

Sascha Kersken

IT-Handbuch für Fachinformatiker

Der Ausbildungsbegleiter



Auf einen Blick

1	Einführung	25
2	Mathematische und technische Grundlagen	57
3	Hardware	109
4	Netzwerkgrundlagen	169
5	Betriebssystemgrundlagen	281
6	Windows	319
7	Linux	365
8	Mac OS X	437
9	Grundlagen der Programmierung	455
10	Konzepte der Programmierung	541
11	Software-Engineering	637
12	Datenbanken	675
13	Server für Webanwendungen	727
14	Weitere Internet-Serverdienste	767
15	XML	791
16	Weitere Datei- und Datenformate	837
17	Webseitenerstellung mit (X)HTML und CSS	865
18	Webserveranwendungen	923
19	JavaScript und Ajax	993
20	Computer- und Netzwerksicherheit	1057

Inhalt

Geleitwort des Fachgutachters	15
Vorwort	17

1 Einführung 25

1.1	Informationstechnik, Informatik und EDV	25
1.1.1	Fachrichtungen der Informatik	26
1.1.2	Überblick über die IT-Ausbildung	27
1.2	Die Geschichte der Rechenmaschinen und Computer	33
1.2.1	Die Vorgeschichte	34
1.2.2	Die Entwicklung der elektronischen Rechner	36
1.2.3	Entwicklung der Programmiersprachen	45
1.3	Digitale Speicherung und Verarbeitung von Informationen	52
1.3.1	Digitale Bilddaten	53
1.3.2	Digitale Audiodaten	54
1.3.3	Digitale Speicherung von Text	55
1.4	Zusammenfassung	56

2 Mathematische und technische Grundlagen 57

2.1	Einführung in die Logik	57
2.1.1	Aussagen	58
2.1.2	Aussageformen	59
2.1.3	Logische Verknüpfungen	60
2.1.4	Mengenoperationen	67
2.1.5	Weitere wichtige Berechnungsverfahren	70
2.2	Informationsspeicherung im Computer	72
2.2.1	Zahlensysteme	72
2.2.2	Bits und Bytes	78
2.3	Elektronische Grundlagen	83
2.3.1	Einfache Schaltungen	83
2.3.2	Zusammengesetzte Schaltungen	86
2.4	Automatentheorien und -simulationen	90
2.4.1	Algorithmen	91
2.4.2	Die Turing-Maschine	96
2.4.3	Der virtuelle Prozessor	100
2.5	Zusammenfassung	106

3	Hardware	109
3.1	Grundlagen	109
3.2	Die Zentraleinheit	113
3.2.1	Aufbau und Aufgaben des Prozessors	115
3.2.2	Der Arbeitsspeicher	125
3.2.3	Das BIOS	127
3.2.4	Bus- und Anschlusssysteme	133
3.3	Die Peripherie	142
3.3.1	Massenspeicher	143
3.3.2	Eingabegeräte	156
3.3.3	Ausgabegeräte	160
3.3.4	Soundhardware	166
3.4	Zusammenfassung	167
4	Netzwerkgrundlagen	169
4.1	Grundlagen	169
4.1.1	Was ist ein Netzwerk?	169
4.1.2	Entstehung der Netzwerke	171
4.2	Funktionsebenen von Netzwerken	176
4.2.1	Das OSI-Referenzmodell	177
4.2.2	Das Schichtenmodell der Internetprotokolle	179
4.2.3	Netzwerkcommunication über die Schichten eines Schichtenmodells	182
4.3	Klassifizierung von Netzwerken	186
4.3.1	Die Reichweite des Netzwerks	186
4.3.2	Die Netzwerktopologie	188
4.3.3	Der Zentralisierungsgrad des Netzwerks	189
4.4	Netzwerkkarten, Netzkabel und Netzzugangsverfahren	196
4.4.1	Die verschiedenen Ethernet-Standards	197
4.4.2	Token Ring	202
4.4.3	Drahtlose Netze	204
4.4.4	Sonstige Zugangsarten	208
4.5	Datenfernübertragung	210
4.5.1	Netzzugang per Modem (analoge Telefonleitung)	211
4.5.2	ISDN	212
4.5.3	DSL-Dienste	214
4.6	Die TCP/IP-Protokollfamilie	217
4.6.1	Netzzugang in TCP/IP-Netzwerken	219
4.6.2	IP-Adressen, Datagramme und Routing	220

4.6.3	Transportprotokolle	246
4.6.4	Das Domain Name System (DNS)	252
4.6.5	Verschiedene Internet-Anwendungsprotokolle	256
4.7	Andere Protokollstapel	272
4.7.1	Die AppleTalk-Protokollfamilie	272
4.7.2	Novell IPX/SPX	275
4.7.3	NetBEUI/SMB	277
4.8	Zusammenfassung	277
5	Betriebssystemgrundlagen	281
5.1	Entwicklung der Betriebssysteme	282
5.1.1	Die Geschichte von Unix	284
5.1.2	PC-Betriebssysteme	286
5.2	Aufgaben und Konzepte	290
5.2.1	Allgemeiner Aufbau von Betriebssystemen	290
5.2.2	Prozessverwaltung	298
5.2.3	Speicherverwaltung	302
5.2.4	Dateisysteme	304
5.3	Die allgegenwärtige Virtualisierung	311
5.3.1	Virtualisierungslösungen im Überblick	312
5.3.2	VMware Workstation als konkretes Beispiel	313
5.4	Zusammenfassung	317
6	Windows	319
6.1	Allgemeine Informationen	319
6.1.1	Die verschiedenen Windows-Versionen	319
6.1.2	Windows-Dateisysteme	323
6.2	Windows im Einsatz	324
6.2.1	Die Windows-Benutzeroberfläche	325
6.2.2	Die Windows-Konsole	333
6.2.3	Die Windows PowerShell	337
6.2.4	Windows-Konfiguration	348
6.3	Windows-Netzwerkkonfiguration	355
6.3.1	Allgemeine Einstellungen	355
6.3.2	TCP/IP-Dienstprogramme	357
6.3.3	Datei- und Druckserver unter Windows	360
6.3.4	Windows-Server	361
6.4	Zusammenfassung	363

7	Linux	365
7.1	Arbeiten mit der Shell	367
7.1.1	Booten und Login	367
7.1.2	Virtuelle Terminals	370
7.1.3	Grundfunktionen der Shell	371
7.1.4	Hilfefunktionen	377
7.1.5	Pipes und Ein-/Ausgabeumleitung	380
7.1.6	Die wichtigsten Systembefehle	383
7.2	Konfigurations- und Administrationsaufgaben	397
7.2.1	Syslog und Log-Dateien	397
7.2.2	Programme automatisch starten	399
7.2.3	Software installieren	401
7.3	Automatisierung	403
7.3.1	Shell-Skripte	403
7.3.2	Weitere Hilfsmittel	407
7.4	Editoren	410
7.4.1	vi	410
7.4.2	Emacs	417
7.5	Grafische Benutzeroberflächen	422
7.5.1	Der X-Server	422
7.5.2	Desktops	423
7.6	Netzwerkkonfiguration unter Linux	428
7.6.1	Grundeinstellungen	428
7.6.2	TCP/IP-Dienstprogramme	429
7.6.3	Datei- und Druckserver unter Linux	430
7.7	Zusammenfassung	434
8	Mac OS X	437
8.1	Mit Aqua arbeiten	440
8.1.1	Die Menüleiste	442
8.1.2	Das Dock	443
8.1.3	Der Finder	443
8.1.4	Exposé und Dashboard	445
8.2	Systemkonfiguration	447
8.2.1	Besonderheiten der Mac-Dateisysteme	448
8.3	Mac OS X-Netzwerkkonfiguration	449
8.3.1	Serverdienste unter Mac OS X	451
8.4	Zusammenfassung	453

9	Grundlagen der Programmierung	455
9.1	Die Programmiersprache C	456
9.1.1	Das erste Beispiel	457
9.1.2	Elemente der Sprache C	460
9.1.3	Die C-Standardbibliothek	479
9.2	Java	483
9.2.1	Grundlegende Elemente der Sprache Java	485
9.2.2	Objektorientierte Programmierung mit Java	490
9.2.3	Dateizugriffe in Java	496
9.3	Perl	497
9.3.1	Das erste Beispiel	499
9.3.2	Elemente der Sprache Perl	500
9.4	Ruby	510
9.4.1	Das erste Beispiel	512
9.4.2	Ruby-Grundelemente	514
9.4.3	Objektorientierung in Ruby	528
9.5	Zusammenfassung	539
10	Konzepte der Programmierung	541
10.1	Algorithmen und Datenstrukturen	541
10.1.1	Ein einfaches Praxisbeispiel	541
10.1.2	Sortier-Algorithmen	544
10.1.3	Such-Algorithmen	548
10.1.4	Ausgewählte Datenstrukturen	550
10.2	Reguläre Ausdrücke	555
10.2.1	Muster für reguläre Ausdrücke	556
10.2.2	Programmierung mit regulären Ausdrücken	560
10.3	Systemnahe Programmierung	563
10.3.1	Prozesse und Pipes	563
10.3.2	Threads	567
10.4	Einführung in die Netzwerkprogrammierung	571
10.4.1	Die Berkeley Socket API	571
10.4.2	Ein praktisches Beispiel	578
10.4.3	Ein Ruby-Webserver	582
10.5	Verteilte Anwendungen mit Java Enterprise Edition	598
10.5.1	Enterprise Java Beans (EJB)	599
10.5.2	Java Servlets	604
10.5.3	Webservices	605
10.6	GUI- und Grafikprogrammierung	608

10.6.1	Zeichnungen und Grafiken erstellen	610
10.6.2	Animation	615
10.6.3	Programmierung fensterbasierter Anwendungen	619
10.6.4	Java-Applets	632
10.7	Zusammenfassung	634

11 Software-Engineering 637

11.1	Überblick	638
11.1.1	Der Entwicklungszyklus	639
11.1.2	Planung und Analyse	640
11.1.3	Entwurf	646
11.1.4	Implementierung und Test	647
11.1.5	Dokumentation	648
11.1.6	Konkrete Entwicklungsverfahren	649
11.2	Werkzeuge	653
11.2.1	UML	654
11.2.2	Entwurfsmuster	660
11.2.3	Unit-Tests	668
11.3	Zusammenfassung	672

12 Datenbanken 675

12.1	Die verschiedenen Datenbanktypen	676
12.1.1	Einzeltabellendatenbanken	678
12.1.2	Relationale Datenbanken	679
12.1.3	Objektorientierte Datenbanken	686
12.2	MySQL – ein konkretes RDBMS	689
12.2.1	MySQL installieren und konfigurieren	690
12.2.2	Erste Schritte mit dem mysql-Client	693
12.3	SQL-Abfragen	694
12.3.1	Datenbanken und Tabellen erzeugen	695
12.3.2	Auswahlabfragen	699
12.3.3	Einfüge-, Lösch- und Änderungsabfragen	703
12.3.4	Transaktionen	705
12.4	MySQL-Administration	706
12.4.1	mysqladmin	706
12.4.2	Benutzerverwaltung	707
12.4.3	Import und Export von Daten, Backups	712
12.4.4	Konfigurationsdateien	715
12.4.5	Log-Dateien	716

12.4.6	Replikation	718
12.5	Grundlagen der Datenbankprogrammierung	720
12.6	Zusammenfassung	724
13	Server für Webanwendungen	727
13.1	HTTP im Überblick	727
13.1.1	Ablauf der HTTP-Kommunikation	728
13.1.2	HTTP-Statuscodes	731
13.1.3	HTTP-Header	734
13.2	Der Webserver Apache	737
13.2.1	Apache im Überblick	738
13.2.2	Apache-Module	739
13.2.3	Apache installieren	741
13.2.4	Apache-Konfiguration	744
13.3	PHP installieren und einrichten	758
13.3.1	Installation	759
13.3.2	Die PHP-Konfigurationsdatei php.ini	763
13.4	Zusammenfassung	765
14	Weitere Internet-Serverdienste	767
14.1	Namens- und Verzeichnisdienste	767
14.1.1	Der DNS-Server BIND	767
14.1.2	Der Verzeichnisdienst OpenLDAP	773
14.2	Sonstige Server	783
14.2.1	vsftpd, ein FTP-Server	783
14.2.2	inetd und xinetd	785
14.3	Zusammenfassung	788
15	XML	791
15.1	Der Aufbau von XML-Dokumenten	793
15.1.1	Die grundlegenden Bestandteile von XML-Dokumenten	793
15.1.2	Wohlgeformtheit	801
15.2	DTDs und XML Schema	804
15.2.1	Document Type Definitions (DTDs)	804
15.2.2	Namensräume	815
15.2.3	XML Schema	816
15.3	XSLT	819
15.4	Grundlagen der XML-Programmierung	825
15.4.1	SAX	827

15.4.2	DOM	834
15.5	Zusammenfassung	836
16	Weitere Datei- und Datenformate	837
16.1	Textdateien und Zeichensätze	837
16.1.1	Das Problem des Zeilenumbruchs	838
16.1.2	Zeichensätze	840
16.1.3	Textbasierte Dateiformate	847
16.2	Binäre Dateiformate	850
16.2.1	Bilddateiformate	853
16.2.2	Multimedia-Dateiformate	857
16.2.3	Archivdateien verwenden	859
16.3	Zusammenfassung	862
17	Webseitenerstellung mit (X)HTML und CSS	865
17.1	HTML und XHTML	866
17.1.1	Die Grundstruktur von HTML-Dokumenten	867
17.1.2	Textstrukturierung und Textformatierung	869
17.1.3	Listen und Aufzählungen	876
17.1.4	Hyperlinks	880
17.1.5	Bilder in Webseiten einbetten	884
17.1.6	Tabellen	888
17.1.7	Formulare	894
17.1.8	Einbetten von Multimedia-Dateien	902
17.1.9	Meta-Tags und Suchmaschinen	905
17.2	Cascading Style Sheets (CSS)	908
17.2.1	Platzieren von Stylesheets	909
17.2.2	Stylesheet-Wertangaben	911
17.2.3	Stylesheet-Eigenschaften	912
17.2.4	Layer erzeugen und positionieren	916
17.3	Zusammenfassung	922
18	Webserveranwendungen	923
18.1	PHP	923
18.1.1	Sprachgrundlagen	924
18.1.2	Klassen und Objekte	936
18.1.3	Include-Dateien, Autoloader und Namespaces	947
18.1.4	Webspezifische Funktionen	948
18.1.5	Zugriff auf MySQL-Datenbanken	954

18.1.6	Unit-Tests mit PHPUnit	966
18.2	Ruby on Rails	973
18.2.1	Grundlagen	973
18.2.2	Ein Praxisbeispiel	974
18.3	Weitere Technologien im Überblick	985
18.3.1	Content-Management, Weblogs & Co.	986
18.3.2	Blogs und Wikis	988
18.4	Zusammenfassung	990
19	JavaScript und Ajax	993
19.1	JavaScript	993
19.2	JavaScript im HTML-Dokument	994
19.2.1	Erstes Beispiel: Ausgabe ins Dokument	995
19.3	Formulare und Event Handler	999
19.3.1	Erstes Beispiel	999
19.3.2	Zugriff auf Formulare und ihre Elemente	1001
19.4	Datums- und Uhrzeit-Funktionen	1008
19.4.1	Datums- und Uhrzeit-Methoden	1009
19.4.2	Timeout – die JavaScript-»Stoppuhr«	1010
19.5	Manipulation von Bildern	1011
19.5.1	Erstes Beispiel: Austauschen eines Bildes auf Knopfdruck	1012
19.5.2	Vorausladen von Bildern	1013
19.5.3	Eine gut funktionierende Rollover-Lösung	1014
19.6	Browser- und Fensteroptionen	1018
19.6.1	Browser-Eigenschaften	1019
19.6.2	Automatische Hyperlinks – History und Location	1020
19.6.3	Neue Browserfenster öffnen	1021
19.7	DHTML und DOM	1025
19.7.1	W3C-DOM im Überblick	1025
19.7.2	Eine DOM-Baum-Anzeige	1028
19.7.3	DOM-Anwendung in der Praxis	1030
19.7.4	Dokumentinhalte verändern und austauschen	1033
19.8	Ajax	1035
19.8.1	Die erste Ajax-Anwendung	1036
19.8.2	Datenaustauschformate: XML und JSON	1042
19.8.3	Größeres Beispiel: eine interaktive Länderliste	1043
19.9	Zusammenfassung	1056

20 Computer- und Netzwerksicherheit 1057

20.1	PC-Gefahren	1058
20.1.1	Viren und Würmer	1058
20.1.2	Trojaner und Backdoors	1064
20.1.3	Weitere Schädlinge	1065
20.2	Netzwerk- und Serversicherheit	1071
20.2.1	Servergefahren	1071
20.2.2	Wichtige Gegenmaßnahmen	1073
20.2.3	Kryptografie	1079
20.3	Zusammenfassung	1082

Anhang 1085

A	Glossar	1087
B	Zweisprachige Wortliste	1101
B.1	Englisch-Deutsch	1101
B.2	Deutsch-Englisch	1103
C	Kommentiertes Literatur- und Linkverzeichnis	1107
C.1	Allgemeine Einführungen und Überblicke	1107
C.2	Mathematische und technische Grundlagen	1108
C.3	Hardware	1108
C.4	Netzwerktechnik	1109
C.5	Betriebssystemgrundlagen	1109
C.6	Windows	1110
C.7	Linux	1111
C.8	Mac OS X	1111
C.9	Grundlagen der Programmierung	1112
C.10	Konzepte der Programmierung	1113
C.11	Software-Engineering	1113
C.12	Datenbanken	1114
C.13	Server für Webanwendungen	1115
C.14	Weitere Internet-Serverdienste	1115
C.15	XML	1116
C.16	Webseitenerstellung mit HTML und CSS	1116
C.17	Webserveranwendungen	1116
C.18	JavaScript und Ajax	1117
C.19	Computer- und Netzwerksicherheit	1118

Index	1119
-------------	------

Die Zeit vergeht nicht schneller als früher, aber wir laufen eiliger an ihr vorbei.
– George Orwell

Vorwort

Sind wirklich schon wieder zwei Jahre vergangen? So fragte ich mich, als es hieß, die vorliegende Neuauflage in Angriff zu nehmen.

Sind wirklich erst zwei Jahre vergangen? So fragte ich mich dann, als ich an der Arbeit saß und mir klarmachte, was seit der letzten Auflage in der IT-Welt alles passiert ist. Internetzugang über Mobilgeräte ist inzwischen fast allgegenwärtig, das Cloud Computing ist in aller Munde, HTML5 und CSS3 revolutionieren die Art und Weise, wie Webseiten gestaltet werden, und NoSQL-Datenbanken wie CouchDB werden benutzt, wo relationale Datenbanken nicht passen. Und das sind nur die Neuerungen, die ich in diese Neuauflage aufgenommen habe.

Der Rest wurde überarbeitet, von bekannten Fehlern befreit und auf den neuesten Stand gebracht. Ich hoffe, dass die Mischung von Theorie- und Praxisthemen, von allgemeinen Konzepten und aktuellen Technologien wieder einmal gelungen ist, und wünsche Ihnen viel Spaß beim Lesen und viel Erfolg während Ihrer IT-Ausbildung.

Worum geht es in diesem Buch?

Dieses Buch beschreibt die wesentlichen Ausbildungsinhalte im Fachkunde-Unterricht in den Ausbildungsberufen Fachinformatiker Anwendungsentwicklung und Systemintegration, IT-Systemelektroniker, IT-Systemkaufleute und Informatikkaufleute. Anders als herkömmliche Bücher handelt es sich jedoch nicht um eine Tabellensammlung zum Auswendiglernen, sondern um praxisorientierte Anleitungen, die Sie weit über Ihre Ausbildung hinaus bei Ihrer alltäglichen Arbeit mit Computersystemen einsetzen können.

Sie lernen in den folgenden Kapiteln zahlreiche Geräte, Betriebssysteme, Programmiersprachen und Anwendungsprogramme kennen und erfahren viel Wissenswertes über Netzwerke und das Internet. Bei allen Themen habe ich versucht, die Balance zwischen Theorie und Praxis zu wahren: Es werden weder die technischen und theoretischen Details verschwiegen, wie in zahlreichen Büchern für absolute Einsteiger, noch kommen die

praktischen Anwendungsbeispiele zu kurz, was in manchen akademischen Lehrwerken der Fall ist.

Soweit es möglich ist, habe ich versucht, konkrete Software aus dem Open-Source-Bereich auszuwählen. Zum einen, weil ich selbst seit Jahren Open-Source-Software einsetze und entwickle, vor allem aber auch, weil Sie auf diese Weise fast jedes in diesem Buch behandelte Programm kostenlos und ohne jegliche Einschränkungen herunterladen, installieren und benutzen können.

- [»] Die Ausbildung zum Fachinformatiker oder für andere IT-Berufe enthält neben den fachspezifischen Inhalten auch Aspekte wie Wirtschafts- und Sozialwissenschaften, Rechtskunde oder Betriebsorganisation. Solche Themen können in diesem Buch nicht berücksichtigt werden. Für das Fach Technisches Englisch finden Sie in Anhang B immerhin eine Stichwortliste in beiden Richtungen.

Kapitelübersicht

Die einzelnen Kapitel dieses Buches widmen sich den folgenden Themen:

- ▶ **Kapitel 1**, »Einführung«, behandelt die Geschichte und grundlegende Funktionsweise des Computers. Außerdem werden einige wichtige Grundlagen der Informationstechnik erläutert; sie bilden die Voraussetzung für das Verständnis späterer Kapitel.
- ▶ In **Kapitel 2**, »Mathematische und technische Grundlagen«, werden zunächst die mathematischen und logischen Prinzipien erläutert, auf denen der Computer basiert. Anschließend lernen Sie die wichtigsten elektrotechnischen Grundbausteine kennen. Abgerundet wird das Kapitel durch die Vorstellung von Automatentheorien und Rechnersimulationen.
- ▶ **Kapitel 3**, »Hardware«, beschäftigt sich mit den diversen Bauteilen, aus denen ein Computer besteht, sowie mit zahlreichen wichtigen Peripheriegeräten. Sie erfahren nicht nur die wichtigsten technischen Details über Elemente wie den Mikroprozessor, verschiedene Laufwerke und Datenträger oder andere Ein- und Ausgabegeräte, sondern es werden auch praktische Informationen vermittelt. Zum Beispiel werden Einbau und Anschluss von Komponenten oder das BIOS-Setup angesprochen.
- ▶ In **Kapitel 4**, »Netzwerkgrundlagen«, wird die Entwicklung der Netzwerke und des Internets beschrieben, und Sie erhalten eine Einführung in die Begriffswelt der Netzwerke. Beispielsweise werden Schichtenmodelle und Netzwerkarchitekturen vorgestellt. Danach wird die genaue Funktionsweise verschiedener Arten von Netzwerkkarten und -anschlüssen erläutert; außerdem werden diverse Netzwerkprotokolle beschrieben. Den Schwerpunkt bildet die TCP/IP-Protokollfamilie, die für das Internet entwickelt wurde und inzwischen der wichtigste Kommunikationsstandard für alle Arten von Netzwerken ist.

- ▶ In **Kapitel 5**, »Betriebssystemgrundlagen«, werden allgemeine Konzepte des Betriebssystemaufbaus erklärt, zum Beispiel die Verwaltung von Prozessen, das Speicher-
management oder die Dateiverwaltung.
- ▶ **Kapitel 6**, »Windows«, beschreibt die theoretischen und praktischen Grundlagen des verbreitetsten PC-Betriebssystems der Welt; den Schwerpunkt bildet die aktuelle Ver-
sion Windows 7, wobei die Verhältnisse in den recht ähnlichen älteren Versionen Vista
und XP oft mit behandelt werden. Neben der grafischen Benutzeroberfläche wird auch
die Arbeit mit der klassischen Windows-Kommandozeile sowie mit der neueren Win-
dows PowerShell behandelt. Den Abschluss bildet ein Einstieg in die Windows-Konfi-
guration.
- ▶ In **Kapitel 7**, »Linux«, wird eine zweite Betriebssystemfamilie praktisch beleuchtet.
Schwerpunktmäßig wird die Arbeit mit der Linux-Shell *bash* und den zahlreichen Sys-
temprogrammen behandelt; daneben werden die Grundlagen der Administration und
des Umgangs mit klassischen Linux/Unix-Texteditoren beschrieben.
- ▶ **Kapitel 8**, »Mac OS X«, beschäftigt sich mit dem Betriebssystem, mit dem Apple seine
diversen Macintosh-Rechner liefert. Besonders die komfortable Benutzeroberfläche
und ihre Einstellmöglichkeiten kommen zur Sprache, da Mac OS X ein Unix-System ist
und der Konsolenbereich somit demjenigen von Linux sehr ähnlich ist.
- ▶ In **Kapitel 9**, »Grundlagen der Programmierung«, werden vier verschiedene wichtige
Programmiersprachen eingeführt, die unterschiedliche Entwicklungsstufen und As-
pekte des Programmierens abdecken: C, Java, Perl und Ruby.
- ▶ **Kapitel 10**, »Konzepte der Programmierung«, baut auf diese Grundlagen auf und erläu-
tert verschiedene Aspekte, die die Programmierung in der Praxis ausmachen: Algorith-
men und Datenstrukturen, reguläre Ausdrücke (mächtige Suchmuster), Elemente der
systemnahen Programmierung oder die Entwicklung von Programmen für grafische
Oberflächen. Zudem erfahren Sie in Grundzügen, wie Netzerkennungen pro-
grammiert werden, und Sie erhalten eine Einführung in die Programmierung verteilter
Anwendungen mit der Java 2 Enterprise Edition und in Webservices.
- ▶ Das in **Kapitel 11** behandelte »Software-Engineering« geht weit über reine Program-
miertechniken hinaus: Sie lernen viele unterschiedliche Methoden zur Bearbeitung
und Verwaltung von Softwareprojekten kennen. Unter anderem werden die Anwen-
dungsmodellierung mit der UML, Grundlagen des Projektmanagements und verschie-
dene Ansätze des Software-Engineerings wie der Unified Process oder Extreme Pro-
gramming vorgestellt. Für die in den IT-Ausbildungsberufen fällige Projektarbeit sind
diese Themen überlebenswichtig.
- ▶ **Kapitel 12**, »Datenbanken«, befasst sich mit einer der wichtigsten Funktionsgrundla-
gen vieler Programmierprojekte und Anwendungen. Nach der üblichen Erläuterung
von Begriffen und Konzepten wird als konkretes Datenbanksystem der weitverbreitete

Open-Source-Datenbankserver MySQL eingeführt. Am Ende des Kapitels steht noch ein kurzer Überblick über die NoSQL-Datenbank CouchDB.

- ▶ In **Kapitel 13**, »Server für Webanwendungen«, wird zunächst das Webprotokoll HTTP beschrieben. Danach geht es um die Installation und Konfiguration des verbreiteten Open-Source-Webservers Apache 2 mitsamt der Web-Programmiersprache PHP.
- ▶ **Kapitel 14**, »Weitere Internet-Serverdienste«, befasst sich mit diversen weiteren Servern für TCP/IP-basierte Netzwerke: dem Nameserver BIND, dem Verzeichnisdienst-Server OpenLDAP, dem FTP-Server `vsftpd` sowie dem Universal- oder Metaserver `xinetd`.
- ▶ **Kapitel 15**, »XML«, stellt die eXtensible Markup Language vor, eine Sprache, die der Definition beliebiger hierarchisch gegliederter Dokumentformate dient. In zahlreichen Anwendungen wird XML inzwischen eingesetzt, so dass es nützlich ist, die Konzepte dieses Formats zu kennen. Sie erfahren das Wichtigste über wohlgeformte Dokumente, über Formatbeschreibungen mit Hilfe von DTDs und XML Schema, die Umwandlung von XML-Dokumenten mit XSLT sowie über die Programmierung XML-basierter Anwendungen.
- ▶ **Kapitel 16**, »Weitere Datei- und Datenformate«, widmet sich den wichtigsten Formaten für Text, Bild und Multimedia. Es geht zunächst um Text und Zeichensätze, anschließend werden verschiedene textbasierte und binäre Dateiformate konkret erläutert.
- ▶ In **Kapitel 17**, »Webseitenerstellung mit (X)HTML und CSS«, wird zunächst HTML beziehungsweise XHTML vorgestellt, die Sprache, in der Webseiten verfasst werden. Hier werden verschiedene konkrete Aspekte der Webseitenerstellung erläutert, etwa die Textstrukturierung, der Listen- und Tabellensatz, das Einbetten von Bildern, Hyperlinks und Webformulare. Wichtige Neuerungen aus HTML5 werden jeweils an den passenden Stellen genannt. Der zweite Teil beschreibt das konsistente Layout von Webseiten mit Cascading Style Sheets (CSS), samt einer Einführung in die wichtigsten Neuerungen von CSS3.
- ▶ In **Kapitel 18**, »Webserveranwendungen«, erfahren Sie, wie Sie Websites erstellen, die nicht nur aus statischen HTML-Dokumenten, sondern auch aus dynamisch generierten Inhalten bestehen. Zuerst wird die Programmiersprache PHP 5 behandelt. Neben den allgemeinen Konzepten wird die Programmierung konkreter, datenbankbasierter Anwendungen beschrieben. Danach geht es um das moderne Web-Framework Ruby on Rails.
- ▶ **Kapitel 19**, »JavaScript und Ajax«, stellt die wichtigste clientseitige Programmiersprache vor, mit der Sie die Inhalte einer Webseite »zum Leben erwecken« können. Zunächst werden die klassischen Anwendungen wie die Ausgabe ins Dokument selbst, die Verarbeitung von Formularen oder der Austausch von Bildern behandelt, anschlie-

ßend erfahren Sie das Wichtigste über DOM, die Standardtechnik, mit der Sie die Elemente eines Dokuments nachträglich modifizieren können, sowie Ajax, um die Inhalte für diese Änderungen ohne Neuaufbau der Seite vom Server nachzuladen. Zuletzt gibt es noch einen Schnellüberblick über jQuery als Beispiel für eine der immer häufiger verwendeten komfortablen JavaScript-Ajax-Bibliotheken.

- **Kapitel 20**, »Computer- und Netzwerksicherheit«, befasst sich mit verschiedenen Themen der immer wichtiger werdenden Sicherheit: Der Schutz vor Viren, Würmern und Trojanern wird ebenso behandelt wie Kryptographie, Firewalls oder Fragen der Datensicherheit.

Nach den Kapiteln folgen noch drei Anhänge:

- **Anhang A**, »Glossar«, enthält kurze Beschreibungen der wichtigsten IT-Stichwörter.
- In **Anhang B**, »Zweisprachige Wortliste«, finden Sie ein deutsch-englisches und ein englisch-deutsches Verzeichnis wichtiger Fachbegriffe.
- **Anhang C**, »Kommentiertes Literatur- und Linkverzeichnis«, empfiehlt weiterführende Bücher und Webressourcen zu den Themen der verschiedenen Kapitel.

Für wen ist dieses Buch geeignet?

In erster Linie können Sie mit diesem Handbuch etwas anfangen, wenn Sie eine Ausbildung im IT-Bereich oder in verwandten Berufen absolvieren. Es wurde insbesondere für den im Titel genannten Ausbildungsberuf des Fachinformatikers in den beiden Fachrichtungen Anwendungsentwicklung und Systemintegration geschrieben, dürfte aber auch für IT-Systemelektroniker oder IT-Systemkaufleute gut geeignet sein. Genau wie in Ihrer Ausbildung werden auch in diesem Buch viele verschiedene Themenbereiche behandelt: Da Computer sehr komplexe Maschinen sind, ist es erforderlich, sich ein großes Spektrum verschiedener Kenntnisse anzueignen, selbst dann, wenn Sie nur in einem bestimmten Fachbereich arbeiten möchten.

Auch für Studenten im Grundstudium der Informatik oder in den Informatikkursen anderer Studienrichtungen ist das Buch durchaus geeignet. Gerade der Mittelweg zwischen Darstellung der theoretischen Grundlagen und praktischer Anleitung dürfte für Sie eine wichtige Lücke schließen.

Zu guter Letzt ist dieses Buch aber natürlich auch für alle anderen geeignet, die an Computern, Programmierung oder Netzwerken interessiert sind. Es eignet sich nicht nur als Unterrichtsbegleiter, sondern auch zum Selbststudium einzelner Themen. Die einzige Voraussetzung ist im Grunde genommen, dass Sie einen Computer zur Verfügung haben und grundsätzlich wissen, wie Sie damit umgehen.

Zum Komplexitätsniveau ist noch anzumerken, dass dies weder ein Buch für absolute Neueinsteiger noch ein Begleiter für vollkommene Experten ist. Wenn Sie noch nie mit einem Computer gearbeitet haben, benötigen Sie eine grundlegendere Anleitung; im vorliegenden Buch erfahren Sie nicht, wie Sie das Gerät in Betrieb nehmen, Ihre Arbeit als Datei speichern, einen Ordner anlegen oder einen Webbrowser bedienen. All diese Dinge (und noch einige mehr) müssen klar sein, bevor Sie etwas Sinnvolles mit diesem Handbuch anfangen können. Wenn Sie dagegen bereits Erfahrung haben, gibt es wahrscheinlich einige Themen, über die Sie noch nicht Bescheid wissen. In diesem Fall finden Sie möglicherweise hier, was Sie suchen.

