/etc/hosts`-Datei wurde bereits im Routingbeispiel auf Seite 472 gezeigt.

Bei der Namensauflösung wird normalerweise zunächst in der Datei `/etc/hosts` nachgesehen. Genauer gesagt, wird die Reihenfolge in der Datei `/etc/host.conf` bzw. `/etc/nsswitch.conf` festgelegt (siehe DNS; Seite 495). Das bedeutet, dass man Namen, die durch einen zentralen Server festgelegt werden sollen, hier nicht eintragen sollte. Würde beispielsweise die Datei `/etc/hosts` folgendermaßen aussehen, dann könnte der Webserver **www.galileo.de** nicht mehr erreicht werden, weil er der lokalen IP-Adresse von `gaston` zugeordnet wäre:¹⁶

Normalerweise
geht die Datei
`hosts` vor

¹⁶ Das braucht Sie nicht zu bekümmern, denn der Verlag mit den fantastischen Büchern hat die Webadresse www.galileocomputing.de.


```
127.0.0.1      localhost
192.168.109.144 gaston.willemer.edu    gaston
192.168.109.144 www.galileo.de
```

Hostnamen
sollte man
klein schreiben

Sie machen sich übrigens keine Freunde im TCP/IP-Umfeld, wenn Sie Ihre Hostnamen in Großbuchstaben setzen. In den meisten Fällen unterscheiden die Netzprotokolle zwischen Groß- und Kleinschreibung, sodass später jeder, der mit diesen Rechnern zu tun hat, mit den Großbuchstaben hantieren muss.

18.5.3 Die Datei `/etc/services`

Für eine Netzwerkanfrage reicht es nicht aus, nur den Namen des Servers zu kennen. Wenn eine Anfrage an einen Netzdienst wie HTTP, FTP oder den Druckdienst gestellt wird, muss man den entsprechenden Prozess wie `httpd`, `ftpd` oder `lpd` auf diesem Rechner erreichen, der die Anfrage beantwortet.

Sockets Um eine Verbindung zwischen zwei Prozessen aufzubauen, wird auf jeder Maschine ein Socket (dt. *Steckdose*) verwendet. Die Sockets eines Rechners sind durchnummeriert. Der Client braucht einen Socket, um über diesen später eine Antwort zu bekommen. Dazu bekommt er irgendeine Socketnummer zugeteilt, die gerade frei ist. Um einen bestimmten Dienst und damit dessen Server zu erreichen, muss der Client den Socket kontaktieren, der von dem Serverprozess betreut wird.

Für Standarddienste hat man die Nummern der Sockets festgelegt. Sie erreichen den Webserver eines Rechners im Regelfall über den Socket 80. Das Protokoll heißt HTTP (Hypertext Transfer Protocol), und der Serverprozess, der Anfragen über dieses Protokoll beantwortet, heißt `httpd`. In Abbildung 18.9 sehen Sie zwei Clientprozesse namens `mozilla` und `netscape`, die auf dem Rechner 192.168.109.137 laufen. Sie stellen beide Anfragen an den Webserver auf dem Rechner 192.168.109.144. Die Sockets, die die Clients verwenden, haben eine beliebige Nummer, die ihnen zufällig vom System zugeordnet wird. Sie rufen den Socket 80 auf dem Zielrechner an. Diese Nummer hat der Webserverprozess `httpd` beim Start angefordert.

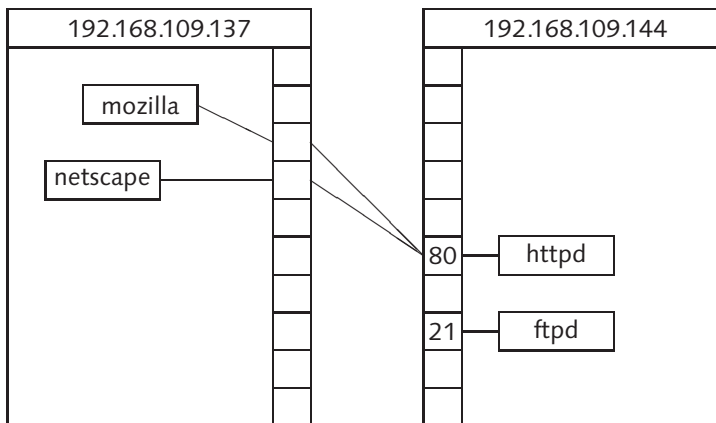


Abbildung 18.9 Kommunikation über Sockets

Im Zusammenhang mit den festgelegten Serversockets wird von einem Port gesprochen. Um den Nummern Namen zuzuordnen, gibt es die Datei **/etc/services**. Hier sehen Sie einen Ausschnitt:

```
ftp-data      20/tcp      # File Transfer [Default Data]
ftp-data      20/udp      # File Transfer [Default Data]
ftp           21/tcp      # File Transfer [Control]
telnet        23/tcp      # Telnet
telnet        23/udp      # Telnet
http          80/tcp      # World Wide Web HTTP
http          80/udp      # World Wide Web HTTP
www           80/tcp      # World Wide Web HTTP
www           80/udp      # World Wide Web HTTP
www-http      80/tcp      # World Wide Web HTTP
www-http      80/udp      # World Wide Web HTTP
```

Unter der Nummer 80 finden Sie den Dienst http und unter 21 den Dienst ftp. Und da ftp in der Regel unter der Portnummer 21 zu erreichen ist, spricht man von einem »well known port« (übersetzt etwa: »wohl bekannter Hafen«).

well known port

Das tcp hinter dem Schrägstrich ist die Kennung für das Protokoll. Neben tcp findet sich in der **/etc/services** noch das Protokoll udp. Es kann für dieselbe Nummer beide Protokolle nebeneinander geben. Ein weiteres Protokoll wurde im Zusammenhang mit dem Befehl ping bereits vorgestellt: ICMP.

Die Protokolle
tcp und udp

TCP (Transmission Control Protocol) hat die Aufgabe, Daten sicher durch das Netzwerk zu transportieren. Dabei überprüft es die Netzpakete anhand einer Prüfnummer auf ihre Unversehrtheit. Beschädigte Pakete wer-

TCP

den neu angefordert. Geraten Pakete in der Reihenfolge durcheinander, weil sie beispielsweise unterschiedliche Wege genommen haben oder defekte Pakete neu angefordert wurden, sorgt TCP für die korrekte Reihenfolge. Den Anwendungsprogrammen gegenüber stellt es einen Datenstrom zur Verfügung. Das Programm muss die Aufteilung auf Pakete also nicht selbst durchführen. Zu guter Letzt veranlasst TCP den geregelten Verbindungsabbau.

UDP UDP (User Datagram Protocol) ist da sehr viel einfacher. Es wird keine Verbindung aufgebaut, sondern es werden einfach Pakete versendet. UDP prüft nicht den Empfang oder die Reihenfolge der Pakete. Es wird lediglich gewährleistet, dass die Pakete unversehrt sind. Dieser fehlende Komfort kommt der Geschwindigkeit zugute. So arbeitet beispielsweise das verteilte Dateisystem NFS auf Basis von UDP.

ICMP ICMP (Internet Control Message Protocol) ist noch einfacher. Es transportiert keine Daten, sondern Fehlermeldungen und Diagnoseinformationen. Das Protokoll wird meist unsichtbar von den Netzwerkschichten verwendet. Lediglich durch die Befehle `ping` und `traceroute` hat der Anwender mit diesem Protokoll direkt zu tun.

18.5.4 Netzgruppen: `/etc/netgroup`

/etc/netgroup Netzgruppen sind eine besondere Zusammenfassung von Benutzern. Sie werden in der Datei **/etc/netgroup** definiert. Jede Netzgruppe hat einen Namen und wird durch ein oder mehrere eingeklammerte Tripel festgelegt, die aus Rechner, Benutzer und Domäne bestehen. Beispiel:

```
awfriends    (,arnold,) (gaston,,) (,willemer.edu)
```

Dieser Eintrag bedeutet, dass die Netzgruppe `awfriends` aus allen Benutzern mit dem Namen `arnold` besteht. Hinzu kommen alle Benutzer eines Rechners `gaston`, ganz gleich in welcher Domain, und alle Rechner der Domäne `willemer.edu`.

Zu Netzgruppen werden Benutzer zusammengefasst, denen gemeinsame Rechte zugewiesen werden sollen. Beispielsweise können in den Konfigurationsdateien von NFS (siehe Seite 581) oder in den Dateien **.rhosts** der `r`-Kommandos (siehe Seite 549) auch Netzgruppen genannt werden. Um sie dort von Benutzern zu unterscheiden, wird ihnen das `@`-Zeichen vorangestellt.

18.5.5 Domain Name Service: DNS

Der Domain Name Service dient zur Auflösung von Host- und Domainnamen in IP-Adressen und umgekehrt. Er übernimmt damit die Aufgabe, die die Datei **/etc/hosts** lokal erfüllt. Der Vorteil von DNS liegt auf der Hand: Änderungen müssen nur an einer Stelle durchgeführt werden und gelten für alle Rechner im Netz.

Ersatz für
/etc/hosts

Die Bedeutung von DNS reicht aber weit über das lokale Netzwerk hinaus. Der Dreh- und Angelpunkt für einen Zugriff auf das Internet ist die Anbindung an einen DNS-Server. Sobald diese Anbindung erfolgt ist, erhalten Namen wie `www.apple.de` oder `arnold@willemer.de` erst die Zuordnung zu den richtigen IP-Adressen.

DNS macht das
Internet erst
erreichbar

Eine Domäne ist nicht unbedingt deckungsgleich mit einem physikalischen Netzwerk, sondern bezeichnet den Bereich, für den der Namensserver eingesetzt wird. Die kleine Firma, die als Beispiel für das Routing diente, hat zwar mehrere TCP/IP-Netze, wird aber sicher nur eine Domäne einrichten.

BIND (Berkeley Internet Name Domain) ist die wichtigste Implementation des DNS. Die Konfiguration des Clients erfolgt in **resolv.conf** und **host.conf** bzw. **nsswitch.conf**. Alle Dateien befinden sich im Verzeichnis **/etc**. Der Server verwendet eine Datei zur allgemeinen Steuerung des Dämons und zwei weitere Dateien je Domäne zur Namens- bzw. zur Nummernauflösung.

Wichtige Dateien

Als Primary Server bezeichnet man denjenigen Namensserver, der die Autorität für die Domäne hat. Um sich gegen einen Ausfall des Primary Servers abzusichern, stellt man einen Secondary Server ins Netz. Dieser übernimmt die Namensauflösung, wenn der Primary Server ausfällt. Seine Informationen übernimmt der Secondary Server über das Netz vom Primary Server. Damit muss nur der Primary Server aktualisiert werden, wenn beispielsweise neue Rechner hinzukommen.

Primary und
Secondary Server

Als weitere Variante gibt es den Cache Only Server. Cache Only Server kopieren sich die Informationen des Primary Servers. Sie werden beispielsweise in Netzen verwendet, die zur Namensauflösung ansonsten eine Wahlleitung in Anspruch nehmen müssten.

Cache Only Server

DNS-Client

Der Client wird als Resolver (engl. *resolve*: auflösen) bezeichnet. Dabei ist der Resolver kein eigenständiger Prozess, sondern ist durch eine Funktionsbibliothek quasi in den einzelnen Programmen enthalten. Al-

le Programme, die mit dem Namen von Rechnern arbeiten, verwenden den Systemaufruf `getservbyname()`, um die IP-Adresse dieses Rechners zu ermitteln (siehe Seite 968). Sobald UNIX diesen Aufruf erhält, prüft `getservbyname()` anhand der Einträge in der Datei **host.conf**, ob die Namensauflösung zuerst per **hosts** oder per DNS erfolgt. In neueren UNIX-Systemen wird auch die Datei **/etc/nsswitch.conf** verwendet. Soll der Name per DNS ermittelt werden, wird in der **resolv.conf** nachgesehen, welche DNS-Server zuständig sind. Dann wird direkt vom Programm aus der DNS-Server kontaktiert. Alle drei Dateien befinden sich im Verzeichnis **/etc**.

/etc/host.conf In der Datei **host.conf** legt die Zeile, die mit `order` beginnt, fest, in welcher Reihenfolge die Namensauflösungsverfahren aufgerufen werden. Im unteren Beispiel wird zuerst in der Datei **/etc/hosts** nachgesehen und anschließend BIND verwendet. Diese Reihenfolge ist normalerweise auch sinnvoll, da der Zugriff auf die lokalen Dateien effizienter ist als die Suche im Netz.

```
order hosts bind
multi on
```

/etc/nsswitch.conf In einigen Systemen wird die Reihenfolge der Informationen aus Diensten oder Konfigurationsdateien aus der Datei **/etc/nsswitch.conf** entnommen. Die Datei vermittelt nicht nur zwischen **/etc/hosts** und DNS, sondern bezieht auch NIS (siehe Seite 505) ein. Für nähere Informationen gibt es eine Manpage, die mit `man nsswitch.conf` aufgerufen werden kann. Der relevante Eintrag für das DNS würde so aussehen:

```
hosts:          files dns
```

Damit werden zuerst die Dateien (`files`) und dann der Namensdienst (`dns`) befragt.

/etc/resolv.conf In der Datei **/etc/resolv.conf** wird definiert, zu welcher Domäne der Rechner gehört und welche Rechner Namensserver sind. Um meinen Arbeitsplatzrechner an den DNS-Dienst des Internets zu koppeln, reicht eine kleine **resolv.conf**. Beispiel:

```
domain    willemer.edu
nameserver 194.25.2.129    # frage den ISP
```

Eine Namensanfrage wird aufgrund der Datei **host.conf** zunächst lokal in der Datei **/etc/hosts** bedient, und alle anderen Namen werden beim DNS des Providers gesucht.

Wollen Sie das TCP/IP-Netz der Beispielfirma an das Internet anbinden, reicht es, eine etwas größere **resolv.conf** zu verwenden. Dabei wird hier angenommen, dass die Firma die Domäne `firma.de` hätte.

```
domain      firma.de
nameserver  192.168.108.201 # frage zuerst unseren Server
nameserver  194.25.2.129    # dann frage den ISP
```

Zuerst werden die Namen lokal geprüft. Damit gehen lokale Namensauflösungen nicht ins Internet. Erst wenn der Name hier unbekannt ist, wird der DNS-Server des Internet Service Providers (ISP) gefragt.

Um für eine vollständige Anbindung zu sorgen, müssen Sie natürlich ein geeignetes Routing einrichten. Da die meisten Firmen über keine eigenen internetfähigen IP-Adressen verfügen, muss noch ein Proxy (siehe Seite 571) oder ein Masquerading (siehe Seite 570) eingerichtet werden. Beides sind unterschiedliche Mechanismen, um über einen anderen Rechner, der eine Verbindung zum Internet hat, einen Zugang zu erhalten, obwohl der eigene Computer keine internetfähige Adresse hat.

Der DNS-Server `named`

Der Serverprozess des DNS heißt `named`. Er verwaltet die Namenszuordnungen der IP-Adressen einer Zone. Eine Zone ist quasi eine Ebene einer Domäne. Die oberste Zone ist die Domäne selbst ohne ihre Subdomänen. Für jede der Subdomänen ist wiederum ein eigener DNS-Server zuständig.

`/etc/named.conf`

Der DNS-Server verwendet mehrere Dateien zur Konfiguration. Die Ausgangsdatei heißt **`/etc/named.conf`**. Sie legt die Rolle des Rechners im DNS fest und definiert die Namen und Orte der weiteren Dateien. Dann gibt es für jede Zone zwei weitere Dateien. Die eine bildet die Namen auf IP-Adressen ab, und die andere nennt den Standardnamen für eine IP-Adresse.

Die folgende Konfigurationsdatei verwaltet die Domäne `willemer.edu` mit der IP-Adresse `192.168.109.0` und stellt eine Verbindung zum Internet und dessen DNS-Server unter der IP-Adresse `194.25.2.129` her. Da `willemer.edu` keine Subdomäne besitzt, ist hier die Zone mit der Domäne identisch.

`named.conf`

```
options {
    directory "/var/lib/named";
    forward only;
    forwarders { 194.25.2.129; };
};
```

```

zone "willemer.edu" {
    type master;
    file "willemer.edu.zone";
};

zone "109.168.192.in-addr.arpa" {
    type master;
    file "192.168.109.zone";
};

zone "localhost" {
    type master;
    file "localhost.zone";
};

zone "0.0.127.in-addr.arpa" {
    type master;
    file "127.0.0.zone";
};

zone "." {
    type hint;
    file "root.hint";
};

```

options Im Bereich options werden Einstellungen vorgenommen, die den Namensserver als Ganzes betreffen. Mit dem Schlüsselwort `directory` wird festgelegt, wo sich die weiteren Konfigurationsdateien des Namensservers befinden. Für jede Zone und jedes Netzwerk wird eine eigene Datei benötigt.

Weiterleiter Mit dem Schlüsselwort `forward` wird festgelegt, ob Anfragen, die dieser Server nicht beantworten kann, an weitere DNS-Server weitergeleitet werden. Der Wert `only` könnte zwar assoziieren, dass nur Namensinformationen weitergereicht werden. Gemeint ist allerdings, dass der DNS-Server nur Kontakt mit den unter `forwarder` aufgeführten Rechnern aufnehmen soll. In der geschweiften Klammer hinter `forwarders` werden die IP-Adresse der Rechner angegeben, die angefragt werden sollen, wenn der lokale DNS-Server nicht weiter weiß. Wichtig ist das Semikolon am Ende der Nummern. Soll das lokale Netz gar nicht mit dem Internet oder einem anderen Netz verbunden werden, sind die beiden Einträge nicht erforderlich.

Die erste Zone wird für die Domäne **willemer.edu** definiert. Mit dem Schlüsselwort `type` wird festgelegt, dass dieser Server der Master der Domäne ist. Hinter dem Schlüsselwort `file` wird der Dateiname festgelegt, in dem sich die Zuordnung der Namen zu den IP-Adressen befindet. Der Name ist frei wählbar. Es erleichtert natürlich die Übersicht, wenn der Domänenname Bestandteil des Dateinamens ist.

Zonen

Sollen die Rechnernamen aus den IP-Adressen ermittelt werden, muss es eine zweite Tabelle geben. Diese Informationen werden verwendet, wenn IP-Verbindungen als Namen angezeigt werden sollen oder wenn es notwendig ist, eine Anfrage mit einem angegebenen Hostnamen zu vergleichen. Enthält die Domäne mehrere Netzwerke, wird für jedes Netzwerk eine eigene Datei angelegt.

Nummern-
auflösung

Die älteren BIND-Versionen verwendeten eine Datei namens **named.boot**. Die folgende Datei hat in etwa die gleiche Wirkung wie die oben gezeigte **named.conf**, unterscheidet sich allerdings syntaktisch sehr deutlich:

BIND 4:
named.boot

```
primary   willemer.edu           /var/named/willemer.edu.zone
primary   109.168.192.IN-ADDR.ARPA /var/named/192.168.109.zone
cache     .                      /var/named/named.root
forwarders 194.25.2.129
```

Primary Server wird ein DNS-Server genannt, der die Hostnamen für die Domäne verantwortlich verwaltet. Die erste Zeile drückt aus, dass dieser Rechner ein Primary Server für die Domäne **willemer.edu** ist. Die Namensinformationen der Domäne werden in der Datei **willemer.edu.zone** gepflegt. Für den umgekehrten Weg, um also aus den Nummern die Namen zu ermitteln, wird **192.168.109.zone** verwendet. Die Cachedatei dient als Basis für das Vorhalten der Namen aus dem Internet. Wie Sie an diese Datei gelangen, wird später beschrieben. Der letzte Eintrag gibt den Namensserver an, an den die Anfrage weitergeleitet werden soll, wenn der Name aus einer fremden Domäne stammt. Beispielsweise kann hier der DNS-Server des Internet Providers stehen. Ein Secondary Server hätte eine ganz ähnliche **/etc/named.boot**:

```
directory /var/named
secondary willemer.edu           192.168.109.144 willemer.edu.zone
secondary 109.168.192.IN-ADDR.ARPA 192.168.109.144 192.168.109.zone
```

Der Unterschied besteht in der ersten Spalte der zweiten und dritten Zeile. Hier sehen Sie, dass dieser Rechner ein Secondary Server für **willemer.edu** ist. Ferner ist die IP-Adresse des Primary Servers angegeben. Schließlich wird definiert, in welcher lokalen Datei die Informationen vom Primary Server abgelegt werden. Die erste Zeile (`directory`) ist

lediglich eine Abkürzung, damit man den Pfadnamen nicht bei jeder Datei-angabe wiederholen muss.

secondary = slave Ab der Version 8 wird ein Secondary Server als Slave bezeichnet. Die Bezeichnung Slave findet sich in den Konfigurationsdateien nicht mehr. Stattdessen wird eine `forward-Option` auf den Primary Server und die Option auf `forward only` gesetzt. Damit ist es funktional ein Secondary Server.

Auflösung: Name nach Nummern

Die Zuordnung der Namen und IP-Adressen erfolgt weder in der Datei **named.conf** noch in **named.boot**, sondern in den Dateien, auf die dort verwiesen wird.

**Tabelle Name
nach Nummer**

Die folgende Datei ermittelt aus den Namen die IP-Adressen. In den Beispielen wurde diese Datei als **willemer.edu.zone** bezeichnet. Die Datei ist bei der neueren BIND-Version gleich geblieben. Lediglich die erste Zeile ist neu:

```
$TTL 2W
@      IN SOA  mail.willemer.edu.  root.mail.willemer.edu. (
                                200111303      ;serial
                                3600000         ;refresh every 100 hours
                                3600           ;retry after 1 hour
                                3600000        ;expire after 1000 hours
                                360000        ;default ttl is 100 hours
                                )
;      Wer ist der zuständige Namensserver
;      IN NS  gaston.willemer.edu.

;      Wer sind die zuständigen Mail-Server
;      IN MX 10 mail.willemer.edu.

localhost      IN A      127.0.0.1

;      Alle Rechner in der eigenen Domain
mail           IN A      192.168.109.137
asterix        IN CNAME  mail
gaston         IN A      192.168.109.144
powermac       IN A      192.168.109.141
```

TTL Neu ist, dass der TTL-Wert (*Time To Live*; engl. Lebenszeit) bereits am Anfang der Datei stehen muss. Dieser Wert gibt an, wann die Daten von einem anderen Server erneut gelesen werden müssen. In der Version 8 muss er noch nicht zwingend dort stehen; in der Version 9 wird der

Eintrag allerdings verlangt. Er befindet sich entweder am Anfang der Datei vor dem SOA-Eintrag in der folgenden Form:

```
$TTL 1W
```

Oder Sie setzen den Wert im SOA-Eintrag direkt vor das Schlüsselwort IN:

```
@      86400 IN SOA ns hostmaster (
```

Fehlt dieser Eintrag, finden Sie in den syslog-Protokollen (siehe Seite 420), also beispielsweise in der **messages**-Datei, die Meldung »no TTL specified«.

Die zweite Zeile beginnt mit einem @. Es steht für den lokalen Domännennamen. Hier könnte also auch willemer.edu stehen. Die Abkürzung SOA bedeutet »Start Of zone Authority«. Die so bezeichnete Zeile gibt den Standardnameserver der Domäne und die E-Mail-Adresse des Verantwortlichen an. Allerdings wird das @ in der Mail-Adresse durch einen Punkt ersetzt. gaston ist der Namensserver (IN NS). Der Mail-Server der Domäne ist der Rechner mail (IN MX), der aber auch unter dem Namen asterix im Netz agiert.

Ein Fallstrick ist, dass bei den neuen Versionen von BIND die öffnende Klammer hinter dem SOA-Eintrag noch in der gleichen Zeile stehen muss und nicht in der Folgezeile stehen darf.

Achtung: Offene Klammer

In der Klammer werden Parameter gesetzt, die Informationen über das Update-Verhalten festlegen. Die Zeile mit dem Kommentar *serial* kann irgendeine Versionsnummer sein, die allerdings bei jeder Änderung steigen muss. Es hat sich eingebürgert, hier das Datum in der Darstellung YYYYMMDD, gefolgt von einer mehrstelligen Zahl, zu verwenden, die hochgezählt wird, wenn am selben Tag mehrere Änderungen durchgeführt werden. In diesem Fall wurde die Datei zuletzt am 13.11.2001 und an diesem Tag das dritte Mal geändert.

Versionsinformationen

Die anderen Zeilen liefern folgende Informationen:

► **IN NS**

Der zuständige Namensserver für die Domäne.

► **IN MX Nummer**

Der Mail-Server. Es können mehrere Server angegeben werden. Die Nummer gibt die Rangfolge an. Je kleiner die Nummer ist, desto höher ist der Rang.

► **IN A**

definiert die IP-Adresse eines Hostnamens. Ein Hostname wird nur einmal auf eine IP-Adresse abgebildet. Weitere Namen der gleichen IP-Adresse werden per CNAME angegeben.

► **IN CNAME**

definiert für einen Rechner, der bereits mit IN A angegeben ist, einen weiteren Namen. Man spricht hier von einem Nickname, also einem Spitznamen.

Es werden die Namen mail, gaston und powermac auf IP-Adressen abgebildet, und für mail wird der Nickname asterix festgelegt.

Bei den in dieser Datei vorkommenden Namen wird vom System der Domänenname ergänzt, wenn der Name nicht mit einem Punkt beendet wird. Darum steht hinter mail.willemer.edu ein Punkt. Taucht bei einem Test der Domänenname doppelt auf (also ein Hostname wie max.willemer.edu.willemer.edu), dann fehlt sicher irgendwo ein Punkt.