Icons

Um das Buch für Sie übersichtlicher und damit noch nützlicher zu machen, finden Sie an manchen Stellen die folgenden zwei Symbole in der Randspalte:

- [>>] Dieses Icon markiert einen hilfreichen Tipp, eine Abkürzung oder eine nützliche Zusatzinformation.
- [!] Mit diesem Symbol werden Warnungen gekennzeichnet. Sie sollten sie ernst nehmen, um keine unnötigen Datenverluste oder Fehlfunktionen zu erleiden.

IHK-Fragen und Beispiellistings im Internet

Unter <http://www.buecher.lingoworld.de/fachinfo/> finden Sie alle im Buch abgedruckten Beispiellistings, so dass Sie diese nicht abtippen müssen. Außerdem finden Sie dort Fragen, mit denen Sie sich auf die IHK-Prüfung vorbereiten können, ein Diskussionsforum sowie jeweils aktualisierte Literatur- und Linklisten.

Danksagungen

Zunächst einmal, wie immer, vielen Dank an das Team von Galileo Press. Dass dieses Buch so gut aussieht, angenehm in der Hand liegt, ordentlich gebunden ist und sogar ein Lesebändchen besitzt, ist den Bemühungen des Verlags zu verdanken. Besonderer Dank gebührt wieder einmal meinem Lektor Stephan Mattescheck, der für die inhaltliche Seite zuständig ist und dieser Verantwortung stets mit Bravour gerecht wird. Vielen Dank auch an die Korrektorin Petra Biedermann, die mir bei der Korrektur weit über die reine Rechtschreib- und Grammatikkorrektur hinausgehende, wertvolle, inhaltliche Hinweise gegeben hat.

Ich danke auch den zahlreichen Lesern der ersten vier Auflagen, die mir nette Kommentare und auch Hinweise auf vorhandene Fehler zugeschickt haben. Durch diesen wertvollen Beitrag konnte ich viele Stellen noch entscheidend verbessern oder klarer formu-

lieren. Die fast durchweg positive Resonanz auf die vier Voraufgaben hat nicht nur zu meinem Wohlbefinden beigetragen, sondern auch die Motivation zum Schreiben des vorliegenden Buches gesteigert.

Eine Danksagung wäre nicht komplett, wenn sie nicht auch die wichtigsten Menschen in meinem Leben einschließen würde: meine Familie. Vor allem mein Sohn und meine Frau haben wieder einmal unzählige Stunden auf mich verzichten müssen und waren immer für mich da. Meine Frau hat zudem diverse Infografiken für dieses Buch angefertigt.

Sascha Kersken

*In der Informatik geht es genauso wenig um Computer
wie in der Astronomie um Teleskope.
– Edsger W. Dijkstra*

1 Einführung

In diesem Kapitel erhalten Sie einen Überblick über die wichtigsten Grundlagen der Informationstechnik: eine Abgrenzung der verschiedenen Fachrichtungen und Ausbildungsgänge der Informatik, die Entwicklungsgeschichte der Computer sowie eine allgemeine Übersicht über die digitale Speicherung verschiedener Arten von Informationen.

1.1 Informationstechnik, Informatik und EDV

Allgemein gesprochen geht es in diesem Buch um *Informationstechnik* (englisch: *information technology* oder kurz IT). Der traditionelle Begriff für diese Art der Technik lautet *elektronische Datenverarbeitung* (EDV). *Daten* oder *Informationen* sind Werte, die im Zusammenhang mit beliebigen Sachverhalten angelegt werden oder die im Rahmen von Mess- oder Rechengvorgängen anfallen. *Datenverarbeitung* ist der Vorgang der Sammlung, Speicherung und Manipulation dieser Informationen. Im Lauf der Zeit haben sich unterschiedliche Verfahren der Datenverarbeitung entwickelt:

- ▶ Die *manuelle Datenverarbeitung* führt Berechnungen und Datenmanipulationen ohne jegliche Hilfsmittel durch; sie basiert auf Kopfrechnen und Auswendiglernen. Das äußerste erlaubte Hilfsmittel ist ein Schreibblock, um Daten oder Zwischenergebnisse zu notieren.
- ▶ Die *mechanische Datenverarbeitung* verwendet mechanische Hilfsmittel für die Verarbeitung von Informationen, beispielsweise eine mechanische Schreibmaschine oder einen Rechenschieber.
- ▶ Bei der *elektrischen Datenverarbeitung* werden elektrisch betriebene Geräte als Hilfsmittel eingesetzt, zum Beispiel elektrische Schreibmaschinen oder klassische Registrierkassen.
- ▶ Die *elektronische Datenverarbeitung* verwendet schließlich elektronisch gesteuerte Arbeitsmittel, also Elektronenrechner oder Computer.

Arten der
Datenver-
arbeitung

1.1.1 Fachrichtungen der Informatik

Die wissenschaftliche Fachrichtung, die sich mit den verschiedenen Aspekten der Computertechnik auseinandersetzt, wird seit den 60er Jahren des 20. Jahrhunderts als *Informatik* (englisch: *computer science*) bezeichnet; es handelt sich um ein Kunstwort aus Information und Mathematik. Die akademische Informatik wird üblicherweise in vier Fachrichtungen unterteilt:

- ▶ Die *theoretische Informatik* betrachtet insbesondere die mathematisch-logischen Grundlagen, die der Verwendung und Programmierung von Computern zugrunde liegen. Es geht beispielsweise um die *Berechenbarkeit* (ist ein Problem überhaupt durch Berechnung lösbar?) und um *Automatentheorien* – die mathematisch-formalen Modelle, auf denen Rechner unabhängig von der elektronischen Machbarkeit aufbauen.
- ▶ Die *technische Informatik* beschreibt die elektronisch-technischen Eigenschaften der Bauteile, aus denen Computer zusammengesetzt sind. Ein wichtiges Teilgebiet der technischen Informatik ist die *Schaltalgebra*, die Umsetzung logischer Operationen durch elektronische Schaltungen.
- ▶ In der *praktischen Informatik* geht es im Großen und Ganzen um die *Programmierung* von Computern und die Mittel, die dazu erforderlich sind. Die Erforschung des Aufbaus von Betriebssystemen und Programmiersprachen-Compilern sowie deren Implementierung (praktische Umsetzung) sind die wichtigsten Teilgebiete.
- ▶ Die *angewandte Informatik* kümmert sich gewissermaßen um alles andere, nämlich um sämtliche Nutzanwendungen von Computern. Das reicht von Datenbanken über die Netzwerkkommunikation bis hin zu Grafik, Animation und Audio-/Videobearbeitung.

Da es sich bei diesem Buch um ein Praxisbuch handelt, das nicht für das trockene Auswendiglernen von Lehrsätzen geschrieben wurde, sondern für die alltägliche Nutzung von Computern, ist es kaum verwunderlich, dass sich fast alle Kapitel mit Aspekten der angewandten Informatik beschäftigen. In den Kapiteln 5 bis 10 werden auch die wichtigsten Ansätze der praktischen Informatik, nämlich Betriebssysteme und diverse Themen der Programmierung, behandelt.

Einige grundlegende Aspekte der theoretischen Informatik lernen Sie in Kapitel 2, »Mathematische und technische Grundlagen«, kennen: Dort werden die wichtigsten mathematischen und logischen Operationen besprochen, die Computer ausführen. Außerdem wird beispielhaft auf die Realisierung einiger dieser Funktionen durch elektronische Bauteile eingegangen, also auf einige Ansätze der technischen Informatik.

Im Übrigen gibt es noch ein eigenes Kapitel zum Thema Hardware, die ebenfalls dem Gebiet der technischen Informatik zugeordnet werden kann (Kapitel 3, »Hardware«).

1.1.2 Überblick über die IT-Ausbildung

Das Berufsfeld der Informationstechnik ist vielfältig. Deshalb gibt es zahlreiche unterschiedliche Möglichkeiten einer Ausbildung in diesem Bereich. Sie lassen sich zum einen nach Fachgebieten wie Anwendungsentwicklung, Systemintegration oder kaufmännischem EDV-Einsatz unterscheiden. Zum anderen gibt es zwei grundlegende Ausbildungsformen: Berufsausbildung und Studium.

Aus diesen beiden Unterteilungen ergeben sich unter anderem folgende konkrete Ausbildungsgänge:

Überblick
Ausbildungs-
gänge

- ▶ Ausbildungsberufe
 - ▶ Fachinformatiker/in mit den Fachrichtungen Anwendungsentwicklung und Systemintegration
 - ▶ IT-Systemelektroniker/in
 - ▶ IT-Systemkaufmann/frau
 - ▶ Informatikkaufmann/frau
- ▶ Studiengänge
 - ▶ Informatik
 - ▶ Informatik (FH)
 - ▶ Wirtschaftsinformatik
 - ▶ Medieninformatik
 - ▶ Bioinformatik
 - ▶ medizinische Informatik

Ausbildungsberufe

Bis etwa Mitte der 1990er Jahre galt die Informationstechnik als zu komplex, um in einer praxisorientierten Berufsausbildung im klassischen dualen System (Ausbildungsbetrieb – Berufsschule) gelehrt zu werden. Die beiden angebotenen Ausbildungen zum EDV-Kaufmann und Büromaschinenelektroniker hatten andere Schwerpunkte, nämlich einen kaufmännischen beziehungsweise elektrotechnischen Fokus.

Erst 1996 wurden die IT-Berufe in der Bundesrepublik durch ein Übereinkommen von Arbeitgeberverbänden und Gewerkschaften unter Beratung des Bundesinstituts für Berufsbildung (BiBB) neu geordnet. Seitdem gibt es vier verschiedene IT-Ausbildungsberufe. Sie bereiten auf Tätigkeiten im Bereich der Informations- und Telekommunikationstechnik vor, die früher Bewerberinnen mit Studienabschluss vorbehalten waren oder in Einzelfällen an qualifizierte Quereinsteiger vergeben wurden. Nach 15 Jahren Erfahrung hat sich das System bewährt, und ein signifikanter Anteil von Mitarbeitern in der IT rekrutiert sich aus Absolventen dieser Ausbildungsgänge.

IT-Berufe

Im Einzelnen handelt es sich um folgende Ausbildungsgänge:

- ▶ Fachinformatiker/in (Fachrichtungen Anwendungsentwicklung und Systemintegration)
- ▶ IT-Systemelektroniker/in
- ▶ IT-Systemkaufmann/frau
- ▶ Informatikkaufmann/frau

Fachinformatiker

Aufgabe der *Fachinformatiker* ist es gemäß offizieller Definition, »fachspezifische Anforderungen in komplexe Hard- und Softwaresysteme« umzusetzen. Diese recht ungenaue Beschreibung läuft in der Praxis auf sehr vielfältige Anforderungen hinaus, da sich nicht nur die beiden Fachrichtungen Anwendungsentwicklung und Systemintegration, sondern vor allem auch Ausbildungsbetriebe und Einsatzumgebungen oft stark voneinander unterscheiden. Gemeinsam ist ihnen allen lediglich der Umgang mit Computersystemen, mit Netzwerken, mit der unterschiedlichsten Software und dem komplexen Zusammenspiel dieser Komponenten. Verallgemeinert kann man sagen, dass der Ausbildungsgang Anwendungsentwicklung auf den Beruf des Entwicklers oder Programmierers vorbereitet, während sich aus den Absolventen der Systemintegration vor allem Systemadministratoren rekrutieren.

Die beiden Berufsbilder sollen sich im Laufe der drei Ausbildungsjahre allmählich auseinanderentwickeln. Bei Ausbildungsbeginn stehen gemeinsame Grundlagen im Vordergrund. Dazu gehören nicht nur fachspezifische Themen, wie sie in diesem Buch behandelt werden, sondern auch wichtige Informationen zum Arbeitsablauf und zum betrieblichen Umfeld. Dies sind insbesondere Grundlagen der Betriebswirtschaftslehre, der Projekt- und Betriebsorganisation, des Arbeits- und Ausbildungsrechts sowie der betrieblichen Buchführung. Diese Kenntnisse werden vor allem in der Berufsschule (60 Tage pro Ausbildungsjahr) vermittelt.

Anwendungsentwicklung

Der Schwerpunkt des *Fachbereichs Anwendungsentwicklung* ist die Erstellung von Software für den eigenen Betrieb oder für Kunden. Der erste Schritt ist die Entwicklung neuer oder die Anpassung vorhandener Programme gemäß den Anforderungen der späteren Anwender. Es folgt die Einrichtung der Software auf den gewünschten Systemen, die gegebenenfalls dafür angepasst werden müssen. Anschließend sollen die Benutzer informiert oder gar geschult werden. In das Umfeld des Berufsbildes gehört demzufolge auch die Erstellung brauchbarer Dokumentationen für Entwickler, Administratoren und Endanwender.

Systemintegration

Im *Fachbereich Systemintegration* liegen die Hauptaufgaben im Bereich der Hardware-, Software- und Netzwerkeinrichtung. Hier sind gute Kenntnisse der verschiedenen Hardwarekomponenten, Betriebssysteme sowie der Netzwerkgeräte, -dienste und -protokolle gefragt. Auch die Schnittstellen zwischen der Informations- und der Telekommunika-

tionstechnik spielen in diesem Beruf eine wichtige Rolle. Die Fachinformatiker im Fachbereich Systemintegration müssen das projektorientierte Arbeiten beherrschen und dabei alle Aufgaben planen, durchführen, testen und dokumentieren.

Ein weiteres, sehr wichtiges Arbeitsgebiet ist die Automatisierung administrativer Aufgaben. Dazu benötigen die Fachinformatiker Systemintegration umfangreiche Kenntnisse der Shells (Kommandozeileninterpreter) der jeweiligen Betriebssysteme sowie in verschiedenen Skriptsprachen. Unter Windows beginnt dies mit DOS/Windows-Batchdateien und CMD-Dateien; für komplexere Aufgaben sind auch der Windows Scripting Host und die Windows PowerShell nützlich. Im Unix-Bereich ist die bash-Shell das grundlegende Werkzeug; größere Aufgaben werden dagegen mit Editor- und Skriptsprachen wie sed, awk und Perl gelöst, wobei Perl die wichtigste dieser Sprachen ist und bei fortgeschrittenen Kenntnissen alle anderen ersetzen kann. In Kapitel 6 bis 10 dieses Buches werden die wichtigsten Kenntnisse zu diesen Shells und Sprachen vermittelt.

Der Beruf des *IT-Systemelektronikers* ist elektrotechnischer orientiert als der des Fachinformatikers. Mehr als beim Fachinformatiker der Fachrichtung Systemintegration liegt der Schwerpunkt hier im Bereich der Hardware. Zu den wichtigsten Aufgaben gehört die Verkabelung von Computern und Netzwerken, angefangen bei der Stromversorgung über den Anschluss von Peripheriegeräten bis hin zu Besonderheiten der Netzwerkinfrastruktur. Dazu benötigen IT-Systemelektroniker ein umfangreiches Know-how über Hardwarekomponenten und Treiber sowie über deren Zusammenarbeit mit Betriebssystemen und Anwendungsprogrammen. Hinzu kommen Kenntnisse über DSL, ISDN und andere Telekommunikationstechniken. Zu den wichtigsten Arbeitgebern für Systemelektroniker gehören große Telefongesellschaften, so dass deren spezifische Anforderungen dieses Berufsbild wesentlich geprägt haben.

IT-System-
elektroniker

IT-Systemkaufleute sind im Wesentlichen die Kundenberater für Hard- und Softwareprojekte. Sie verfügen über technisches Know-how, so dass sie die Einzelheiten von IT-Projekten verstehen und den Kunden erklären können, und benötigen umfangreiche betriebswirtschaftliche und kaufmännische Kenntnisse, um die Kalkulation für ein solches Projekt durchführen zu können. Sie sind für alle Phasen der Kundenbetreuung für IT-Lösungen zuständig: angefangen bei der Marktanalyse über Beratung und Marketing bis hin zum Vertrieb sowie zur Einarbeitung und Schulung.

IT-System-
kaufleute

Bei den *Informatikkaufleuten* stehen die kaufmännischen Aufgaben stärker im Vordergrund, während die technischen Aspekte etwas weniger ausgeprägt sind. Sie bilden das Bindeglied zwischen wirtschaftlichen und technischen Abteilungen. Typischerweise vermitteln sie zwischen den Kunden, die vor allem wirtschaftliche Fragen haben, und den Beratern der Hard- und Softwarehersteller (oft den oben genannten IT-Systemkaufleuten), die eine eher technische Sicht auf Vorgänge benötigen. Beispielsweise könnten Informatikkaufleute die technische Umsetzung von Geschäftsprozessen anleiten und entspre-

Informatik-
kaufleute

chende Pflichtenhefte erstellen. Eine weitere Aufgabe wäre es, die Unterschiede verschiedener IT-Lösungen in kaufmännisch kalkulierbaren Zahlen auszudrücken.

Alle IT-Ausbildungen – mit Ausnahme zweijähriger Umschulungsmaßnahmen – beinhalten eine (ausschließlich theoretische) Zwischenprüfung. Es besteht Teilnahmepflicht, aber das Ergebnis ist rein informativ – sowohl für die Auszubildenden als auch für die Betriebe.

Abschlussprüfung Die Abschlussprüfung besteht aus einer selbstgewählten Projektarbeit, die von der jeweils zuständigen Industrie- und Handelskammer genehmigt werden muss. Ihre Dauer beträgt bei den Fachinformatikern des Fachbereichs Anwendungsentwicklung 72 Stunden, wobei auch ein Pflichtenheft erstellt werden muss. Bei den anderen Ausbildungsgängen beträgt die Dauer lediglich 36 Stunden, und der Projektablauf beschränkt sich hier auf die vier Phasen Planung, Durchführung, Test und Abnahme.

Innerhalb der veranschlagten Zeit müssen sowohl das Projekt selbst als auch eine umfangreiche Dokumentation dazu erstellt werden; Letzteres sollte etwa acht Stunden der Projektdauer ausmachen. Die Dokumentation wird bei der IHK eingereicht; zudem müssen die Auszubildenden ihre Projekte dort präsentieren und ein Fachgespräch dazu führen. Wichtige Hinweise zum projektorientierten Arbeiten erhalten Sie in Kapitel 11, »Software-Engineering«.

Die zweite Säule der Abschlussprüfung bildet die bundeseinheitliche schriftliche Prüfung. Sie besteht aus drei Teilen:

- ▶ Zwei *ganzheitliche Aufgaben* (GA) beschreiben je ein virtuelles IT-Szenario und stellen sechs komplexe Fragen dazu. Die Auszubildenden müssen eine dieser Fragen, mit deren Thema sie sich am wenigsten auskennen, streichen, und die übrigen fünf ausführlich beantworten. Maximal sind dabei für jede beantwortete Frage 20 Punkte, das heißt insgesamt 100 für jede GA erreichbar. Die GA 1 enthält fachspezifische Fragen für den jeweiligen Ausbildungsgang, während die GA 2 die für alle IT-Ausbildungsberufe identische Kernqualifikation abfragt. Für jede GA sind 90 Minuten angesetzt.
- ▶ Der dritte Teil ist eine 60 Minuten dauernde Prüfung zu den Themen Wirtschaft und Soziales einschließlich Arbeits- und Ausbildungsrecht. Diese Prüfung besteht aus isolierten Fragen, wobei einige offen, viele dagegen im Multiple-Choice-Verfahren zu beantworten sind.

Ausbildung und dieses Buch In Tabelle 1.1 sehen Sie, welche Kapitel dieses Buches für die Auszubildenden der einzelnen Fachrichtungen besonders nützlich (×) oder unerlässlich (+) sind. Wenn ein Kapitel für Ihren Beruf nicht angekreuzt wurde, bedeutet dies nicht, dass Sie sich gar nicht mit diesem Thema beschäftigen sollten! Wie bereits im Vorwort erwähnt, ist ein möglichst gründliches Allgemeinwissen eine der wichtigsten Voraussetzungen für den beruflichen Erfolg. Zudem verstehen Sie Zusammenhänge der IT- und Medienwirtschaft umso besser, je größer Ihr Überblick ist.

Kapitel	Kurzname	Fachinf. Anwend.	Fachinf. Systemint.	Kaufl.	IT-System- elektr.
1	Einführung	+	+	+	+
2	Math.-Tech.	+	+	×	+
3	Hardware	×	+	×	+
4	Netzwerk	+	+	×	+
5	Betriebssyst.	+	+	×	+
6	Windows	+	+	×	+
7	Linux	+	+	×	+
8	Mac OS X	+	+	×	+
9	Grundl. Prog.	+	+	×	+
10	Konzepte Prog.	+	×		×
11	Softw.-Engin.	+	+	+	+
12	Datenbanken	+	+	×	×
13	Webserver	×	+		×
14	Internet-Serv.	×	+		×
15	XML	+	+	×	×
16	Datenformate	+	+	×	×
17	HTML/CSS	+	×	×	×
18	Webserv.-anw.	+	×		
19	JavaScript	+	×		
20	Sicherheit	×	+	×	+

Tabelle 1.1 Die einzelnen Kapitel dieses Buches und ihre Relevanz für die verschiedenen Ausbildungsberufe

Studiengänge

Die Studiengänge im Bereich der Informationstechnik sind wesentlich gründlicher, dafür aber auch viel theoretischer orientiert als die betrieblichen Ausbildungsgänge. Neben den hier behandelten Studienfächern der Informatik gibt es übrigens auch angrenzende Fachgebiete wie Telematik, Elektrotechnik, Mechatronik oder angewandte Mathematik.¹

Das allgemeinste Spektrum bietet das Studium der *Informatik*. Das Grundstudium umfasst Einführungen in alle vier Disziplinen des Faches (theoretische, technische, praktische und angewandte Informatik). Im Hauptstudium ist dagegen eine Spezialisierung auf einen bestimmten Bereich innerhalb eines dieser Zweige vorgesehen. Denkbar wären etwa Betriebssysteme aus dem Gebiet der praktischen Informatik oder Datenbanken, die zur angewandten Informatik zählen.

Informatik

¹ Für Fachinformatiker Anwendungsentwicklung ist das gesamte Kapitel relevant, für die anderen Ausbildungsberufe dagegen vor allem die Ausführungen zum Projektmanagement.

Das Informatikstudium wird sowohl an Universitäten als auch an Fachhochschulen angeboten. Naturgemäß ist Letzteres etwas praktischer orientiert; das Lernen erfolgt besonders im Grundstudium schulähnlicher. In beiden Lehreinrichtungen überwiegen seit wenigen Jahren die internationalen Abschlüsse Bachelor und Master; die zuvor dominierenden klassischen Diplomabschlüsse laufen zurzeit aus.

Nebenfächer Wie in jedem mathematisch-naturwissenschaftlichen Studium müssen auch bei der Informatik Nebenfächer belegt werden. Die Auswahl ist so vielfältig wie die Anwendungsbereiche der Informationstechnik: Für theoretischer ausgerichtete Forschungen ist etwa die Mathematik interessant; ein Fach aus dem Bereich der Wirtschaftswissenschaften eröffnet den Zugang zur Entwicklung kaufmännischer Systeme und Anwendungen. In diversen Bereichen der angewandten Informatik könnten auch Fächer wie Biologie oder Physik von Interesse sein. Spannend ist auch die Kombination mit philologischen Fächern, besonders mit den Sprachwissenschaften oder der philosophischen Logik.

Umgekehrt kann Informatik auch für Studiengänge in allen genannten Fächern als Nebenfach belegt werden.

Spezielle Studiengänge Einen Schritt weiter als die Wahl eines bestimmten Nebenfaches gehen gezielte Kombinationsstudienfächer. Die wichtigsten von ihnen sind folgende:

- ▶ *Wirtschaftsinformatik*: Eine gezielte Kombination aus Themen der (insbesondere angewandten) Informatik und der Betriebswirtschaftslehre. Wirtschaftsinformatiker können in allen Bereichen der kaufmännischen IT-Anwendung arbeiten, vor allem in Banken, Versicherungen oder den Fachabteilungen großer Unternehmen, aber auch in der Entwicklung kaufmännischer Softwarelösungen.
- ▶ *Medieninformatik*: Dieser Studiengang stellt den Bereich der multimedialen IT-Anwendung in den Mittelpunkt. Neben den Grundlagen der Informatik lernen die Studierenden vor allem den praktischen IT-Einsatz kennen: Grafik- und Bildbearbeitung, Audio- und Videoschnitt und die Erstellung multimedialer Präsentationen. Von allen IT-Studiengängen erfordert dieser das größte kreative Potential.
- ▶ *Bioinformatik*: Die Forschungsergebnisse der modernen Biologie, besonders der Mikrobiologie und der Gentechnik, könnten ohne umfangreiche IT-Unterstützung nicht ausgewertet werden. Bioinformatiker bilden die Schnittstelle zwischen Biologie und Informationstechnik; sie erstellen die Anwendungen zur Analyse biologischer Forschungen, etwa der Sequenzanalyse von Genen. Die Bioinformatik ist möglicherweise der wachstumsstärkste Sektor in der gesamten IT-Landschaft.
- ▶ *Medizinische Informatik*: Auch in der modernen Medizin spielt Computertechnik eine überaus wichtige Rolle – von der Verwaltung der Patienten- und Diagnosedaten über die automatisierte Analyse von Laborproben bis hin zum halbautonom arbeitenden OP-Roboter ist der Praxis- und Klinikalltag ohne IT nicht mehr denkbar. In der medi-

zistischen Informatik erwerben die Studierenden sowohl fundierte medizinische als auch wichtige informationstechnische Fachkenntnisse, so dass sie in den verschiedensten Bereichen der Medizintechnik und -praxis arbeiten können.

Die Tiefe des für ein Studium erforderlichen Wissens geht weit über das hinaus, was ein einzelnes allgemeines Grundlagenbuch wie das vorliegende leisten kann. Dennoch kann dieses Buch auch im Studium ein unentbehrlicher Begleiter sein: Hier können Sie die Grundlagen der Theorie und Praxis der unterschiedlichen Themen schnell und übersichtlich nachschlagen – als letzte Absicherung vor der entscheidenden Fachklausur oder auch als Anleser für die Veranstaltungsplanung des kommenden Semesters.

Studium und
dieses Buch

1.2 Die Geschichte der Rechenmaschinen und Computer

In diesem Buch ist von Computern die Rede, und natürlich wissen Sie ganz genau, was das ist. Wenn Sie allerdings den Versuch machen sollten, einen Computer mit allen Dimensionen seiner heutigen Möglichkeiten griffig zu definieren – wie würde diese Definition lauten? Vergangene Generationen sprachen etwa von *programmgesteuerten Rechenautomaten*, und gerade Fachleute scheinen noch heute lieber »Rechner« als »Computer« zu dieser Maschine zu sagen.

Aber ist Rechnen heutzutage die wichtigste Aufgabe der Computer? Es scheinen schließlich mehr Leute einen Webbrowser zu verwenden als die Tabellenkalkulation Excel, und das Bildbearbeitungsprogramm Photoshop ist erheblich populärer als spezielle Mathematiksoftware wie Mathematica oder Maple.

Sind
Computer nur
Rechner?

Trotzdem ist ein Computer ein Gerät, das Probleme durch Berechnungen löst: Er kann nur diejenigen Sachverhalte »verstehen«, die man in Form von Zahlen und mathematischen Formeln darstellen kann. Dass es sich dabei heute auch um Bilder, Töne, Animationen, 3D-Welten oder Filme handeln kann, liegt einfach an der enormen Rechengeschwindigkeit und Kapazität moderner Rechner.

Sehen Sie sich den Begriff »programmgesteuerter Rechenautomat« noch einmal genau an: Ein »Rechenautomat« ist ein Gerät, das automatisch etwas berechnet, so dass man dies nicht manuell erledigen muss. Das kann auch ein Taschenrechner oder sogar eine mechanische Rechenmaschine. Das Besondere, das ein Computer zu bieten hat, beschreibt der Begriff »programmgesteuert«. Ein Computerprogramm ist eine Abfolge von Rechenvorschriften, die aufeinander aufbauen können und Schritt für Schritt ausgeführt werden.

Mit anderen Worten ist ein Computer nicht nur ein Rechenautomat, sondern ein Algorithmen-Automat. Ein *Algorithmus*, benannt nach dem arabischen Mathematiker *Mohamed Ibn Musa Al Chwarismi*, ist eine Schritt-für-Schritt-Anleitung zum Lösen mathematischer Probleme. Jeder Computer versteht eine oder mehrere formale Sprachen, in denen

Algorithmen

man ihm solche Algorithmen einprogrammieren kann. Ist ein Algorithmus erst einmal im Computer gespeichert, kann er immer wieder mit anderen Daten ausgeführt werden. Betrachten Sie etwa den folgenden Algorithmus aus dem Alltagsleben, der die Internet-Nutzungskosten verschiedener Kunden eines Providers berechnet:

1. Eingabe Tarif: Flatrate oder minutenbasiert?
2. War es die Flatrate? Macht 19,99 €. Berechnung beendet.
3. Minutenbasiert: Eingabe der Minuten.
4. Multipliziere die Minuten mit 0,01 €.
5. Addiere die Grundgebühr von 2,49 € hinzu. Berechnung beendet.

Dieser einfache Algorithmus kann natürlich ohne weiteres von einem Menschen abgearbeitet werden. Er wird zwar langsamer rechnen als ein heutiger Computer, aber durchaus innerhalb einer annehmbaren Zeit damit fertig werden. Andererseits kann der Computer Millionen solcher Berechnungen in der Sekunde ausführen. Zwar wird er gewisse Leistungen des menschlichen Geistes wahrscheinlich niemals erreichen, aber schneller rechnen kann er allemal.

Darüber hinaus gibt es Algorithmen, an denen wir Menschen schlichtweg verzweifeln würden – oder hätten Sie Lust, jedes einzelne Pixel eines 10×10 cm großen hochauflösenden Bildes anhand der Farben seiner acht umgebenden Pixel neu zu berechnen, um das Bild zu vergrößern oder zu verkleinern? Ohne Sie entmutigen zu wollen: Ein solches Bild besteht, wenn es für den Druck geeignet sein soll, aus etwa 1.392.400 Pixeln – viel Spaß beim Rechnen!²

So macht der Computer sich selbst immer unentbehrlicher: Sobald Computersysteme eine gewisse Komplexität erreichen, wird eine Anwendung erfunden, die ohne sie nicht zu bearbeiten wäre. Um diese Anwendung dann schneller und effizienter ausführen zu können, entsteht die nächste Computergeneration, für die dann wieder neue Anwendungsgebiete gefunden werden. Auf diese Weise ist der Fortschritt in der Computerentwicklung nicht aufzuhalten, und was auch immer als unüberwindbare Grenze galt, wurde von cleveren Ingenieuren irgendwann überschritten. Der vorliegende Abschnitt versucht, diesen Weg ein wenig konkreter zu beschreiben.

1.2.1 Die Vorgeschichte

Seit Menschen überhaupt sesshaft in größeren Gemeinschaften zusammenleben, sind sie zum Rechnen gezwungen, um diese Gemeinschaften zu organisieren. Zu diesem Zweck wurden im Laufe der Zeit immer komplexere und ausgeklügeltere Geräte erfunden.

² Zugrunde gelegt wurde hier die gängige Druckauflösung von 300 Pixeln/Inch, was etwa 118 Pixeln/cm entspricht (1 Inch = 2,54 cm).

Das erste Rechenhilfsmittel in der Geschichte war die Rechentafel oder der Abakus. Ähnliche Geräte wurden unabhängig voneinander zum Beispiel in China und im alten Rom entwickelt. Es handelte sich um eine Tafel mit verschiebbaren Steinen, die in mehreren Spalten angeordnet waren. Das Ganze ähnelte den heutigen kugelbestückten Rechenhilfen, wie sie etwa in Grundschulen eingesetzt werden. Interessant ist, dass die Römer gar nicht erst versuchten, ihr recht umständliches Zahlensystem auf den Abakus zu übertragen, sondern ein modernes Stellenwertsystem darauf verwendeten.

Überhaupt kann die Erfindung der Stellenwertsysteme zum Schreiben von Zahlen als eine der wichtigsten Errungenschaften auf dem Weg zum Computer betrachtet werden: Die brillante Idee, den Wert einer Ziffer von ihrer Position innerhalb der ganzen Zahl abhängig zu machen, hatten die Inder; die heutigen »arabischen« Zahlen wurden in Indien erfunden und später von den Arabern – erstmals durch den bereits genannten Al Chwarismi im 9. Jahrhundert – übernommen. Die wichtigste Erfindung überhaupt ist in diesem Zusammenhang die Null. Die indischen Mathematiker notierten sie zunächst als Punkt und dann bis heute als Kreis. Ohne die Null ist es nicht möglich, den Wert einer einzelnen Ziffer zu vervielfachen: Auch wenn sie für sich allein keinen mathematischen Wert besitzt, werden Sie zugeben, dass 2.000 etwas völlig anderes bedeutet als 2.

Stellenwert-
systeme

Die Erfindung der Stellenwertsysteme war der Ausgangspunkt für die Konstruktion immer ausgefeilterer mechanischer Rechenmaschinen: Im 17. Jahrhundert konstruierte der französische Mathematiker und Philosoph *Blaise Pascal* eine Addiermaschine, 1722 erfand *Gottfried Wilhelm Leibniz* eine komplexere Maschine, die bereits sämtliche Grundrechenarten beherrschte. Solche mechanischen Rechenapparaturen arbeiteten mit einem komplexen Gefüge aus Zahnrädern und ähnlichen Bauteilen, die aus der Uhrmacherei übernommen wurden.