Tabelle Nummer
nach Name

Die Datei **192.168.109.zone** ist das Gegenstück zur Namenstabelle. Hier werden die Hostanteile der IP-Adressen auf Namen abgebildet. Bei einem Class-B-Netz würden also zwei durch einen Punkt getrennte Nummern in der linken Spalte stehen. Diese Umsetzung von Nummern auf Namen wird im Allgemeinen bei Berechtigungsprüfungen oder bei Anzeigen verbundener Rechner und Ähnlichem benötigt.

Der erste Teil der Datei entspricht der Datei, die zur Namensauflösung verwendet wird. Erst zum Ende der Datei erscheinen zuerst die Hostnummern und durch IN PTR getrennt die vollständigen Namen der Rechner.

```
$TTL 2W
@      IN SOA      mail.willemer.edu. root.mail.willemer.edu. (
                                1999062702      ;serial
                                360000            ;refresh every 100 hours
                                3600              ;retry after 1 hour
                                3600000           ;expire after 1000 hours
                                360000            ;default ttl is 100 hours
                                )
;
                                Wer ist der zuständige Namensserver
                                IN NS mail.willemer.edu.

137    IN PTR      mail.willemer.edu.
144    IN PTR      gaston.willemer.edu.
```

Testen

Das Testen einer DNS-Konfiguration erfolgt mit dem Befehl `nslookup`. Nach dem Aufruf meldet sich `nslookup` mit dem zuständigen DNS-Router. Anschließend können Sie Hostnamen eingeben, und `nslookup` zeigt die ermittelte IP-Adresse an:

```
gaston > nslookup
Default Server:  www-proxy.KI1.srv.t-online.de
Address:  212.185.254.170

> www.willemer.de
Server:  www-proxy.KI1.srv.t-online.de
Address:  212.185.254.170

Non-authoritative answer:
Name:  www.willemer.de
Address:  212.227.118.90

> www.apple.de
Server:  www-proxy.KI1.srv.t-online.de
Address:  212.185.254.170

Non-authoritative answer:
Name:  www.germany.euro.apple.com
Address:  17.254.3.153
Aliases:  www.apple.de
> exit
```

Hier wurde in einer Internetsitzung nach den Adressen `www.willemer.de` und der Adresse `www.apple.de` gefragt. Beendet wird die Sitzung mit dem Befehl `exit`. Der wiederholt auftretende Kommentar »Non-authoritative answer« bedeutet, dass der angerufene Server (hier von T-Online) kein autorisierter DNS-Server der Domäne `willemer.de` oder `apple.de` ist.

`nslookup` kennt die Option `server`, mit der angegeben werden kann, welcher DNS-Server gefragt werden soll. Diese Möglichkeit kann genutzt werden, um festzustellen, ob der eigene DNS-Server seine Informationen auch an fremde DNS-Server exportiert. Dies wird man in privaten Netzen vermeiden wollen.

Das Programm `dig` (domain information groper) wird als moderne Alternative zu `nslookup` angesehen. Im Gegensatz zu diesem hat `dig` keinen interaktiven Modus, sondern wird mit jedem Ziel einzeln aufgerufen. Dafür sind die Ergebnisse sehr viel umfassender.

Alternative `dig`

```

libo> dig www.apple.de

; <<>> DiG 9.3.2 <<>> www.apple.de
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 60368
;; flags:qr rd ra; QUERY:1, ANSWER:2, AUTHORITY:6, ADDITIONAL:0

;; QUESTION SECTION:
;www.apple.de.                IN      A

;; ANSWER SECTION:
www.apple.de.                 53295   IN      CNAME  euro-red.apple.com.
euro-red.apple.com. 53295   IN      A      17.254.3.122

;; AUTHORITY SECTION:
apple.com.                    390913  IN      NS      nserver.asia.apple.com.
apple.com.                    390913  IN      NS      nserver.apple.com.
apple.com.                    390913  IN      NS      nserver4.apple.com.
apple.com.                    390913  IN      NS      nserver3.apple.com.
apple.com.                    390913  IN      NS      nserver2.apple.com.
apple.com.                    390913  IN      NS      nserver.euro.apple.com.

;; Query time: 354 msec
;; SERVER: 192.168.109.1#53(192.168.109.1)
;; WHEN: Thu Sep 20 14:36:18 2007
;; MSG SIZE rcvd: 223

libo>

```

Root-Cache-Dateien

Für einen Cache Only Server benötigen Sie eine Ausgangsdatei, in der die Adressen der DNS-Server der Rootzone verzeichnet sind. Diese Datei können Sie unter dem Namen **named.root** per ftp vom Server **rs.internic.net** mit der Adresse 198.41.0.7 aus dem Verzeichnis **domain** herunterladen. Diese Datei muss als Ausgangsdatei für den Cache verwendet werden. Im Folgenden sehen Sie ein Protokoll, das zeigt, wie man die Datei herunterlädt:

```

gaston # ftp rs.internic.net
Connected to rs.internic.net.
220-*****
220-*****
220-***** InterNIC Public FTP Server
220-*****

```

```

220-***** Login with username "anonymous" *****
220-***** You may change directories to the following: *****
220-***** *****
220-***** domain - Root Domain Zone Files *****
220-***** *****
220-***** Unauthorized access to this system may *****
220-***** result in criminal prosecution. *****
220-***** *****
220-***** All sessions established with this server are *****
220-***** monitored and logged. Disconnect now if you do *****
220-***** not consent to having your actions monitored *****
220-***** and logged. *****
220-***** *****
220-*****
220-
220 FTP server ready.
Name (rs.internic.net:root): anonymous
331 Guest login ok, send your e-mail address as password.
Password:
230 User ftp logged in. Access restrictions apply.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> cd domain
250 CWD command successful.
ftp> get named.root
local: named.root remote: named.root
500 'EPSV': command not understood.
227 Passive Mode (198,41,0,6,177,154)
150 Opening BINARY mode data connection for named.root.
100% |*****| 2769 3.91 KB/s 00:00 ETA
226 Transfer complete.
2769 bytes received in 00:00 (3.07 KB/s)
ftp> quit
221-You have transferred 2769 bytes in 1 files.
221-Total traffic for this session was 4586 bytes in 1 transfer.
221 Thank you for using the FTP service on rs.internic.net.
gaston #

```

Die Datei **named.root** enthält die DNS-Server der Rootdomain, also die letzte Instanz.

18.5.6 Network Information Service: NIS

NIS (Network Information Service) hat die Aufgabe, in einem UNIX-Netzwerk die Informationen auf allen Maschinen konsistent zu halten. NIS

NIS zentralisiert
Konfigurations-
dateien

verwaltet Konfigurationsdateien und stellt sie netzweit zur Verfügung. Dadurch ist NIS nur für Maschinen mit UNIX interessant, da andere Systeme beispielsweise mit der Datei **/etc/passwd** wenig anfangen können.

Die Konfigurationsdateien werden auf dem Zentralrechner, dem Master, in sogenannte Maps umgewandelt. Anschließend kann der Client seine Informationen per Netz vom NIS holen. Da die NIS-Informationen auf dem Netz nicht besonders gesichert sind, eignet sich NIS nicht für Netze mit besonderen Sicherheitsanforderungen.

Zentrales Passwort In den meisten Fällen wird man die Passwortdatei von NIS verwalten lassen, um die Benutzerkonten der Anwender über alle Maschinen konsistent zu halten. Die Datei **/etc/passwd** wird also zentralisiert. Damit haben Sie auf allen Rechnern im Netz das gleiche Passwort und müssen es nur an einer Stelle verwalten.

Yellow Pages Die Namen der Befehle von NIS beginnen meist mit `yp`. Das hängt damit zusammen, dass Sun NIS ursprünglich »yellow pages«, also »Gelbe Seiten«, nennen wollte. Die britische Telefongesellschaft hatte allerdings ein Markenrecht auf den Begriff, sodass Sun den Dienst in NIS umbenennen musste.

Aufbereiten der Dateien

Makefile als Konfigurationstool Im ersten Schritt muss festgelegt werden, welche Dateien per NIS verwaltet werden. Diese Dateien werden dann in die sogenannten Maps überführt. Der zentrale Befehl dazu lautet `makedbm`. Er wird allerdings normalerweise nicht direkt aufgerufen, sondern über ein Makefile, das sich im Verzeichnis **/var/yp** befindet. Hier werden später auch die Maps liegen. Ein Makefile ist eine editierbare Datei zum Generieren bestimmter Zielformate und wird normalerweise in der Programmierung verwendet (siehe Seite 868).

Ändern Sie die Datei **Makefile** mit Ihrem Lieblingseditor. Sie finden hinter dem Label `all`: alle Dateien, die für das NIS umgestellt werden sollen. Mit dem Kommentarzeichen `#` können Sie die Dateien ausschließen, die Sie nicht mit NIS in dieser Domäne verwalten wollen:

```
# If you don't want some of these maps built, feel free to
# comment them out from this list.
```

```
#all: passwd group hosts rpc services netid protocols \
# netgrp mail shadow publickey networks ethers \
# bootparams printcap amd.home auto.master auto.home \
# auto.local passwd.adjunct timezone locale netmasks
all: passwd group rpc services netid
```

Nachdem **Makefile** angepasst ist, werden durch einen Aufruf von `make` mit dem Parameter `all` die Map-Dateien erstellt (auf manchen Systemen gibt es auch ein spezielles `ypmake`):

```
make all
```

Anschließend sollten im Verzeichnis **/var/yp** die Maps zu finden sein, mit deren Hilfe NIS die Anfragen später beantworten wird.

NIS-Server starten

Im nächsten Schritt wird die NIS-Datenbank für den Master initialisiert:

```
gaston# cd /var/yp
gaston# domainname willemer.edu
gaston# ypserv
gaston# /usr/lib/yp/ypinit -m
```

```
At this point, we have to construct a list of the hosts which
will run NIS servers.  gaston.willemer.edu is in the list of
NIS server hosts.  Please continue to add the names for the
other hosts, one per line.  When you are done with the
list, type a <control D>.
```

```
    next host to add: gaston.willemer.edu
```

```
    next host to add:
```

```
The current list of NIS servers looks like this:
```

```
gaston.willemer.edu
```

```
Is this correct? [y/n: y] y
```

```
We need some minutes to build the databases...
```

```
Building /var/yp/willemer.edu/ypservers...
```

```
Running /var/yp/Makefile...
```

```
gmake[1]: Entering directory `/var/yp/willemer.edu'
```

```
Updating passwd.byname...
```

```
Updating passwd.byuid...
```

```
Updating group.byname...
```

```
Updating group.bygid...
```

```
Updating rpc.byname...
```

```
Updating rpc.bynumber...
```

```
Updating services.byname...
```

```
Updating services.byservicename...
```

```
Updating netid.byname...
```

```
gmake[1]: Leaving directory `/var/yp/willemer.edu'
```

```
gaston#
```


ypinit Nach dem Start von `ypinit` fragt das Programm nach allen NIS-Servern. Der eigene Rechner wird automatisch angezeigt. Hier können Sie weitere Slave-Server eintragen, die bei einem Ausfall des Masters einspringen. Wenn keine weiteren NIS-Server in der Domäne arbeiten sollen, wird die Eingabe mit der Tastenkombination **ctrl+D** abgeschlossen.

Der Passwortserver Damit kann dieser NIS-Server bereits Anfragen beantworten. Passwörter haben eine besondere Position. Sie sollen leicht von den Benutzern auf jedem beliebigen Client geändert werden können. Die Änderungen sollen dann sofort auf dem Masterserver aktualisiert werden. Das erledigt der zum NIS gehörige Passwortserver. Er wird wie folgt auf dem Masterserver gestartet:

```
rpc.yppasswdd -s /etc/yp/shadow -p /etc/yp/passwd -e chsh
ypserv
```

Nun sollte ein Client mit dem Aufruf von `yppasswd` sein Passwort domänenweit ändern können.

Überprüfen des Servers NIS basiert auf dem RPC (Remote Procedure Call), der durch den Prozess `portmap` zur Verfügung gestellt wird. Dieser Prozess wird bei jedem auf RPC basierenden Server benötigt, da er die Umsetzung der Funktionsnummern durchführt. Das Programm `rpcinfo` hilft zu prüfen, ob der NIS-Server aktiv und sauber eingebunden ist:

```
gaston # rpcinfo -u localhost ypserv

program 100004 version 1 ready and waiting
program 100004 version 2 ready and waiting

gaston #
```

Die oben angegebene Meldung zeigt an, dass der NIS-Server korrekt in das RPC eingebunden ist und läuft.

Starten eines NIS-Clients

Zunächst muss die Domäneninformation stimmen. Dies kann durch einen Aufruf des Befehls `domainname` geprüft werden. Übrigens muss die Domäne des NIS nicht mit der Domäne des DNS übereinstimmen. Anschließend wird der Client-Dämon gestartet. Er heißt `ypbind`:

```
silver# domainname willemer.edu
silver# ypbind -broadcast
```

Diese beiden Befehle sollten in den rc-Dateien des Systems stehen, damit NIS nach dem Start bereits zur Verfügung steht. Wer Daten aus der **/etc/passwd** per NIS ermitteln will, sollte als letzten Eintrag in der lokalen **passwd**-Datei die folgende Zeile eintragen:

```
+:::~:
```

Anschließend kann mit dem Befehl `yycat` getestet werden, ob die Verteilung der **passwd**-Datei durch den NIS-Server funktioniert:

```
silver# yycbind -broadcast
silver# yycat passwd
```

Der Befehl `yycat` muss eine Liste von Einträgen der **passwd**-Datei liefern: Dann ist alles in Ordnung.

18.5.7 Portable Verzeichnisse LDAP

Ein Informationsdienst wie NIS ist in größeren Netzwerken allein für die zentrale Benutzerverwaltung unverzichtbar. Leider ist NIS sehr auf UNIX-spezifische Konfigurationsdateien festgelegt. In heutigen Netzwerken erwartet aber der Anwender, dass er sich mit seinem UNIX-Kennwort auch am Windows-Netz anmelden kann. Eine solche Verwaltung netzwerkweiter Daten ist mit LDAP (Lightweight Data Access Protocol) auch in heterogenen Netzwerken möglich.

LDAP ist aber nicht auf die Benutzerverwaltung beschränkt. Es kann beliebige Daten speichern, die netzwerkweit gebraucht werden. Beispielsweise kann auch das Kundenverzeichnis einer Firma über LDAP realisiert werden. Diese Flexibilität hat dazu geführt, dass immer mehr Netzwerkdienste die Option bieten, ihre Konfigurationsdaten zumindest teilweise durch LDAP zu verwalten.

Flexible Struktur

Von allen LDAP-Implementationen dürfte OpenLDAP am weitesten verbreitet sein. Aus diesem Grund wird das Vorgehen auf dieser Basis beschrieben.

OpenLDAP

Verzeichnisstruktur

In der Tabelle 18.7 sind die Objektklassen aufgeführt.

Kürzel	Objektklasse
c	Country, also Land
l	Location
o	Organisation
ou	Organisatorische Einheit
cn	Datenblatt

Tabelle 18.7 Objektklassenkürzel

Die Daten werden bei LDAP in einer Struktur abgelegt. Die Grundstruktur ist vorgegeben und unterliegt gewissen Regeln.

- ▶ Das Verzeichnis beginnt mit der Wurzel. Es existiert also immer ein Objekt namens Root.
- ▶ Unterhalb von Root liegt entweder ein Country (c) oder eine Organisation (o).
- ▶ Das Country-Objekt muss nicht, aber darf nur einmal existieren.
- ▶ Die Organisation muss direkt unter dem Country-Objekt stehen. Fehlt das Country, steht die Organisation direkt unter Root.
- ▶ Es darf mehrere Organisationen nebeneinander geben.
- ▶ Unterhalb einer Organisation muss ein Blatt (cn) oder eine organisatorische Einheit (ou) stehen.
- ▶ Es darf beliebig viele organisatorische Einheiten auf gleicher Ebene geben.
- ▶ Blätter dürfen nur an Organisationen (o) oder organisatorischen Einheiten (ou) hängen.

Bestimmte Objektklassen kommen immer wieder vor und sind bereits vordefiniert.

Person Ein Objekt vom Typ Person hat immer einen CommonName (cn) und einen Surname (sn) und kann noch eine Description (desc), eine telephoneNumber, ein userPassword und ein seeAlso besitzen.

Ein Beispiel

Von der Wurzel ausgehend, wird eine Organisation namens MeineOrg definiert. Abteilungen sollen »Verkauf« und »Entwicklung« sein.

LDAP-Server-Konfiguration

Die grundlegende Konfiguration des Serverprozesses `slapd` erfolgt in der Datei `/etc/openldap/slapd.conf`. Eine Zeile besteht aus einem Parameter und dessen Wert.

Mit dem Parameter `include` wird eine weitere Datei in die Konfigurationsdatei hineingeladen. Das macht es möglich, die Konfiguration auf mehrere Dateien zu verteilen und etwas übersichtlicher zu halten. include

```
include <DateiMitPfad>
```

Das ist auch bitter nötig. Denn mit LDAP kommt ein Rudel Schemata, die bestimmte Objekte vordefinieren, so dass nicht jeder Administrator das Rad neu erfinden muss. In den meisten Fällen gibt es auch eine Beispielkonfiguration, die einfach angepasst wird. In dieser Umgebung waren folgende Dateien standardmäßig eingebunden.

```
include    /etc/openldap/schema/core.schema
include    /etc/openldap/schema/cosine.schema
include    /etc/openldap/schema/inetorgperson.schema
include    /etc/openldap/schema/rfc2307bis.schema
include    /etc/openldap/schema/yast.schema
```

Der Parameter `schemacheck` legt fest, ob die Konfiguration sich an Richtlinien halten muss (on) oder völlig frei ist (off). schemacheck

```
schemacheck    off
```

Dieser Parameter legt fest, welches Format für die Datenbank verwendet wird. Welche Datenbank verwendet wird, hängt vom Zielsystem ab. Bei einigen Maschinen lautet der Eintrag `ldbm`, bei anderen `bdb`. Auch hier lohnt es sich, in der Beispielkonfiguration zu spicken. database

Der Parameter `directory` legt fest, in welchem Verzeichnis sich die Datenbank von LDAP befindet. In den meisten Fällen wird das `/var/lib/ldap` sein. directory

`lastmod` legt fest, ob der LDAP-Server automatisch notiert, wer das Objekt zuletzt bearbeitet hat. Das kann erst einmal off bleiben. lastmod

Der Parameter `suffix` bezeichnet die Bereiche, für die der Server die Informationen halten soll. In unserem Fall wird hier die Organisation eingetragen. suffix

```
suffix      "o=MeineOrg"
```

Für den Anfang wird ein root-Zugang eingebaut. `rootdn` bezeichnet die Benutzerkennung des Administrators, `rootpw` dessen Passwort.

```
rootdn      "cn=admin, o=MeineOrg"
rootpw      GehHeim
```

Zugriffe Für die Zugriffe gibt es die Parameter `defaultaccess` und `access`. Der Parameter `defaultaccess` beschreibt, welche Rechte jemand hat, der ohne Kennung und Passwort zugreift. Der Parameter kann die Werte `none`, `compare`, `search`, `read`, `write` oder `delete` annehmen oder eine Kombination aus diesen Rechten, die durch einen senkrechten Strich getrennt werden. Der Parameter `access` kann mehrfach auftreten. Er beschreibt, wer was mit welchen Objekten tun darf.

```
access to {<Objekt>|<Attribut>}
by <Objekt> <Recht>
[ by <Objekt> <Recht> ]*
```

Hier ist die Konfigurationsdatei im Überblick:

```
include      /etc/openldap/schema/core.schema
include      /etc/openldap/schema/cosine.schema
include      /etc/openldap/schema/inetorgperson.schema
include      /etc/openldap/schema/rfc2307bis.schema
include      /etc/openldap/schema/yast.schema

schemacheck  off
database     bdb
directory    /var/lib/ldap
lastmod      off
suffix       "o=MeineOrg"
rootdn       "cn=admin, o=MeineOrg"
rootpw       GehHeim
defaultaccess read
access to attrs=userPassword
            by self write
            by * none

access to dn="cn=admin,o=MeineOrg"
            by * none

access to *
            by * read
```

Das Passwort ist hier im Klartext. Üblicherweise wird hier ein verschlüsseltes Passwort hinterlegt. Der Aufruf von `slappasswd` erstellt Passwörter nach der RFC2307.

```
libo # slappasswd
New password:
Re-enter new password:
{SSHA}TuZRRgpwrjdI63xhs/ap4Gh/AyfvPuLZ
libo #
```

Nun wird der Eintrag entsprechend angepasst:

```
rootpw {SSHA}TuZRRgpwrjdI63xhs/ap4Gh/AyfvPuLZ
```

Erster Test

Der LDAP-Dämon wird der Einfachheit halber über den Startskript gestartet.

```
libo# /etc/init.d/slaped start
```

Probleme werden in den Dateien **messages** oder **syslog** im Verzeichnis **/var/log** abgelegt. Die Fehlermeldungen geben insbesondere über Fehler in der Konfigurationsdatei Auskunft. Fehler

Daten per LDIF

LDAP speichert seine Daten in Binärdateien, auf die kein direkter Zugriff besteht. Sie werden befüllt, indem LDIF-Dateien importiert werden. LDIF-Dateien können Sie mit einem normalen Editor bearbeiten. Der Aufbau eines Eintrags in der LDIF-Datei folgt dem Muster:

```
dn: <Distinguished Name>
<Attribut>: <Wert>
[ <Attribut>: <Wert> ]*
```

Nun wird eine Datei namens **MeineOrg.ldif** erzeugt, die folgenden Inhalt hat:

```
dn: o=MeineOrg
objectclass: organization
o: MeineOrg

dn: cn=admin, o=MeineOrg
objectclass: person
cn: admin
description: LDAP-Administrator
```

```
dn: cn=arnold, o=MeineOrg
objectclass: person
cn: arnold
telephonenumber: 1024
mail: arnold.willemer@gmx.de
description: Arnold Willemer
```

Diese Datei kann mit dem Befehl `ldapadd` in die Datenbank übernommen werden.

```
ldapadd -h libo -D "cn=admin, o=MeineOrg" -W -f MeineOrg.ldif
```

Daraufhin fragt der Befehl nach dem Passwort und fügt die Daten in die Datenbank ein.

LDAP-Client-Konfiguration

Ein LDAP-Client muss wissen, in welcher Domäne er sich bewegt und welcher Rechner für ihn zuständig ist. Diese Informationen finden die Client-Prozesse in der Datei **/etc/openldap/ldap.conf**.

```
BASE    o=MeineOrg
HOST    libo
```

ldapsearch Für die Suche nach Einträgen wird das Kommando `ldapsearch` aufgerufen. Um den Eintrag für `arnold` zu suchen, lautet der Befehl:

```
ldapsearch "cn=arnold"
```

Es können für die Suche Wildcards verwendet werden.

ldapdelete Einträge können durch `ldapdelete` gelöscht werden.

```
ldapdelete -D "cn=admin,o=MeineOrg" -W "cn=arnold,o=MeineOrg"
```

18.6 Dynamische IP-Adressen (DHCP)

DHCP dient wie das ältere bootp dazu, die IP-Adresse eines Netzes an einem zentralen Ort zu verwalten. DHCP ist in RFC 2131 und 2132 beschrieben.

DHCP findet seine Anwendung in großen Netzwerken, bei denen möglichst ohne Eingriffe des Administrators kurzfristig neue Rechner einge-

hängt werden sollen. Insbesondere beim Einsatz von Laptops, die man schnell ins Netz einbinden will, ist dieses Verfahren sehr hilfreich.¹⁷

Für den Anwender ist DHCP äußerst bequem. Der Rechner fragt beim Booten nach der Netzkonfiguration. Nichts muss eingestellt werden. Wie so oft geht auch hier die Bequemlichkeit auf Kosten der Sicherheit. So kann jeder Mitarbeiter der Firma ohne Rückfrage an die Administration sein privates Notebook in das Netzwerk einbinden.

18.6.1 DHCP-Clients

In den meisten Fällen wird der Anwender nicht direkt mit dem DHCP-Client konfrontiert. Wenn Sie Ihr Betriebssystem installieren, werden Sie meist gefragt, ob Ihr Rechner per DHCP seine Netzinformationen holen soll. Falls Sie diese Auswahl bestätigen, erledigt die Installationsroutine den Rest.

In den Start-Skripten wird das Programm `dhclient` gestartet, das sich per Broadcast nach dem DHCP-Server umsieht. Von dort bekommt er die Informationen über die IP-Adresse, die Default-Route und den DNS-Server. Über den Aufruf von `ifconfig` (siehe Seite 463) wird die Schnittstelle konfiguriert. Mit dem Befehl `route` wird die Route gesetzt. Falls ein DNS-Server angegeben wird, wird die passende Datei **`/etc/resolve.conf`** erzeugt.

Die Suche nach einem DHCP-Server verzögert das Booten vor allem dann etwas, wenn kein DHCP-Server im Netzwerk existiert. Daher kann das Setzen einer festen IP-Adresse den Bootvorgang beschleunigen.