Erst im 19. Jahrhundert entstand die Idee zu einer Apparatur, die mit heutigen Computern vergleichbar ist. Der britische Wissenschaftler *Charles Babbage* hatte bereits eine klassische mechanische Rechenmaschine namens *Differential Engine* gebaut und plante um 1850 den Bau eines weiteren mechanischen Geräts namens *Analytical Engine*. Es sollte der erste frei programmierbare Rechenautomat der Welt werden. Ohne es zu ahnen, nahm Babbage einige der wichtigsten Designprinzipien des Computers vorweg: Das Gerät sollte aus einem Rechenwerk namens »mill«, einem Programm- und Datenspeicher und einer Ein- und Ausgabeeinheit bestehen. Auch wenn spätere Rekonstruktionen auf der Basis von Babbages Aufzeichnungen ergaben, dass die Maschine funktioniert hätte, war Babbage selbst nicht in der Lage, sie fertigzustellen, weil die damalige Mechanik keine Bauteile in ausreichender Präzision liefern konnte.

Babbages
Analytical
Engine

Interessant ist dagegen, dass es Grundideen einer Programmiersprache für die *Analytical Engine* gab. Babbage engagierte *Ada Lovelace*, die Tochter des berühmten Dichters Lord Byron, für die Entwicklung einer solchen Sprache. Ihr zu Ehren wurde eine in den 1970er Jahren entwickelte Programmiersprache *Ada* genannt.

Die Zuse Z3 Während die Rechenmaschinen in der zweiten Hälfte des 19. Jahrhunderts immer leistungsfähiger und komplexer wurden, beachtete dennoch niemand den Gedanken der Programmierbarkeit. Alle Rechenmaschinen bis etwa 1940, auch die elektrischen, waren auf die Berechnung einzelner eingegebener Rechenaufgaben beschränkt. Erst um diese Zeit erfand der Bauingenieur *Konrad Zuse* in Berlin programmierbare elektromechanische Rechenautomaten. Sein dritter Versuch, die Z3, funktionierte tatsächlich. Die Maschine arbeitete mit Relais, elektromagnetischen Schaltern aus der Telefontechnik. Sie verwendete eine binäre Fließkommaarithmetik mit einer Wortbreite von 22 Bit: 1 Bit für das Vorzeichen, 14 Bit für die Mantisse und 7 Bit für den Exponenten. Der Hauptspeicher besaß eine Kapazität von 64 Maschinenwörtern, also 64×22 Bit. Programmiert wurde die Maschine (und einige ihrer Nachfolger) in einer von Zuse entwickelten Sprache namens *Plankalkül*.

Elektrizität, Elektromechanik und Elektronik

Bereits Anfang des 20. Jahrhunderts wurden die ersten elektrischen Rechenmaschinen konstruiert. Die Rechner, die Konrad Zuse ab den 1930er Jahren baute, waren elektromechanisch. In den 1940er Jahren begann man in den USA und Großbritannien mit der Entwicklung elektronischer Rechner.

In diesem Zusammenhang ist es wichtig, den Unterschied zwischen Elektrizität, Elektromechanik und Elektronik herauszustellen:

- ▶ Ein *elektrisches* Gerät ist jede Maschine, bei der mechanische Bauteile durch elektrischen Strom angetrieben werden.
- ▶ *Elektromechanisch* werden alle Schaltelemente genannt, die durch einen Elektromagneten gesteuert werden, beispielsweise die Relais, die Zuse für seine ersten Computer verwendete. Mit anderen Worten geschieht die Steuerung des Geräts durch elektrischen Strom, allerdings indirekt durch mechanische beziehungsweise magnetische Bauteile.
- ▶ *Elektronische Geräte* schließlich werden allein durch elektrischen Strom gesteuert – Bauteile wie die klassische Elektronenröhre oder der neuere Transistor dienen als elektrisch gesteuerte Schalter ohne mechanische Teile.

1.2.2 Die Entwicklung der elektronischen Rechner

Computer im heutigen Sinne sind alle programmierbaren Rechner, die elektronisch arbeiten. Die elektronischen Computer lassen sich grob in vier Generationen einteilen:

- ▶ 1. Generation: *Röhrenrechner*
Ab den 1940er Jahren wurden Rechner auf der Basis von *Elektronenröhren* entwickelt.
- ▶ 2. Generation: *Transistorrechner*
Ab den 1950er Jahren wurden die teuren, stromhungrigen und störungsanfälligen Röhren durch *Transistoren* ersetzt.

► 3. Generation: *Rechner mit integrierten Schaltkreisen*

In den 1960er Jahren gelang es, durch fotolithografische Verfahren große Mengen von Transistorschaltungen auf Halbleiterplatten unterzubringen – es begann die Entwicklung der *integrierten Schaltkreise* (englisch: *integrated circuits*, abgekürzt ICs, auch »Chips« genannt).

► 4. Generation: *Rechner mit Mikroprozessor*

Die frühen ICs waren fest verdrahtet – sie konnten nur eine einzelne, festgelegte Aufgabe erfüllen. Solche Chips für spezielle Anwendungszwecke gibt es noch heute. Anfang der 1970er Jahre wurden zusätzlich frei programmierbare ICs entwickelt, die man *Mikroprozessoren* nannte. Es dauerte allerdings noch über fünf Jahre, bis die ersten Computer mit diesen Prozessoren konstruiert wurden.

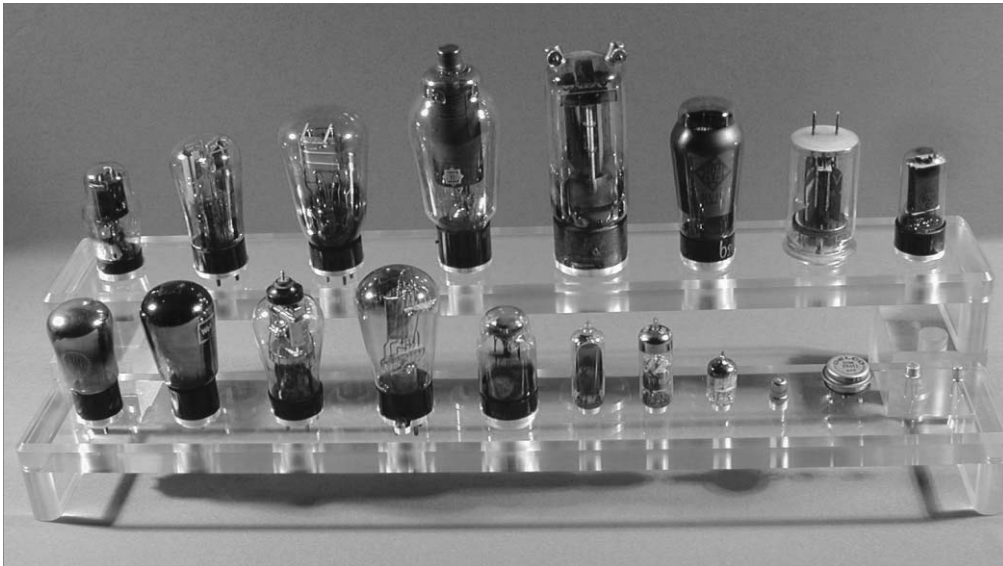


Abbildung 1.1 Eine kleine Auswahl verschiedener Elektronenröhren

Röhrenrechner

Auf der Grundlage der Forschungen von *John von Neumann*, der das theoretische Modell eines Computers formulierte, wurden in den USA Geräte wie Harvard Mark I und Mark II oder der berühmte ENIAC gebaut. Diese erste Generation elektronischer Computer arbeitete mit *Elektronenröhren* als Schaltelementen. Diese mit den Glühlampen verwandten Vakuumbauteile wurden ab Ende des 19. Jahrhunderts entwickelt und dienten verschiedenen Zwecken – denken Sie beispielsweise an das Röhrenradio, in dem die Elektronenröhre als Verstärkerelement eingesetzt wird. Abbildung 1.1 zeigt eine Auswahl verschiedener Elektronenröhren, die zwischen 1927 und 1960 hergestellt wurden. Das Bild stammt übrigens aus dem virtuellen Physikmuseum der Universität Innsbruck (<http://>

exphys.uibk.ac.at/museum/); Herr Professor Denoth stellte es mir freundlicherweise zur Verfügung. Auf der angegebenen Website finden Sie noch viele weitere interessante Bilder und Informationen zur Physikgeschichte, darunter auch weitere Röhren, die ab 1870 entwickelt wurden.

Das für die Computertechnik interessanteste Röhrenmodell war die *Triode*, die mit ihren drei Anschlüssen die früheste Verwirklichung eines rein elektronischen Schalters darstellt: Einer der drei Anschlüsse dient der Steuerung; wenn dort Spannung anliegt, fließt Strom durch die beiden anderen Anschlüsse.

Programmiert wurden die meisten Röhrenrechner durch Schalter und Steckverbindungen an großen Schalttafeln. Die einzige Möglichkeit, ein bestimmtes Programm für die spätere erneute Ausführung zu »speichern«, bestand darin, den Zustand der Schalttafel aufzumalen oder zu fotografieren. So dauerte es oft mehrere Stunden, den Computer in die Lage zu versetzen, komplexe Aufgaben zu erfüllen.

Erst allmählich begann man mit der Verwendung von *Lochkarten* zur Programm- und Dateneingabe. Die Lochkarte selbst wurde im 19. Jahrhundert erfunden, um mechanische Webstühle zu steuern. Der Ingenieur *Hermann Hollerith*, ein Mitbegründer der späteren IBM, setzte sie um 1900 zur Speicherung von Daten einer US-Volkszählung ein, was die Dauer der Ergebnisberechnung mit Hilfe von Rechenmaschinen von den erwarteten Jahren auf wenige Wochen reduzierte.

Transistorrechner

Röhrenrechner hatten einige gravierende Nachteile: Sie waren zu groß, zu stromhungrig und wegen der gewaltigen Hitzeentwicklung zu störanfällig. Aus diesen Gründen wurde bald der 1947 erfundene *Transistor* für die Computerentwicklung eingesetzt. Transistoren sind *Halbleiterbauteile*, die prinzipiell dieselben Schaltaufgaben erledigen können wie die verschiedenen Arten von Röhren, aber kleiner, billiger und weniger stromhungrig sind. Der Begriff »Halbleiter«, der im Zusammenhang mit Computern immer wieder genannt wird, bezeichnet übrigens ein Material, dessen elektrische Leitungsfähigkeit etwa in der Mitte zwischen den Leitern (vielen Metallen) und Isolatoren liegt. Das beliebteste chemische Element zur Fertigung von Halbleiterbauteilen ist Silizium, was der berühmten Gegend in Kalifornien, in der sich in den 1960er Jahren zahlreiche Elektronikfirmen ansiedelten, den Namen »Silicon Valley« eingebracht hat.

Miniaturisierung durch Transistoren

Erst durch die Einführung des Transistors gelangte die Elektronik zu ihrer vollen Blüte. Dies zeigte sich vor allem an den Transistorradios, die ab den 1950er Jahren zuhauf verkauft wurden. Auch für die entstehende Computerindustrie ergaben sich neue Impulse: Durch die Transistortechnik ließen sich kleinere, leistungsfähigere und weniger störanfällige Rechner konstruieren. Natürlich ist »klein« und »leistungsfähig« relativ. Angesichts eines heutigen PCs oder gar Notebooks waren auch die Transistorrechner monströs, mindestens jedoch so groß wie eine ganze Reihe gewaltiger Kleiderschränke.

Auch die Transistorrechner wurden anfangs vor allem durch Lochkarten gesteuert. Mitte der 1960er Jahre begann man allerdings mit der Entwicklung von *Terminals* für den direkten Dialog mit dem Computer. Ein Programmierer konnte über eine Tastatur Befehle eingeben und ein unmittelbares Feedback erhalten, anfangs über einen Fernschreiber-Endlosdrucker, später über einen Monitor. Die Geschichte der Steuerung von Computern wird in Kapitel 5, »Betriebssystemgrundlagen«, genauer behandelt.

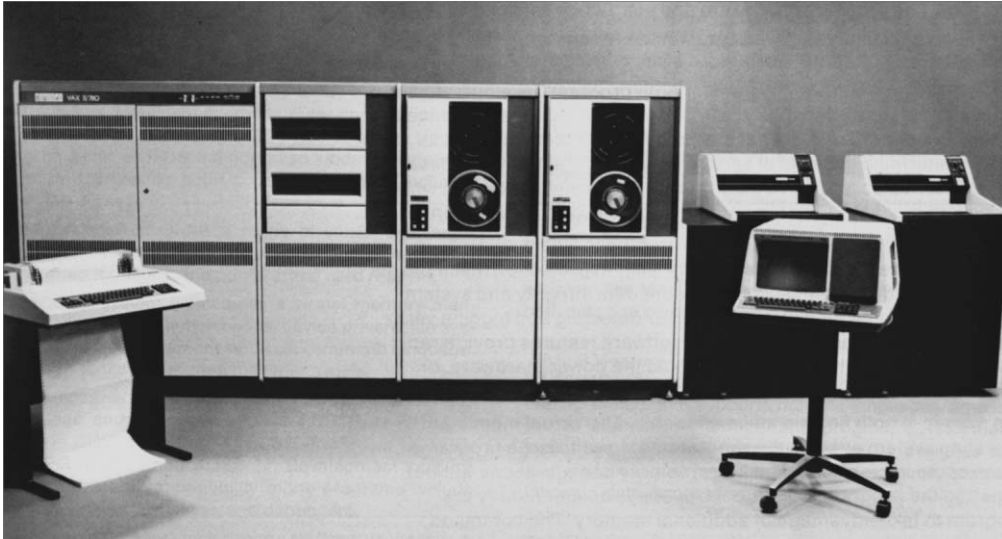


Abbildung 1.2 Eine VAX von Digital Equipment mit Terminals verschiedener Generationen;
Quelle: Wikipedia

Computer mit integrierten Schaltkreisen

Der Übergang von Transistorrechnern zu Computern mit ICs verlief unspektakulär und allmählich. Bereits in den frühen 1970er Jahren waren deshalb Computer in verschiedenen Größen verfügbar: Die Großrechner oder Mainframes – vor allem von IBM produziert – bildeten die Rechenzentren von großen Behörden, Versicherungskonzernen oder Universitäten. Daneben kamen die sogenannten Kleincomputer auf (noch immer größer als die meisten Kleiderschränke); einer der führenden Hersteller war die Digital Equipment Corporation (DEC). Die Kleincomputer waren zwar nicht ganz so leistungsfähig wie Mainframes, dafür aber flexibler und sogar – mit entsprechend großen LKWs – transportfähig. Besonders wichtige Beispiele für Kleincomputer sind die Geräte der DEC-PDP-Baureihe, die untrennbar mit der Geschichte des Betriebssystems Unix verknüpft ist.

Durch die Verwendung von ICs kam eine weitere Klasse von Geräten hinzu: die sogenannten Minicomputer. Sie waren etwa so groß wie eine größere Kommode und verhältnismäßig leicht, aber durch die fortschreitende Miniaturisierung nicht viel weniger leis-

tungsfähig als Kleincomputer. Ein herausragendes Beispiel für den Minicomputer sind spätere Ausgaben der ebenfalls von DEC stammenden VAX-Baureihe (ein älteres Modell sehen Sie in Abbildung 1.2).

Für Klein- und Minicomputer wurden übrigens die ersten standardisierten Betriebssysteme und Anwendungsprogramme entwickelt. Dies erforderte die Entwicklung leicht kopierbarer Speichermedien. Ein wichtiger Schritt in diese Richtung war die Verwendung von Magnetbandspulen als Datenspeicher; ihr Aussehen und ihre Funktionsweise entsprachen den etwa zur selben Zeit verbreiteten Tonbändern.

Mikrocomputer

1971 wurde der erste programmierbare Mikrochip entwickelt, genannt *Mikroprozessor*. Allgemein wird der Intel 4004 als erster Prozessor betrachtet; später stellte sich aber heraus, dass ein anderes Unternehmen bereits einige Monate vor Intel einen ähnlichen Chip entwickelt hatte. Allerdings wussten die Intel-Entwickler nichts davon.

Der 4004 war ein 4-Bit-Mikroprozessor. Er konnte also Informationen verarbeiten, die aus einer Abfolge von vier Einsen oder Nullen bestanden. Mit dieser *Wortbreite* lassen sich sechzehn verschiedene Werte darstellen, zum Beispiel die Zahlen 0 bis 15. Der Prozessor verstand verschiedene Arten von grundlegenden Befehlen: Er beherrschte arithmetische Operationen, also Berechnungen in den Grundrechenarten, und konnte logische Verknüpfungen und Vergleiche durchführen und auf der Basis ihrer Ergebnisse die »Entscheidung« treffen, an einer anderen Stelle im Programm fortzufahren. Die mathematischen Grundlagen und ihre Verwirklichung durch elektronische Bauteile werden im nächsten Kapitel, »Mathematische und technische Grundlagen«, vorgestellt.

Die ersten Personal Computer

Intel unterschätzte zu Anfang die Möglichkeiten des Mikroprozessors; er wurde in Rechenmaschinen und Ampelanlagen eingebaut, aber nicht in einen Computer. Erst 1975 baute die Rechenmaschinenfirma MITS einen einfachen Mikrocomputer-Bausatz, den Altair 8800. Er war mit einer Reihe von DIP-Schaltern für die Eingabe der einzelnen Bits und einer Reihe von Leuchtdioden zur Anzeige der Ergebnisse ausgestattet. Der verwendete Mikroprozessor war der Intel 8080, der bereits 8 Bit verarbeiten konnte. Für dieses zunächst nicht besonders nützliche Gerät entwarfen Bastler Schnittstellen für Monitor und Tastatur, und ein junger Programmierenthusiast schrieb einen Interpreter (zeilenweisen Übersetzer) für die einfache Großrechner-Programmiersprache BASIC, der auf dem Gerät lief. Der Programmierer war *Bill Gates*. Die Entwicklung von Programmiersprachen-Paketen für die beginnende Personal- und Homecomputer-Industrie war das erste Geschäftsfeld seiner 1976 gegründeten Firma *Microsoft*.

Große Verdienste im Zusammenhang mit der PC-Entwicklung haben übrigens die Forscherinnen und Forscher des *Xerox PARC* (der Name PARC steht für »Palo Alto Research Center«). Diese Forschungseinrichtung in Kalifornien hat eine interessante Geschichte: Zu

Beginn der 1970er Jahre bekamen die Manager der renommierten Papier-, Druckmaschinen- und Kopiererfirma Xerox langsam Angst, denn sie sahen angesichts der schnellen Weiterentwicklung der Computer das Schreckgespenst des »papierlosen Büros« auf sich zukommen. Natürlich weiß man es heute besser; der Papierverbrauch in den Büros hat sich seitdem vervielfacht, weil es einfach und billig ist, Dokumente mal eben auszudrucken und dann auf Papier zu kontrollieren.

Aber damals konnte das natürlich niemand ahnen. Und so beschloss die Firma Xerox: Wenn die Leute im Büro kein Papier mehr brauchen, dann müssen wir ihnen eben das liefern, was sie stattdessen dort benötigen. Dies war der Grundgedanke für die Einrichtung dieses Forscherparadieses. Einige der brilliantesten Köpfe der Computertechnik und Informatik wurden eingeladen, jeweils fünf Jahre lang, mit fast unbegrenzten finanziellen Mitteln ausgestattet, zu erforschen, was immer sie wollten. So kam es, dass hier schon Mitte der 1970er Jahre Neuerungen wie eine grafische Benutzeroberfläche, die Dokumentenbearbeitung nach dem Prinzip WYSIWYG (»What You See Is What You Get«) oder ein Laserdrucker entwickelt wurden.

Im Prinzip existierte dort ein marktreifes Personal-Computer-System, dessen technische Fähigkeiten erst über 15 Jahre später zum allgemeinen Standard wurden. Der einzige Nachteil dieses Systems, des *Alto*, war sein außerordentlich hoher Preis von ca. 100.000 US-Dollar. Jedenfalls wurde Xerox PARC so auch die Wiege des lokalen Netzwerkes: Die Vision war, dass jeder Mitarbeiter eines Unternehmens ein solches Gerät auf seinem Schreibtisch stehen hat und der Austausch von Daten ganz einfach ist. So entstand Ethernet, diejenige Form des lokalen Netzes, die heute – neben dem drahtlosen WLAN – am häufigsten genutzt wird.

Sollten Sie sich jetzt wundern, warum keine dieser bahnbrechenden Entwicklungen heute den Namen Xerox trägt, so liegt dies daran, dass die Führungsetage des Unternehmens im fernen New York deren Brillanz und zukunftsweisende Eigenschaften nicht erkannte. So wurden die Entwicklungsergebnisse der meisten Forscher von diesen mitgenommen, jahrelang in die Schublade gelegt und später oftmals zur Gründung anderer Firmen verwendet.

John Warnock etwa, eine Zeit lang Forschungsleiter von PARC, gründete die Firma *Adobe* und konzentrierte sich besonders auf die Bereiche WYSIWYG und den Laserdrucker. Andere Ergebnisse wurden freimütig an Fremde weitergegeben, so etwa die Geheimnisse der grafischen Oberfläche an den *Apple*-Mitbegründer *Steve Jobs*, der darin die Grundlage für den Computer der Zukunft erkannte. Zusammen mit dem Bastelgenie *Steve Wozniak* entwarf er den ersten verbreiteten »richtigen« Personal Computer auf Mikroprozessorbasis: *Apple II* hieß das 1977 entstandene Gerät, das sich in seinen verschiedenen Versionen bis 1984 Millionen Mal verkaufte. Dieser Computer definierte für fast zehn Jahre, was ein Homecomputer haben musste: einen eingebauten BASIC-Interpreter, eine

fest in das Gerät integrierte Tastatur, gewöhnliche Audiokassetten und später Disketten als Datenspeicher sowie einen Anschluss für die Bildausgabe auf einem gewöhnlichen Fernseher (die privaten Anwender oder kleine Unternehmen konnten sich nicht auch noch einen Monitor leisten, zumal es diesen zu dem Zeitpunkt nur einfarbig gegeben hätte).

Der IBM-PC 1981 stieg auch der Großrechner-Multi *IBM* in das Geschäft mit Personal Computern ein. Es war zwar untypisch für dieses Unternehmen, nicht alle Bauteile eines Computers selbst zu entwickeln, aber aus Gründen des Zeitdrucks kauften sie sich die Bestandteile ihres Geräts auf dem freien Markt zusammen. Der IBM-PC war nicht unbedingt besser als seine Vorgänger, aber allein der gute Name des Herstellers schuf das Vertrauen der Wirtschaft und der Industrie, dass PCs eine solide und zukunftssträchtige Technologie seien.

Einige Jahre war IBM Marktführer im Bereich der *Personal Computer*. Da jedoch immer mehr Hersteller kompatible Nachbauten auf den Markt brachten, die zunehmend billiger wurden, verlor der Konzern irgendwann den ersten Platz. Zudem zerstritten sie sich 1990 mit ihrem Betriebssystemlieferanten Microsoft. Seitdem wurde nicht mehr von IBM definiert, wie ein »richtiger PC« auszusehen hat, sondern von Intel und Microsoft – wegen der Kombination aus Intel-Prozessoren und Microsoft-Windows-Betriebssystemen wurden PCs in den 1990er Jahren mitunter als »WinTel«-PCs bezeichnet.

8-Bit-Home-computer Parallel blühte in der ersten Hälfte der 1980er Jahre der Markt mit reinen Homecomputern ohne professionelle Ambitionen. Interessant war auf diesem Gebiet zunächst der ZX81 (siehe Abbildung 1.3), den der Brite *Clive Sinclair* entwickelt hatte. Trotz seiner bescheidenen Ausstattung mit 1 KByte Arbeitsspeicher, Schwarzweißgrafik und Folientastatur verkaufte sich das Gerät vorzüglich, weil es der erste Computer war, der weniger als 100 £ (damals etwa 400 DM, also ca. 200 EUR) kostete.



Abbildung 1.3 Der Sinclair ZX81 (Quelle: Wikipedia)

Der beliebteste Homecomputer der 8-Bit-Generation wurde allerdings der 1983 auf den Markt gebrachte *Commodore C64*. Eines der wichtigsten Verkaufsargumente war die reichhaltige Auswahl an Software, vor allem Computerspielen. In Abbildung 1.4 wird dieses Gerät gezeigt. Ein nicht ganz so beliebtes, aber durchaus konkurrenzfähiges Gerät war der *Atari 800 XL*; für diesen gab es zwar weniger Spiele, aber dafür war er besser programmierbar. Die Unterschiede zwischen diesen beiden Geräten waren allerdings relativ gering. Beide waren mit dem 8-Bit-Prozessor 6502 von MosTek ausgestattet, verfügten über 64 KByte RAM, eine Grafik mit 320 × 240 Pixeln und 16 Farben (Atari bei einer geringeren Auflösung sogar 256) sowie mehrstimmigen Synthesizer-Sound, der an das angeschlossene Fernsehgerät übertragen wurde. Obwohl die beiden Rechner sich also technisch sehr ähnlich waren, tobten zu jener Zeit »Systemkriege«, die sich ohne weiteres mit den heutigen PC/Macintosh- oder Linux/Windows-»Glaubenskriegen« vergleichen lassen.³



Abbildung 1.4 Commodore C64, der berühmte »Brotkasten« (Quelle: Wikipedia)

Die nächste Generation von Homecomputern, die in der zweiten Hälfte der 1980er Jahre erschien, basierte auf dem 16-Bit-Prozessor 68000 von Motorola und seinen Nachfolgern und war mit »richtigen« Betriebssystemen mit grafischer Benutzeroberfläche ausgestattet. Wieder stritten sich Commodore und Atari um den ersten Platz; auch die verschiedenen Modelle von Commodore Amiga und Atari ST waren mit vergleichbaren Features ausgestattet. Schon früher, nämlich 1984, war der ähnlich gestaltete *Apple Macintosh* erschienen; allerdings war er nicht für Heimanwender konzipiert.

16-Bit-Home-
computer

³ Um mich selbst zu outen: Mein eigener erster Rechner war ein ZX81, dem ein Atari 800 XL folgte. Später hatte ich dann sehr lange nur noch IBM-kompatible PCs. Inzwischen sind noch ein MacBook und ein Netbook (Samsung NC10) dabei. Sämtliche in diesem Buch beschriebene Betriebssysteme sind auf diesen Rechnern im Einsatz und kommen prima miteinander aus.

Bemerkenswert ist, dass Standard-PCs erst viele Jahre später mit Multimedia-Fähigkeiten ausgestattet wurden, die auch nur ansatzweise mit Geräten wie Amiga oder ST vergleichbar waren. Noch heute sind diese Rechner bei manchen Künstlern oder Musikern beliebt, und zumindest im Fall des Amigas gibt es immer wieder Gerüchte um eine Wiederbelebung durch verschiedene Firmen.

32-Bit-Rechner

Zu Beginn der 1990er Jahre wurden die Home- und Personal Computer nach und nach mit 32-Bit-Prozessoren ausgestattet. Den Anfang machte der IBM-PC/AT mit Intels 80386-Prozessor, dem wie bei den Vorgängermodellen wieder zahlreiche kompatible Nachbauten folgten. Apple ersetzte dagegen die Motorola 680xx-Prozessoren der frühen Macs durch Power-PC-Prozessoren, was die Familie der PowerMacs einleitete. Bemerkenswerterweise wurde die Macintosh-Prozessorbasis ab 2005 nochmals gewechselt; statt Power PC werden nun Intel-Multicore-Prozessoren eingesetzt.

Die Leistungssteigerung der PC- und Mac-Prozessoren sorgte dafür, dass diese Rechner – besonders die günstigeren IBM-kompatiblen PCs – die klassischen Homecomputer verdrängten und nach den Büros auch den Heimbereich eroberten. Dazu wurden insbesondere im Bereich der Multimedia-Fähigkeiten große Fortschritte erzielt. Ein gewisses Problem bestand allerdings darin, dass die Betriebssysteme der frühen 1990er Jahre nicht für die modernen 32-Bit-Architekturen ausgelegt waren. 1995 brachte Microsoft mit Windows 95 ein massentaugliches Kompromiss-System auf den Markt, das zwar noch immer einen MS-DOS-Unterbau besaß, aber innerhalb der grafischen Oberfläche echtes 32-Bit-Multitasking beherrschte. Erst im Jahr 2001 verabschiedete sich Microsoft mit Windows XP ganz von der MS-DOS-Vergangenheit im Heimsektor; im selben Jahr erschien die erste Version von Apples Neuentwicklung Mac OS X.

Beide Systemwelten hatten übrigens viel weniger Schwierigkeiten beim Wechsel von 32- auf die immer weiter verbreiteten 64-Bit-Architekturen, genauso wenig wie das freie, für viele Plattformen verfügbare Betriebssystem Linux. Näheres zur Geschichte der Betriebssysteme lesen Sie übrigens in Kapitel 5, »Betriebssystemgrundlagen«.

Ausblick in die nähere Zukunft

Offiziell zählen alle heutigen Computer zur vierten Generation, zu den Computern mit Mikroprozessoren. Selbst mächtige Servermaschinen oder Großrechner werden durch zahlreiche parallel betriebene Prozessoren realisiert. Eine offizielle fünfte Generation ist niemals ausgerufen worden.

Dennoch zeichnen sich zurzeit verschiedene Entwicklungen ab, die in Zukunft zu einem Wandel auf dem Computermarkt führen könnten. Die wichtigsten sind folgende:

- Es entstehen immer mehr spezielle mobile Geräte, die Teile der Funktion des universellen PCs übernehmen. Dazu gehören Mobiltelefone mit Internet- und Multimedia-Funktionen (*Smartphones*) wie Apples iPhone oder die zahlreichen Google-Android-

und Windows-Phone-Geräte, Tablets (iPad, Samsung Galaxy Tab & Co.), PDAs oder spezielle Geräte, die wie eine Armbanduhr oder ein Gürtel getragen werden können oder in Kleidungsstücke eingebaut sind (*Wearable Computers*). Im gleichen Maße wird das Internet, besonders in seiner drahtlosen Form, immer wichtiger.

- ▶ Über die klassische binäre Elektronik hinaus entstehen neue Ansätze für den Bau von Computern. Dazu gehören beispielsweise DNA-basierte »Bio-Rechner« oder Computer auf der Basis von *Lichtwellenleitern* – diese auch als *Glasfaserkabel* bezeichneten Leitungen transportieren Licht statt elektrischen Strom und kommen bereits seit langem zur Datenübertragung zum Einsatz (Kabelfernsehen, Netzwerke und so weiter). Ein weiteres interessantes Projekt ist der *Quantencomputer*: Da die Quantenmechanik besagt, dass ein Teilchen mehrere Zustände zur selben Zeit aufweisen kann, können sogenannte *QBits* (*Quanten-Bits*) codiert werden, die »1 und 0 gleichzeitig« enthalten – das Durchprobieren zahlreicher verschiedener Fälle muss nicht mehr nacheinander geschehen, sondern kann gleichzeitig erfolgen.
- ▶ Überraschend ruhig ist es dagegen in den letzten Jahren in der Öffentlichkeit um die »künstliche Intelligenz« geworden. Inzwischen hat sich die Forschung vorläufig von dem Gedanken verabschiedet, das komplexe menschliche Gehirn nachzuahmen. Stattdessen betreibt man erst einmal Grundlagenforschung und simuliert beispielsweise das Zusammenspiel weniger einzelner Nervenzellen in sogenannten *neuronalen Netzen*.

1.2.3 Entwicklung der Programmiersprachen

Damit ein Computer nützliche Aufgaben erledigen kann, muss er programmiert werden. Da in der kurzen Übersicht über die Computergeschichte bereits von Programmiersprachen die Rede war, folgt hier ein kurzer Abriss über deren Entwicklung.

Die Maschinensprache des Prozessors

Die einzige Sprache, die ein Mikroprozessor wirklich versteht, ist seine *Maschinensprache*. Sie besteht aus nichts weiter als Zahlen: Jeder Befehl, den der Prozessor »versteht«, besitzt einen bestimmten numerischen Code. Je nach Art des Befehls folgen auf die Befehlsnummer ein oder mehrere Argumente verschiedener Länge.

Ein Maschinenprogramm ist für Menschen so gut wie unlesbar und schon gar nicht schreibbar. Wenn Sie eine binäre Programmdatei mit einem Texteditor öffnen, werden die gespeicherten Zahlen als Zeichen interpretiert; es erscheint merkwürdiger Zeichensalat, an manchen Stellen unterbrochen von kleinen Textblöcken, wenn das Programm normalen Text enthält.

Angenommen, ein Programm enthält den folgenden (fiktiven) Befehl:

65 0 0 0 98

Bei dem Befehl 65 könnte es sich beispielsweise um die Anweisung handeln, einen bestimmten Wert auf einen Speicherstapel zu legen; der Wert wird als 32-Bit-Ganzzahl angegeben, hier 98. Im Texteditor sähe dies etwa so aus:

```
A    b
```

Das große A besitzt den Zeichencode 65, das kleine b den Code 98. Die drei Null-Bytes werden in manchen Editoren als merkwürdige Sonderzeichen, in anderen als Leerzeichen angezeigt. Ein Rückschluss auf die tatsächlichen Befehle ist so gut wie unmöglich.

Wenn Sie überhaupt jemals gezwungen sein sollten, Maschinensprachdateien von Hand zu modifizieren (zum Beispiel, um ein Computerspiel zu überlisten), verwenden Sie besser einen *Hex-Editor*, der die einzelnen Werte nicht nur als ASCII-Zeichen, sondern zusätzlich hexadezimal darstellt. Hier könnte das Ganze folgendermaßen aussehen:

```
41 00 00 00 52    A    b
```

Assembler – die »benutzerfreundliche« Maschinensprache

Um Maschinensprache halbwegs benutzbar zu machen, wurde der *Assembler* entwickelt. Statt die Befehle mit ihren tatsächlichen Zahlencodes zu schreiben, werden sie durch Kürzel dargestellt, die man sich mehr oder weniger gut merken kann – daher auch der Name »Mnemonics«. In der Regel werden diese Namen für die jeweiligen Assembler-Befehle unmittelbar vom Prozessorhersteller selbst festgelegt, um jegliches Chaos zu vermeiden.

Die Assembler-Sprache ist von Prozessor zu Prozessor völlig verschieden. Jede Prozessorarchitektur versteht ihre ganz eigenen Arten von Befehlen, die entsprechend unterschiedlich in Assembler umgesetzt werden.

Assembler ist sowohl der Name für diese vereinfachte Schreibweise der Maschinensprache als auch der Name für das Programm, das diese Sprache in die eigentliche Maschinensprache umsetzt (im Englischen wird die Sprache allerdings eher als *assembly language* bezeichnet).

Das Assembler-Programm führt gegenüber der eigentlichen Maschinensprache oft eine Reihe von Erleichterungen ein. Viele Assembler beherrschen etwa die Definition sogenannter *Makros*: Immer wiederkehrende Abfolgen von Befehlen erhalten einen eindeutigen Namen und können dann später unter diesem Namen aufgerufen werden.

Assembler – praktischer Einsatz

Assembler wird heutzutage kaum noch zur Programmierung verwendet, zumindest nicht zur Erstellung vollständiger Programme. Wichtige Ausnahmen sind folgende:

1. In Betriebssystemen sind einige der besonders hardwarenahen Kernroutinen im Assembler des jeweiligen Prozessors geschrieben, und zwar vor allem deswegen, damit der gesamte Rest des Systems so weit von der Hardware abstrahiert wird, dass er vollständig in einer höheren Sprache – meist C – geschrieben werden kann. Diese Vorge-

hensweise wurde in den 1970er Jahren bei der Implementierung von Unix entwickelt und gilt noch heute.

2. Auch Gerätetreiber, die zu den wichtigsten Bestandteilen der Betriebssysteme gehören, müssen manchmal in Assembler geschrieben werden.
3. Bestimmte Teile von Computerspielen werden hin und wieder in Assembler geschrieben. Spieleprogrammierern, besonders von schnellen 3D-Spielen, kommt es vor allem auf Geschwindigkeit an.
4. Besonders systemnahe Computerviren (Bootsekturviren, die den Startbereich eines Datenträgers infizieren, und Programmviren, die ausführbare Programme befallen) sind meistens vollständig in Assembler geschrieben.

Die ersten höheren Programmiersprachen (Fortran, Cobol, BASIC)

Als praktischer Ersatz für die maschinenorientierten Sprachen wurden seit Mitte der 1950er Jahre die problem- oder benutzerorientierten Programmiersprachen eingeführt. Ihr Vorrat an möglichen Befehlen und ihre Syntax orientierten sich eher an den Bedürfnissen der Programmierer als an denen des Rechners.

Diese Programmiersprachen müssen in die Maschinensprache des konkreten Prozessors übersetzt werden. Dazu wurden zwei grundlegende Vorgehensweisen entwickelt:

- ▶ Der *Compiler* erzeugt ein dauerhaft lauffähiges Maschinensprachprogramm und speichert es als ausführbares Programm (*Binary Executable*) ab.
- ▶ Der *Interpreter* übersetzt den Quellcode dagegen Zeile für Zeile; der Code wird also während der Ausführung (zur Laufzeit) übersetzt. Interpretierte Sprachen werden häufig auch als *Skriptsprachen* bezeichnet.

Die erste Generation der höheren Programmiersprachen war sehr einfach. Beispielsweise gab es noch keine echte Programmstrukturierung. Das ursprüngliche BASIC verwendete etwa Zeilennummern, zu denen gesprungen werden konnte; Fortran benutzte spezielle Sprungmarken.

Jede dieser frühen Sprachen hatte eine spezielle Ausrichtung oder einen besonderen Verwendungszweck:

- ▶ *Fortran*, entwickelt in den 1950er Jahren, ist die Abkürzung für »Formula Translator«. Es handelt sich um eine besonders im Hinblick auf mathematische Bedürfnisse geschriebene Sprache. Sie wird zum Teil auch heute noch von Ingenieuren oder Mathematikern gern verwendet.
- ▶ *Cobol* wurde ebenfalls in den 1950er Jahren entwickelt. Der Name ist die Abkürzung für »Common Business-oriented Language«; es handelt sich also um eine Sprache für kaufmännische Anwendungszwecke, für Handel und Wirtschaft. Cobol ist eine ziemlich »geschwätzige« Sprache. Es braucht relativ viele Wörter, um verhältnismäßig kurze

Anweisungen auszudrücken. Beispielsweise können Sie in fast jeder Programmiersprache folgendermaßen den Wert der Variablen `b` durch 7 teilen und das Ergebnis in einer zweiten Variablen namens `a` speichern:

```
a = b / 7
```

In Cobol wird dagegen folgende Anweisung verwendet:

```
DIVIDE B BY 7 GIVING A
```

- **BASIC** wurde 1960 am Dartmouth College entwickelt. Der Name steht für »Beginner's All-Purpose Symbolic Instruction Code«. Es handelt sich also um eine eher einfache Sprache für Anfänger. Die Sprache fand ab der zweiten Hälfte der 1970er Jahre, als die neu gegründete Firma Microsoft sie zum ersten Mal für Personal Computer anpasste, eine gewaltige Verbreitung. Fast jeder Homecomputer in den 1980er Jahren hatte irgendeine BASIC-Variante im ROM eingebaut.

Ein BASIC-
Beispiel

Zur Verdeutlichung sehen Sie hier ein kleines Beispielprogramm in »allgemeinem« BASIC, das fast jeder klassische BASIC-Interpreter verstehen würde:

```
10 PRINT "Wie heißt du?"
20 INPUT A$
30 PRINT A$; " ist ein interessanter Name."
40 PRINT "Noch mal (j/n)?"
50 INPUT J$
60 IF J$ = "j" THEN GOTO 10
```

Die einzelnen Programmzeilen bedeuten Folgendes:

- (10) Ausgabe des Textes »Wie heißt du?«.
- (20) Eingabe der Variablen `A$`, die durch das Dollarzeichen als String-Variable (Textinhalt) gekennzeichnet wird.
- (30) Der eingegebene Name wird ausgegeben, gefolgt von »ist ein interessanter Name.«.
- (40) Der Benutzer wird gefragt, ob er einen weiteren Durchgang wünscht.
- (50) Eingabe der Variablen `J$`.
- (60) Hat `J$` den Wert »j«, geht es weiter bei Zeile 10; das Programm wird erneut ausgeführt. Andernfalls endet das Programm, da keine weiteren Befehle mehr folgen.

Imperative oder prozedurale Programmiersprachen (Pascal, C)

Diese Programmiersprachen erlauben eine *Strukturierung* von Programmen, darüber hinaus ist eine gewisse *Modularisierung* möglich: Programme können in kleinere logische Einheiten eingeteilt werden, *Prozeduren* oder *Funktionen* genannt. Diese sind bis zu einem gewissen Grad wiederverwendbar.

Pascal wurde ab 1968 von dem Schweizer Mathematikprofessor *Niklaus Wirth* ausdrücklich als Lehrsprache entwickelt. Noch heute ist Pascal eine der beliebtesten Sprachen, um

Schülern oder Studenten das Programmieren beizubringen, weil die Sprache zu einer klaren Programmstrukturierung zwingt.

Die Programmiersprache *C* wurde 1971 von *Dennis Ritchie* und *Brian Kernighan* bei AT&T entwickelt, insbesondere um eine portierbare (auf andere Rechnerplattformen übertragbare) Version des Betriebssystems Unix zu schreiben. Ursprünglich wurde C für einen ganz bestimmten Computer und dessen Besonderheiten entwickelt, den DEC PDP-7. Aus diesem Grund ist C erstaunlich nah an den Fähigkeiten von Assembler, ohne so benutzerunfreundlich zu sein wie dieses.

Hier sehen Sie zunächst ein einfaches Pascal-Programm:

Ein Pascal-
Beispiel

```
PROGRAM tagesgruss;
VAR name: STRING;
    zeit: INTEGER;

BEGIN
    writeln ('Hallo. Gib deinen Namen ein!');
    readln (name);
    writeln ('Gib die Uhrzeit ein - nur Stunde!');
    readln (zeit);
    IF zeit < 12 THEN
        writeln ('Guten Morgen ', name)
    ELSE
        IF zeit < 18 THEN
            writeln ('Guten Tag ', name)
        ELSE
            writeln ('Guten Abend ', name);
    END.
END.
```

Das Programm begrüßt den Benutzer freundlich mit »Hallo«, und dieser wird aufgefordert, seinen Namen einzugeben. Daraufhin wartet es auf die Eingabe, die in der Variablen *name* gespeichert wird. Als Nächstes fragt das Programm nach der Uhrzeit oder genauer nach der Stunde.

In den verschachtelten IF-ELSE-Bedingungen wird dann je nach Tageszeit »Guten Morgen«, »Guten Tag« oder »Guten Abend« ausgegeben.

Die Programmiersprache C wird in Kapitel 9, »Grundlagen der Programmierung«, ausführlich behandelt. Trotzdem sehen Sie hier als Kontrast zu Pascal bereits ein kleines C-Beispielprogramm:

Ein C-Beispiel

```
#include <stdio.h>

int main ()
{
```

```

int a, b;
printf ("Geben Sie die erste Zahl ein: ");
scanf ("%d", &a);
printf ("Geben Sie die zweite Zahl ein: ");
scanf ("%d", &b);
if (a < b)
    printf ("%d ist kleiner als %d.\n", a, b);
else if (a > b)
    printf ("%d ist groesser als %d.\n", a, b);
else
    printf ("Zweimal die Zahl %d.\n", a);
return 0;
}

```

Es wird die Eingabe zweier Zahlen erwartet. Anschließend werden die beiden Zahlen verglichen; je nach Größe der beiden Zahlen wird eine entsprechende Meldung ausgegeben.

Die wichtigste Besonderheit ist, dass es kein *Hauptprogramm* gibt wie in Pascal, sondern nur die spezielle Funktion `main()`, die vom Betriebssystem aufgerufen wird. Die weiteren Eigenheiten von C werden, wie gesagt, in Kapitel 9, »Grundlagen der Programmierung«, erläutert.

Objektorientierte Programmiersprachen (Smalltalk, C++, Java, C#)

In einer *objektorientierten Sprache* wird in wiederverwendbaren Paketen programmiert, den sogenannten *Klassen*. Eine Klasse ist eine allgemeine Vorlage für die Konstruktion von Objekten. Ein *Objekt* ist eine Datenstruktur, die selbst Funktionen enthält, um sich auf die gewünschte Art und Weise zu »verhalten«.

Die wichtigsten Vorteile dieser Art der Programmierung sind die folgenden:

- ▶ *Kapselung*: Datenstrukturen außerhalb eines Objekts können nicht direkt dessen innere Daten manipulieren, sondern nur seine offiziellen Methoden (Schnittstellen nach außen) benutzen. Dies sorgt für ein klares Programmdesign und schützt vor vielen Fehlern.
- ▶ *Vererbung*: Klassen können ihre Eigenschaften und Methoden an »Kindklassen« abgeben, in denen nur noch die Unterschiede programmiert werden müssen. Dies beschleunigt die Softwareentwicklung und macht Programme noch einmal erheblich übersichtlicher.

Zu den wichtigsten objektorientierten Programmiersprachen gehören folgende:

- ▶ *Smalltalk* war die erste vollständig objektorientierte Sprache überhaupt. Sie wurde in den 1970er Jahren zur Programmierung der ersten grafischen Benutzeroberfläche entwickelt.

- ▶ C++ ist die objektorientierte Erweiterung der Programmiersprache C und wurde von *Bjarne Stroustrup* entwickelt. Da C++ abwärtskompatibel mit C ist, wurde die Sprache bald von C-Programmierern eingesetzt; ihre besonderen Vorteile als objektorientierte Sprache wurden aber erst langsam angenommen.
- ▶ *Java* besitzt vor allem die Besonderheit, dass es sich um eine plattformunabhängige Sprache handelt. Sie brauchen ein Java-Programm nur einmal zu kompilieren, es läuft innerhalb eines speziellen Programms für die verschiedenen Plattformen, der virtuellen Java-Maschine (JVM).
- ▶ *C#* ist eine objektorientierte Sprache von Microsoft. Es handelt sich um eine der Sprachen, mit denen Anwendungen für das .NET Framework entwickelt werden können. Zwar betont Microsoft aus Marketing-Erwägungen stets die Verwandtschaft mit C++, aber dennoch hat C# mehr mit Java gemeinsam als mit C++.

Logische Programmiersprachen (LISP, Prolog, Logo)

Einen völlig anderen Ansatz als die bisher behandelten Sprachen versprechen die *logischen Programmiersprachen*: Die Grundidee besteht darin, nicht den fertigen Algorithmus einzutippen und mit wechselnden Wertbelegungen vom Computer berechnen zu lassen, sondern das Grundproblem selbst zu formulieren und die Erzeugung einer möglichst optimalen Lösung dem Compiler zu überlassen.

Die Idee der logischen Programmierung stammt bereits aus den 1960er Jahren. Wegweisend waren die Arbeiten des Linguisten *Noam Chomsky*, der als Erster eine Verbindung zwischen Sprachwissenschaft und Informationstechnik herstellte. Auf dieser Grundlage entwickelten sich sogenannte Expertensysteme, die Fragen durch logische Schlussfolgerungen aus bereits bekannten Informationen beantworten können. Die logischen Programmiersprachen erwiesen sich als ideales Hilfsmittel zur Implementierung solcher Systeme.

Logische Sprachen im engeren Sinne verwenden die *Prädikatenlogik* zur Formulierung von Ausdrücken oder Termen; der wichtigste Vertreter ist *Prolog*.

Sprachen mit ähnlichem Ansatz, aber anderer Syntax werden auch als *funktionale Sprachen* bezeichnet; dazu gehört vor allem *LISP*, aber auch die Sprache *Logo*, die hauptsächlich entworfen wurde, um Kindern die Denkweise der Computerprogrammierung beizubringen. Als gemeinsamen Oberbegriff verwendet man manchmal die Bezeichnung *deklarative Programmiersprachen*.

In den letzten Jahren kommen logische und funktionale Programmiersprachen allmählich wieder in Mode, weil sich der Ansatz besonders gut zur Programmierung moderner Mehrkernprozessoren eignet. So entstanden die moderneren funktionalen Sprachen *Erlang* (logische Sprache, von Prolog inspiriert), *Clojure* (funktional, im Grunde ein moder-

ner LISP-Dialekt) und *Scala* (eine sogenannte Multiparadigmen-Sprache, die neben funktionalen auch objektorientierte und imperative Aspekte enthält).

1.3 Digitale Speicherung und Verarbeitung von Informationen

In diesem Abschnitt wird kurz erläutert, wie ein Computer Informationen speichert und verarbeitet. Es besteht hierbei ein grundsätzlicher Unterschied zwischen analogen und digitalen Daten: Analoge Informationen lassen sich in einer kontinuierlichen Wellenform wie in Abbildung 1.5 darstellen, die in immer kleinere Einheiten unterteilt werden können.

In der Natur liegen alle Informationen zunächst in analoger Form vor: Das Bild, das Sie sehen, oder der Ton, den Sie hören, besitzt prinzipiell keine kleinste Informationseinheit oder Auflösung. Mit dieser Art von Informationen kann ein Computer heutiger Bauart nichts anfangen. Die besonderen Eigenschaften der Elektronik haben dazu geführt, dass Computer digital entworfen wurden. »Digital« stammt vom englischen Wort *digit* (»Ziffer«); dieses Wort ist wiederum vom lateinischen *digitus* (»Finger«) abgeleitet, da die Finger von jeher zum Zählen eingesetzt wurden.

Digital sind Informationen also immer dann, wenn sie in Form von Zahlen dargestellt werden können. Genauer gesagt werden die Daten *binär* gespeichert, das heißt als Abfolge von Einsen und Nullen. Das ist nicht genau dasselbe wie *dual*. Das Dualsystem ist das mathematische Zweiersystem, während binär allgemein die Speicherung beliebiger Daten durch zwei verschiedene Zustände bezeichnet.

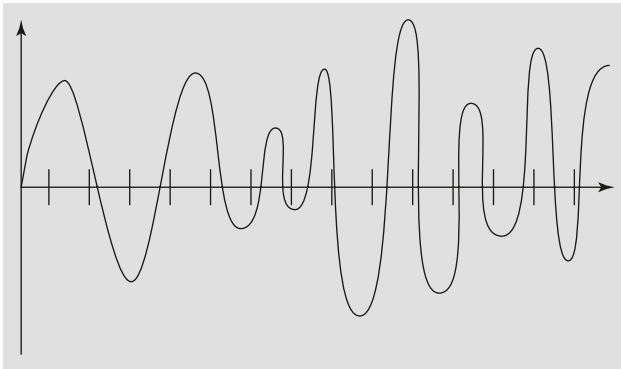


Abbildung 1.5 Schematische Darstellung von Analogdaten als Welle

Die Speicherung von Zahlen erfolgt übrigens in der Tat dual, solange es sich um ganze Zahlen handelt. Eine Besonderheit gilt dabei für vorzeichenbehaftete (positive oder negative) Zahlen, bei denen das vorderste Bit für das Vorzeichen steht. Kompliziert wird es da-

gegen bei Fließkommazahlen, die in Exponentialschreibweise gespeichert werden. Auf die Darstellung von Zahlen im Computer wird in Kapitel 2, »Mathematische und technische Grundlagen«, näher eingegangen.

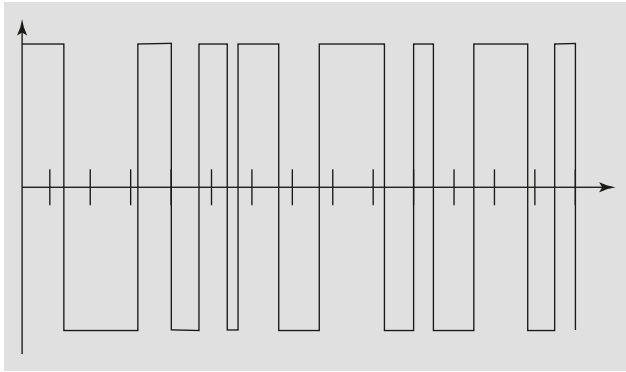


Abbildung 1.6 Schematische Darstellung binärer Digitaldaten

Charakteristisch für digitale Daten ist, dass es eine kleinste Informationseinheit gibt und dass die Information nicht mehr weiter aufgelöst werden kann. Während Analogdaten also durch die Wellenform gekennzeichnet sind, lassen sich Digitaldaten durch eine rechteckige Form darstellen (Abbildung 1.6). In der Mathematik werden Objekte mit einem solchen »Mindestabstand« als *diskrete Mengen* bezeichnet. Ein Alltagsbeispiel ist die Folge der ganzen Zahlen – im Unterschied zu den rationalen oder gar reellen Zahlen, bei denen zwischen zwei beliebigen Elementen immer noch unendlich viele Zwischenwerte liegen.

Die Umwandlung der analogen Eindrücke aus der Realität in computergeeignete digitale Daten wird als *Digitalisierung* bezeichnet. Je nach Datenart wird sie zum Beispiel von einem Scanner oder einer Digitalkamera bei Bildern oder von einer Soundkarte bei Tönen durchgeführt. Die folgenden Abschnitte sollen einen groben Eindruck davon vermitteln, wie verschiedene Arten von Daten grundsätzlich gespeichert werden.

Digit-
alisierung

1.3.1 Digitale Bilddaten

Wie Hardware zur Bilddigitalisierung funktioniert, also Scanner und Digitalkameras, wird kurz in Kapitel 3, »Hardware«, angeschnitten. Beachten Sie, dass es zwei grundlegende Arten von Computergrafik gibt: Die *Pixelgrafik* (auch *Bitmap-Grafik* genannt), von der hier die Rede ist, speichert ein Bild als rechteckiges Raster quadratischer Farbinformationen ab, den sogenannten *Pixeln*. Die *Vektorgrafik* speichert dagegen Umrisslinien und Kurven von Zeichnungen in Form mathematischer Formeln.

Die Qualität eines gespeicherten Pixelbildes lässt sich durch die folgenden Sachverhalte charakterisieren:

- ▶ Die *Auflösung* gibt die Größe der einzelnen Pixel an. Die Angabe besagt, wie viele Pixel pro Zentimeter oder Inch gespeichert werden. Beachten Sie, dass für den Druck erheblich höhere Auflösungen erforderlich sind (etwa 300 Pixel pro Inch) als für eine gleich große Bildschirmfläche (gerechnet wird hier – unabhängig von der tatsächlichen Monitorgröße – mit 72 Pixeln pro Inch).
- ▶ Die *Farbtiefe* gibt an, wie viele Bits zur Speicherung der Informationen eines einzelnen Pixels verwendet werden. Je nach Farbtiefe kann eine bestimmte Anzahl verschiedener Farben eingesetzt werden. Beispielsweise ermöglicht eine Farbtiefe von 8 Bit nur 256 verschiedene Farben, 16 Bit bieten 65.536 Farben und 24 Bit sogar mehr als 16,7 Millionen (genauer gesagt 16.777.216).
- ▶ In der Regel werden die einzelnen Farben nicht stur durchnummeriert, sondern aus einzelnen Grundfarben zusammengesetzt. Ohne hier näher darauf einzugehen, gibt es die *additive Farbmischung* der Lichtfarben Rot, Grün und Blau (RGB) sowie die *subtraktive Mischung* der Druckfarben Cyan, Magenta, Gelb und Schwarz (CMYK). Normalerweise wird für die Intensität jeder einzelnen Grundfarbe ein *Farbkanal* gespeichert. Die Farbtiefe wird dann pro Kanal angegeben, bei einem RGB-Bild mit 24 Bit Farbtiefe also beispielsweise 8 Bit (oder 256 Intensitätsstufen) pro Kanal.

1.3.2 Digitale Audiodaten

Töne werden mit Hilfe eines Verfahrens digitalisiert, das man *Sampling* nennt. Das Audiosignal wird in bestimmten Zeitabständen immer wieder abgetastet, genauer gesagt wird die Amplitude (das Volumen) zum jeweiligen Zeitpunkt gemessen. Die Frequenz (Tonhöhe) ergibt sich dabei aus der zeitlichen Verteilung der Amplituden. Jeder einzelne Abtastvorgang (*Sample*) wird als numerischer Wert abgespeichert.

In der Folge erweist sich die digitalisierte Tonabfolge als diskret in zweifacher Hinsicht: auf der X-Achse *zeitdiskret* durch das gleichbleibende Intervall zwischen den Messungen; auf der Y-Achse *wertdiskret* durch die Beschränkung auf einzelne herausgegriffene (Mittel-)Werte.

Genau wie bei Bildern gibt es auch hier verschiedene Merkmale, die die Datenmenge und die Qualität der gespeicherten Daten betreffen:

- ▶ Die *Sampling-Rate* gibt die Anzahl der Messvorgänge pro Sekunde an. Je höher die Frequenz dieser Messungen, desto höher ist die Tonqualität. Audio-CDs haben beispielsweise eine Sampling-Rate von 44,1 kHz (Kilohertz), es wird also 44.100 Mal pro Sekunde gemessen. Multimedia-Produktionen wie Computerspiele oder Infotainment-Titel verwenden dagegen häufig die halbe Sampling-Rate von 22,05 kHz.
- ▶ Die *Sampling-Tiefe* gibt die Datenbreite des einzelnen gespeicherten Tons an, legt also fest, wie viele verschiedene Amplituden unterschieden werden. Bei Audio-CDs sorgt

eine Sampling-Tiefe von 16 Bit (65.536 unterschiedliche Werte) für guten Ton; niedrigere Sampling-Tiefen klingen nicht besonders gut.

- Die Anzahl der *Tonkanäle* besagt, ob irgendeine Art von Raumklang gespeichert wird oder nicht. Wenn Audiodaten mono gespeichert werden, gibt es nur einen einzigen Kanal. Stereo verwendet zwei getrennte Kanäle, die über einen linken und einen rechten Lautsprecher ausgegeben werden können. Eine noch höhere Anzahl von Kanälen wie bei Quadrophonie, Dolby Surround oder 5.1-Sound bewirkt ein noch realistischeres Hörerlebnis.

Tabelle 1.2 stellt die verschiedenen Merkmale von Bilddaten den vergleichbaren Eigenschaften von Tondaten gegenüber, um die entsprechenden Beziehungen zwischen verschiedenen Arten von Digitaldaten zu verdeutlichen.

Informationsart	Bilddaten	Audiodaten
Auflösung	Bildauflösung, in Pixeln pro Zentimeter oder Inch	Sampling-Rate, in Samples pro Sekunde (kHz)
Datenbreite	Farbtiefe	Sampling-Tiefe
Anzahl Kanäle	Farbkanäle	Tonkanäle

Tabelle 1.2 Vergleich der Merkmale von Bild- und Audiodaten

1.3.3 Digitale Speicherung von Text

Reiner Text (nicht der formatierte Text in Textverarbeitungsprogrammen!) wird als Abfolge nummerierter Zeichen eines *Zeichensatzes* gespeichert. Je nach Datenbreite des verwendeten Zeichensatzes können unterschiedlich viele verschiedene Zeichen verwendet werden. Der grundlegende Computerzeichensatz ist noch heute *ASCII*. Es handelt sich um einen 7 Bit breiten Zeichensatz; er enthält also 128 Zeichen. Es sind sämtliche Zeichen für die Darstellung englischsprachiger Texte verfügbar.

Um auch Texte mit den Sonderzeichen der diversen europäischen Sprachen darstellen zu können, werden verschiedene Erweiterungen des ASCII-Codes verwendet. Da die 7 Bit der ASCII-Zeichen gewöhnlich in 8 Bit breiten Feldern gespeichert werden, steht Platz für weitere 128 Zeichen zur Verfügung. Auf diese Weise lassen sich lateinisch geschriebene Sprachen mit Sonderzeichen wie deutschen Umlauten darstellen, aber auch andere Buchstabenalphabete wie Arabisch, Russisch oder Griechisch. Silbenschriften wie Chinesisch oder Japanisch lassen sich dagegen auf diese Weise nicht speichern.

Um viele verschiedene Zeichensätze »unter einen Hut zu bringen«, wurde der *Unicode*-Standard eingeführt. Es handelt sich um einen Zeichensatz mit 16 bis 32 Bit breiten Zeichen; es können also mindestens 65.536 verschiedene Zeichen, in neueren Versionen sogar noch mehr, gespeichert werden. Auf diese Weise bietet Unicode genügend Platz für

die Schriftzeichen der meisten Sprachen der Welt sowie für mathematische, technische und andere Sonderzeichen.

Die verschiedenen Zeichensätze werden in Kapitel 16, »Weitere Datei- und Datenformate«, genauer besprochen.

1.4 Zusammenfassung

Die Wissenschaft vom Umgang mit Computersystemen, um die es in diesem Buch geht, wird allgemein Informatik genannt. Der praktische Einsatz dieser Systeme heißt dagegen meist Informationstechnik (IT). Es gibt verschiedene Ausbildungsgänge, die auf Berufe in diesem Fachbereich vorbereiten: zum einen praktische Berufsausbildungen wie die im Titel dieses Buches erwähnten Fachinformatiker, zum anderen die erheblich umfangreicheren, aber theoretischeren Studiengänge.

Die Entwicklung des Computers war letztlich eine Folge des wissenschaftlichen Fortschritts: Die seit Jahrtausenden bekannten Gesetzmäßigkeiten der Mathematik und der Logik wurden mit modernen Erkenntnissen der Elektrotechnik beziehungsweise Elektronik kombiniert, um einen frei programmierbaren Rechenautomaten zu erhalten. Die fortschreitende Miniaturisierung und eine effiziente industrielle Fertigung sorgten dafür, dass die Computer sich im Laufe von etwa sechzig Jahren von beinahe unbezahlbaren Einzel-exemplaren an Universitäten zum allgegenwärtigen Werkzeug für fast jeden Anwendungszweck und die meisten Berufe entwickeln konnten.

Parallel zu den Computern selbst hat sich auch die Art und Weise ihrer Programmierung weiterentwickelt: Anfangs musste die reine numerische Maschinensprache mit Hilfe von Schalttafeln und Steckverbindungen eingegeben werden; nach und nach wurden sowohl die Eingabemethoden als auch der Funktionskomfort der Sprachen selbst erheblich verbessert.

Interessant ist in diesem Zusammenhang, dass sich die innere Funktionsweise der Rechner von Anfang an kaum verändert hat: Bis heute werden elektronisch gesteuerte Schalter verwendet, die die beiden Zustände 0 und 1 darstellen können. Sämtliche Informationen – ob mathematische Informationen, Texte oder Multimedia-Daten – sowie die Manipulation dieser Daten müssen mit Hilfe dieser Schaltungen codiert werden. Wie dies im Einzelnen geschieht, erfahren Sie im Verlauf dieses Buches noch wesentlich genauer.

*Jeder wandle für sich und wisse nichts von dem andern.
Wandern nur beide gerad', finden sich beide gewiss.
– Johann Wolfgang Goethe/Friedrich Schiller, Xenien*

4 Netzwerkgrundlagen

Internet und lokale Netzwerke haben die Bedeutung des Computers in den letzten Jahren revolutioniert. Viele Anwendungsprogramme kooperieren über das Netzwerk miteinander. Der Datenaustausch in und zwischen Unternehmen erfolgt fast ausschließlich per Vernetzung, und immer mehr Geschäftsabläufe werden online getätigt. Da die Netzwerkfähigkeit zudem eine Grundfunktionalität aller modernen Betriebssysteme geworden ist, steht diese Einführung noch vor dem Kapitel über allgemeine Systemkonzepte.

Nach einer historischen und technischen Einführung erfahren Sie in diesem Kapitel das Wichtigste über gängige Netzwerkhardware; somit wird die Betrachtung der Hardware aus dem vorigen Kapitel hier vervollständigt. Anschließend werden die Netzwerkprotokolle mit dem Hauptaugenmerk auf die seit Jahren dominierenden Internetprotokolle (TCP/IP) beleuchtet.

4.1 Grundlagen

In diesem Abschnitt erfahren Sie zunächst einmal, was Netzwerke eigentlich sind und was verschiedene Netzwerktypen voneinander unterscheidet. Anschließend wird die Entstehungsgeschichte lokaler Netze, der Datenfernübertragung und des Internets betrachtet.

4.1.1 Was ist ein Netzwerk?

Ein Netzwerk ist eine Verbindung mehrerer Computer zum Zweck des Datenaustauschs, für verteilte Anwendungen oder auch für die Kommunikation zwischen ihren Benutzern.

Im Lauf der Computergeschichte haben sich viele verschiedene Möglichkeiten der Verkabelung und der Kommunikationsstrukturen sowie zahlreiche Anwendungsgebiete entwickelt:

- Die Verkabelung oder allgemein die Hardwaregrundlage reicht von der Verwendung gewöhnlicher Telefonleitungen mit besonderen Verbindungsgeräten, den Modems,

über speziell für die Anwendung in lokalen Netzwerken entwickelte Netzwerkkarten und Netzwerkkabel bis hin zu Hochgeschwindigkeitsnetzen, etwa über Glasfaserkabel. Auch die diversen Möglichkeiten der drahtlosen Übertragung werden immer wichtiger.

- ▶ Kommunikationsstrukturen, definiert durch sogenannte Netzwerkprotokolle, gibt es unzählige. Viele sind von einem bestimmten Hersteller, einer Plattform oder einem Betriebssystem abhängig, andere – wie die Internet-Protokollfamilie TCP/IP – sind offen, unabhängig und weitverbreitet.
- ▶ Was die Anwendungsgebiete angeht, so reichen diese vom einfachen Dateiaustausch in Arbeitsgruppen über die gemeinsame Nutzung teurer Hard- und Software bis hin zu hochkomplexen, spezialisierten und verteilten Anwendungen.

Paketvermittelte Datenübertragung

Ein wesentliches Merkmal der meisten Netzwerkformen ist die Übertragung von Daten mit Hilfe sogenannter *Datenpakete*.