Da ein neuer Rechner über das Netz nichts weiß, kann er auch keine spezifischen Anfragen produzieren. Eine DHCP-Anfrage (DHCPDISCOVER) kann daher nur als Broadcast gesendet werden. Dieses Broadcast kann nicht einmal über die Broadcastadresse des IP-Netzes erfolgen, da sie ja ebenfalls unbekannt ist. Also wird die IP-Adresse 255.255.255.255 im IP-Header als Zieladresse angegeben, die Quelladresse wird auf 0.0.0.0 gesetzt. Die IP-Adresse 255.255.255.255 wird in einem Ethernet auf die Broadcastadresse ff:ff:ff:ff:ff:ff umgesetzt. Das gleiche gilt für das die Antwort (DHCPOFFER) des Servers, das DHCPREQUEST des Clients und die endgültige Bestätigung (DHCPACK) des Servers. Alle werden über Broadcast übertragen, was ein Netzwerk sehr stark belasten kann.

¹⁷ vgl. Nemeth/Snyder/Seebass/Hein: UNIX Systemverwaltung. Markt+Technik – Prentice Hall, München, 2001, S. 372–374, und Augouros, Konstantin: Zahlen Meister! Linux Magazin 02/02, S. 54–59.

18.6.2 DHCP-Server

`/etc/dhcpd.conf` Die DHCP-Implementierung des ISC (Internet Software Consortium) gilt als Standard. Zur Konfiguration verwendet der DHCP-Server des ISC die Konfigurationsdatei `/etc/dhcpd.conf`. Der Server `dhcpd` liest die Datei beim Starten und bricht den Start ab, wenn er Fehler entdeckt.

Das folgende Beispiel zeigt den Inhalt einer `/etc/dhcpd.conf`-Datei. Der DHCP-Server verwaltet die Domäne `willemer.edu`.

```
option domain-name "willemer.edu";
option domain-name-servers gaston.willemer.edu;

default-lease-time 600;
max-lease-time 7200;

authoritative;

log-facility local7;

ddns-update-style ad-hoc;

subnet 192.168.109.0 netmask 255.255.255.0 {
    range 192.168.109.10 192.168.109.20;
    option routers gaston.willemer.edu;
}
```

Hinter `option domain-name` wird die eigene Domäne festgelegt. Welches die DNS-Server (siehe Seite 495) der Domäne sind, steht hinter `option domain-name-servers`. Der Namensserver ist wichtig, da der neue Rechner eine passende Bindung zwischen Namen und IP erhalten muss.

Die Ressourcen, die der DHCP-Server vergibt, sind nur geliehen. Darum spricht man hier von Leasing. Der Parameter `default-lease-time` gibt die Zeit in Sekunden an, die die Ressourcen vergeben werden. Ist der vorgegebene Zeitrahmen überschritten, muss sich der Client um eine Verlängerung bemühen. Dadurch ist gewährleistet, dass nicht Ressourcen an Rechner vergeben sind, die sich längst nicht mehr im Netz befinden.

Die gesetzte Option `authoritative` besagt, dass dieser Server der offizielle DHCP-Server des Netzes ist.

`ddns-update-style` beschreibt die Form, in der die Abstimmung mit dem DNS erfolgt, und muss zwingend in der Datei `dhcpd.conf` definiert werden.

Die Option `log-facility` betrifft den `syslog`-Dämon (siehe Seite 420). Damit wird festgelegt, welche Facility, also welche Quelle, die Fehler des DHCP-Servers aus Sicht des `syslog` haben. `local7` ist eine der frei verwendbaren Quellenangaben.

Die `subnet`-Definition beschreibt, dass für das Subnetz 192.168.109.0 mit der passenden Netzwerkmaske der Bereich der IP-Adresse zwischen 10 und 20 verteilt werden kann. Der Router dieses Subnetzes ist `gaston`.

Zentral: `subnet`

Nachdem die Datei **`dhcpd.conf`** fertig gestellt wurde, sollten Sie `dhcpd` zumindest einmal von Hand starten, um zu sehen, ob und welche Fehlermeldungen es gibt. Bei folgendem Aufruf ist alles in Ordnung:

```
gaston # dhcpd
Internet Software Consortium DHCP Server V3.0rc12
Copyright 1995-2001 Internet Software Consortium.
All rights reserved.
For info, please visit http://www.isc.org/products/DHCP
Wrote 0 leases to leases file.
Listening on LPF/eth0/00:00:e8:59:88:0f/192.168.109.0/24
Sending on   LPF/eth0/00:00:e8:59:88:0f/192.168.109.0/24
Sending on   Socket/fallback/fallback-net
gaston #
```

Der Client braucht keine Information darüber, welcher Rechner im Netz der DHCP-Server ist. Sobald er bootet, wird er per Broadcast im lokalen Netz nach dem passenden Server fragen.

Der DHCP-Server ist auch in der Lage, einem `bootp`-Client seine IP-Nummer anhand seiner MAC-Adresse zuzuteilen. Dazu wird in der Datei **`dhcpd.conf`** folgender Eintrag gemacht.

```
group {
    host libo {
        hardware ethernet 00:0C:F1:3C:73:70;
        fixed-address 192.168.109.117;
    }
    host libowlan {
        hardware ethernet 00:0B:5D:47:A1:37;
        fixed-address 192.168.110.117;
    }
    ...
}
```

Hier wurde ein Eintrag für `libo` und einer für `libowlan` vorgenommen. Weitere Rechner könnten folgen. Das deuten die drei Punkte an. Die

beiden Rechner sind dieselben Geräte, allerdings einmal über das Kabel und das andere Mal per WLAN angeschlossen.

Diese MAC-Adressen sind auf verschiedene Weise zu bestimmen. Die einfachste Methode ist, Sie fragen den Rechner mit `ifconfig` selbst.

```
libo # ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0C:F1:3C:73:70
          inet addr:192.168.109.112  Bcast:192.168.109.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:f1ff:fe3c:7370/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:78185 errors:2 dropped:2 overruns:0 frame:0
          TX packets:116 errors:0 dropped:3 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:6778918 (6.4 Mb)  TX bytes:8396 (8.1 Kb)
          Interrupt:11 Base address:0xe000 Memory:d0201000-d0201fff

eth1      Link encap:Ethernet  HWaddr 00:0B:5D:47:A1:37
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
          Interrupt:11 Base address:0xc000

...
```

Sie sehen hinter dem Kürzel `HWaddr` die MAC-Adresse. Eine andere, allerdings etwas indirektere Quelle ist die Datei `/var/lib/dhcp/dhcp.leases`. Dort finden Sie alle MAC-Adressen der Rechner, die aktuell eine Nummer zugeteilt bekommen haben.

18.7 Next Generation IPv6

Als die IP-Adressen definiert wurden, hielt man die Anzahl der Adressen für extrem großzügig. Immerhin waren sie so dimensioniert, dass bei den damaligen Bevölkerungszahlen für jeden Menschen eine IP-Adresse verfügbar war: Erst wenn jeder Mensch einen eigenen Computer besäße, der ans Internet angeschlossen wäre, würden alle Adressen aufgebraucht sein. Diese Berechnung stellte sich bald als zu einfach heraus, da die Adressen netzweise vergeben wurden und dabei immer einige ungenutzte Nummern in Reserve gehalten wurden.

Als die Adressen mit der überraschend zunehmenden Verbreitung von Computern und Internetzugängen immer knapper wurden, begann man über eine Überarbeitung der IP-Adressierung nachzudenken. So liest man heute überall, dass die nächste Generation von IP-Adressen vor der Tür stehe. Nur, wie weit sie von der Tür entfernt ist, scheint noch nicht ganz klar zu sein. In Asien hat sich die Adressierung durchgesetzt und viele Provider bieten die Adressierung bereits an.

Eine Adresse unter IPv6 ist statt 4 Byte (wie bei dem jetzigen IPv4¹⁸) 16 Byte oder 128 Bit lang. Es gäbe damit etwa $3,4 \cdot 10^{38}$ Adressen. Damit ergibt sich eine 24-stellige Zahl von Adressen pro Quadratmeter Erdoberfläche.¹⁹

IPv6-Adressen
sind 16 statt
4 Byte lang

Allerdings soll der Nummernumfang auch genutzt werden, um einige Dinge zu vereinfachen. Es wird immer noch eine Unterteilung in Netz- und Hostanteil der Adresse geben. Allerdings wird sie nicht wie bisher von einer Netzwerkkategorie abhängig gemacht, sondern die Netzkennung wird in Zukunft konstant 45 Bit und die Hostkennung wird immer 64 Bit lang werden.

T	Netzkennung	Sub	Hostkennung
3	45 Bit	16 Bit	64 Bit

Abbildung 18.10 Aufbau einer IPv6-Adresse

Das Teilnetz wird nun in der IP-Adresse codiert (Sub) und muss nicht mehr in den Routingtabellen verwaltet werden. Auch das führt zur Vereinfachung. Hinzu kommen zu Anfang 3 Bits zur Bezeichnung des Adresstyps (T). Die 64 Bit für die einzelnen Rechner sind so groß, dass man die MAC, also die Hardwareadresse der verwendeten Adapter, darin ablegen kann.²⁰

Die großen Anbieter haben die Umstellung als Prestigeobjekt betrachtet und sie bereits recht weit vorangebracht. Neue Geräte sind im Normalfall in der Lage, sofort mit IPv6 zu arbeiten. Auch die Politik, allen voran die EU, hält IPv6 für unterstützenswert. So hat die EU bereits Forschungsnetze auf IPv6 umgestellt.²¹ Auch hier entsteht leicht der Eindruck, dass sich die Politiker gern als Förderer der neuesten Technologie darstellen möchten, das Engagement also mehr mit Prestige als mit dem Verständnis für die Notwendigkeit dieser Technologie zu tun hat.

Ganz ohne Probleme wird eine weltweite Umstellung allerdings nicht ablaufen. Problematischer als die großen Systeme sind die einzelnen Programme und kleinere sowie ältere Systeme. Die Umstellung wirkt sich auf sämtliche Geräte wie Computer, Router und Druckserver bis auf je-

Probleme beim
Umstieg

¹⁸ Ein IPv5 gibt es nicht.

¹⁹ Nemeth/Snyder/Seebass/Hein: UNIX-Systemverwaltung. Markt+Technik – Prentice Hall, München, 2001. S. 357.

²⁰ vgl. Nemeth/Snyder/Seebass/Hein: UNIX-Systemverwaltung. Markt+Technik – Prentice Hall, München, 2001. S. 363f.

²¹ vgl. Heise-Ticker vom 16.1.2004.

des einzelne Netzprogramm wie ftp oder einen Webserver aus. Da die Bitzahl der IP-Adressen geändert wird, wirkt sich die Umstellung auf alle Datenstrukturen aus, die IP-Adressen halten – bis hin zu Datenbanken, bei denen Tabellenstrukturen geändert werden müssen. Da viele Systeme nicht so schnell oder gar nicht umgestellt werden können, gibt es einen Kompatibilitätsmodus, der vermutlich nach einer Umstellung noch recht lange aktiv sein muss.

Notwendigkeit
fragwürdig

Aus diesem Grund ist es nicht ungewöhnlich, dass der Markt zurückhaltend auf IPv6 reagiert. Inzwischen haben Techniken wie das Masquerading (siehe Seite 570) bewirkt, dass die heraufbeschworenen Katastrophenszenarien so schnell nicht eintreten werden. Es gibt sogar kompetente Aussagen, die besagen, dass eine Umstellung gar nicht erforderlich sei.²² Andere Autoren, wie Simson Garfinkel,²³ erwarten, dass Asien und vermutlich auch Europa umsteigen werden, da der noch verfügbare Adressraum für Asien sehr knapp ist. Garfinkel führt aus, dass von den theoretisch möglichen vier Milliarden IPv4-Adressen etwa drei Milliarden auf amerikanische Internetprovider verteilt sind. Für China und Südkorea stünden dagegen nur 38,5 beziehungsweise 23,6 Millionen Adressen zur Verfügung. Garfinkel vermutet, dass die USA, wie schon beim Versuch des Umstiegs auf das metrische System in den Siebzigern, beim Althergebrachten bleiben könnten. Neben den bereits genannten Argumenten verweist er auf Geschwindigkeitseinbußen, die seiner Ansicht nach darauf beruhen, dass viele Netzwerkkomponenten, die heute bereits in der Hardware implementiert sind, dann durch Software emuliert werden müssten. Garfinkel verweist auf Sicherheitsprobleme, da die neue IPv6-Software nicht so ausgetestet auf den Markt kommt, wie es die IPv4-Software durch ihren jahrzehntelangen Einsatz ist. Darüber hinaus verweist er darauf, dass das IP-Masquerading beziehungsweise NAT nicht nur das Problem behebt, dass es zu wenig Internetadressen gibt, sondern auch dazu führt, dass die hinter der Firewall liegenden Rechner von außen nicht mehr angreifbar sind. Er prophezeit ein Anwachsen der Urheberrechtsverletzungen. Da unter IPv6 jeder Rechner seine eigene IP-Adresse hat, würde dann der Aufbau von Tauschnetzen, wie sie erstmals von Napster eingeführt wurden, wesentlich effizienter laufen. Diese Aussage muss man wohl als Polemik betrachten. Im Gegensatz zu seiner Sorge um den Durchsatz von IPv6 führt er hier die bessere Funktionalität von IPv6 als Argumentationsgrundlage an. Mit einer etwas anders gelagerten Argumentation könnte man sogar IPv6 als »Napster-Killer« bezeichnen. Da

22 vgl. Nemeth/Snyder/Seebass/Hein: UNIX-Systemverwaltung. Markt+Technik – Prentice Hall, München, 2001. S. 357f.

23 vgl. Heise-Online-Meldung vom 14.1.2004 »Licht und Schatten bei IPv6«.

nun jeder beteiligte Rechner mit seiner Original-IP-Adresse angeschlossen ist, wird es für die Musik- und Filmindustrie sehr viel einfacher, die beteiligten Rechner zu identifizieren. Auch wenn man die Argumentation von Garfinkel also mit Vorsicht genießen muss, wird deutlich, dass es diverse Bedenken gegen die Einführung von IPv6 insbesondere in den USA gibt.

Zusammenfassend gibt es inzwischen im europäischen und amerikanischen Raum keinen zwingenden Grund für einen Umstieg auf IPv6. Sollte es gelingen, die bisherigen Ressourcen mit den Asiaten zu teilen, könnte das sogar dazu führen, dass es bei IPv4 bleibt. Andernfalls kann man davon ausgehen, dass IPv6 in Asien irgendwann zum Standard wird. Inwieweit die USA diesen Umstieg mitmachen, ist offen. Allerdings sind die USA verpflichtet ab 2008 auf den großen Backbones mit IPv6 zu arbeiten. Der Wechsel zu IPv6 wird sicher nicht problemlos ablaufen. In vieler Hinsicht ähnelt das Szenario dem des Jahr-2000-Problems. Man kann nicht sicher sagen, welche Systeme wirklich davon betroffen sind und wie viele Ausfälle durch den Wechsel entstehen werden. Auf der anderen Seite ist auch beim Jahr 2000 der Untergang der zivilisierten Welt vorhergesagt worden und doch nicht eingetreten.²⁴ Der Hauptunterschied zwischen beiden Situationen ist aber, dass das Jahr 2000 irgendwann einmal kommen musste, für alle Systeme zeitgleich, und das ausgerechnet an Silvester.

²⁴ Zumindest sind die diesbezüglichen Rückschritte der Zivilisation nicht den Computern anzulasten.

Index

.Xauthority 761
.amandahosts 349
.htaccess 690
.htpasswd 692
.netrc 542
.rhosts 549
/bin 37
/dev 89, 277, 278, 280
/dev/null 117, 280
/dev/pilot 403
/etc 37
/etc/aliases 646
/etc/auto_master 588
/etc/default/tar 342
/etc/exports 369, 583, 586
/etc/fstab 41, 286, 294–297, 300, 305, 586
/etc/group 268
/etc/host.conf 495
/etc/hosts 491
/etc/hosts.allow 535
/etc/hosts.deny 535
/etc/hosts.equiv 550
/etc/inetd.conf 533
/etc/inittab 396
/etc/magic 90
/etc/netgroup 494
/etc/nsswitch.conf 495
/etc/passwd 257, 259
/etc/printcap 378, 380, 390
/etc/profile 175
/etc/raidtab 289
/etc/rc 249
/etc/resolv.conf 496
/etc/securetty 548
/etc/services 492
/etc/shadow 262
/etc/skel 267
/etc/syslog.conf 420
/etc/ttys 396
/home 38, 264, 588
/lib 37
/media 41
/mnt 41, 296

/opt 38
/proc 39, 443
/tmp 37, 410
/usr 37
/var 38
/var/adm/wtmp 270
/var/log/messages 274, 320, 420
/var/run/utmp 270
/var/yp 507

A

a2ps 181
Absoluter Pfad 35
accept() 966
Access Point 482
access() 916
Accounting 269
accton 269
ACPI 327
Ad Hoc 482
adb 875
Administrationsaufgaben 272
Administrationstools 217
 AIX 233
 HP-UX 232
 SCO 238
 Webmin 221
 YaST 236
Administrator 47, 213
Advanced Package Tool 360
Advanced Power Management 328
Advisory Locking 923
afpd 610
AIX 987, 995
 Administrationstools 233
Akku 323, 329
Akkubetrieb 328
alarm() 953
alias 69, 175, 217
aliases 646
Alternative source 790
AMANDA 347
amanda.conf 348
Anonymer FTP-Server 543
ANSI 995

Apache 681
 als Proxy 573
 API 905, 995
 Apple 991
 AppleTalk 610
 apsfiler 383
 apt-get (Debian) 360
 Arbeitsverzeichnis 35
 Archivierung 353
 argc 906
 Argument 995
 argv 906
 ARP 458, 995
 arp 458
 ASCII 995
 Assembler 995
 at 181, 185, 330
 AT&T 985, 986
 atalkd 610
 Athena Widget Set 725
 atime 42
 atof() 374
 atq 185
 atrm 185
 Audio-CD 206
 Aufrufparameter 906
 Auslastung 413, 414
 Ausloggen 547
 AutoFS 588
 Automount 264, 310
 automount 587
 Automount einer CD 589
 automountd 588
 awk 134

B

backquotes 119
 Bandgerät
 Rückspulen 336
 Bandlaufwerk 334
 HP-UX 334
 SCO 334
 Solaris 334
 Bandlaufwerk steuern 334
 basename 173, 785
 bash 168
 Befehlsverschachtelung 119
 Bell Laboratories 986
 Benutzer

Überwachung 269
 Festplattenplatz 305
 Profil 265
 Verwaltung 257
 Wechsel 271
 Benutzer-Modus 43
 Benutzerverzeichnis 35, 38, 74
 dynamisch 588
 Berkeley 986
 Betriebssystem installieren 361
 bg 154
 BIND 495
 bind() 966
 biod 582
 bitmap 738
 block device 279, 280
 Blockgröße 293, 409
 Bluetooth 405, 487
 Bonobo 754
 Boot 241, 365, 995
 Boot-CD 245
 Bootdiskette 245
 Bootkonfiguration 365
 Bootmanager
 FreeBSD 367
 GRUB 244
 LILLO 243
 bootp 514
 Bootprobleme 245
 Bootprompt 241, 242
 Bootsystem 215
 Bourne-Shell 161
 break 783
 Bridge 482
 BSD 986, 989, 995
 Bootdateien 253
 Drucken 378

C

C 986, 995
 C++ 995
 C-Shell 165
 Cache 995
 Callback 476
 cancel 179, 385, 388
 cardmgr 325
 Carriage Return 311
 case
 Shellskript 781

- cat 120
- cc 865
- CD
 - brennen* 203, 315
 - Brenner* 206, 312, 315, 316
 - IDE* 317
 - CD-ROM* 311
 - CD-RW* 316
 - kopieren* 316
 - Multisession* 316, 351
 - Rockridge* 311
- cd 62, 73
- CDE 743, 996
 - Panel* 744
- cdparanoia 202
- cdrecord 312, 313, 315, 316, 350
- CGI 697
 - Perl* 813
- character device 279, 280
- chdir() 927
- checkpc 390
- chgrp 77, 78
- chgrp() 917
- chmod 78, 81, 772
- chmod() 917
- chown 77
- chown() 917
- chroot 543
- CIDR 481
- Classless Inter-Domain Routing 481
- Client 996
- Client-Server-Architektur 456
- Client-Server-Programmierung 963
- clock() 959
- close() 909, 911, 965
- closedir() 926
- cnews 669
- Codec 207
- Comer, Douglas 988
- Common UNIX Printing System 390
- Compiler 865
 - Optionen* 866
- Compilerbau 902
- compress 144
- configure 357
- connect() 967
- continue 783
- Controller 277
- Cookies 694
- CORBA 754, 990

- core dump 444, 874
- cp 70, 296
- cpio 343
- CPU 411, 996
- CPU-Last 416, 438
- creat() 911
- cron 181, 183
- crontab 181, 183, 330
- Cross-Compiler 996
- crypt() 257, 976
- csh 165
- ctime 42
- ctlinnd 672
- CUPS 390
 - Client* 395
 - Verwaltung per Browser* 392
- Curses 980, 996
- Cursor 996
- cut 124
- Cut and Paste 730
- CVS 887
 - Client* 891
 - CVSROOT* 887
 - Server* 890
 - Windows* 892

D

- Dämon 930, 963, 996
 - Start und Stopp* 252
- DAS Direct Attached Storage 611
- date 182
- Dateien
 - Anfang anzeigen* 123
 - anlegen* 84
 - auflisten (ls)* 64
 - Ausschnitt* 124
 - Eigenschaften* 76
 - Eigentümer ändern* 77
 - Ende anzeigen* 123
 - im Verzeichnisbaum suchen* 107
 - Inhalt anzeigen* 120
 - Inhalte durchsuchen* 121
 - komprimieren* 144
 - kopieren* 70
 - löschen* 72, 217
 - MS-DOS* 311
 - Rechte ändern* 78
 - sortieren* 129
 - sperrern* 917

- Status ermitteln* 914
- suchen* 113
- Teilen* 124
- temporär* 925
- Transfer über TCP/IP* 536
- transferieren* 551
- Typ ermitteln* 89
- umbenennen* 71
- Zeichen umcodieren* 125
- Zeilen umbrechen* 125
- Dateinamen 33
- Dateirechte 213
- Dateisystem 285
 - abkoppeln* 300
 - Belegung* 303
 - erstellen* 292
 - FAT* 322
 - Journal* 302
 - Konsistenz* 301, 302
 - Reiser* 303
 - XFS* 303
- Dateizugriffe 909
- Datenabgriff aus Pipe 118
- Datenbank 615
- Datenrücksicherung 336
- Datensicherung 331
 - über Netz per tar* 343
 - AMANDA* 347
 - CD brennen* 313
 - dump* 335
 - inkrementell* 332
 - per CD-Brenner* 350
 - tar* 339
- Datenstrom 34
- Datum 182
- dbx 874
- dd 316, 346
- Dead Keys 375
- Debian 360, 363
- Debugger 873
 - adb* 875
 - Breakpoint* 873
 - dbx* 874
 - gdb* 876
- defer_transports 660
- Defragmentierung 408
- Desktop 743
- Device 996
- Devices 277
- df 303
- DHCP 514, 996
- diff 127
- dig 503
- DIN 996
- disable 388
- Diskette 281, 309
 - formatieren* 309
 - MS-DOS* 310
 - sichern* 346
- DISPLAY 219, 757
- Distribution 996
- dmalloc 902
- DNS 495
 - Cache Only Server* 495
 - Mail-Server* 654
 - Primary Server* 495
 - Secondary Server* 495
 - testen* 503
- domainname 508
- dpkg 361
- Druck formatieren 180
- Druckdämon 380
- Drucken
 - Aus dem Programm* 955
 - BSD* 177
 - cancel* 179
 - lp* 179
 - lpq* 178
 - lpr* 177
 - lprm* 179
 - lpstat* 179
 - SAMBA* 597
- Drucker
 - Administration* 377
 - GDI* 378
 - konfigurieren* 378
- Druckerinterrupt 384
- Druckreihenfolge 382
- Druckserver 382
- Druckspooler 379
- du 303
- dump 335
- dumpkeys 375
- dup0 913
- DVD 319
- Dynamische Bibliotheken 451
- Dynamisches Routen 480

E

E-Mail 631
 Format (RFC 822) 640
 Verschlüsseln 635
 echo 772
 Eclipse 882
 EDGE 562
 EDITOR 775
 Editoren 92
 emacs 102
 vi 92
 Effektiver Benutzer 434
 EGP 481
 Einbruch 575
 eject 309
 Electric Fence 901
 emacs 102
 enable 388
 endgrent() 961
 endpwent() 960
 Enigma 209
 Environment-Variable 996
 errno 908
 Ethernet 456, 996
 EtherTalk 610
 Exception 43
 exec() 931
 exit() 907
 export 157, 163, 791
 expr 775
 ext3 303
 extract 336

F

FAQ 996
 Farbbeschreibung 734
 Fat Client 997
 FAT-Dateisystem 322
 fc 164
 fcntl() 918
 fdisk 243
 Fedora 363
 Fehlerprotokoll 956
 Fehlertolerante Festplatten 287
 Fenstermanager 720, 741
 Festplatte 282
 abschalten 329
 Belegung 303

Geometrie 282
 IDE 283
 S-ATA 284
 SCSI 282
 fetchmail 653, 657
 fg 154
 FIFO 955
 file 89
 find 107, 343
 finger 255, 270
 Firewall 524, 564
 flock 922
 Fokus 742
 fold 125
 for 172, 784
 Perl 808
 Python 843
 fork() 930, 972
 Formatierung 285, 365
 forward 646
 Fotokopierer 402
 Free Software Foundation 988
 FreeBSD
 Bootmanager 367
 Installation 366
 Frontend 997
 fsck 301
 fstab 296, 300
 fstat() 914
 FTP 997
 Client 537
 Kommandos 538
 ftp 536
 .netrc 542
 anonymous 543
 fuser 300, 442