Um die *Paketvermittlung* (Packet Switching) zu verstehen, sollten Sie zunächst ihr Gegenteil, die *Schaltkreisvermittlung* (Circuit Switching) der herkömmlichen Telefonleitungen, betrachten. (Hinweis: Inzwischen gilt dies nicht mehr zwingend; durch die Einführung neuer Technik laufen auch immer mehr Telefonverbindungen hinter den Kulissen paketvermittelt ab – per *Voice over IP* (VoIP) sogar bis zum Endkunden. Durch geeignete Kommunikationsprotokolle wird aber dafür gesorgt, dass die Nutzer dies nicht bemerken.) Durch das Wählen einer bestimmten Rufnummer (oder früher durch die Handvermittlung) werden bestimmte Schalter geschlossen, die für die gesamte Dauer des Telefongesprächs eine feste Punkt-zu-Punkt-Verbindung zwischen beiden Stellen herstellen. Über diese dauerhafte Leitung können Sprache oder Daten in Echtzeit und in der korrekten Reihenfolge ohne Unterbrechung übertragen werden. Nachdem die Übertragung beendet ist, wird die Verbindung wieder abgebaut, und die betroffenen Leitungen stehen für andere Verbindungen zur Verfügung.

Ganz anders sieht es bei der Paketvermittlung aus: Zu keinem Zeitpunkt der Datenübertragung wird eine direkte Verbindung zwischen den beiden beteiligten Stellen hergestellt. Stattdessen sind beide nur indirekt über ein loses Netz von Vermittlungsstellen, *Router* genannt, miteinander verbunden. Damit auf diesem Weg Daten übertragen werden können, wird folgender Mechanismus verwendet:

- ▶ Die Daten werden in kleinere Einheiten unterteilt, die *Datenpakete*.
- ▶ Jedes einzelne Datenpaket wird mit der Absender- und der Empfängeradresse versehen.
- ▶ Der Absender übergibt jedes Datenpaket an den nächstgelegenen Router.
- ▶ Jeder beteiligte Router versucht, das Paket anhand der Empfängerangabe an den günstigsten Router weiterzuleiten, damit es letztendlich an seinem Ziel ankommt.
- ▶ Der Empfänger nimmt die Datenpakete entgegen und interpretiert sie je nach Daten- und Übertragungsart auf irgendeine zwischen den beiden Stellen vereinbarte Art und Weise.

Zur reinen Paketvermittlung gehört zunächst einmal kein Mechanismus, der die vollständige Auslieferung aller Datenpakete garantiert. Es wird standardmäßig weder der Erfolg noch das Ausbleiben einer Paketlieferung gemeldet. Im Übrigen wird auch keine verbindliche Reihenfolge festgelegt. Da jedes einzelne Paket einen beliebigen Weg durch das Netzwerk nehmen kann, kommt mitunter ein später abgesendetes Paket noch vor einem früher versandten beim Empfänger an.

Um die potentiell unsichere Datenübertragung per Paketvermittlung für bestimmte Anwendungen zuverlässiger zu machen, wird zusätzlich eine Erfolgskontrolle implementiert. Außerdem werden die Pakete oft durchnummeriert, um die korrekte Reihenfolge wiederherzustellen. Allerdings haben solche Maßnahmen nichts mit der eigentlichen Paketvermittlung zu tun und müssen in diesem Zusammenhang nicht beachtet werden. In der Regel sind die Softwarekomponenten, die sich um die Übertragung der Datenpakete kümmern, gar nicht in der Lage, diese zusätzlichen Kontrollinformationen selbst auszuwerten.

4.1.2 Entstehung der Netzwerke

Wenn Sie sich die Geschichte der Computer anschauen, die in Kapitel 1, »Einführung«, skizziert wurde, fällt auf, dass die Verwendung von Netzwerken anfangs keinen Sinn ergeben hätte: Bei den frühen Großrechnern gab es keine standardisierte Software, die miteinander hätte kommunizieren können. Darüber hinaus wurden sie zunächst über Schalttafeln und später über Lochkarten bedient. Es gab also keine Echtzeit-Interaktion zwischen Benutzer und Programm, so dass es erst recht abwegig war, verschiedene Computer miteinander interagieren zu lassen. Frühestens, als der Dialogbetrieb über Terminals (siehe Kapitel 3, »Hardware«, und Kapitel 5, »Betriebssystemgrundlagen«) eingeführt wurde, war an eine Vernetzung zu denken.

Geschichte des Internets

Der Anstoß für die Entwicklung eines Computernetzwerks kam aus einer eher unerwarteten Richtung: Die atomare Bedrohung des Kalten Krieges schürte die Angst der Verantwortlichen in Politik und Militär in den USA, im Fall eines Atomkrieges handlungsunfähig zu werden, weil die Übermittlung von Informationen nicht mehr funktionieren könnte. Es war schlichtweg zu riskant, sich auf einen einzigen Zentralcomputer mit Terminals zu verlassen. Deshalb begann 1969 der Betrieb eines experimentellen Netzes aus vier Computern an verschiedenen US-amerikanischen Universitäten. Federführend für das Projekt war die (Defense Department's) Advanced Research Projects Agency (ARPA, später auch DARPA), eine Forschungskommission des amerikanischen Verteidigungsministeriums, die 1957 als Reaktion auf den ersten sowjetischen Satelliten Sputnik gegründet worden war. Die USA wollten den Anschluss auf verschiedenen wichtigen Gebieten der Wissenschaft nicht verpassen – und neben der Raumfahrt gehörte auch die Computertechnik zu diesen Gebieten. Folgerichtig hieß dieses erste Netzwerk ARPANET.

ARPANET-
Anwendung
und -Technik

Allgemein sind bei der Betrachtung von Netzwerken immer mindestens zwei Ebenen zu unterscheiden: zum einen der Anwendungszweck des Netzwerks, zum anderen dessen technische Realisierung. Bei näherem Hinsehen sind sogar noch weitere solcher Ebenen auszumachen; diese sogenannten Schichtenmodelle werden im Abschnitt »Funktionsebenen von Netzwerken« ab Seite 176 besprochen. Interessanterweise stellt sich im Entwicklungsverlauf von Netzwerken manchmal heraus, dass der gewünschte Anwendungszweck technisch anders realisierbar ist, aber auch oft, dass eine bestimmte technische Realisation völlig anderen Anwendungen als der ursprünglich geplanten dienlich sein kann. Besonders in der Geschichte des Internets, dessen Vorläufer das ARPANET war, ist dies oft festzustellen.

Die ursprüngliche Anwendung dieses Netzes bestand lediglich darin, Datenbestände auf den unterschiedlichen angeschlossenen Computern automatisch zu synchronisieren, d. h. einfach aus Sicherheitsgründen den gleichen Informationsbestand auf mehreren Rechnern bereitzuhalten.¹

Grundgedanke der Vernetzung selbst war dabei besonders die Fähigkeit jedes beteiligten Computers, Daten, die nicht für ihn selbst bestimmt waren, sinnvoll weiterzuleiten. Hieraus ergeben sich zwei unschätzbare organisatorische und technische Vorteile:

- ▶ Ein Computer muss nicht direkt mit demjenigen verbunden sein, mit dem er Daten austauschen soll.
- ▶ Der Ausfall oder die Überlastung eines bestimmten Verbindungsweges kann durch Alternativen kompensiert werden.

Auf diese Weise konnte das ursprüngliche Ziel, nämlich die Angriffs- und Ausfallsicherheit des Netzes zu gewährleisten, erreicht werden.

Schon unmittelbar nach der Einrichtung des ARPANETs begann die oben erwähnte Weiterentwicklung. Man stellte schnell fest, dass die technische Infrastruktur dieses Netzes für weit mehr Anwendungen zu nutzen war als das vergleichsweise langweilige automatische Synchronisieren von Datenbeständen. So kam bald eine benutzerorientierte Möglichkeit des Dateiaustauschs hinzu. Außerdem war es schon für gewöhnliche Konfigurationsaufgaben unerlässlich, einem entfernten Computer unmittelbar Anweisungen erteilen zu können. Dies war der Ausgangspunkt für die Entwicklung der Terminalemulation, also der Benutzung des eigenen Terminals für einen Computer, an den es nicht unmittelbar, sondern nur indirekt über das Netzwerk angeschlossen ist. Auch wenn diese Anwendungen noch nicht sofort ihre späteren Namen – FTP und Telnet – erhielten und

¹ Auch heutige Serversysteme vervielfältigen wichtige Daten auf diese Weise automatisch. Das Verfahren wird *Replikation* genannt und kommt insbesondere bei Datenbank- oder Verzeichnisdienstservern zum Einsatz. In Kapitel 12, »Datenbanken«, wird es am Beispiel von MySQL beschrieben.

die technischen Details ihrer Implementierung sich noch weiterentwickelten, sind sie dennoch nach wie vor wichtige Nutzungsschwerpunkte des Internets.

Alles in allem wurde dieses Netzwerk schnell populär. Zwei Jahre nach seiner Einrichtung, im Jahr 1971, waren bereits 40 Computer an verschiedenen Universitäten und anderen staatlichen Forschungseinrichtungen angeschlossen, und es war bei weitem nicht nur die militärische Nutzung von Interesse. Auch akademisch hatte das Netz viel zu bieten: Wissenschaftler sind darauf angewiesen, Daten auszutauschen; hier ergab sich eine Möglichkeit, dies sehr schnell und effektiv zu tun.

1972 wurde dann der bis dahin bedeutendste Dienst dieses Netzes erfunden: Ray Tomlinson, ein Mitarbeiter eines Ingenieurbüros in Kalifornien, verschickte die erste E-Mail. Bis heute zählt die E-Mail zu den erfolgreichsten und verbreitetsten Anwendungen des Netzes; sie kann sich nach dem viel jüngeren World Wide Web noch immer auf einem guten zweiten Platz in puncto Beliebtheit von Internetdiensten halten, und es ist auch nicht zu sehen, warum sich dies in absehbarer Zeit ändern sollte.

Erfindung der
E-Mail

Das ursprüngliche ARPANET wuchs immer weiter. Zudem wurden nach dem gleichen Prinzip andere, ähnliche Netze konstruiert. Dies ist nicht zuletzt der Tatsache zu verdanken, dass alle Schritte, die zur Entwicklung des Netzes beigetragen haben, von Anfang an sorgfältig dokumentiert und der Öffentlichkeit zugänglich gemacht wurden. Dieser Dokumentationsstil ist bis heute beibehalten worden; die entsprechenden Dokumente heißen *RFC* (Request For Comments, etwa »Bitte um Kommentare«).

Es gibt bis heute über 6.300 solcher RFC-Dokumente, die alle online zur Verfügung stehen, zum Beispiel unter <http://www.rfc-editor.org/rfc-index2.html>. Die meisten sind technische Beschreibungen von Entwürfen, Protokollen und Verfahrensweisen; nur wenige (in der Regel mit dem Datum 1. April) nehmen sich nicht ganz so ernst – zum Beispiel RFC 2324, in dem das Protokoll HTCPCP zur Steuerung vernetzter Kaffeemaschinen vorgeschlagen wird, oder RFC 1300, ein nettes Gedicht über Namen und Begriffe, die im Zuge der Computer- und Netzwerkentwicklung ihre ursprüngliche Bedeutung verändert haben.

RFC-
Dokumente

Alle Personen, Institutionen und Unternehmen, die etwas Entscheidendes zum ARPANET und späteren Internet beigetragen haben, haben dies in solchen Dokumenten erläutert. Dies ermöglicht es jedem beliebigen Hersteller von Hard- oder Software, mit seinen Produkten diese Standards zu unterstützen, denn sie gehören keinem einzelnen Hersteller und keiner bestimmten Person, und niemand kann den Zugriff darauf beschränken oder Lizenzgebühren fordern – ein entscheidender Grund dafür, warum die Protokolle des Internets heute vom Personal Computer bis zum Großrechner überall dominieren.

In den 1980er Jahren schließlich wurde der militärisch genutzte Teil des ARPANETs als MilNet von ihm abgetrennt, und das restliche ARPANET wurde mit dem NSFNet, dem Netz der National Science Foundation, und einigen anderen Netzwerken zum Internet zu-

ARPANET +
NSFNet
= Internet

sammengeschlossen. Die kommerzielle Nutzung, heute Hauptverwendungsgebiet des Internets, ließ danach aber noch fast 15 Jahre auf sich warten. Denn die Anwendungen des Internets waren zwar robust und wenig störanfällig, aber alles andere als benutzerfreundlich. Abgesehen davon waren die ersten Personal Computer, die in der zweiten Hälfte der 1970er Jahre auftauchten, weder konzeptionell noch von der Leistung her in der Lage, mit den Internetprotokollen etwas anzufangen.

DFÜ Recht früh wurde dagegen die Datenfernübertragung (DFÜ), also der Datenaustausch über Telefonleitungen, für Home- und Personal Computer eingeführt. Seit Ende der 1970er Jahre wurden sogenannte Akustikkoppler verwendet: Geräte, die an den Computer angeschlossen wurden und auf die einfach der Telefonhörer gelegt werden musste. Diese langsamen und störanfälligen Apparate wurden bald durch Modems ersetzt, die eine direkte Verbindung zwischen Computer und Telefonleitung zuließen und im Laufe der Jahre allmählich schneller und zuverlässiger wurden. Hauptanwendungsgebiete waren auf der einen Seite die sogenannten Mailboxen, also Informations- und Datenangebote für Computer, die eine direkte Telefonverbindung zum Mailboxrechner herstellten. Auf der anderen Seite entstanden in den 1980er Jahren die meisten kommerziellen Onlinedienste wie CompuServe, AOL oder in Deutschland BTX (Vorläufer von T-Online), das zunächst über spezielle Terminals statt über PCs mit einer bestimmten Software genutzt wurde.

Das WWW Die Entwicklung des Internets vom exklusiven Wissenschaftler-Netz zum Massenmedium nahm ihren Anfang erst 1989 in der Schweiz, am Europäischen Forschungsinstitut für Kernphysik (CERN) in Genf. Dort machte sich der britische Informatiker Tim Berners-Lee Gedanken darüber, wie man Netzwerke, besonders das Internet, für den einfachen und effizienten Zugriff auf wissenschaftliche Dokumente nutzen könnte. Ergebnis dieser Arbeit war die Grundidee des World Wide Webs, eines hypertextbasierten Informationssystems, das die Infrastruktur des Internets zur Datenübermittlung nutzen sollte.

Hypertext ist nichts anderes als Text mit integrierten Querverweisen, die automatisch funktionieren. Mit anderen Worten: Durch Betätigen des Querverweises, der in diesem Zusammenhang *Hyperlink* heißt, stellt der Text selbst – beziehungsweise das System, das diesen darstellt – die Verbindung mit dem verknüpften Dokument her.

Nun war Hypertext 1989 gewiss nichts Neues. Versuche damit reichen zurück bis in die 1950er Jahre, in Hilfesystemen war er in den 1980er Jahren bereits Alltag. Neu war nur seine Nutzung über ein Netzwerk, genauer gesagt über das Internet.

WWW-Bestandteile So entstand ein äußerst effektives Informationssystem für Wissenschaftler, die auf diese Weise ihre Forschungsergebnisse miteinander austauschten. Der Prototyp dieses Systems, das World Wide Web heißen sollte, umfasste im Einzelnen die folgenden Bestandteile:

- ▶ ein spezielles neues Anwendungsprotokoll, das Hypertext Transfer Protocol (HTTP)
- ▶ einen Serverdienst, der in der Lage ist, Anfragen, die in der Sprache des HTTP formuliert sind, auszuliefern

- ▶ eine neu geschaffene Formatierungs- und Beschreibungssprache für solche Hypertext-Dokumente, die Hypertext Markup Language (HTML)
- ▶ ein Anzeigeprogramm für entsprechend formatierte Dokumente, den Browser

1991 wurde das System der Öffentlichkeit vorgestellt. Es wurde praktisch von Anfang an nicht nur zu ernsthaften wissenschaftlichen Zwecken genutzt, sondern allgemein zur Veröffentlichung von Text, Bildern und den verschiedensten Themen. Zunächst war die Nutzung des Systems beschränkt auf wissenschaftliches Personal sowie interessierte Studenten. Sie störten sich nicht am mangelnden Komfort der ersten Browser oder den geringen Layoutfähigkeiten der ersten HTML-Versionen. Als jedoch immer mehr private Benutzer dazukamen – was durch das allmähliche Entstehen kommerzieller Internetprovider und Browser für PC-Betriebssysteme wie Windows oder Mac OS gefördert wurde –, änderte sich dies. Der berühmt gewordene »Browserkrieg« zwischen Netscape und Microsoft schuf letztendlich Fakten, die niemand für möglich oder auch nur wünschenswert gehalten hätte, die jedoch bis heute das Wesen des World Wide Webs bestimmen.

Zwei Merkmale sind hier besonders wichtig:

- ▶ Die Seitenbeschreibungssprache HTML wurde immer mehr für die Definition des Seitenlayouts genutzt statt nur für die Struktur. Für Websites, die ein möglichst großes Publikum erreichen sollen, das weniger technisch und mehr inhaltlich interessiert ist, ist das Layout wichtiger als die Struktur. (Inzwischen kommt für das Layout allerdings praktisch nur noch CSS zum Einsatz, und HTML konzentriert sich wieder – wie ursprünglich angedacht – auf die Dokumentstruktur.)
- ▶ Der Anteil kommerzieller Websites am gesamten Bestand wurde immer größer und überwiegt heute bei weitem; das Angebot im Web ist den Rundfunkmedien wie etwa dem Fernsehen ähnlicher geworden. Während Tim Berners-Lee sich ursprünglich ein Netz vorgestellt hatte, in dem alle Teilnehmer sowohl Anbieter als auch Konsumenten von Inhalten sein sollten, wird das Web heutzutage von vielen weitgehend passiv als Medium genutzt. Erst die neu entstehenden »Web 2.0«-Tools wie Blogs, Wikis und andere kommen Berners-Lees eigentlichen Ideen näher, wobei die gleichzeitig zu beobachtende Kommerzialisierung samt Aufkauf der wichtigsten Sites durch große Medienkonzerne sicherlich nicht in seinem Sinne ist.

Lokale Netze

Einen vollkommen anderen Anstoß zur Entwicklung von Netzwerken gab das Aufkommen des sogenannten *Outsourcings* in der Computertechnik, also der Verlagerung der Rechenleistung von einem Zentralcomputer auf den einzelnen Schreibtisch.

Die fortschreitende Ausstattung von Büros mit Personal Computern führte mangels anderer Optionen zunächst zur Blüte des »Turnschuhnetzwerks«: Anwender liefen mit Datenträgern bewaffnet durch das ganze Gebäude, um Daten miteinander auszutauschen oder

zum Beispiel einen speziellen Drucker zu verwenden. Auch zwischen verschiedenen Unternehmen erfreute sich der sogenannte Datenträgeraustausch großer Beliebtheit: Die Datensätze von Geschäftsvorfällen wurden auf Disketten oder Magnetbändern zwischen den einzelnen Unternehmen hin und her gereicht.

Lokale Firmennetzwerke wurden zwar bereits Mitte der 1970er Jahre bei XEROX PARC erfunden, aber erst Ende der 80er Jahre rückten sie stärker ins allgemeine Interesse. Es war ein Bedürfnis der Anwender von PCs, miteinander Daten auszutauschen, einfach deshalb, weil die meisten Vorgänge der Datenverarbeitung von mehreren Mitarbeitern erledigt werden. So entstanden viele verschiedene Arten der Netzwerkhardware. Neben dem bereits genannten Ethernet mit seinen vielfältigen Varianten gab es beispielsweise auch Token Ring von IBM, ARCnet oder auch einfache serielle Direktverbindungen zwischen Computern über die sogenannten Nullmodemkabel.

Was die Software angeht, so wurden die eigentlich nicht dafür geeigneten PC-Betriebssysteme um Netzwerkfähigkeiten erweitert. Hinzu kamen spezielle Betriebssysteme für Server, also solche Rechner, die anderen im Netzwerk verschiedene Ressourcen zur Verfügung stellen. Bekannt sind hier etwa Novell NetWare, IBM OS/2 oder später auch Windows NT Server.

Wenn Sie in diesem Zusammenhang Linux und andere Unix-Varianten vermissen, dann liegt das daran, dass Unix als PC-Betriebssystem und als Serversystem für PC-Netzwerke erst einige Jahre später populär wurde. Ein gewisses Grundverständnis für Unix ist übrigens unerlässlich, um die Funktionsweise der Internetprotokolle nachvollziehen zu können. Einige Grundlagen dieses Systems werden in Kapitel 5, »Betriebssystemgrundlagen«, und 7, »Linux«, erläutert.

4.2 Funktionsebenen von Netzwerken

Wie bereits in der Einleitung mehrfach angedeutet wurde, besteht ein gewisses Problem beim Verständnis von Netzwerken darin, dass einige sehr verschiedene Aspekte zu ihrem Funktionieren beitragen. Schon ganz zu Beginn haben Sie eine grobe Unterteilung in die drei Ebenen Verkabelung oder allgemeine Netzwerkhardware, Kommunikationsstrukturen oder Netzwerkprotokolle und schließlich Anwendungen eines Netzwerks kennengelernt.

Eine so ungenaue Einteilung lässt die grundsätzliche Schwierigkeit erkennen, reicht aber nicht ganz aus, um Netzwerke in all ihren Bestandteilen zu begreifen, und schon gar nicht, um verschiedene Arten von Netzwerken miteinander zu vergleichen. Auch die Tatsache, dass ein und dieselbe Komponente auf einer bestimmten Ebene wahlweise mit mehreren unterschiedlichen Elementen einer anderen Funktionsebene zusammenarbeiten kann, wird so noch nicht transparent genug.

4.2.1 Das OSI-Referenzmodell

Um die Ebenen, die ein Netzwerk ausmachen, ganz genau auseinanderhalten zu können, bedient man sich sogenannter *Schichtenmodelle* (Layer Models). Das bekannteste und verbreitetste von ihnen ist das OSI-Referenzmodell der internationalen Standardisierungsorganisation ISO. OSI steht für »Open Systems Interconnect«, also etwa »Verbindung zwischen offenen Systemen«. Das Modell wurde 1978 entworfen und besteht aus sieben übereinander angeordneten Schichten, die jeweils einen Aspekt der Netzwerkkommunikation beschreiben. Ganz unten ist die Hardware angesiedelt, ganz oben befindet sich die Anwendung des Netzes. Hier zunächst die Schichten des OSI-Modells im Überblick, die Beschreibung folgt danach:

1. Bitübertragungsschicht (Physical Layer)
2. Sicherungsschicht (Data Link Layer)
3. Vermittlungsschicht (Network Layer)
4. Transportschicht (Transport Layer)
5. Kommunikationssteuerungsschicht (Session Layer)
6. Darstellungsschicht (Presentation Layer)
7. Anwendungsschicht (Application Layer)

Die einzelnen
OSI-
Schichten

Die Bezeichnung »OSI-Referenzmodell« deutet bereits darauf hin, dass es sich nicht um einen Standard handelt, der konkrete Netzwerkprotokolle definiert. Das OSI-Modell definiert nur die Funktionen der einzelnen Schichten und ist somit ein Schema zur Definition solcher Standards, beispielsweise für die weiter unten vorgestellten IEEE-802.3-Standards. Jeder Standard deckt dabei immer nur Teilaspekte des OSI-Modells ab.

Die Bedeutung der einzelnen Schichten des OSI-Modells

1. Die *Bitübertragungsschicht* oder auch *physikalische Schicht* beschreibt nur, wie die reine Übertragung der Daten elektrisch beziehungsweise allgemein physikalisch vonstattengeht. OSI-basierte Netzwerkstandards beschreiben in dieser untersten Schicht die Struktur der Signale. Dazu gehören unter anderem folgende Aspekte:
 - ▶ zulässiger Amplitudenbereich
 - ▶ Versand- und Empfangsmethoden für Bitfolgen
 - ▶ Operationen zur Umwandlung dieser Bitfolgen in Daten für die nächsthöhere Schicht (und umgekehrt)
 - ▶ Verarbeitungsgeschwindigkeit der Bitfolgen
 - ▶ Start- und Stoppsignale
 - ▶ Erkennung beziehungsweise Unterscheidung der Signale bei gemeinsam genutzten Medien

- ▶ Übertragungseigenschaften der Medien (Kabel, Lichtwellenleiter, Funk o. Ä.)
- ▶ Die Medien selbst sowie Netzwerkkarten oder Onboard-Netzwerkchips sind kein Bestandteil der Definitionen auf der ersten Schicht. Die Hersteller müssen selbst dafür Sorge tragen, dass ihre Produkte den Spezifikationen genügen.

2. Die *Sicherungsschicht* beschreibt alle Vorkehrungen, die dafür sorgen, dass aus den einzelnen zu übertragenden Bits, also dem reinen physikalischen Stromfluss, ein verlässlicher Datenfluss wird. Dazu gehören die beiden Teilaufgaben Media Access Control (MAC) – die Regelung des Datenverkehrs, wenn mehrere Geräte den gleichen Kanal verwenden – sowie Logical Link Control (LLC), wobei es um die Herstellung und Aufrechterhaltung von Verbindungen zwischen den Geräten geht.

Viele Protokolle dieser Schicht implementieren eine Fehlerkontrolle, bei Ethernet wird zum Beispiel CRC verwendet. In manchen Fällen wird auch Quality of Service (QoS), eine Art Prioritätsinformation, benutzt. In der Sicherungsschicht werden die Bitfolgen in Einheiten einer bestimmten Größe unterteilt und mit einem Header aus Metainformationen versehen. Je nach Standard werden auf dieser Ebene unterschiedliche Namen für diese Datenpakete verwendet. Bei Ethernet und Token Ring ist beispielsweise von »Frames« die Rede, bei ATM von »Zellen«. Der Payload (Nutzdateninhalt) eines Frames beziehungsweise einer Zelle beginnt in aller Regel mit dem Header eines hineinverschachtelten Pakets der nächsthöheren Schicht. Es kann aber auch vorkommen, dass Pakete verschiedener Protokolle der zweiten Schicht ineinander verschachtelt werden. Dies ist zum Beispiel bei PPP over Ethernet, PPP over ATM oder ATM over SDH der Fall.

3. Die *Netzwerkschicht* oder *Vermittlungsschicht* definiert diejenigen Komponenten und Protokolle des Netzwerks, die an der indirekten Verbindung von Computern beteiligt sind. Hier ist sogenanntes *Routing* erforderlich, das Weiterleiten von Daten in andere logische oder auch physikalisch inkompatible Netzwerke. So gehören zum Beispiel auch alle diejenigen Protokolle zur Netzwerkschicht, die die logischen Computeradressen der höheren Schichten in die physikalischen Adressen umsetzen, bei Ethernet zum Beispiel ARP. Auch auf der Netzwerkschicht werden die Daten in Pakete unterteilt, deren Namen sich je nach konkretem Protokoll unterscheiden. Das mit Abstand verbreitetste Protokoll dieser Ebene, das weiter unten ausführlich vorgestellte IP-Protokoll, bezeichnet sie als »IP-Datagramme«.
4. Die Protokolle der *Transportschicht* lassen sich in verbindungsorientierte Protokolle wie TCP und verbindungslose Protokolle wie etwa UDP unterteilen. Auf dieser Schicht werden vielfältige Aufgaben erledigt. Ein wichtiger Aspekt sind Multiplex-Mechanismen, die die Anbindung der Datenpakete an konkrete Prozesse auf den kommunizierenden Rechnern ermöglichen, bei TCP und UDP beispielsweise über Portnummern, bei SPX über Connection-IDs. Verbindungsorientierte Transportprotokolle wie TCP oder SPX sind zudem meist mit einer Fluss- und Fehlerkontrolle ausgestattet, um zu gewährleisten, dass Pakete vollständig am Ziel ankommen und dort in der richtigen

Reihenfolge verarbeitet werden. Auch auf der vierten Schicht verwenden verschiedene Protokolle jeweils eigene Bezeichnungen für die Datenpakete; so ist etwa von »UDP-Datagrammen«, »TCP-Sequenzen« und »SPX-Paketen« die Rede.

5. Die *Kommunikationssteuerungsschicht* oder *Sitzungsschicht* stellt die Kommunikation zwischen kooperierenden Anwendungen oder Prozessen auf verschiedenen Rechnern sicher.
6. Die *Darstellungs-* oder *Präsentationsschicht* dient der Konvertierung und Übertragung von Datenformaten, Zeichensätzen, grafischen Anweisungen und Dateidiensten.
7. Die *Anwendungsschicht* schließlich definiert die unmittelbare Kommunikation zwischen den Benutzeroberflächen der Anwendungsprogramme, kümmert sich also um die Verwendung derjenigen Dienste über das Netzwerk, die Benutzer unmittelbar zu Gesicht bekommen.²

Da das OSI-Modell eine Zusammenstellung von möglichen Fähigkeiten für viele verschiedene Arten von Netzwerken darstellt, kann es natürlich vorkommen, dass die eine oder andere Schicht in einem bestimmten Netzwerk wichtiger ist als eine andere oder dass zum Beispiel ein Protokoll Funktionen zweier Schichten abdeckt oder auch nur eine Teilfunktion einer Schicht erbringt. Um diese Umstände deutlich zu machen, wendet man häufig anders aufgeteilte Schichtenmodelle an mit meist weniger, selten mehr Schichten. Dies gerade dann, wenn es um die konkrete Beschreibung einer bestimmten Art von Netzwerk geht.

Unterschied
zur Praxis

4.2.2 Das Schichtenmodell der Internetprotokolle

Im Bereich der TCP/IP-Netzwerkprotokolle, die unter dem Betriebssystem Unix und im Internet den Standard darstellen, wird zum Beispiel häufig ein Modell aus nur vier Schichten verwendet. Dies wird dem Wesen dieser Protokolle auch wesentlich eher gerecht als das OSI-Modell, denn die Internetprotokolle sind bereits einige Jahre älter als OSI. Es handelt sich um das Schichtenmodell des ursprünglichen ARPANETs, das vom US-Verteidigungsministerium finanziert wurde. Deshalb wird es meist als *DoD-Modell* (»Department of Defense«), manchmal auch als *DDN-Modell* (»Department of Defense Network«) bezeichnet.

Die vier Schichten bei TCP/IP-Netzwerken nach dem DDN Standard Protocol Handbook sind:

1. Netzzugangsschicht (Network Access Layer oder Link Layer)
2. Internetschicht (Internet Layer)

² Halb scherzhaft wird der Anwender des vernetzten Rechners manchmal als achte Schicht bezeichnet, und Probleme, die durch fehlerhafte Benutzung zustande kommen, nennt man entsprechend »Layer-8-Fehler«.

3. Host-zu-Host-Transportschicht (Host-to-Host Transport Layer oder einfach Transport Layer)
4. Anwendungsschicht (Application Layer)

Diese vier Schichten sind den konkreten Gegebenheiten von TCP/IP-Netzwerken angepasst, bei denen es zum Beispiel nur theoretisch möglich ist, von einer separaten Sitzungsschicht zu sprechen.

Das OSI-Referenzmodell kann mit dem DDN-Schichtenmodell deshalb nur grob in Beziehung gesetzt werden. Tabelle 4.1 zeigt, wie ein solcher Vergleich ungefähr aussehen könnte.

OSI-Modell	DDN-Modell
7. Anwendungsschicht	4. Anwendungsschicht
6. Darstellungsschicht	
5. Sitzungsschicht	
4. Transportschicht	3. Host-zu-Host-Transportschicht
3. Vermittlungsschicht	2. Internetschicht
2. Sicherungsschicht	1. Netzzugangsschicht
1. Bitübertragungsschicht	

Tabelle 4.1 Vergleich zwischen dem OSI-Referenzmodell und dem DDN-Schichtenmodell der Internetprotokolle

Die Bedeutung der Schichten des DDN-Modells

1. Die Netzzugangsschicht (*Network Access Layer* oder *Link Layer*) beschreibt, wie die physikalische Datenübertragung erfolgt. Die Aufgaben, die auf dieser Schicht zu erledigen sind, werden durch viele recht unterschiedliche Protokolle erbracht, einfach deshalb, weil es kaum eine Sorte von Netzwerkhardware gibt, auf der die Internetprotokolle noch nicht implementiert worden wären. Die eigentlichen Kernprotokolle, zu denen besonders die Namensgeber der Protokollfamilie gehören – also das Transmission Control Protocol (TCP) und das Internet Protocol (IP) – kümmern sich überhaupt nicht um die physikalischen Verhältnisse. Damit dies möglich ist, müssen auf dieser untersten Schicht die Bitübertragung und die Transportsicherung zuverlässig zur Verfügung gestellt werden.

Auf diese Weise entspricht die Netzzugangsschicht der Internetprotokolle den beiden untersten Schichten von OSI.

2. Die Internetschicht (*Internet Layer*), die im Wesentlichen der Vermittlungsschicht des OSI-Modells ähnelt, kümmert sich um die logische Adressierung der Rechner im Netz, durch die die grundsätzliche Identifizierbarkeit des jeweiligen Rechners sichergestellt wird. Eine weitere wichtige Aufgabe auf dieser Ebene ist das Routing, also die Weiterleitung von Daten über verschiedene physikalisch und/oder logisch getrennte Netze hinweg. Grundlage dieser Tätigkeiten ist das IP-Protokoll (Internet Protocol). Es definiert die IP-Adressen, 32 (in einer neueren Version 128) Bit breite Nummern, die den einzelnen Rechnern zugewiesen werden und die der Unterscheidung der einzelnen Netzwerke und der Rechner in diesen Netzen dienen. Außerdem versieht es jedes Datenpaket mit einem Header, also einer Zusatzinformation, die insbesondere die Quelladresse des sendenden Rechners und die Zieladresse des empfangenden Hosts enthält. Ein Datenpaket dieser Ebene wird als *Datagramm* bezeichnet.
3. Die Host-zu-Host-Transportschicht (*Host-to-Host Transport Layer*) kümmert sich um den zuverlässigen Datenaustausch zwischen den kommunizierenden Rechnern. Im Wesentlichen sind hier zwei verschiedene Protokolle verantwortlich (neben anderen, selten verwendeten). Diese beiden Protokolle werden jedoch niemals gleichzeitig, sondern immer alternativ verwendet. Das einfachere und weniger robuste *UDP* (User Datagram Protocol) stellt einen schlichten und wenig datenintensiven Mechanismus zur Verfügung, der die direkte Nutzung der IP-Datagramme für die Host-zu-Host-Kommunikation erlaubt. Hierbei wird keine virtuelle Verbindung zwischen den beiden Rechnern hergestellt; es findet also keine Kontrolle über einen kontinuierlichen Datenstrom statt. Das erheblich komplexere *TCP* (Transmission Control Protocol) hat zwar einen deutlich größeren Overhead (Daten-Mehraufwand durch Verwaltungsinformationen) als UDP, stellt aber dafür einen zuverlässigen Transportdienst dar: Es wird eine virtuelle Verbindung zwischen den beiden Hosts hergestellt. Sie besteht darin, dass die Datenpakete durchnummeriert werden und eine Übertragungskontrolle und eventuelle Neu-Übertragung jedes einzelnen Pakets stattfinden. Ob eine Anwendung nun UDP oder TCP verwendet, ist ihre eigene Entscheidung. Allgemein benutzen Dienste, die kontinuierlich größere Datenmengen transportieren müssen, eher TCP, während etwa Verwaltungs- und Konfigurationsdienste zu UDP tendieren.

Im Vergleich zum OSI-Modell nimmt die Host-zu-Host-Transportschicht insbesondere die Aufgaben der OSI-Transportschicht wahr; je nach konkretem Protokoll können auch Funktionen der Sitzungsschicht ausgemacht werden.

Der Begriff *Host* (Gastgeber) bezeichnet übrigens jeden Computer, der an ein Netzwerk angeschlossen ist und mit anderen Geräten kommuniziert. Es ist keine Bezeichnung für einen expliziten Dienstleistungsrechner, dieser (oder vielmehr die darauf ausgeführte Software) wird *Server* genannt. Der Host muss lediglich vom Router abgegrenzt werden, der Pakete nicht für sich selbst entgegennimmt, sondern an andere Netze weiterleitet. Da das Routing jedoch eine Ebene weiter unten stattfindet, ist es ein auf dieser

Schicht unsichtbares Detail – die Transportschicht ist nur für Rechner relevant, die Daten für den Eigenbedarf benötigen.³

4. Die Anwendungsschicht (*Application Layer*) schließlich definiert die Kommunikation zwischen den Anwendungsprogrammen auf den einzelnen Rechnern; hier arbeiten Protokolle wie HTTP für Webserver, FTP zur Dateiübertragung oder SMTP für den E-Mail-Versand. Die Schicht entspricht im Wesentlichen der gleichnamigen obersten Schicht des OSI-Referenzmodells, wobei auch einige Komponenten von dessen Darstellungsschicht mit hineinspielen. Beispielsweise bedarf HTML-Code, der von einer Webserver-Anwendung ausgeliefert wird, der Interpretation durch einen Browser; hier entspräche der HTML-Code selbst eher der Darstellungsschicht, die Browseranwendung aber der OSI-Anwendungsschicht. Sitzungsmanagement ist dagegen vom ursprünglichen Design her gar nicht vorgesehen; falls es benötigt wird, muss es durch die Anwendungen selbst bereitgestellt werden. In Kapitel 18, »Webserveranwendungen«, erfahren Sie beispielsweise, wie Sie mit Hilfe der Programmiersprache PHP Websessions verwalten.

4.2.3 Netzwerkkommunikation über die Schichten eines Schichtenmodells

In diesem Abschnitt wird erläutert, wie die Kommunikation über die Schichten von Schichtenmodellen funktioniert. Dazu werden zwei Beispiele angeboten: Das erste ist ein Alltagsbeispiel, das mit Computernetzwerken nichts zu tun hat, während das zweite ein einfaches Beispiel der Netzwerkkommunikation darstellt.

Ein Alltagsbeispiel

Neben der Datenübertragung im Netzwerk lassen sich auch völlig andere Arten der Kommunikation in Schichten gliedern. Beispielsweise kann die Kommunikation zwischen Gesprächspartnern am Telefon folgendermaßen unterteilt werden:

1. Die beiden Telefone sind physikalisch über eine Telefonleitung miteinander verbunden.
2. Die Verbindung zwischen den Telefonanschlüssen kommt dadurch zustande, dass einer der beiden Teilnehmer die eindeutige Nummer des anderen wählt und der andere das Gespräch annimmt.
3. Über die Telefonleitung werden Informationen in Form von elektromagnetischen Impulsen übertragen, beim klassischen Telefonnetz analog, bei ISDN, Mobilfunk oder VoIP dagegen digital.

³ Router verständigen sich allerdings auch mit Hilfe der weiter unten in diesem Kapitel vorgestellten Routing-Protokolle miteinander; diese Protokolle werden durchaus auf der Host-zu-Host-Transportschicht ausgeführt. Trotzdem ist dieses technische Detail für die eigentlichen Anwendungshosts uninteressant und unsichtbar.

4. An den beiden Endpunkten der Kommunikation sprechen die Gesprächspartner in ihre jeweilige Sprechmuschel hinein; die akustischen Signale werden in elektromagnetische Impulse umgewandelt (bei den digitalen Varianten kommt noch die Analog-digital-Wandlung hinzu). Umgekehrt hört ein Teilnehmer aus der Hörmuschel wiederum akustische Signale, die aus den übertragenen Impulsen zurückverwandelt wurden.
5. Die akustischen Signale, die die Gesprächspartner miteinander austauschen, werden zu Silben, Wörtern und schließlich Sätzen kombiniert.
6. Aus den einzelnen Bestandteilen der Sprache ergibt sich schließlich der eigentliche Inhalt der Nachrichten, die miteinander ausgetauscht werden.

Möglicherweise besteht dieses Kommunikationsmodell sogar aus noch mehr Schichten: Angenommen, die beiden Gesprächspartner sind leitende Angestellte oder gar Direktoren eines großen Unternehmens. Dann wird in aller Regel auf beiden Seiten eine Sekretärin die Gesprächsvermittlung vornehmen, möglicherweise ist sogar noch eine Telefonzentrale involviert. Noch komplizierter wird es beispielsweise, wenn Dolmetscher mitwirken.

Dieses einfache Beispiel zeigt deutlich, dass alle Ebenen, die sich oberhalb der untersten, also der physikalischen Ebene befinden, nur Abstraktionen darstellen, durch die den übertragenen Signalen jeweils ein neuer Sinnzusammenhang zugeordnet wird. Die eigentliche Verbindung erfolgt nämlich stets nur auf dieser untersten Ebene! Für jedes Schichtenmodell gilt daher zusammenfassend Folgendes:

Höhere
Schichten als
Abstraktion

Die Daten, die über einen Kommunikationskanal übertragen werden sollen, werden auf der Seite des Senders zunächst Schicht für Schicht nach unten weitergereicht und jeweils mit den spezifischen Zusatzinformationen für diese Schicht versehen. Schließlich werden sie über die unterste Schicht, die eigentliche physikalische Verbindung, übertragen. Auf der Empfängerseite werden sie dann wieder schichtweise nach oben weitergeleitet. Jede Schicht ermittelt die für sie bestimmten Informationen und regelt die Weiterleitung an die nächsthöhere Schicht.

Was in dem Telefonbeispiel geschieht, wird schematisch in Abbildung 4.1 dargestellt: Während die Gesprächspartner den Eindruck haben, in einem direkten Gespräch miteinander zu kommunizieren, geschieht in Wirklichkeit etwas erheblich Komplexeres: Die gesprochene Sprache (die eigentlich aus Silben beziehungsweise einzelnen Lauten besteht, die letztlich einfach nur Schallwellen sind) wird in eine andere Form von Information umgewandelt, über eine elektrische Leitung übertragen und auf der Empfängerseite wieder zusammengesetzt.

Wichtig ist außerdem, dass jede Schicht immer nur die für sie selbst bestimmten Informationen auswertet. Beim Empfang von Daten haben die niedrigeren Schichten überhaupt erst dafür gesorgt, dass die Informationen auf der entsprechenden Schicht angekommen sind, die Spezialinformationen der höheren Schichten sind der aktuellen Schicht dagegen

unbekannt. Sie muss lediglich anhand ihrer eigenen Informationen dafür sorgen, dass die Daten an die korrekte Stelle einer höheren Schicht ausgeliefert werden. Auf diese Weise ist jede Schicht virtuell mit der Schicht der gleichen Stufe auf der anderen Seite verbunden; das tatsächliche Zustandekommen dieser Verbindung ist für die jeweilige Schicht dagegen absolut unsichtbar.

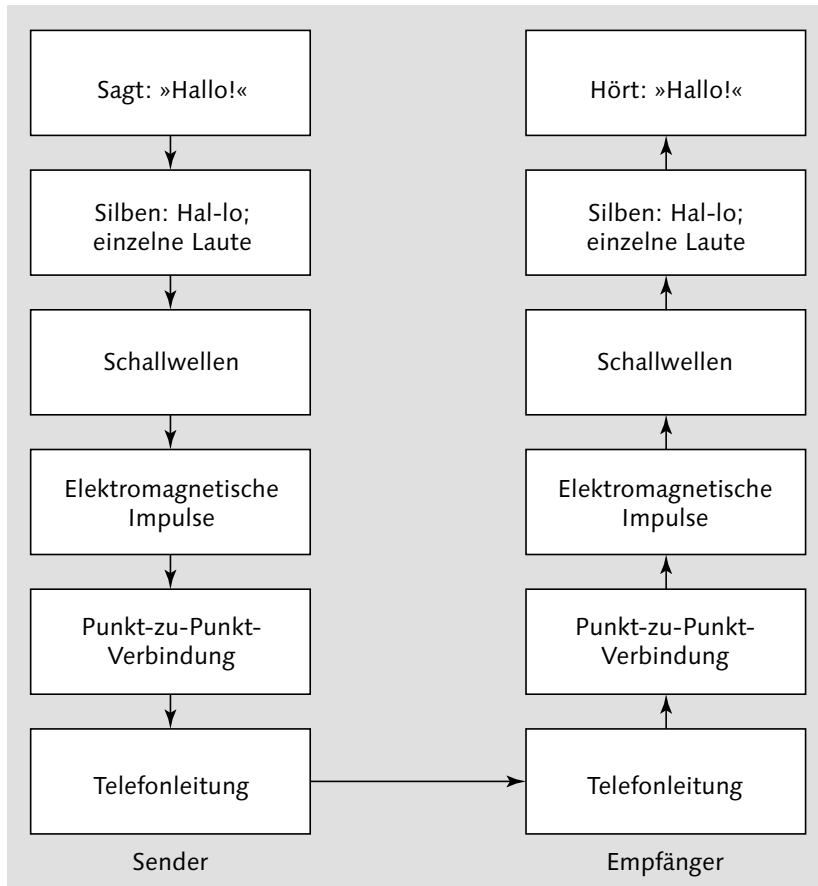


Abbildung 4.1 Schichtenmodell eines Telefongesprächs. Die tatsächliche Verbindung besteht nur auf der Ebene der Telefonleitung!

Ein Netzwerkbeispiel

Dieses zweite Beispiel – der Versand einer E-Mail an Galileo Press – zeigt, wie sich die beim Telefonbeispiel erläuterten Sachverhalte wieder auf Netzwerke übertragen lassen. Schematisch geschieht Folgendes:

1. In meinem E-Mail-Programm, zum Beispiel Mozilla Thunderbird, verfasse ich den eigentlichen Inhalt der Mail, als Empfänger setze ich *info@galileo-press.de* ein. Nachdem ich alles fertig geschrieben habe, drücke ich auf den Absendeknopf.
2. Da eine E-Mail eine in sich geschlossene Dateneinheit darstellt, die in ihrer ursprünglichen Reihenfolge beim Empfänger ankommen muss, wird der Transport durch das TCP-Protokoll übernommen, dessen eingebaute Datenflusskontrolle dafür sorgt, dass alle Daten vollständig und in der richtigen Reihenfolge übertragen werden.
3. Die Datenpakete, die durch das TCP-Protokoll angelegt wurden, werden nun durch das IP-Protokoll mit der korrekten Absender- und Empfängeradresse versehen. Diese Adressen haben nichts mit den nur auf der Anwendungsebene wichtigen E-Mail-Adressen zu tun. Vielmehr geht es darum, dass mein Rechner die Daten an den zuständigen Mailserver beziehungsweise an einen Vermittlungsrechner versendet.
4. Die fertig adressierten Datenpakete werden nun dem eigentlichen physikalischen Netzwerk anvertraut und entsprechend übertragen.

Auf der Empfängerseite – das heißt auf dem Serverrechner, der das Postfach *info@galileo-press.de* verwaltet – kommen die Daten dann folgendermaßen an:

1. Über die physikalische Netzwerkverbindung des Serverrechners treffen Datenpakete ein.
2. Auf der Ebene des IP-Protokolls werden die Datenpakete nach den zuständigen Transportdiensten sortiert und an diese weitergereicht – die E-Mail wird dem TCP-Dienst übergeben.
3. Der TCP-Dienst stellt fest, dass die entsprechenden Datenpakete für den Mailserver (gemeint ist das Programm, nicht der Rechner selbst) bestimmt sind, und reicht sie an diesen weiter.
4. Der Mailserver wertet die E-Mail-Adresse des Empfängers aus und speichert die Mail in dem Postfach *info@galileo-press.de*. Dort kann sie jederzeit vom berechtigten Empfänger abgeholt werden.

Die Übertragung der E-Mail vom empfangenden Server an das E-Mail-Programm des eigentlichen Empfängers funktioniert im Großen und Ganzen genauso, wenngleich auf der Anwendungsebene ein anderes Protokoll zum Einsatz kommt.

Abbildung 4.2 zeigt noch einmal schematisch, wie die Übertragung funktioniert. Mehr über die grundlegenden E-Mail-Protokolle erfahren Sie im Abschnitt »Verschiedene Internet-Anwendungsprotokolle« ab Seite 256.

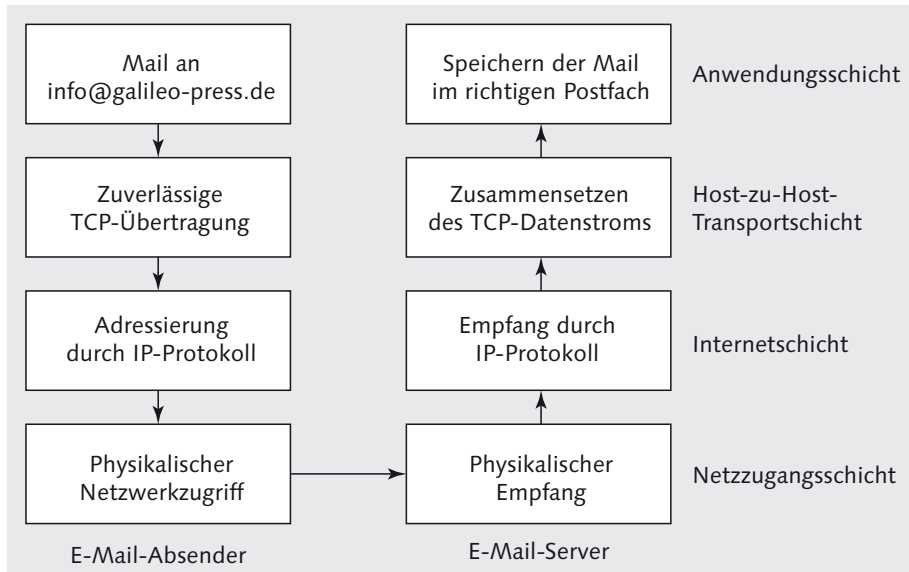


Abbildung 4.2 Übertragung einer E-Mail vom E-Mail-Programm des Absenders auf den Postfachserver des Empfängers. In der Regel sind allerdings mehrere Zwischenstationen beteiligt.

4.3 Klassifizierung von Netzwerken

Nachdem Sie nun mit Hilfe der Schichtenmodelle eine Möglichkeit kennengelernt haben, unterschiedliche Netzwerke in ihren Funktionen miteinander zu vergleichen, sollten Sie auch verstehen, worin sie sich unterscheiden. Es gibt diverse Unterscheidungsmerkmale, die zwar nicht genau den Schichten der Modelle entsprechen, aber doch ebenfalls mehrere Aspekte der einzelnen Netzwerke betreffen. Es handelt sich um die Unterscheidung nach der Reichweite des Netzwerks, der physikalischen Grundstruktur oder Topologie und zuletzt nach der zentralen oder dezentralen Verwendung des jeweiligen Netzes.

4.3.1 Die Reichweite des Netzwerks

Bei der Unterteilung der Netzwerke entsprechend der Reichweite – also nach der geographischen Größenordnung, die das Netzwerk überbrückt – werden insgesamt vier Stufen unterschieden:

- Das *Local Area Network* (LAN) – lokales Netzwerk – beschreibt ein Netzwerk, das an ein einzelnes zusammenhängendes Areal gebunden ist, also etwa einen Raum, ein Gebäude oder maximal ein zusammenhängendes (Firmen-)Gelände. LANs sind heutzutage weitverbreitet in Wirtschaftsunternehmen, Schulen und Universitäten oder anderen Organisationen und Instituten.

- ▶ Das *Metropolitan Area Network* (MAN) – Stadtgebiets-Netzwerk – bezeichnet ein Netz, das eine Stadt oder Gemeinde umfasst. Ein Beispiel wären die verschiedenen eigenen Netze von NetCologne in Köln.
- ▶ Das *Wide Area Network* (WAN) – Fernnetzwerk – ist ein Netz, das mehrere Städte, eine ganze Region oder sogar ein ganzes Land umfasst. In Deutschland gibt es beispielsweise das Deutsche Forschungsnetz (DFN).
- ▶ Das *Global Area Network* (GAN) – weltweites Netzwerk – ist über mehrere Länder, einen ganzen Kontinent oder sogar die ganze Welt verbreitet. Das bei weitem größte GAN ist heutzutage natürlich das Internet – im engeren Sinne ist ein GAN allerdings ein homogenes Netzwerk, während das Internet aus zahllosen Einzelnetzen mit unterschiedlichen Architekturen zusammengesetzt ist.

Es sei noch angemerkt, dass die drei Netzwerkarten, die größere Entfernungen überbrücken – also MAN, WAN und GAN – oftmals einfach unter dem Sammelnamen WAN zusammengefasst werden. Dies umso mehr, als alle drei Typen von Fernnetzen im Wesentlichen die gleiche Art von Technologie verwenden – oder genauer gesagt: Alle Arten von Technologien für Fernnetze werden von allen drei Netzarten benutzt.

Es gibt Fernnetze, die Wählleitungen, also einfache Telefonverbindungen, verwenden, sowohl das klassische Analog- als auch das digitale ISDN-Netz. Immer beliebter werden auch die neuen diversen DSL-Dienste, bei denen durch die Verwendung besonders hochfrequenter Signale über die normalen Kupferdrähte der Telefonleitungen wesentlich höhere Datenübertragungsraten erzielt werden. Zu einer besonderen Form der Wählleitung zählen Verbindungen über die digitalen GSM-Mobilfunknetze und deren Nachfolger GPRS und UMTS. Daneben existieren unterschiedliche Arten von Standleitungen, die für besonders häufig beanspruchte oder besonders zuverlässige Leitungen verwendet werden. Hierbei gibt es unter anderem spezielle DSL-Standleitungen oder Glasfasernetze. Auch drahtlose Übertragung, etwa über Funk- oder Satellitenverbindungen, spielt eine immer größere Rolle.

WAN-Technologien

Zu beachten ist allerdings, dass DSL, Wireless LAN und Mobilfunknetze im Wesentlichen die Technologien für den Zugang einzelner Hosts zu einem MAN oder WAN darstellen. Im Backbone-Bereich, also in der eigentlichen Netzwerkinfrastruktur, kommen vor allem Zeitmultiplexing-Verfahren über Glasfasernetze zum Einsatz. Sprache, Video und sonstige Daten werden dabei über Gigabit-Ethernet, SDH/SONET, ATM oder manchmal auch Frame-Relay übertragen. Eine zunehmende Bedeutung erlangten in den letzten Jahren auch DWDM-Verfahren (Dense Wavelength Division Multiplexing). Dabei werden über ein und denselben Lichtwellenleiter mehrere Signale mit unterschiedlicher Wellenlänge gleichzeitig versandt, was für extrem hohe Datenraten sorgt.

Lokale Netzwerke verwenden ebenfalls viele unterschiedliche technische Übertragungsarten. Allein für Ethernet, die häufigste Form der lokalen Vernetzung, werden unterschied-

LAN-Technologien

liche Arten von Koaxial-, Twisted-Pair- oder Glasfaserkabeln benutzt. Diese zeichnen sich durch verschiedene Übertragungsgeschwindigkeiten, mögliche maximale Entfernungen und natürlich auch unterschiedliche Kosten aus. Wireless LAN – der Betrieb von lokalen Netzwerken ohne Kabel über Funk, Infrarot oder Mikrowellen – erfreut sich auch zunehmender Beliebtheit. Abgesehen davon gibt es neben Ethernet viele andere Formen lokaler Netzwerke.

Die technologischen Grundlagen der Verkabelung, der Signalübermittlung und des Netzzugangs werden im Abschnitt »Netzwerkkarten, Netzwurkkabel und Netzzugangsverfahren« ab Seite 196 ausführlicher besprochen.

4.3.2 Die Netzwerktopologie

Die *Topologie* eines Netzwerks beschreibt, in welcher physikalischen Grundform die einzelnen Geräte organisiert sind. Manche Arten von Netzwerkhardware setzen eine bestimmte Topologie voraus, andere überlassen dem Einrichtenden die Entscheidung zwischen mehreren Möglichkeiten. Topologie ist normalerweise eine Eigenschaft lokaler Netzwerke oder gar einzelner Netzsegmente. Die meisten Fernnetze verbinden ohnehin nicht einzelne Rechner, sondern ganze Netzwerke an unterschiedlichen Orten miteinander.

Es werden im Wesentlichen folgende Grundformen unterschieden:

- ▶ Die *Bus-Topologie* beschreibt ein Netzwerk, bei dem die einzelnen Knoten (Anschlüsse) hintereinander an einem einzelnen Kabelstrang angeschlossen sind, dessen Enden nicht miteinander verbunden werden dürfen (sonst würde es sich um eine Ring-Topologie handeln!). Häufig werden die beiden Enden des Kabelstrangs durch Abschlusswiderstände (Terminatoren) abgeschlossen. Ein Beispiel für echte busförmige Netzwerke ist Ethernet über Koaxialkabel.
- ▶ Die *Stern-Topologie* ist die Form eines Netzes, bei dem alle Knoten mit jeweils eigenem Kabel an einem zentralen Gerät miteinander verbunden werden. Dieses zentrale Bindeglied heißt, je nach seiner genauen Funktionsweise, *Hub* oder *Switch*. Die Stern-Topologie wird zum Beispiel von Ethernet über Twisted-Pair-Kabel verwendet.
- ▶ Die *Ring-Topologie* ähnelt der Bus-Topologie insofern, als auch hier alle Knoten an einem zentralen Strang aufgereiht sind. Dieser zentrale Kabelstrang bildet jedoch einen geschlossenen Ring. Daraus ergibt sich automatisch eine Datenstromrichtung, in die die Datenpakete grundsätzlich weitergereicht werden. Bekanntestes Beispiel der ringförmigen Vernetzung ist Token Ring.
- ▶ Die *Baum-Topologie* schließlich ist eher ein Standard für den Zusammenschluss verschiedener Netzsegmente. Von einem zentralen Kabelstrang, gewissermaßen dem »Stamm« des Baums, gehen nach beliebigen Richtungen einzelne Verästelungen ab, an denen entweder eine einzelne Station oder ein ganzes Netz hängt.

Wichtig ist zu guter Letzt, dass ein Unterschied zwischen einer physikalischen und einer logischen Topologie bestehen kann, denn die äußere Form der Verkabelung (physikalische Topologie) kann einfach aus praktischen Erwägungen heraus gewählt worden sein, obwohl von der Funktion her eine völlig andere Struktur herrscht, nämlich die logische Topologie.

Ein gutes Beispiel für eine unterschiedliche physikalische und logische Struktur sind neuere Token-Ring-Varianten: Die eigentliche Vernetzung erfolgt sternförmig, logisch gesehen handelt es sich jedoch um einen Ring. Auch Ethernet über Twisted-Pair-Kabel verwendet physikalisch gesehen die Stern-Topologie, die logische Funktionsweise hängt von der Art des zentralen Verteilers ab: Ein Hub erzeugt letztlich die Funktion eines busförmigen Netzes, da es einen durchgehenden Strang enthält, an dem alle Stationen angeschlossen sind. Ein Switch dagegen stellt jeweils eine gesonderte Verbindung zwischen zwei Stationen her, die miteinander Daten austauschen; mithin handelt es sich hier auch logisch um die echte Sternform.

4.3.3 Der Zentralisierungsgrad des Netzwerks

Ein weiteres wichtiges Kriterium bei der Einteilung von Netzwerken in unterschiedliche Gruppen ist die Frage nach der Arbeitsaufteilung in ihnen. Kleine Arbeitsgruppen, die jeweils mit ihren Arbeitsplatzrechnern untereinander Dateien austauschen möchten, haben hier sicherlich andere Bedürfnisse als riesige Organisationen, in denen Tausende von Anwendern auf bestimmte Datenbestände zugreifen müssen. Deshalb werden die sogenannten Client-Server-Netzwerke, in denen zentrale Dienstleistungsrechner, die Server, arbeiten, von den Peer-to-Peer-Netzwerken unterschieden, in denen die einzelnen Computer gleichberechtigt Ressourcen freigeben und verwenden können.

- ▶ Das *Client-Server-Netzwerk* unterscheidet generell zwei Arten von beteiligten Rechnern: Der Server (Dienstleister) ist ein Computer, der den Arbeitsstationen der einzelnen Anwender an zentraler Stelle Ressourcen und Funktionen zur Verfügung stellt; der Client (Kunde) nimmt diese Dienstleistungen in Anspruch. Die Dienste, die von Servern angeboten werden, sind sehr vielfältig: Sie reichen vom einfachen Dateiserver, der Dateien im Netzwerk verteilt oder Festplattenplatz für andere freigibt, über Druckserver, Mail- und andere Kommunikationsserver bis hin zu ganz speziellen Diensten wie Datenbank- oder Anwendungsserver.
- ▶ Das *Peer-to-Peer-Netzwerk* besteht aus prinzipiell gleichberechtigten Arbeitsplatzrechnern (*peer* heißt etwa »Kollege«). Jeder Anwender ist in der Lage, Ressourcen seines eigenen Rechners an andere im Netzwerk freizugeben. Das heißt, dass alle Rechner im Netz bis zu einem gewissen Grad Serverdienste wahrnehmen.

In der Praxis sind allerdings Mischformen häufiger anzutreffen als reine Client-Server- oder absolute Peer-to-Peer-Netze. Beispielsweise könnte man sich in einem Unternehmen

die folgende Situation vorstellen: Aufgaben wie die direkte Kommunikation (E-Mail), der Zugang zum Internet (über einen Proxyserver oder einfach einen Router) oder Lösungen zum Backup (Datensicherung) werden durch zentrale Server zur Verfügung gestellt; der Zugang zu Dateien innerhalb der Abteilungen oder auf Drucker der Kollegen innerhalb eines Büros wird dagegen im Peer-to-Peer-Verfahren, unter Umgehung von Servern, geregelt.

Wichtig ist außerdem, zu verstehen, dass die Begriffe »Client« und »Server« im engeren Sinne nicht unbedingt spezifische Rechner, sondern besondere Softwarekomponenten bezeichnen.

**Definition
eines Servers**

Ein *Server* ist einfach ein Programm, das meist automatisch gestartet wird und im Hintergrund darauf »lauert«, irgendeine Dienstleistung zur Verfügung zu stellen. Allgemeiner werden solche Programme zum Beispiel im Unix-Umfeld als *Daemon* bezeichnet, unter Windows NT und seinen Nachfolgern (Windows 2000, XP, Vista und Windows 7) heißen sie Dienst (*Service*). Grundsätzlich kann ein solcher Serverdienst auf jedem beliebigen Rechner laufen – vorausgesetzt natürlich, er ist für die Hardwareplattform und das Betriebssystem dieses Rechners bestimmt. Der Grund für den Einsatz besonders leistungsfähiger Hardware (eben der Serverhardware) und spezialisierter Betriebssysteme liegt einfach in ihrer höheren Belastbarkeit, wenn viele Benutzer gleichzeitig diese Dienste benötigen.

**Definition
eines Clients**

Ein *Client* ist zunächst eine Software, die in der Lage ist, mit der Serversoftware zu kommunizieren; üblicherweise stellt sie dem Benutzer auch eine Schnittstelle zur Verfügung, um diese Kommunikation in Anspruch zu nehmen. So ist beispielsweise ein Webbrowser ein Client für das HTTP-Anwendungsprotokoll, er kommuniziert also mit HTTP-Servern. Interessanterweise können Webserver und Browser auch beide auf demselben Rechner laufen. Dies ist nützlich, um Webanwendungen zunächst lokal auszuprobieren.

Arten von Servern

Im Folgenden sollen einige Serverarten genauer vorgestellt werden. Sie sollten auf jeden Fall das oben Gesagte im Hinterkopf behalten: Es spielt überhaupt keine Rolle für die allgemeine Funktion, ob ein Serverdienst, also die Software, die diesen Dienst zur Verfügung stellt,

- ▶ mit anderen Diensten zusammen auf dem gleichen Rechner läuft,
- ▶ allein auf einem separaten Serverrechner ausgeführt wird oder
- ▶ sogar auf mehrere Server verteilt ist, weil ansonsten die Belastung zu groß wäre.

Letzteres ist insbesondere im Bereich öffentlicher WWW-Server sehr häufig zu finden, da populäre Sites wie etwa Suchmaschinen oder große Webshops sehr viel Datenverkehr zu verkraften haben. Hier werden sogenannte Load-Balancing-Systeme eingesetzt, die die

hereinstürmenden Anfragen automatisch möglichst gerecht auf mehrere physikalische Server verteilen.

Im Wesentlichen gibt es die folgenden wichtigen Arten von Serverdiensten:

- ▶ Fileserver
- ▶ Printserver
- ▶ Mailserver
- ▶ Webserver
- ▶ Verzeichnisdienst-Server
- ▶ Anwendungsserver und Serveranwendungen

In den folgenden Unterabschnitten wird jeder dieser Servertypen kurz vorgestellt; in späteren Kapiteln lernen Sie auch konkrete Beispiele für viele von ihnen kennen.

Fileserver

Der Fileserver (Dateiserver) stellt anderen Rechnern im Netzwerk freigegebene Verzeichnisse zur Verfügung. Auf diese Weise können sich die Anwender über einen zentralen Austauschpunkt gegenseitig Dateien zukommen lassen. Der Fileserver ist relativ stark an ein bestimmtes Betriebssystem oder eine Plattform gebunden. Erst allmählich setzen sich neuere Möglichkeiten durch, die in der Lage sind, auch unterschiedliche Rechner gleichzeitig zu bedienen. Denn die Besonderheit eines Fileservers ist, dass die Benutzer ihn völlig transparent genau so benutzen können wie die lokalen Dateisysteme ihres Arbeitsplatzrechners. In einem idealen (lokalen) Netzwerk sollte es dem normalen Anwender vollkommen egal sein, ob seine Dateien am Arbeitsplatz oder auf einem Fileserver zu finden sind.

Sehr wichtig ist im Zusammenhang mit Fileservern die Verwaltung von Zugriffsrechten, da nicht jede Datei für alle Benutzer gedacht ist.

Der Internetdienst FTP (File Transfer Protocol) ist übrigens kein vollwertiger Fileserver, sondern dient lediglich der einfachen Dateiübertragung. Die Informationen über die Dateien des entfernten Rechners sind nicht vollständig genug, um das Äquivalent eines Dateisystems abzubilden.

Informationen über Fileserver für die verschiedenen Systemplattformen finden Sie in den Kapiteln 6, »Windows«, 7, »Linux«, und 8, »Mac OS X«. FTP wird dagegen zusammen mit anderen Arten von Internetservern in Kapitel 14, »Weitere Internet-Serverdienste«, behandelt.

Printserver

Der Printserver (oder Druckserver) erlaubt mehreren Anwendern beziehungsweise Arbeitsstationen den gemeinsamen Zugriff auf einen Drucker. Die größte Herausforderung besteht darin, den einzelnen Arbeitsstationen automatisch den passenden Druckertreiber für ihr jeweiliges Betriebssystem zur Verfügung zu stellen, so dass diese den Drucker einfach verwenden können, ohne dass der Treiber zuvor noch einmal lokal installiert werden müsste.

Der Betrieb von Printservern ist besonders in Windows-Netzwerken weitverbreitet, da hier der Drucker gewöhnlich über ein USB-Kabel an einen einzelnen Rechner angeschlossen wird. Dieser Rechner wird dann so eingerichtet, dass er den Zugriff auf den Drucker auch den anderen Computern erlaubt.