G

gated 481
 Gateway 469, 997
 gdb 876
 GDI-Drucker 378
 Gerätedateien 278, 286
 Gerätenamen 281
 Gerätetreiber 280
 getcwd() 927
 getenv() 908
 geteuid() 928
 getgrent() 961

getgroups() 962
 gethostbyname() 968, 975
 getnodebyname() 975
 getpgrp() 934
 getpid() 928
 getppid() 928
 getpwent() 960
 getservbyname() 968
 getty 397
 getuid() 928, 959
 getwd() 927
 GhostScript 377, 383
 GIF 196
 Gnokii 405
 GNOME 752, 990, 997
 GNU 988, 997
 GNU-Compiler 865
 GNU-Debugger 876
 GPA 639
 gpasswd 268
 gpg 637
 GPL 997
 Grafischer Login 248
 grep 121
 group (Datenstruktur) 961
 growsisofs 319
 GRUB 244, 365
 Gruppe wechseln 269
 Gruppenpasswort 268
 Gruppenverwaltung 268
 GUI 997
 gunzip 145
 gzip 145

H

Halt 248
 halt 255
 Handle 997
 Hardcopy unter X 734
 Hardware 277
 Hardware-RAID 288
 hdparm 284, 329
 head 123
 Headache 199
 help 778
 Herunterfahren 254
 Hewlett Packard 987
 Hilfe 53
 Hintergrundbild 738

Hintergrundprozess 147
 HOME 774
 Host 998
 hostname 490
 hosts 491
 Hotplug 285, 288, 998
 HP-UX 987
 Administrationstools 232
 Bandlaufwerk 334
 Bootdateien 253
 Festplattengröße 308
 Installation 372
 SD-UX 358
 Software Agent 356
 VXFS 302
 Zugriff auf BSD-Drucker 389
 HTML 681, 683, 998
 Formular 684
 hton() 969
 htpasswd 692
 HTTP 693
 httpd 681
 httpd.conf 687
 Hub 457, 998

I

i-node 42, 998
 i4l 476
 IBM 987
 icecream 903
 ICMP 494, 523, 998
 Icon 998
 IDE 881
 IDE-Brenner 317
 IDE-Festplatten 283
 IDS 575
 IEEE 987, 998
 if
 Perl 806
 Python 841
 Shellskript 777
 ifconfig 463
 iftop 530
 IMAP 651
 inetd 533, 603
 info 56, 102
 infocmp 400
 init 246, 247, 396
 init 0 254

init.d 250
 inittab 246, 247, 396
 Inkrementelle Datensicherung 332
 inn 669
 Installation 355
 HP-UX 372
 Solaris/86 370
 Internet 986, 998
 Internet Printing Protocol (IPP) 390
 Internetanschluss 559
 Internetfilter 574
 Interrupt 278, 444, 998
 Intranet 998
 ioctl 278
 IP 998
 IP-Adressen 458
 private 462
 ipchains 566
 ipcrm 939
 ipcs 939
 ipfw 568
 iptables 566, 567
 IPv6 518
 Programmierung 975
 IPX 608
 isdnctrl 476
 ISO 998
 ISO 9660 311, 313
 ISO-8859-1 374

J

Java 878, 999
 Archiv (jar) 881
 Compiler (javac) 879
 Debugger (jdb) 880
 Java Server Pages 702
 JavaScript 705
 Job-Nummer 147
 jobs 154
 Joliet 313
 Journal-Dateisystem 302, 999
 JPEG 196

K

K3B 203
 KDE 747, 990, 999
 Kontrollzentrum 749
 Panel 748

KDevelop 884
 Kernel 43, 447
 kernel panic 445
 Kernel-Modus 43
 Kernelparameter 43, 307
 kill 153, 154, 439
 top 439
 kill() 953
 Knoppix 216
 Knuth, Donald 190
 Kommandointerpreter 60, 161
 Komprimieren 144
 Konfigurationsdateien 253
 Konqueror 607, 749
 Konsistenz 301, 999
 Dateisysteme 302
 Konsole 752, 999
 Kontextmenü 999
 Kontrollterminal 935
 Kontrollzentrum
 KDE 749
 kooka 402
 Korn-Shell 162
 ksh 162
 Kubuntu 363

L

Löschen 72
 lame 199
 LAN (Local Area Network) 999
 LANG 158, 374
 LaTeX 190
 LC_TYPE 374
 LDAP 509
 ldap.conf 514
 LDIF 513
 LeakTracer 902
 less 121
 let 163
 lex 902
 Lexikalische Analyse 902
 LILO: Linux Loader 243
 Line Feed 311
 Link 84
 hart 84
 Symbolisch 40
 symbolisch 87, 250, 253, 296, 314
 link() 924
 LinNeighborhood 607

Linux 356, 382, 989
 SAMBA 591
 XFS 303
 listen() 966
 Lizenz 987
 ln 84
 ln -s 87
 loadkeys 375
 locate 113
 lockd 582
 lockf 921
 locking 923
 Login 51, 271
 grafisch 248
 LOGNAME 775
 logout 547
 logrotate 432
 Lokale Mail lesen 631
 Look and Feel 999
 loopback 457, 463
 lost+found 301
 lp 179, 385
 lpadmin 385
 lpc 382
 lpd 378, 380
 Netzwerkprotokoll 381
 lpoptions 391
 lpq 178, 378, 381
 lpr 177, 378, 380
 lprm 179, 378, 381, 382
 LPRng 389
 lpshut 385
 lpstat 179, 385, 388
 ls 64
 lseek() 912
 lsmod 320
 lsof 255, 442
 lsusb 321
 ltrace 900

M

Mülleimer 117, 280
 MacOS X 991
 Desktop 755
 Macintosh 610, 991
 Mail
 Weiterleiten 646
 mail 631
 Mailingliste 647

mailq 645
 Mailqueue 645
 main() 905
 major number 279, 280
 make 355–357, 448, 868, 887, 904
 Makro 871
 Suffixregeln 872
 Ziele 872
 Makefile 506
 man 53
 Sektionen 54
 Mandatory Locking 923
 Marke 987
 Mars 608
 Masquerading 570
 Master Boot Record 243, 326, 365
 Maus 324
 MBR 243, 326, 365, 999
 MBR sichern 347
 Medien kopieren 346
 Mehrbenutzersystem 45
 memory leak 902
 Memory Mapped I/O 980
 Message Queues 943
 messages 429
 Metazeichen 54
 metric 470
 MIME 999
 MINIX 988
 Dateisystem 293
 minor number 279, 280
 Mirroring 287
 MIT 999
 mkdev (SCO) 592
 mkdir 62, 75
 mkdir() 928
 mkfifo() 955
 mkfs 292
 mkisofs 313, 315, 316, 350
 mknod 281
 mkswap 294
 Mobiltelefon 405
 Modem 401
 Module 449
 more 118, 120
 Motif 715, 719, 747, 999
 mount 41, 286, 292, 295, 309, 584
 mountd 582
 MP3 198, 199
 MP3-CDs brennen 206

MPEG 207
 mpg123 199
 mqueue 645
 MS-DOS-Disketten 310
 msgctl() 945
 msgget() 944
 msgrcv() 945
 msgsnd() 944
 mt 334
 mtime 42
 MTU 526
 MULTICS 985
 Multiprocessing 929
 Multisession 316
 Multitasking 147, 435, 1000
 Multithreading 929
 Multiusermodus 248
 mv 71
 MySQL 620
 Administration 622
 Benutzerverwaltung 621
 Datensicherung 623
 Installation 620
 Perl 831
 Python 855

N

named 495, 497
 Named Pipe 955
 Namensauflösung 489
 NAS 611
 Nationale Besonderheiten 374
 Nautilus 753
 ncfs 608
 NDAS 611
 ndiswrapper 487
 NDMP 611
 netatalk 610
 NetBeans 883
 netdate 612
 netgroup 494
 Netiquette 665
 netstat 255, 525–527
 Network File System 581
 Network Information System 505
 Netzadapter
 anzeigen 526
 konfigurieren 463
 Netzgruppe 494

Netzwerk beobachten 528
 Netzwerkdrucker 379
 Netzwerkkarten 327
 Netzwerkklassse 459
 Netzwerkmaske 477, 481
 Neutrino 209
 newfs 292
 newgrp 269
 Newsgroups 663
 Administration 672
 Newsreader
 Pan 666
 NextStep 991
 NFS 369, 581, 987, 1000
 nfsd 582
 nice 150
 NIS 505, 589, 1000
 NIS-Client starten 508
 NIS-Server starten 507
 nmbd 598
 NNTP 677
 nohup 547
 Notebook 323
 notlame 199
 Notsysteme 215
 Novell 608, 992
 nslookup 503
 NTFS 299
 ntoh() 969
 NTP 613
 ntpdate 613
 NVRAM 242

O

obsolete 989
 OCR 402
 od 128
 ODBC
 Perl 831
 Offene Dateien 441
 oktal 1000
 OLDPWD 774
 open() 909
 opendir() 926
 OpenLDAP 509
 OpenOffice.org 188
 OSF 987, 1000
 OSI-Referenzmodell 1000

P

- Paging 294
- Paketfilter 569
- Palm 403
- Pan 666
- Panel
 - CDE* 744
 - KDE* 748
- panic 445
- Parallele Schnittstelle 377
- Parameterverarbeitung 906
- Partition 285, 290
 - Belegung* 303
 - Bezeichnung* 281
 - Größe* 290
 - sichern* 346
 - Swap* 291
 - Windows* 297
- passwd 257, 259
- passwd (Datenstruktur) 959
- Passwort 257, 259
 - unsicher* 258
 - verschlüsselt für SAMBA* 600
- Patente 1000
- PATH 159, 216, 267, 774
- pause() 953
- pccardd 324
- PCMCIA 323, 324
- PDA 403
- PDF 1001
- PDP-11 986
- Performance 1001
- Perl 793
 - ARGV* 804
 - Array* 799
 - Aufrufparameter* 804
 - Ausgabe* 803
 - Bedingungen* 805
 - CGI* 813
 - chomp* 803
 - Dateizugriffe* 815
 - Datenbank* 831
 - Eingabe* 803
 - for* 808
 - foreach* 810
 - Funktionen* 814
 - Hash* 801
 - if* 806
 - keys* 810
 - lokale Variablen* 815
 - my* 797, 815
 - MySQL* 831
 - ODBC* 831
 - PostgreSQL* 831
 - print* 803
 - Reguläre Ausdrücke* 802
 - Skalar* 794
 - sort* 810
 - split()* 800
 - strict* 797
 - suchen* 806
 - Tk* 818
 - UNIX* 818
 - until* 811
 - while* 811
- perror() 908
- Personal Firewall 564
- PGP 1001
- PHP 700
- PID 147, 151, 152
- ping 467, 523
- Pipe 89, 115, 118, 315, 343, 1001
 - Programmierung* 954
- pipe() 954
- pkgadd 357
- PNG 196
- Polling 384, 980, 1001
- POP3 647
- popen() 955
- portmap 582
- POSIX 161, 987, 1001
 - Sperren* 918
- Postfix 658
- PostgreSQL 624
 - Benutzerverwaltung* 626
 - Datenbank anlegen* 627
 - Datensicherung* 627
 - Installation* 625
 - Perl* 831
 - Python* 855
- PostScript 377, 383, 1001
- PPID 151
- PPP 563
- pr 180
- Präfix 1001
- Primary Server (DNS) 499
- printenv 167
- PRINTER 177
- Priorität 150

- profile 265
- Programmabbruch 154
- Programmierung
 - Client-Server* 963
 - Socket* 963
- Prompt 1001
- Protokolldateien
 - paralleles Schreiben* 910
- Proxy 571, 1001
- Prozess 147, 434, 928, 1001
 - anzeigen* 151
 - Erneuerung* 931
 - Im Hintergrund starten* 147
 - Nacheinander starten* 149
 - Priorität* 150
 - terminieren* 439
- Prozessgruppe 933
- Prozessstart 931
- prstat 439
- ps 148, 151, 435
- PS1 159, 775
- PS2 160, 775
- pthread_create 948
- pthread_exit 949
- pthread_join 949
- pty 280
- putenv() 908
- PWD 774
- pwd 73
- Python 833
 - Ausnahmebehandlung* 834
 - Boolesche Variablen* 843
 - Boolesche Verknüpfung* 842
 - Dateienbehandlung* 852
 - Datenbank* 854
 - Dictionary* 849
 - elif* 841
 - else* 841
 - for-Schleife* 843
 - Funktionen* 845
 - Grafische Oberfläche* 858
 - Http-Zugriff* 857
 - if* 841
 - input()* 834
 - Klassen* 850
 - Liste* 847
 - Logische Operatoren* 842
 - MySQL* 855
 - Numerische Operatoren* 835
 - PostgreSQL* 855