Bei anderen Plattformen gibt es das Problem in dieser Form seltener. In klassischen Macintosh-Netzwerken ist es beispielsweise üblich – und viel bedienungsfreundlicher –, den Drucker unmittelbar per Ethernet ans Netzwerk anzuschließen, denn damit ist er automatisch für alle freigegeben.

In heterogenen Netzen war es bis vor wenigen Jahren verhältnismäßig schwierig, über Betriebssystemgrenzen hinweg gemeinsam auf einen Drucker zuzugreifen. Inzwischen ist jedoch beispielsweise das Drucksystem CUPS für alle Unix-Varianten verfügbar, das sogar Windows-Clients relativ problemlos bedienen kann.

Da Druckserver, genau wie Dateiserver, an das jeweilige Betriebssystem gebunden sind, werden die wichtigsten Varianten in Kapitel 6, »Windows«, und Kapitel 7, »Linux«, beschrieben.

Mailserver

Ein Server für elektronische Post (E-Mail) muss nicht immer bei einem Internetprovider installiert sein, sondern kann auch im lokalen Netz seinen Dienst verrichten. Denn erstens ist es in Unternehmen oder Organisationen oft von Vorteil, wenn die Mitarbeiter untereinander per E-Mail kommunizieren können, und zweitens ist es manchmal schon allein deshalb erforderlich, einen internen Mailserver zu betreiben, weil der Zugang zum Internet aus Sicherheitsgründen stark eingeschränkt ist und etwa die Kommunikation eines Arbeitsplatzrechners mit einem externen Mailserver gar nicht zulässt.

Internet- E-Mail

Obwohl im Lauf der Netzwerk-Entwicklungsgeschichte verschiedene Formen der elektronischen Post entstanden sind, gibt es heute eigentlich keine Alternative mehr zu Internet-E-Mail. Diese verwendet verschiedene Serverdienste zum Senden und Empfangen der E-Mail: Das SMTP-Protokoll (Simple Mail Transport Protocol) bestimmt, wie zu versendende E-Mails zu transportieren sind; POP3 (Post Office Protocol Version 3) oder das modernere, komfortablere IMAP (Internet Message Access Protocol) beschreiben ein Benutzerkonto (Postfach) für eingehende E-Mails sowie den Vorgang der »Abholung«.

Rein theoretisch kann Internet-E-Mail direkt zum einzelnen Host gesendet werden. Das ist aber insofern problematisch, als normale Arbeitsplatzrechner manchmal ausgeschaltet werden und private Einzelplatzrechner meist nur temporär über Wählleitungen mit dem Internet verbunden sind. Dies ist überhaupt der wichtigste Grund dafür, warum sich Posteingangsserver etabliert haben, auf denen die Mail für einen bestimmten Anwender im Prinzip vorgehalten wird, bis dieser sie abrufen.

Da die gewöhnliche Form der E-Mail auf den Standard-Internetprotokollen aufsetzt, gibt es übrigens kein Problem, sie plattform- und betriebssystemübergreifend zu verwenden.

Webserver

Ein Webserver (die exakte Bezeichnung ist eigentlich »HTTP-Server«) liefert auf Anfrage Webseiten über ein Netzwerk aus. In der Regel ist dieses Netzwerk das Internet. In den lokalen Netzen von Unternehmen und Institutionen setzt sich diese Form der Informationsübermittlung aber auch immer mehr durch. Ein solches lokales Netz, das Technologien und Dienste der Internetprotokolle verwendet, wird *Intranet* genannt. Der Anwender verwendet ein Anzeigeprogramm für Webseiten, den sogenannten *Browser*, um Webseiten anzufordern, anzusehen und auch, um die enthaltenen Hyperlinks – also Verknüpfungen zu anderen Dokumenten auf dem gleichen oder einem anderen Server – per Mausklick zu folgen.

Webseiten sind prinzipiell Textdokumente, die in der Strukturierungssprache HTML geschrieben werden. Viele dieser Dokumente liegen statisch auf dem Server und werden einfach auf Anfrage ausgeliefert. Eine wachsende Anzahl solcher Dokumente wird aber auch aus Vorlagen und dynamischen Daten, etwa aus einer Datenbank, kombiniert und dann an den anfragenden Host geschickt. Diese Entwicklung ist für Websites mit umfangreichem, schnell wechselndem Inhalt, etwa Online-Tageszeitungen oder die Kataloge in E-Commerce-Sites, unvermeidlich.

Webserver sind im Übrigen schon von ihrer Grundidee her dafür gedacht, Clients unter vielen verschiedenen Betriebssystemen zu bedienen. Falls es Inkompatibilitäten geben sollte, liegt das höchstens daran, dass bei der Erstellung des HTML-Codes Steuerbefehle verwendet wurden, die nicht jeder Browser versteht.

Der praktische Einsatz eines Webserver wird in Kapitel 13, »Server für Webanwendungen«, am wichtigsten Beispiel Apache beschrieben; die Kapitel 17 bis 19 kümmern sich dagegen um die Erstellung von Webinhalten und -anwendungen.

Verzeichnisdienst-Server

Verzeichnisdienste (*Directory Services*) gewinnen in der IT seit längerer Zeit stark an Bedeutung. Ein Verzeichnis ist in diesem Zusammenhang kein Dateisystem, sondern ein datenbankähnlicher, standardisierter Katalog von Benutzern, Computern, Peripheriegeräten

und Rechten in einem Netzwerk. Durch den Eintrag in das Verzeichnis können diese Informationen netzwerkweit abgerufen werden, so dass Verzeichnisdienste eine praktische Grundlage für zahlreiche Dienste legen, die in einer größeren Netzwerkumgebung die Arbeit der Administratoren und das Leben der Anwender erleichtern. Hier nur einige Beispiele:

- ▶ automatisierte Softwareverteilung und -installation
- ▶ mobile Benutzerprofile (Roaming User Profiles)
- ▶ zentralisierte Anmeldedienste (Single-Sign-on)
- ▶ rechner-, benutzer- und eigenschaftsbasierte Rechtekontrolle

In Kapitel 14, »Weitere Internet-Serverdienste«, wird OpenLDAP als Praxisbeispiel für einen Verzeichnisdienst vorgestellt.

Anwendungsserver und Serveranwendungen

Ein Anwendungsserver (*Application Server*) erlaubt den Benutzern die Verwendung von Anwendungsprogrammen, die sich eigentlich auf dem Server befinden, über das Netzwerk.

Einfache Anwen- dungsdienste

Bei der einfachsten Form des Anwendungsservers liegt der Datenbestand der Anwendung auf den Datenträgern des Servers, die Anwendung wird über das Netzwerk in den Arbeitsspeicher des Clients geladen und dort lokal ausgeführt. Der Unterschied zum Fileserver ist hier minimal: Es muss der Anwendung lediglich klar sein, dass eventuell notwendige Zusatzkomponenten oder Konfigurationsdaten nicht auf dem Rechner liegen, auf dem sie ausgeführt wird, sondern auf der Maschine, von der sie geladen wurde.

Bei vielen normalen Einzelplatz-Anwendungsprogrammen kann eine solche Einstellung vorgenommen werden. Diese Verwendung von Software hat vor allem zwei Vorteile: Erstens kann es weniger Arbeit bedeuten, ein Programm einmal auf dem Server statt auf mehreren Arbeitsplatzrechnern zu installieren, und zweitens können Kosten gespart werden – die meisten Softwarelizenzen gelten jeweils pro Rechner, auf dem das jeweilige Programm installiert ist. Wird eine Anwendung auf mehreren Rechnern genutzt, aber nicht gleichzeitig, so kann die Software auf dem Server installiert werden; damit werden die Lizenzgebühren dann nur einmal fällig.

Echte Client- Server- Anwen- dungen

Bei komplexeren Formen von Anwendungsservern werden Teile des Programms – oder unter Umständen auch das ganze Programm – direkt auf dem Server ausgeführt. Die möglichen Gründe hierfür sind im Einzelfall genauso vielfältig wie die verschiedenen Formen der Verwirklichung. Beispielsweise ist es bei großen Datenbanken üblich, dass der Datenbestand als solcher auf einem Server liegt, ebenso die grundlegende Datenverwaltungssoftware. Auf den Clients existieren dann in der Regel sogenannte *Frontends*, also Softwarekomponenten, die den Benutzern eine Bedienoberfläche für die eigentliche

Datenbank bereitstellen. (Den Gegenbegriff zum Frontend bildet das *Backend*, wobei es sich um einen nur für spezielle, angemeldete Benutzer zugänglichen Teil des Clients handelt, der der Verwaltung des Servers dient.)

Noch einen Schritt weiter gehen die sogenannten verteilten Anwendungen oder Enterprise-Anwendungen. Sie basieren in der Regel auf einem oder mehreren Datenbankservern für den Datenbestand, einem Anwendungsserver für die Geschäftsabläufe und diversen Client-Frontends (sowohl native Programme für bestimmte Betriebssysteme als auch Webanwendungen).

Eine andere Form der Serveranwendung existiert bei der Verwendung der sogenannten Terminalserver. Die einfachste Form, der Internetdienst Telnet, stellt dem Anwender eine Konsolenoberfläche zur Verfügung, über die sich von fern auf dem Server selbst mittels Kommandoeingabe arbeiten lässt. Das heißt, die Ein- und Ausgabe von zeilenorientierten Kommandos und Anwendungsprogrammen erfolgt auf dem Client, die eigentliche Ausführung auf dem Server – der eigene Rechner wird somit zu einem Terminal für den entfernten Server.

Eine sehr merkwürdige Form der Serversoftware ist in diesem Zusammenhang der aus dem Unix-Bereich stammende X-Window-Server oder einfach X-Server (siehe Kapitel 7, »Linux«). Die Bezeichnung »Server« für diese Software erscheint zunächst sehr irreführend, handelt es sich doch einfach um die Grundlage der grafischen Benutzeroberfläche (GUI) unter Unix. Der X-Server stellt den Anwendungsprogrammen seine Dienste zur Verfügung, die darauf zugreifen, um Fenster und andere Komponenten der GUI darzustellen.

Der X-Server

Die Tatsache, dass hier ein Dienst verfügbar gemacht wird, ist es übrigens, die den Begriff »Server« rechtfertigt. Dabei müssen Anwendung und X-Server auch nicht unbedingt auf dem gleichen Rechner laufen. Erstaunlicherweise läuft aber der X-Server auf dem Anwendungsclient! Denn da die Programmausführung auf dem entfernten Rechner stattfindet, aber die grafische Darstellung auf dem lokalen Rechner, muss hier dieser Dienst angeboten werden.

Terminalserver gibt es auch unter Windows Server 2008 und anderen Microsoft-Systemen; auch hier läuft die eigentliche Anwendung auf dem Server, der Client erlaubt deren Bedienung und Anzeige. Das Angebot solcher Anwendungsdienste über das Internet wird allmählich beliebter. Ein ASP (Application Service Provider) lässt Anwendungen wie beispielsweise Bürosoftware auf seinen Servern laufen; über eine spezielle Clientsoftware oder sogar über einen Webbrowser kann der Kunde darauf zugreifen und die angebotene Software von der ganzen Welt aus benutzen. Eine wesentlich einfachere Form solcher Serveranwendungen, die über das Web verwendet werden und die Sie wahrscheinlich gut kennen, ist der weitverbreitete webbasierte E-Mail-Dienst mit diversen Zusatzfunktionen, wie ihn GMX, Google Mail oder web.de anbieten.

**Thin Clients
statt PCs?**

Die Quintessenz der Verwendung von Anwendungsservern ist die von einigen Firmen (Sun Microsystems, Oracle) seit Jahren angestrebte Abschaffung der gewöhnlichen Personal Computer und deren Ersatz durch sogenannte Thin Clients – Rechner ohne Festplatte, die ihr Betriebssystem und die Anwendungsprogramme vollständig aus dem Netzwerk oder aus dem Internet beziehen. Allerdings konnte sich das Konzept bisher nicht recht durchsetzen. Das Hauptargument der entsprechenden Unternehmen, nämlich die geringeren Kosten, lässt sich angesichts des massiven Preisverfalls bei den »ausgewachsenen« PCs nicht aufrechterhalten.

4.4 Netzwerkkarten, Netzkabel und Netzzugangsverfahren

Im Laufe der Entwicklungsgeschichte der Netzwerke, die weiter oben in diesem Kapitel skizziert wurde, haben sich viele verschiedene Formen der Netzwerkhardware entwickelt. Jede von ihnen hatte zum Zeitpunkt ihrer Entstehung ihre Berechtigung, und dennoch haben sich einige auf breiter Front durchgesetzt, während andere schnell wieder vom Markt verschwunden sind. Die verbreitetste Art der Netzwerkhardware ist heute Ethernet in seinen vielfältigen Varianten.

**Die
IEEE-802-
Standards**

Analog zu den oben beschriebenen Schichtenmodellen – vor allem dem standardisierten OSI-Referenzmodell – gibt es auch Standards, die speziell die Netzwerkhardware und den Netzzugang betreffen, also die beiden untersten Ebenen des OSI-Modells. Die umfangreichste Sammlung ist IEEE 802 des Institute of Electrical and Electronics Engineers. Die Nummer 802 bezeichnet Jahr und Monat der ursprünglichen Festlegung, nämlich den Februar 1980. Innerhalb dieser Sammlung existiert eine Reihe verschiedener Unterstandards beziehungsweise Arbeitsgruppen. Zu den wichtigsten gehören 802.1 (allgemeine Netzwerkstandards), 802.3 (Netzzugangsverfahren CSMA/CD, besonders Ethernet) und 802.11 (drahtlose Netze). Tabelle 4.2 zeigt die vollständige Liste. Einige dieser Standards werden im Folgenden näher beschrieben.

IEEE-Gruppe	Bezeichnung
802.1	Internetworking
802.2	Logical Link Control (LLC)
802.3	CSMA/CD, Ethernet
802.3u	Fast Ethernet
802.3z	Gigabit Ethernet über Glasfaser
802.3ab	Gigabit Ethernet über Twisted Pair
802.4	Token-Bus-Zugriffsverfahren
802.5	Token-Ring-Zugriffsverfahren

Tabelle 4.2 Die IEEE-802-Arbeitsgruppen im Überblick

IEEE-Gruppe	Bezeichnung
802.6	Metropolitan Area Network (MAN)
802.7	Breitbandübertragungstechnologie
802.8	Glasfaserübertragungstechnologie
802.9	integrierte Sprach- und Datendienste
802.10	Netzwerksicherheit
802.11	drahtlose Netze
802.12	Demand-Priority-Verfahren
802.14	Breitband-Kabelfernsehen (CATV)
802.15	Wireless Personal Area Network (WPAN)
802.16	Broadband Wireless Access (BWA)
802.17	Resilient Packet Ring (RPR)
802.18	Radio Regulatory Technical Advisory Group (RRTAG)
802.19	Coexistence TAG
802.20	drahtlose Breitbandnetze
802.21	medienunabhängiges Handover
802.22	drahtlose Regionalnetze (WRAN)
802.30	100BaseX, 100BaseT, Fast Ethernet

Tabelle 4.2 Die IEEE-802-Arbeitsgruppen im Überblick (Forts.)

4.4.1 Die verschiedenen Ethernet-Standards

Ethernet ist heute der verbreitetste Standard für lokale Netze (LANs). Zehntausende von Hersteller weltweit unterstützen diese Art von Netzwerken mit ihrer Hard- und Software.

Jede Ethernet-Schnittstelle, also die Netzkarte oder der fest eingebaute Anschluss, ist mit einer weltweit einmaligen Identifikationsnummer ausgestattet, der sogenannten *MAC-Adresse* (für Media Access Control, einer der beiden Bestandteile der OSI-Netzzugangsschicht). Es handelt sich um eine 48 Bit lange Zahl, die in sechs hexadezimalen Blöcken zwischen 0 und 255 (00 bis FF hex) geschrieben wird, zum Beispiel 00-A0-C9-E8-5F-64.

Identifikation per
MAC-Adresse

Die Datenpakete – auf der Netzzugangsschicht *Frames* genannt – werden mit den MAC-Adressen der sendenden und der empfangenden Station versehen und in der Regel an alle Stationen im Segment versandt. Jede Station überprüft daraufhin, ob die Daten für sie bestimmt sind. Im Übrigen kann man Ethernet-Schnittstellen auch in den »Promiscuous Mode« schalten, in dem sie ohne Unterschied alle Daten entgegennehmen. Auf diese Weise kann der gesamte Datenverkehr in einem Netzsegment überwacht werden.

Die MAC-Adresse wird normalerweise nicht über das jeweilige Teilnetz hinaus weiterverbreitet.⁴ Nach außen ergäbe ihre Verwendung auch keinen Sinn, da das nächste Teilnetz auf einer Route womöglich noch nicht einmal zum Ethernet-Standard gehört.

Das Netzzugangsverfahren CSMA/CD

Es ist wichtig zu verstehen, dass mit dem Namen »Ethernet« gar keine einheitliche Netzwerkhardware bezeichnet wird. Vielmehr handelt es sich um einen Sammelnamen für diverse Netzwerkstandards, die ein bestimmtes Netzzugangsverfahren verwenden. Insofern sind alle Ethernet-Varianten auf der OSI-Schicht 2 identisch, unterscheiden sich aber auf der untersten Schicht.

Als der Vorläufer von Ethernet Ende der 60er Jahre des vorigen Jahrhunderts an der Universität von Hawaii konzipiert wurde (anfangs unter dem geografisch passenden Namen ALOHAnet), handelte es sich zunächst um Datenfunk. Diesem Umstand ist übrigens auch der endgültige Name zu verdanken: »ether«, zu Deutsch Äther, ist das gedachte Medium, durch das sich Funkwellen fortpflanzen. Erst in den 1970er Jahren wurde dasselbe Netzzugangsverfahren auch für die Datenübertragung per Kabel eingesetzt, und zwar zunächst über Koaxialkabel.

CSMA/ CD-Schema

Das gemeinsame Netzzugangsverfahren aller Ethernet-Formen trägt den Namen CSMA/CD: Carrier Sense Multiple Access with Collision Detection. Schematisch gesehen funktioniert dieses Verfahren wie folgt:

1. Ein Gerät, das Daten senden möchte, lauscht den Netzabschnitt ab, um festzustellen, ob dieser gerade frei ist, das heißt, ob gerade kein anderes Gerät sendet (*Carrier Sense*).
2. Wurde in Schritt 1 festgestellt, dass der Netzabschnitt frei ist, beginnt die Station mit dem Senden der Daten. Möglicherweise hat auch eine andere Station festgestellt, dass das Netz frei ist, und beginnt gleichzeitig ebenfalls mit dem Senden (*Multiple Access*).
3. Falls auf die beschriebene Art und Weise zwei Stationen gleichzeitig mit dem Senden begonnen haben, findet eine sogenannte Datenkollision statt, die von den beteiligten Stationen entdeckt wird (*Collision Detection*). Eine Station, die eine Kollision bemerkt, stellt das Senden von Nutzdaten ein und versendet stattdessen eine Warnmeldung (Jam Signal).
4. Eine Station, die wegen einer Datenkollision das Senden abgebrochen hat, beginnt nach einer zufällig gewählten Zeitspanne von wenigen Millisekunden erneut mit dem Senden. Genau diese Zufälligkeit der Zeitspanne, die nach einem komplizierten Verfahren berechnet wird, ist enorm wichtig, damit die beiden Stationen beim nächsten Versuch nicht wieder genau gleichzeitig mit dem Senden beginnen.

4 Ausnahmen: Das weiter unten beschriebene IPX/SPX-Protokoll verwendet die MAC-Adresse auch für die Adressierung auf der Netzwerkschicht, und die IP-Weiterentwicklung IPv6 benutzt die MAC-Adresse als Teil der 128 Bit langen IP-Adresse.

Das große Problem von Ethernet besteht darin, dass das CSMA/CD-Verfahren umso ineffektiver wird, je frequenter der jeweilige Netzabschnitt ist: Ab einem gewissen Grenzwert überschreitet die Anzahl der Datenkollisionen die Menge der Nutzdaten. Heutzutage umgeht man dieses Problem in der Regel durch die Verwendung sogenannter *Switches*, die für zwei miteinander kommunizierende Stationen jeweils eine exklusive Punkt-zu-Punkt-Verbindung einrichten. Wo diese Möglichkeit aufgrund veralteter, inkompatibler Hardware nicht zur Verfügung steht, muss ein Netz mit viel Datenverkehr stattdessen segmentiert, das heißt in kleinere Abschnitte unterteilt werden.

Ethernet-Hardware

Die Bezeichnungen der verschiedenen Arten der Hardware, die für Ethernet-Netzwerke verwendet werden, setzen sich aus der Übertragungsgeschwindigkeit des jeweiligen Netzes in MBit/s und einer spezifischen Bezeichnung für den Kabeltyp oder die maximal zulässige Kabellänge zusammen.

Wie bereits erwähnt, waren Koaxialkabel die ersten für Ethernet verwendeten Kabel. Der Aufbau dieser Kabel ist folgender: Im Zentrum befindet sich ein leitender Draht, der von einer Isolationsschicht umgeben ist, darüber befindet sich ein weiterer Ring aus leitendem Metall und außen natürlich wiederum eine Isolationsschicht. Das bekannteste Alltagsbeispiel für ein Koaxialkabel ist ein handelsübliches Fernsehantennenkabel.

Koaxialkabel

Es gibt zwei Arten von Koaxialkabeln, die für Ethernet eingesetzt werden:

- 10Base2: dünnes schwarzes Koaxialkabel

Die 10 steht für die maximale Datenübertragungsgeschwindigkeit des Netzes, in diesem Fall 10 MBit/s. Die nähere Spezifikation, die durch die 2 angegeben wird, betrifft die maximal zulässige Gesamtlänge eines 10Base2-Netzsegments von etwa 200 Metern (eigentlich 200 Yard, was ca. 185 Metern entspricht). In einem Segment dürfen sich maximal 30 Stationen befinden. Um eine größere Entfernung zu überbrücken oder mehr Stationen zu betreiben, muss eine Signalverstärkung durch sogenannte Repeater durchgeführt werden.

Alternative Bezeichnungen für diese Ethernet-Form sind »Thinnet Coaxial« oder »Cheapernet«, weil es sich früher um die billigste Art der Vernetzung handelte.

An der Netzwerkkarte wird an eine BNC-Buchse ein T-Adapter angeschlossen. An diesen beiden Seiten werden wiederum über BNC-Stecker die Koaxialkabel angeschlossen, die zu den T-Stücken der Netzwerkkarten der benachbarten Rechner führen. Der Mindestabstand zwischen zwei T-Stücken, das heißt die minimale Länge eines einzelnen Kabels, beträgt 50 cm. Das Netzwerk ist in einer Bus-Topologie organisiert; die T-Stücke des ersten und des letzten Rechners im Netzwerk werden auf je einer Seite mit einem Abschlusswiderstand oder Terminator versehen.

► 10Base5: dickes gelbes Koaxialkabel

Der Vorteil dieser auch »Thicknet Coaxial« genannten Variante besteht in der größeren zulässigen Länge des Netzsegments, nämlich – wie die Zahl 5 vermuten lässt – 500 Yard (knapp 460 m). Andererseits ist dieses erheblich dickere Kabel weniger flexibel als das dünnere 10Base2. Beispielsweise ist es schwieriger, solche Kabel durch verwinkelte Kabelkanäle zu ziehen.

Auf dem Kabel sitzen bei dieser Ethernet-Form sogenannte *Transceiver*, die über 15-polige Buchsen an die Netzwerkkarten angeschlossen werden. Zwischen zwei Transceivern muss ein Mindestabstand von 2,5 Metern eingehalten werden; das Kabel enthält ab Werk Markierungen in diesem Abstand. Die Transceiver werden an diesen Stellen einfach in das Kabel hineingebohrt (deshalb werden sie als »Vampirabzweige« bezeichnet). In einem Segment dürfen sich maximal 100 davon befinden. Auch dieses Netz ist busförmig, und beide Enden müssen durch Abschlusswiderstände terminiert werden.

Twisted-Pair-Kabel

Heutzutage wird Ethernet viel häufiger über Twisted-Pair-Kabel betrieben. Bei dieser Kabelsorte handelt es sich um einen verdrehten Kupfer-Zweidrahtleiter: Je zwei isolierte Kupferdrähte werden umeinandergewickelt. Dies verhindert die gegenseitige Beeinträchtigung der Signalqualität, die bei parallel zueinander verlaufenden Kabeln durch die elektromagnetischen Felder aufträte. In einem Twisted-Pair-Kabel verlaufen üblicherweise vier, manchmal auch acht solcher Doppeladern nebeneinander. Sie enden auf beiden Seiten in einem RJ-45-Stecker, der auch für ISDN-Anschlüsse verwendet wird. Bekannt sind solche Kabel vor allem durch ihre Verwendung als Telefonleitungen.

Man unterscheidet zwei verschiedene Grundarten von Twisted-Pair-Kabeln: UTP oder Unshielded Twisted Pair ist ein nicht abgeschirmter Zweidrahtleiter, STP (Shielded Twisted Pair) ein abgeschirmter, der eine höhere Signalqualität aufweist, so dass er zum Beispiel größere Entfernungen überbrücken kann.

Außerdem werden Twisted-Pair-Kabel in verschiedene Kategorien unterteilt, die unterschiedliche maximale Datenübertragungsraten zulassen. Diese sind in Tabelle 4.3 aufgelistet.

Kategorie	Übertragungsrate	Verwendungszweck
1	nicht festgelegt	Telefonie
2	4 MBit/s	ISDN
3	10 MBit/s	Ethernet; Token Ring
4	16 MBit/s	verschiedene
5	100 MBit/s	Fast Ethernet; allgemeiner Standard
6	200 MBit/s	verschiedene
7	600 MBit/s	verschiedene

Tabelle 4.3 Die verschiedenen Kategorien von Twisted-Pair-Kabeln

Alle über Twisted Pair verkabelten Arten von Ethernet weisen eine sternförmige Topologie auf, zumindest im physischen Sinn: Alle Stationen werden jeweils über ein eigenständiges Kabel an einen zentralen Verteiler angeschlossen. Der Vorteil dieser Form der Vernetzung besteht grundsätzlich darin, dass der Ausfall einer einzelnen Verbindung zwischen einem Rechner und dem Verteiler nicht zur Unterbrechung des gesamten Netzes führt, wie es beim busförmigen Koaxialkabel-Ethernet der Fall ist.

Der zentrale Verteiler wird in seiner einfacheren Form *Hub* genannt, die etwas teurere, aber leistungsfähigere Bauweise heißt *Switching Hub* oder kurz *Switch*. Die innere Struktur des Hubs ist letztlich busförmig, so dass es genau wie bei der Vernetzung über Koaxialkabel zu Datenkollisionen kommen kann. Ein Switch stellt dagegen für zwei Stationen, die miteinander kommunizieren möchten, eine exklusive Punkt-zu-Punkt-Verbindung bereit. Dies geschieht dadurch, dass ein Switch die MAC-Adressen aller Schnittstellen zwischenspeichert, an die er bereits Daten ausgeliefert hat, und auf diese Weise die restlichen Stationen nicht mehr mit Daten behelligen muss, die gar nicht für sie bestimmt sind. Da die Preise für Netzwerkzubehör in den letzten Jahren stark gesunken sind, gibt es eigentlich keinen Grund mehr, etwas anderes als einen Switch einzusetzen.

Hub versus
Switch

Bei einem Hub teilen sich alle Stationen die gesamte Übertragungsgeschwindigkeit, beim Switch steht sie dagegen jeder einzelnen Verbindung zur Verfügung.

Im Übrigen gibt es besondere Hubs, die als *Bridges* bezeichnet werden. Sie verbinden Ethernet-Netzwerke verschiedenen Typs miteinander, beispielsweise besitzen sie eine Reihe von RJ-45-Ports für Twisted-Pair-Kabel und zusätzlich einen Anschluss für 10Base2-BNC-Kabel; oder sie unterstützen einfach verschiedene maximale Übertragungsgeschwindigkeiten.

Hubs oder Switches weisen in der Regel 5 bis 24 Anschlüsse (Ports) auf, an die jeweils ein Gerät angeschlossen werden kann. Um Netzwerke mit mehr Geräten zu betreiben, sind diese Geräte kaskadierbar: Die meisten Hubs oder Switches besitzen einen speziellen Port, den sogenannten Uplink-Port, der über ein Kabel mit einem normalen Port eines weiteren Verteilers verbunden werden kann. Bei vielen Hubs/Switches kann ein einzelner Port über einen Schalter zwischen »Normal« und »Uplink« umgeschaltet werden.

Die einzige Ausnahme von der allgemeinen Regel, dass ein Hub oder Switch benötigt wird, bildet der Sonderfall, in dem nur zwei Rechner miteinander vernetzt werden sollen: Die beiden Stationen können unmittelbar über ein sogenanntes Crosslink-Kabel verbunden werden. Dieses spezielle Kabel besitzt überkreuzte Anschlusspaare anstelle der geradlinig verlaufenden bei normalen Twisted-Pair-Kabeln.

Zwei-
Rechner-
Netze
ohne Hub

Historisch betrachtet existieren zwei Arten von Ethernet über Twisted Pair, die unterschiedliche Übertragungsgeschwindigkeiten unterstützen:

- ▶ 10BaseT: Die Datenübertragungsrate beträgt 10 MBit/s.
- ▶ 100BaseT (auch Fast Ethernet genannt): Daten werden mit bis zu 100 MBit/s übertragen; hierzu sind mindestens UTP-Kabel der Kategorie 5 erforderlich. Genauer gesagt gibt es zwei Unterarten: 100BaseTX ist voll kompatibel mit 10BaseT, so dass das Netz schrittweise umgerüstet werden kann. 100BaseT4 verwendet dagegen alle vier Kupferdrahtpaare eines Twisted-Pair-Kabels und ist mit den anderen Standards inkompatibel; in der Praxis spielt es keine Rolle mehr.

Die meisten Netzwerkkarten, Hubs und Switches, die heute verkauft werden, unterstützen beide Übertragungsraten. Der zu verwendende Wert kann bei vielen Netzwerkkarten per Software eingestellt werden, häufiger wird er automatisch gewählt. Natürlich sollten Sie prinzipiell darauf achten, keine reine 10-MBit-Hardware mehr zu kaufen. Aber möglicherweise hat 100-MBit-Hardware der ersten Generation, die nicht auf 10 MBit/s heruntergeschaltet werden kann, sogar noch schlimmere Einschränkungen zur Folge. Zwar ist es bei normalen Standard-PCs ein Leichtes, die Netzwerkkarte gegen ein neueres Modell auszutauschen, um die Kompatibilität zu einer aktualisierten Netzwerkumgebung aufrechtzuerhalten, aber bei anderen Geräten wie beispielsweise Netzwerkdruckern oder kompakten Router-Boxen ist das eventuell nicht möglich. Solche Geräte sind mit einem reinen 100er-Netz eventuell nicht mehr kompatibel.

Neue Ethernet-Versionen

Noch neuere Formen von Ethernet erreichen Übertragungsraten von 1.000 MBit/s (Gigabit-Ethernet), entweder über Lichtwellenleiter (1000BaseFL für »Fiber Logic«) oder über mehradrige Twisted-Pair-Kabel (1000BaseTX). Bereits entwickelt, aber noch nicht weit verbreitet, sind Ethernet-Varianten mit 10 oder gar 100 GBit/s – anfangs nur über verschiedene Arten von Lichtwellenleitern, aber inzwischen ebenfalls über Twisted Pair.

4.4.2 Token Ring

Das Token-Ring-Netzwerk wurde ursprünglich 1972 von IBM konzipiert. Es besitzt grundsätzlich eine ringförmige Topologie, wobei dies heute nur noch den logischen Aufbau betrifft, während die physikalische Topologie sternförmig ist: Die Rechner werden über die bei Ethernet beschriebenen Twisted-Pair-Kabel an einen speziellen Token-Ring-Hub angeschlossen; innerhalb dieses Gerätes sind die Anschlüsse ringförmig miteinander verbunden.

Token Passing

Die Datenübertragungsrate von Token Ring betrug anfangs 4 MBit/s, wurde jedoch später auf 16 MBit/s erhöht. Auch wenn diese Geschwindigkeiten im Vergleich zu Fast Ethernet sehr gering erscheinen, kann Token Ring in gewisser Hinsicht dennoch damit konkurrieren. Es verwendet ein wesentlich ausgeklügelteres Netzzugangsverfahren als Ethernet, das Token-Passing-Verfahren. Dieses funktioniert schematisch gesehen folgendermaßen:

1. In Datenstromrichtung reichen die einzelnen Stationen ein spezielles Datenmuster weiter, das sogenannte Frei-Token.
2. Eine Station kann nur senden, wenn sie das Token erhalten hat. Sie wandelt das Frei-Token in ein Besetzt-Token um, hängt die zu übertragenden Daten an und sendet sie in Datenstromrichtung weiter.
3. Die Empfängerstation sendet nach erfolgreichem Empfang der Daten eine Bestätigung an den Absender.
4. Nach Erhalt der Bestätigung erzeugt der ursprüngliche Absender ein neues Frei-Token und versendet es wiederum in Datenstromrichtung.