- print* 833
- range()* 843
- raw_input()* 834
- read()* 854
- readline()* 853
- Referenz* 851
- Schleifen* 843
- seek()* 854
- Sequenz* 847
- String-Bibliothek* 839
- Tk* 858
- Tupel* 848
- Verzweigung* 841
- while-Schleife* 844
- Zeichenketten* 838

Q

- qmake 884
- quota 305
 - Gnadenfrist* 306
 - quotacheck* 306
 - quotaoff* 306
 - quotaon* 306

R

- RAID 286, 1001
 - Hardware* 288
 - Linux* 289
 - Software* 289
- RAID 0 287
- RAID 1 287
- RAID 10 288
- RAID 5 287
- RAM 411
- RAM-Disk 410, 1001
- raw device 286
- rc-Dateien 249
- rcmd 553
- rcp 549, 551
- RCS 886
- read
 - Shellskript* 789
- read() 911
- readdir() 926
- Realer Benutzer 434
- Reboot 248
- reboot 255
- Rechtereiterteigabe 213

- recode 311
- recv() 967
- Red Hat 356, 990
- Red Hat Package Manager 359
- Register 277
- Regulärer Ausdruck 138, 978, 1002
- Reiser 303
- rekursiv 68, 1002
- Relativer Pfad 35
- Relay 642
- relayhost 660
- Remote Procedure Call (RPC) 980
- rename() 925
- renice 151, 440
 - top* 439
- Ressourcen 411, 718, 738
- restore 335, 336
- rexec 553
- RFC 59, 1002
- RFC 1179: Netzwerkdrucker
 - (lpd) 390
- RFC 1519: CIDR 481
- rhosts 549
- RISC 1002
- Ritchie, Dennis 986
- rlogin 549, 552
- rlp (SCO) 388
- rm 72, 217
- rmdir 76
- rmdir() 928
- RMT
 - Umgebungsvariable* 337
- rmt 337
- Rockridge 311
- root 47, 213, 1002
- ROT13 126
- route 470
- routed 481
- Routing 469
 - dynamisch* 480
 - MS Windows* 474
 - statisch* 470
 - Tabelle* 527
- RPC 582
- rpm 355, 356, 359
- RS-232 1002
- RS/6000 987
- RSA-Authentifizierung 555
- rsh 549, 553
- rshd 337, 343, 552

- Ruhezustand 325
- Runlevel 246

S

- S-ATA 284
- sam 219, 232, 758
- SAMBA 590
 - Client* 606
 - Drucken* 597
 - Freigaben* 596
 - Linux* 591
 - SCO* 592
 - Solaris* 591
 - swat* 604
 - Verbindungsstatus* 606
 - verschlüsseltes Passwort* 600
- SAN Storage
 - Area Network 611
- sar 414
- Satellit 562
- Save to Disk S2D 326
- Scannen 402
- SCCS 885
- Scheduler 435
- Schriften unter X 735
- SCO
 - Administrationstools* 238
 - Bandlaufwerk* 334
 - BSD-Drucksystem installieren* 388
 - SAMBA* 592
 - Santa Cruz Operation Inc.* 992
 - tar* 342
 - ULIMIT* 308
- scp 553
- screen 547
- SCSI 315, 1002
 - ID* 282
 - Terminierung* 283
 - Wide* 283
- SCSI-Festplatten 282
- SD-UX 358
- sed 130
- select() 974
- Semaphore 940
- semctl() 941
- semget() 940
- semop() 940
- send() 967
- sendmail 642, 645

- serielle Schnittstelle 280
- Server 1002
- Servlets 702
- Session 934
- setenv 157, 167
- setgrent() 961
- setpgid() 934
- setpwent() 960
- setsid() 934
- SetUID-Bit 214
- SGI 987
 - XFS 303
- Shadow Password 262
- Shared Memory 935
- Shell 43, 60, 161, 1002
- Shell-Variablen 157
- Shellskript 772
 - Aufrufparameter* 773
 - case* 781
 - Ein- und Ausgabe* 789
 - export* 791
 - expr* 775
 - for* 784
 - Funktionen* 786
 - if* 777
 - let* 776
 - Programmierung* 771
 - Rückgabewert* 780
 - rc-Datei* 251
 - read* 789
 - Rechnen* 775
 - Schleife* 782
 - shift* 783
 - test* 778
 - Umgebungsvariablen* 774
 - Variablen* 772
 - while* 782
- Shift 1002
- shift 783
- shmat() 936
- shmctl() 936
- shmget() 935
- showmount 585
- shutdown 254
 - per login* 275
- Sicherheit 48
- Sicherheitsloch
 - USB-Stick* 323
- Sicherheitsproblem 246
- SIGCONT 441
- SIGHUP 440, 547
- SIGINT 154
- SIGKILL 441
- signal() 951
- Signale 439, 951
 - ignorieren* 954
- Signieren
 - E-Mail* 635
- SIGSTOP 441
- SIGTERM 441
- SIGTSTP 154
- Silicon Graphics Inc. 987
- Single-User-Modus 241, 247, 256
- SINIX
 - tar* 342
- Sitzung 934
- skel 267
- slapd.conf 511
- sleep() 974
- SMB 590
- smb.conf 593
- smbclient 606
- smbd 598
- smbstatus 606
- smit 233
- SMTP 642, 1003
- Snort 575
- Socket 89, 1003
- socket() 965
- Socketprogrammierung 963
- Software-Interrupt 43
- Software-RAID 289
- Softwarepackages 355
- Solaris 990
 - admintool* 229
 - Bandlaufwerk* 334
 - dump und restore* 335
 - Packages* 356
 - SAMBA* 591
 - Solaris Management Console* 230
 - UFS* 302
- Solaris Packages 357
- Solaris/86
 - Installation* 370
- sort 129
- source 790
- Spam 642
- SPARCbook 323
- special files 89
- Speicherleck 901, 902

Speichermedien 39
 Speicherverwaltung 417
 Sperren 917
 Advisory 923
 Mandatory 923
 POSIX 918
 Spiegelung 287
 split 124
 Spool 1003
 sqid 573
 SQL 615, 1003
 Data Definition Language
 (*DDL*) 616
 Data Manipulation Language
 (*DML*) 618
 Daten 618
 Index 617
 SELECT 618
 Tabelle 616
 View 617
 squidguard 574
 ssh 220, 553, 760
 ssh-keygen 555
 SSH-Tunnel 557
 sshd_config 554
 Stallmann, Richard M. 988
 StarOffice 188, 730
 Start eines Programms 932
 startx 723
 stat() 914
 statd 582
 Statusloser Server 678, 694, 973
 stderr 116
 stdin 115
 stdout 115
 strace 900
 Streams 980
 strerror() 908
 Striping 287
 stty 400
 su 271, 669
 su - 271
 Subnetting 477
 Subversion 893
 suck 675
 sudo 272, 275
 sudoers 272
 SUID 81
 Sun 987
 Superuser 1003

SUSE 363
 rc.config 253
 YaST 236
 svn 893
 Swap-Partition 291
 Größe 294
 swapon 294
 Swapping 293, 1003
 Datei 295
 Partition erzeugen 294
 Partition vs. Datei 294
 starten 294
 swat 604
 swinstall 358
 Switch 1003
 swlist (HP-UX) 358
 swremove (HP-UX) 358
 Symbolischer Link 253, 296
 symlink() 924
 sync 310, 442
 Syntaktische Analyse 903
 Syntax 1003
 syslog 274, 420, 429
 syslog() 956
 syslog-ng 423
 syslog.conf 421
 System V 987
 system() 932
 Systemabschluss 254
 Systemaufrufe 900, 905
 Systemstart 241

T

tail 123, 429
 Tanenbaum, Andrew S. 988, 989
 tar 141, 296, 310, 339, 355, 356
 Besonderheiten SINIX und SCO 342
 per Netzwerk 343
 Tastaturbelegung 375
 TCP 493, 1003
 tcp 493
 TCP/IP 455, 986, 1003
 tcpd 534
 tcpdump 528
 tee 118
 telnet 220, 544
 Client 544
 Server 548
 Sitzung 545

TERM 398, 775
 termcap 375, 398
 Terminal 395, 1003
 anpassen 398, 399
 virtuell 544, 552
 Zurücksetzen 400
 Terminalemulation 731
 terminfo 399
 auslesen 400
 compiler 400
 test 778
 testparm 598
 TeX 190
 TFTP 544
 tgz 143
 Thin Client 767
 Thompson, Ken 986
 Thrashing 293
 Thread 929, 947, 1003
 Thunderbird 667
 tic 400
 TIFF 196
 time to live 468
 time() 958
 tmpnam() 925
 Tomcat 701
 top 438
 Torvalds, Linus 989
 touch 83, 305
 Touchpad 324
 tr 125, 311
 traceroute 528
 Transaktion 1004
 Transceiver 456
 Treiber 277, 1004
 Trojanisches Pferd 216
 TTL 500
 ttl 468
 TTY 1004
 Tuning 407
 Twisted Pair 457

U

Ubuntu 363
 UDP 494, 1004
 udp 493
 UFS 302
 ufsdump 335
 ufsrestore 335

Uhrzeit 182
 ULIMIT
 SCO 308
 ulimit 175, 265
 umask 82, 266
 umask() 917
 Umbenennen 71
 Umgebungsvariablen 157, 267, 774,
 907
 Umleitung der Ein- und Ausgabe 116
 umount 286, 295, 300
 UMTS 562
 uname 419
 UNICODE 374
 UNICS 986
 UNIX
 API 905
 UnixWare 992
 unlink() 924
 unzip 145
 updatedb 113
 uptime 433
 URL 1004
 USB 320, 377, 1004
 USB-Stick 322
 usb_storage 321, 322
 USER 775
 User-ID
 effektiv 434
 real 434
 User-ID-Bit 81
 useradd 264
 UTF-8 374
 utmp 270

V

Variable 1004
 Verbindung prüfen 467
 Verschachtelung von Komman-
 dos 119
 Verschlüsseln 976
 E-Mail 635
 Versionsfeststellung 419
 Versionsverwaltung 884
 Verteiltes Kompilieren 903
 Verzeichnis 35, 73
 anlegen 75, 928
 auslesen 926
 erzeugen 75

- löschen* 76, 928
- wechseln* 73
- Verzeichnisbaum 36
- Verzeichnisfunktionen 926
- Verzeichnisprototyp skel 267
- Verzeichnistrenner 35
- vi 92, 986
- virtual device 280
- Virtuelle Geräte 280
- Virtueller Speicher 294
- Virus 48
- visudo 272
- vmstat 413
- VPN 1004
- VXFS 302

W

- wait() 932
- WAN 1004
- Wartbarkeit 48
- Warteschlange 278
- wc 129
- Webmin 221
 - Hardware* 227
 - Netzwerk* 225
 - Server* 224
 - System* 222
- Webserver 681
- Wechselmedien 286
- well known port 492, 493
- WEP 484
- whereis 114
- which 114
- while
 - Python* 844
- who 255, 270
- whodo 270
- Wide-SCSI 283
- Widget 718, 1004
- Widget Set 715, 718
- Wiederbeschreibbare CDs 316
- Wildcard
 - Fragezeichen* 90
 - rechteckige Klammern* 91
 - Stern* 90
- WinCVS 892
- Window Manager 741
- Windows 1005
- Windows Mobile 404

- Windows-Netz 590
- Windows-Partition 297, 364
- WinModem 327
- WINS 595
- Wireshark 529
- WLAN 482, 1005
- Workstation 713, 1005
- WPA 484
- write() 912
- wtmp 270

X

X

- Bitmaps* 738
- Farben* 734
- Hintergrundbild* 738
- Netzwerkverteilung* 756
- Ressourcen* 738
- Schriften* 735
- Standardoptionen* 729
- starten* 721
- X Konsortium 1005
- X Toolbox Intrinsics 718
- X Window System 44, 711
- X-Bibliotheken 717
- X-Client 717
- X-Server 716, 762, 1005
 - freigeben* 757
- X-Terminal 713, 716
- X-Terminalbetrieb unter Linux 766
- X/Open 988, 1005
- Xaccess 764
- xauth 761
- xcalc 726
- xclock 727
- xdm 723, 763
- xdm-kontrollierter X-Server 767
- xedit 727
- xeyes 727
- xfd 736
- XFS 303
- xhost 757
- xinetd 535
- xinit 722, 723
- XINU 988
- xkill 734
- Xlib 717, 718
- xload 726
- xlsfonts 735

xman 725
xpaint 727
xprop 739
xsane 402
Xservers 765
xsetroot 738
xterm 731
xwd 734

Y

yacc 902
YaST 236
Yellow Pages 506, 1005
ypbind 508
ypinit 507

yppasswd 508
ypserv 507

Z

Zeichensatzkonvertierung 311
Zeitabgleich 612
Zeitfunktionen 958
Zeitversetzte Kommandos 185
zgrep 145
zip 145
zless 145
Zombies 954, 973
Zonen 499
Zuteilung Festplattenplatz 305