Durch dieses Konzept können in einem solchen Netz grundsätzlich keine Datenkollisionen auftreten. Andere Probleme müssen dagegen speziell behandelt werden:

- Eine Station muss in der Lage sein, zu erkennen, dass ihr Nachfolger in Datenstromrichtung ausgefallen ist, um einen neuen Nachfolger zu bestimmen. Dazu wartet sie nach der Weitergabe des Tokens eine bestimmte Zeitspanne – die sogenannte Slot-Time – auf eine Bestätigung ihres Nachfolgers. Bleibt diese aus, so sendet sie ein neues Token aus. Wird auch dieses nicht bestätigt, so versendet die Station ein spezielles Who_Follows-Frame (»Wer folgt«) mit der Adresse ihres bisherigen Nachfolgers. Jeder Rechner kontrolliert diese Adresse; derjenige, um dessen Vorgänger es sich handelt, sendet ein Set_Successor-Frame (»Setze Nachfolger«) an den ursprünglichen Absender, der diesen Rechner als seinen neuen Nachfolger einträgt.
- Es muss die Möglichkeit geben, nachträglich hinzugekommene Rechner in den Ring aufzunehmen. Dazu sendet die Station, die gerade das Token besitzt, von Zeit zu Zeit eine Solicit_Successor-Anfrage aus. Eine neu in Betrieb genommene Station ist in der Lage, normale Frames zu empfangen, kann jedoch nur auf dieses spezielle Frame antworten, und zwar mit der Set_Successor-Meldung.
- Falls mehrere Stationen gleichzeitig dem Netz beitreten möchten und deshalb auf dieselbe Solicit_Successor-Anfrage antworten, versendet deren Absender ein Resolve_Contention-Frame (»Beendet Auseinandersetzung«). Daraufhin berechnet jeder Rechner aus seiner Adresse eine spezifische Wartezeit, so dass die Stationen letztlich nacheinander bei späteren Solicit_Successor-Anfragen dem Netzwerk beitreten.

Heute werden kaum noch neue Token-Ring-Netzwerke eingerichtet, im LAN-Bereich wird fast nur noch Ethernet eingesetzt. Die Technik, die sich als nächste allgemein durchzusetzen scheint, sind drahtlose Netzwerke nach IEEE 802.11, die im nächsten Abschnitt behandelt werden.

4.4.3 Drahtlose Netze

Schon seit sehr langer Zeit werden über drahtlose Technologien wie Funk, Mikrowellen, Satellit oder Infrarot nicht nur Sprache, Radio- und Fernsehsignale, sondern auch Daten übertragen. Die digitale (!) Datenübertragung per Funk war sogar die erste Anwendung der drahtlosen Nachrichtentechnik überhaupt: Der Funkpionier Guglielmo Marconi erfand die drahtlose Telegrafie mit Hilfe des binären Morse-Alphabets⁵ lange vor dem Sprechfunk.

Im Bereich der Netzwerke gibt es immer mehr Anwendungsfälle, bei denen sich der Einsatz drahtloser Techniken anbietet. Die folgenden Beispiele können als Anhaltspunkte dienen:

- ▶ In Privathaushalten wird WLAN inzwischen häufiger eingesetzt als kabelbasierte Netze. Da viele Menschen Laptops und/oder WLAN-fähige Mobiltelefone haben, ist dies auch viel praktischer. Für den Internetzugang kommen entsprechend oft WLAN-DSL-Router zum Einsatz, die eine Verbindung zwischen dem Internet und den Endgeräten vermitteln.
- ▶ In einem Unternehmen werden viele Außendienstmitarbeiter beschäftigt. Sie sind mit Notebooks ausgestattet und kommen nur gelegentlich in die Firmenzentrale.
- ▶ Eine Firma zieht in ein denkmalgeschütztes Haus ein, an dessen Bausubstanz nichts geändert werden darf – an das Verlegen von Kabelkanälen oder gar das Aufstemmen von Wänden für die Vernetzung ist nicht zu denken.
- ▶ Zwischen zwei Gebäuden eines Unternehmens verläuft eine öffentliche Straße; für die Überbrückung durch ein Kabel müsste ein langfristiges Genehmigungsverfahren mit ungewissem Ausgang eingeleitet werden.
- ▶ Auf LAN-Partys (Treffen von Netzwerkspielern), Messen, Kongressen oder ähnlichen Veranstaltungen müssen Unmengen von Computern für kurze Zeit vernetzt werden.

Für den Betrieb drahtloser Netzwerke kommen die verschiedensten Übertragungsmethoden zum Einsatz. Sie lassen sich nach folgenden Kriterien unterscheiden oder für den praktischen Einsatz auswählen:

- ▶ Welche maximale Entfernung zwischen zwei Stationen muss überbrückt werden?
- ▶ Besteht zwischen den einzelnen Standorten Sichtkontakt, oder befinden sich Wände oder andere Hindernisse zwischen ihnen?
- ▶ Soll eine freie Funkfrequenz genutzt werden, oder kann es auch eine lizenzpflichtige sein (Letzteres kann teuer werden)?
- ▶ Sind die vernetzten Geräte selbst stationär oder mobil?

⁵ Ganz und gar binär ist das Morsealphabet übrigens nicht: Neben »Lang« und »Kurz« muss die Pause zwischen zwei Zeichen als drittes mögliches Signal betrachtet werden, da die einzelnen Zeichen (übrigens gemäß ihrer Häufigkeit in englischen Texten) aus unterschiedlich vielen Einzelsignalen bestehen.

Diese diversen Auswahlkriterien zeigen bereits, dass es so etwas wie »das« drahtlose Netz nicht gibt. Für jeden Anwendungszweck bieten sich verschiedene Lösungen an, die sorgfältig geprüft werden müssen.

Genau wie bei der verkabelten Konkurrenz lassen sich auch hier verschiedene Kategorien von Reichweiten unterscheiden. Das WLAN (Wireless LAN, auch WiFi genannt) nach IEEE 802.11 ist ein drahtloses Netz für den Nahbereich, also für die Vernetzung innerhalb einer einzelnen Institution. Das WWAN (Wireless Wide Area Network) dagegen ist ein drahtloses Fernnetzwerk. Dazu zählen unter anderem Satellitenverbindungen.

In diesem Abschnitt wird nur das 802.11-kompatible WLAN beschrieben, da es sich seit seiner Einführung 1997 sehr schnell verbreitet hat und heute von allen Wireless-Technologien am häufigsten eingesetzt wird. 802.11 besteht aus mehreren Unterstandards, die sich in den Punkten Frequenzspektrum, Übertragungsrate und Funktechnologie unterscheiden. Sie alle werden jedoch über Funk betrieben; eine ursprünglich ebenfalls spezifizierte Infrarotvariante hat sich nicht durchgesetzt. Infrarot wird größtenteils für den drahtlosen Anschluss von Peripheriegeräten wie Mäusen oder Tastaturen verwendet. Tabelle 4.4 zeigt eine Übersicht über die wichtigsten gebräuchlichen 802.11-Varianten.

Standard	Frequenzbereich	Übertragungsrate	Funktechnik
802.11	2,4 GHz	1 oder 2 MBit/s	FHSS/DSSS
802.11a	5 GHz	bis zu 54 MBit/s	OFDM
802.11b	2,4 GHz	5,5/11/22 MBit/s	HR/DSSS
802.11g	2,4 GHz	bis zu 54 MBit/s	OFDM
802.11n6	2,4 und 5 GHz	bis zu 600 MBit/s	MIMO

Tabelle 4.4 Verschiedene Varianten von IEEE 802.11

Die Trägerfrequenz von 2,4 GHz wird vor allem deshalb am häufigsten verwendet, weil sie nicht lizenzpflichtig ist. Es handelt sich nämlich um diejenige Frequenz, mit der Mikrowellenherde arbeiten, da diese Wellenlänge Wassermoleküle am effektivsten erhitzt.

Die diversen Funkverfahren arbeiten alle mit verschiedenen Varianten der Frequency-Hopping-Methode, die auch im Mobilfunk eingesetzt wird: Nach einem bestimmten Schema werden die Funkwellen über mehrere Frequenzen übertragen, die mehrmals in der Sekunde wechseln. Dies ist erheblich weniger störanfällig als die Verwendung einer einzelnen Frequenz. Die grundlegende Technik wurde Mitte der 1930er Jahre von der österreichischen Schauspielerin Hedy Lamarr erfunden. Ihr damaliger Ehemann war Rüstungsfabrikant, und diese Funktechnik sollte helfen, Torpedos der Alliierten fernzusteuern, ohne dass die Signale abgefangen und verfälscht werden konnten. Im Einzelnen werden folgende Verfahren unterschieden:

Frequency
Hopping

- ▶ FHSS (Frequency Hopping Spread Spectrum): Die Frequenzen wechseln nach einem zufälligen Muster.
- ▶ DSSS (Direct Sequence Spread Spectrum): Es werden erheblich mehr Einzelfrequenzen verwendet; die Verteilung erfolgt nach einem komplexen mathematischen Verfahren.
- ▶ HR/DSSS (High Rate Direct Sequence Spread Spectrum): Entspricht DSSS mit speziellen Erweiterungen, die eine höhere Übertragungsrate ermöglichen.
- ▶ OFDM (Orthogonal Frequency Division Multiplexing): Jeder Kanal wird in mehrere Teilkanäle unterteilt, die Signale werden über alle Teilkanäle parallel übertragen. Aus diesem Grund ist OFDM das Übertragungsverfahren mit der höchsten Datenrate, andererseits aber auch das aufwendigste, so dass die entsprechende Hardware noch vor wenigen Jahren vergleichsweise teuer war.
- ▶ MIMO (Multiple Input/Multiple Output): Im Wesentlichen eine nochmals verbesserte OFDM-Variante, die wiederum erheblich höhere Übertragungsraten ermöglicht. Die Datenübertragung kann gleichzeitig über mehrere Frequenzbänder erfolgen.

WLAN-Hardware Der größte Teil der Wireless-LAN-Hardware, der momentan verkauft wird, basiert auf den Standards 802.11b und 802.11g (die meisten Geräte unterstützen wahlweise beide). Die Preise für Hardware dieser Variante sind in den letzten Jahren stark gefallen. Ein WLAN-Adapter ist inzwischen ab etwa 20 € erhältlich, sowohl als PCI-Karte als auch als PCMCIA- oder USB-Adapter. Außerdem sind Notebooks (und meist auch Desktop-PCs) ab Werk standardmäßig mit einer WLAN-Schnittstelle ausgestattet. Vorreiter dürften das PowerBook und das iBook von Apple gewesen sein; Apple fördert diese Technologie unter dem Namen AirPort seit vielen Jahren.

CSMA/CA Als Netzzugangsverfahren in 802.11-Netzen kommt CSMA/CA zum Einsatz (Carrier Sense Multiple Access with Collision Avoidance) – wie der Name vermuten lässt, werden Datenkollisionen von vornherein vermieden. Anders als bei CSMA/CD sendet eine Station, die ein freies Übertragungsmedium (in diesem Fall den entsprechenden Funkkanal) vorfindet, nicht einfach ihre Daten, sondern eine Sendeankündigung (RTS). Daraufhin warten andere sendebereite Stationen; und die erste Station, die das RTS gesendet hat, sendet ihre Daten, nachdem ihr die Empfängerstation ihre Empfangsbereitschaft (CTS) signalisiert hat. Abgeschlossen wird die Datenübertragung durch ein ACK-Signal, daraufhin kann die nächste Station ihren Sendewunsch bekanntgeben.

Das einfachste denkbare 802.11-WLAN besteht nur aus mehreren Rechnern mit entsprechender Schnittstelle, die auf direktem Weg miteinander kommunizieren. Ein solcher Aufbau wird als *Basic Service Set* (BSS) bezeichnet. Die Entfernung zwischen zwei beliebigen Stationen darf die maximale Reichweite des Funksignals nicht überschreiten, da jede Station die Signale nur senden und empfangen, aber nicht verstärken und weiterleiten kann. Da ein solches Netzwerk nicht mit anderen Netzen kommunizieren kann, wird es als *unabhängiges BSS* (Independent BSS oder kurz IBSS) bezeichnet. Derartige Netzwerke

sind sinnvoll für die sogenannte Ad-hoc-Vernetzung temporärer Zusammenkünfte wie Messen oder LAN-Partys.

Ein wenig komplexer wird der Aufbau eines BSS, wenn ein Access Point hinzugefügt wird. Im Grunde funktioniert ein Access Point wie ein Ethernet-Hub, denn sobald er vorhanden ist, kommunizieren die Stationen nicht mehr direkt miteinander, sondern senden die Frames an den Access Point, der sie an den gewünschten Empfänger weitergibt. Die Identifikation der einzelnen Stationen erfolgt wie bei Ethernet anhand einer 48 Bit langen MAC-Adresse. Ein BSS mit einem Access Point wird als *Infrastruktur-BSS* bezeichnet. Für die Reichweite des Netzes ist nur noch die Entfernung zwischen einer Station und dem Access Point ausschlaggebend.

Access Points

Die wichtigste Aufgabe eines Access Points besteht in seiner Funktion als Bridge. Er verbindet das WLAN mit einem Backbone-Netzwerk – meistens Twisted-Pair-Ethernet. Auf diese Weise kann das WLAN mit stationären Teilen des Netzes verbunden werden oder Zugang zu Servern und Routern erhalten, ohne dass diese selbst mit WLAN-Schnittstellen ausgestattet werden müssten.

Im Übrigen bildet ein Verbund aus miteinander vernetzten Access Points (entweder ebenfalls über Funk oder über Ethernet) ein sogenanntes *Extended Service Set* (ESS). Eine Station kann sich innerhalb eines ESS frei bewegen, weil die Access Points einander darüber auf dem Laufenden halten, welche Stationen sich gerade in ihrem Bereich befinden. Eine Station kann immer nur genau mit einem Access Point verbunden sein; sobald das Signal eines anderen Access Points stärker wird als das des bisherigen, meldet die Station sich bei ihrem alten Access Point ab und bei dem neuen an. Auf diese Weise werden Frames immer über den jeweils aktuellen Access Point an eine Station gesendet.

Ein zusätzlicher Nutzen von Access Points besteht darin, dass sie in der Lage sind, Frames zu puffern, die an bestimmte Stationen adressiert sind. Gerade Notebooks schalten im Standby-Modus oft auch die WLAN-Schnittstelle ab, um Strom zu sparen; sobald die Verbindung wieder aufgebaut wird, werden die zwischengespeicherten Frames ausgeliefert.

Das ESS-Modell wird immer häufiger für öffentlich verfügbare Netzwerkzugänge eingesetzt. In Bahnhöfen, Flughäfen oder Gaststätten stehen öffentlich zunehmend WLAN-Access-Points (auch *Hotspots* genannt) zur Verfügung, in die sich Notebook-Benutzer ohne weiteres einwählen können. Mittlerweile werden sogar die ersten Innenstädte fast flächendeckend mit einander überlappenden Access Points ausgestattet. Irgendwann könnte ein ähnlich dichtes Netz entstehen, wie es die Mobilfunkzellen inzwischen bilden.

Eine der größten Herausforderungen beim Einsatz von Wireless-Technologien bleibt die Sicherheit. Es ist zwar auch nicht weiter schwierig, das Signal von Ethernet-Kabeln abzuhören, aber immerhin ist es vergleichsweise einfach, den physikalischen Zugang zu ihnen zu kontrollieren. Bei WLAN kann dagegen im Grunde genommen jeder die Signale mit einer kompatiblen Antenne auffangen und analysieren, um unberechtigt Informationen zu

WLAN-Sicherheit

erhalten oder gar zu manipulieren. Das gilt umso mehr, als man die Grenzen der Funkreichweite niemals ganz genau auf die Größe des zu vernetzenden Gebäudes oder Geländes abstimmen kann; es ist also durchaus möglich, die Funkwellen außen zu empfangen.

Um ein Mindestmaß an Sicherheit zu gewährleisten, bot die ursprüngliche 802.11-Spezifikation eine optionale Verschlüsselung der Frames an. Allerdings ist diese Methode nicht besonders sicher; Sicherheitsexperten haben bereits bewiesen, dass die Verschlüsselung verhältnismäßig leicht zu knacken ist. Schon der Name dieser Technik, WEP (Wired Equivalent Privacy), sagt allzu deutlich aus, dass es nicht um mehr geht, als etwa dasselbe Maß an Sicherheit zu gewährleisten wie beim rein physikalischen Schutz verkabelter Netzwerke. Der Hauptverwendungszweck besteht auch gar nicht in der Geheimhaltung, sondern in der Abgrenzung eines Wireless-Netzes von benachbarten Netzen: Es ist ärgerlich, wenn jedes vorbeifahrende Fahrzeug, in dem sich zufälligerweise ein Laptop mit 802.11-Schnittstelle befindet, diesen vorübergehend automatisch ins Netz einbucht und wieder daraus verschwindet. Dies lässt sich allerdings zuverlässiger verhindern, indem der Access Point mit einer Whitelist zugelassener MAC-Adressen konfiguriert wird.

Inzwischen stehen mit WPA und WPA2 (WiFi Protected Access) stark verbesserte WLAN-Verschlüsselungsverfahren zur Verfügung.

4.4.4 Sonstige Zugangsarten

Einige weitere Formen von Netzwerkhardware und Netzzugang sollen hier kurz erwähnt, aber nicht weiter vertieft werden. Es handelt sich zum Teil um nicht mehr gebräuchliche, zum Teil aber auch um seltene Zugangsverfahren.

ATM

Der Asynchronous Transfer Mode (ATM) verwendet als physikalisches Trägermedium Telefonleitungen. Es handelt sich um einen Standard für Breitband-WAN-Verbindungen, der seit Anfang der 1990er Jahre von Telefongesellschaften angeboten wird. Die Vernetzung erfolgt über ATM-Switches, die genau wie Ethernet-Switches direkte Verbindungen zwischen den Stationen herstellen und auf diese Weise eine feste Übertragungsrate garantieren. Bei ATM beträgt diese 155 MBit/s.

TCP/IP
über ATM

ATM wurde kurz vor dem Siegeszug des Internets und der TCP/IP-Netze eingeführt und definierte ursprünglich nicht nur Hardware- und Netzzugangsstandards, sondern auch eigene Vermittlungs- und Transportverfahren, die mit IP-Netzen inkompatibel sind. Insbesondere scheiterte IP-Datentransport über ATM an der außerordentlich geringen ATM-Datenpaketgröße, die gerade einmal 53 Byte beträgt: 48 Byte Nutzdaten und ein nur 5 Byte großer Header.

Dies macht ATM besonders schnell und flexibel, gerät aber für TCP/IP zu einem fast unüberwindbaren Hindernis, da oft noch nicht einmal ein IP-Paket-Header in ein ATM-Paket

hineinpasst. Die IP-Pakete müssen auf komplizierte Art und Weise fragmentiert und am Ziel wieder zusammengesetzt werden. Zu diesem Zweck wurde eine Reihe von konkreten Verfahren entwickelt. Deshalb spielt ATM heute durchaus eine gewisse Rolle im Bereich der Breitband-Internetzugänge für Unternehmen, praktischere und kostengünstigere Technologien wie das im nächsten Abschnitt vorgestellte DSL liefern ihm jedoch sehr schnell den Rang ab.

In Backbones spielt ATM dagegen noch eine gewisse Rolle, aber selbst dort erfolgt allmählich ein Umstieg auf Technologien wie Gigabit Ethernet. Selbst Cisco Systems als wichtigster Hersteller von Backbone-Technologien rät seinen Kunden inzwischen zur Migration von ATM nach Ethernet: <http://www.cisco.com/warp/public/cc/pd/si/casi/ca2900xl/index.shtml>.

FDDI

Das Fiber Distributed Data Interface (FDDI) ist ein LAN- und WAN-Standard, der Glasfaserkabel verwendet. Als dieser Standard Anfang der 1990er Jahre konzipiert wurde, sollte er hauptsächlich als Backbone-Netz für Ethernet in Firmen eingesetzt werden, wurde jedoch in manchen Umgebungen auch als normales LAN eingeführt. Wie Token Ring ist FDDI in einer ringförmigen Topologie angeordnet. Allerdings existieren zwei unabhängige Ringe, von denen einer als Ersatzring fungiert, falls der Hauptring ausfällt. Genauer gesagt verläuft die Datenstromrichtung des zweiten Rings andersherum als die des ersten, so dass ein größerer Ring aus dem Zusammenschluss von beiden gebildet werden kann, falls ein bestimmtes Stück ausfällt.

Innerhalb eines FDDI-Rings können bis zu 500 Stationen betrieben werden. Der maximale Umfang des gesamten Rings beträgt 100 Kilometer, wobei jedoch spätestens alle 2 Kilometer ein Repeater eingesetzt werden muss. Die Übertragungsrate beträgt 100 MBit/s. Eine noch höhere Geschwindigkeit ist erreichbar, wenn beide Ringe parallel für die Datenübertragung verwendet werden, dann allerdings ohne zusätzliche Ausfallsicherheit.

Als Netzzugangsverfahren wird eine spezielle Form von Token Passing eingesetzt, das Timed-Token-Protokoll. Bei dieser Variante wird unmittelbar, nachdem eine Station mit dem Senden begonnen hat, wieder ein neues Frei-Token erzeugt, so dass gleichzeitig mehrere Datenübertragungsvorgänge im Netz stattfinden können.

ARCNet

Bei Attached Resource Computer Network (ARCNet) handelt es sich um eine bereits in den 1970er Jahren entwickelte proprietäre LAN-Lösung der Firma Datapoint. Daher existiert für sie kein IEEE-Standard. In Büronetzwerken wird ARCNet so gut wie gar nicht eingesetzt, es konnte sich jedoch einen Platz im Bereich der Industrie erobern, wo es insbesondere zur Vernetzung von Steuercomputern dient.

Es wird über Twisted-Pair-Kabel und Hubs in einer Stern-Topologie betrieben; die maximale Datenübertragungsrate beträgt 2,5 MBit/s. Ein neuerer Standard, ARCNet Plus, beherrscht 20 MBit/s. Als Netzzugangsverfahren wird Token Passing verwendet.

4.5 Datenfernübertragung

Nachdem im vorigen Abschnitt die verschiedenen Formen der LAN-Vernetzung und der WANs über Standleitungen beschrieben wurden, sollen nun diverse Verfahren der Datenfernübertragung (DFÜ) geschildert werden. Wie bereits angesprochen, wurde DFÜ bereits eingesetzt, als sie lediglich der Punkt-zu-Punkt-Kommunikation zwischen einzelnen Rechnern über eine direkte Telefonverbindung diente. Heute geht es in der Regel darum, den Zugang zu einem bestehenden Netzwerk oder (über einen kommerziellen Provider) zum Internet herzustellen.

DFÜ-Varianten Die erste Generation der DFÜ-Hardware, der umständliche und störanfällige Akustikkoppler, muss hier nicht mehr beschrieben werden. Die drei wesentlichen Technologien sind heute Modems für den Netzzugang über analoge Telefonleitungen, der Zugang über die digitale Telefonleitung ISDN sowie Hochfrequenz-Verbindungen über verschiedene DSL-Dienste. Diese verschiedenen Zugangsverfahren werden im Folgenden dargestellt.

Eine Gemeinsamkeit aller DFÜ-Netzwerkverbindungen besteht in der Notwendigkeit, die Datenübertragung über diese Leitungen zu standardisieren und bestimmte Grundlagen für die Protokolle der Vermittlungsschicht zu schaffen. Dafür werden spezielle Protokolle verwendet, die den Netzzugang über relativ langsame serielle Leitungen ermöglichen. Das traditionelle Protokoll für die Vernetzung über Wählleitungen war SLIP (Serial Line Interface Protocol). Allerdings besitzt es eine Reihe organisatorischer und technischer Mängel und wurde deshalb weitgehend durch PPP (Point-to-Point Protocol) ersetzt.

Aufgaben von PPP PPP kümmert sich um die Authentifizierung des Benutzers nach der Einwahl, indem Benutzername und Passwort übermittelt werden; anschließend verhandeln die beiden direkt miteinander verbundenen Punkte die eigentlichen Netzwerkdetails. Eine der wesentlichsten Fähigkeiten des Protokolls für Internetverbindungen besteht darin, dass der Einwahlknoten dem anwählenden Rechner automatisch eine IP-Adresse zuweisen kann, über die diese Netzwerkschnittstelle im gesamten Internet identifiziert wird.

Im Einzelnen erfolgen bei PPP also die folgenden Schritte:

- ▶ Wird eine Wählleitung (analog oder ISDN) verwendet, so stellt der Rechner des Benutzers über die entsprechende Schnittstelle eine Telefonverbindung her. Falls die Leitung besetzt sein sollte, werden spezielle frei konfigurierbare Maßnahmen getroffen; in der Regel erfolgt nach einer gewissen Wartezeit ein erneuter Wählversuch. Bei DSL-Leitun-

gen wird ebenfalls die Verbindung aktiviert, auch wenn man dies nicht als »Wählen« im klassischen Sinne bezeichnen kann.

- ▶ Der Einwahlknoten verlangt eine Authentifizierung, in der Regel in Form von Benutzername und Passwort. Die meisten PPP-Implementierungen in modernen Betriebssystemen übermitteln diese Daten nach einmaliger Konfiguration automatisch, ohne Zutun des Benutzers.
- ▶ Nachdem die Daten überprüft wurden, erfolgen entweder die Ablehnung des Benutzers und der Verbindungsabbau, oder die Netzwerkparameter werden ausgehandelt. Auch wenn PPP als Netzzugangsgrundlage für alle möglichen Protokolle der Vermittlungsschicht dienen kann, wird heute fast nur noch TCP/IP aufgesetzt. Zu diesem Zweck weist der PPP-Knotenpunkt des Internetproviders der seriellen Verbindung auf der Einwahlseite eine IP-Adresse zu, eine im gesamten Internet einmalige Identifikationsnummer. Ihr Konzept wird im nächsten Abschnitt genau beschrieben.

4.5.1 Netzwerkzugang per Modem (analoge Telefonleitung)

Für Modems wurden im Laufe der Zeit viele verschiedene Standards entwickelt, die sich insbesondere bezüglich ihrer maximalen Datenübertragungsrate voneinander unterscheiden. Der aktuelle Standard heißt V.90 und überträgt bis zu 56.600 Bit/s. Gemessen an den üblichen Geschwindigkeiten fest verdrahteter Netzwerke ist das natürlich sehr langsam, aber kein Vergleich zu den Modem-Geschwindigkeiten vergangener Jahrzehnte. Rein physikalisch scheint mit 56,6 KBit/s die Leistungsgrenze erreicht zu sein; etwas höhere Übertragungsraten lassen sich durch die heutzutage häufig verwendete Datenkomprimierung erzielen – je nach übertragener Datenart etwa bis zur doppelten Leistung.

Das Wort »Modem« ist eine Zusammensetzung aus den Abkürzungen für »Modulator« und »Demodulator«, weil es die digitalen Signale des Computers in frequenzmodulierte Analogimpulse umwandelt, diese über die Telefonleitung überträgt und am Ziel wieder zurückverwandelt. Zu diesem Zweck muss es auf der einen Seite mit dem Computer verbunden werden, zum Beispiel über USB, klassisch auch über die alte serielle Schnittstelle oder als PCI-Steckkarte. Auf der anderen Seite wird das Modem über einen TAE-Stecker an die Telefonbuchse angeschlossen. Praktisch sind in diesem Zusammenhang Dreifach-TAE-Dosen, die leicht nachgerüstet werden können: Sie verfügen über einen speziellen Anschluss (TAE-F) für ein Telefon in der Mitte und zwei Anschlüsse (TAE-N) für Zusatzgeräte – Faxgerät, Anrufbeantworter oder eben Modem – außen.

Das Modem wird vom Computer über ein einfaches ASCII-basiertes Protokoll gesteuert. Heutzutage verwenden praktisch alle Modems den sogenannten Hayes-Befehlssatz (benannt nach einem längst vergessenen Modem-Hersteller). Da die Befehle dieses Protokolls alle mit der Zeichenfolge »AT« beginnen, wird er mitunter auch als AT-Befehlssatz bezeichnet. Wichtige Befehle sind etwa folgende:

Der Hayes-Befehlssatz

- ▶ ATDT <Rufnummer>: DT steht für »Dial Tone« – eine Rufnummer wird im Tonwahlverfahren (Mehrfrequenzverfahren) angewählt.
- ▶ ATDP <Rufnummer>: »Dial Pulse« – eine Rufnummer wird im Pulswahlverfahren angewählt (heute sehr selten).
- ▶ ATH: »Hangup« – die Telefonverbindung wird unterbrochen, es wird »aufgelegt«.
- ▶ ATZ: Das Modem wird auf den Einschaltzustand zurückgesetzt (Reset).

Telefon- Wahl- verfahren

Das Tonwahlverfahren verwendet mehrere Töne unterschiedlicher Frequenzen, die zusammen die verschiedenen Ziffern und Funktionen des Telefons repräsentieren. Das Pulswahlverfahren sendet dagegen eine Reihe von »Klicktönen« – einen für eine 1, zwei für eine 2 und so weiter, bis zehn für eine 0. Seitdem alle Vermittlungsstellen in den deutschen Telefonnetzen digital sind, benötigt niemand mehr das langsamere und unzuverlässigere Pulswahlverfahren. Verwechseln Sie übrigens digitale Vermittlung nicht mit digitaler Signalübertragung – Letztere findet beispielsweise bei ISDN statt, das im nächsten Abschnitt behandelt wird.

Bevor eine Datenkommunikation überhaupt denkbar ist, müssen sich beide Seiten darüber einig sein, auf welche Weise sie die aufeinanderfolgenden einzelnen Bits überhaupt als Datenbits interpretieren sollen, was Datenbits, Stoppbits und eventuelle Parity-Bits angeht. Die meisten Internet-Einwahlpunkte verwenden heutzutage den Standard 8N1 (8 Datenbits, kein Parity-Bit, 1 Stoppbit). Dies muss in den Modem-Konfigurationsdaten eingetragen werden. Eine Beschreibung der verschiedenen Formen der seriellen Datenübertragung finden Sie in Kapitel 3, »Hardware«.

Der Hand- shake

Nach der Herstellung der eigentlichen Telefonverbindung findet der sogenannte Handshake (»Handschlag«) zwischen den beiden Gegenstellen statt. Es wird eine Übertragungskapazität ausgehandelt, die beide Seiten verwenden können. Erst nachdem die grundlegende Datenübertragung funktioniert, wird PPP eingesetzt, um die eigentliche Netzwerkverbindung über die Telefonverbindung herzustellen, wie weiter oben beschrieben wurde.

4.5.2 ISDN

Das Integrated Services Digital Network (etwa »Digitalnetzwerk mit integrierten Diensten«) oder kurz ISDN wurde in den 1980er Jahren von verschiedenen europäischen Telefongesellschaften eingeführt; die aktivste von ihnen dürfte die in Deutschland damals noch zuständige Deutsche Bundespost gewesen sein. Es handelt sich im Prinzip um die Übertragung digitaler Signale über klassische Kupferdraht-Telefonleitungen. Einem reinen Telefonkunden bietet ISDN zunächst die folgenden unmittelbaren Vorteile:

- ▶ Es werden zwei voneinander unabhängige Kanäle zur Verfügung gestellt; über beide kann gleichzeitig telefoniert, gefaxt oder Datenfernübertragung betrieben werden.

- ▶ Die Rufnummer eines Anrufers, der ebenfalls ISDN verwendet, wird übermittelt und im Display eines entsprechend ausgerüsteten Telefons angezeigt (dies funktioniert natürlich auch bei Standard-Telefonleitungen mit Digitalvermittlung).
- ▶ Während eine Verbindung besteht, kann ein weiterer Anruf angenommen werden. Entweder wird der jeweils andere Gesprächspartner in den Wartezustand versetzt (Makeln), oder der neue Anrufer wird mit dem bisherigen in dieselbe Verbindung aufgenommen (Dreierkonferenz).

Inzwischen stehen die meisten dieser Dienste dank der flächendeckend digitalen Vermittlung in Deutschland auch Analogkunden zur Verfügung; lediglich die beiden separaten Kanäle bleiben ISDN vorbehalten. Im Übrigen erhalten ISDN-Benutzer üblicherweise drei unabhängige Rufnummern (bei einigen Telefongesellschaften sogar noch mehr), die frei auf die jeweiligen Geräte verteilt werden können.

Technisch betrachtet werden sogar drei Kanäle zur Verfügung gestellt; die beiden B-Kanäle übertragen Telefongespräche, Faxe oder Daten mit jeweils 64 KBit/s, während der D-Kanal Dienstinformationen wie Rufnummernübermittlung oder Anklopfen mit 16 KBit/s überträgt. Für Internetverbindungen und andere Datenübertragungsmethoden besteht die Möglichkeit, beide B-Kanäle zu bündeln und auf diese Weise insgesamt eine Datenübertragungsrate von 128 KBit/s zu gewährleisten; natürlich entstehen dafür auch doppelte Kosten.

Die ISDN-Kanäle

In der Praxis funktioniert ISDN folgendermaßen: An die normale TAE-Telefonsteckdose wird ein spezielles ISDN-Endgerät namens NTBA angeschlossen. Es stellt zwei sogenannte S₀-Basisanschlüsse zur Verfügung. Diese verwenden die auch vom Twisted-Pair-Ethernet bekannten RJ-45-Stecker. An jeden dieser Anschlüsse kann ein ISDN-Endgerät angeschlossen werden, beispielsweise ein Telefon, ein Faxgerät oder ein ISDN-Adapter zur Computerdatenübertragung. Damit niemand seinen kompletten Telekommunikations-Gerätepark umstellen muss, werden als spezielle Form von ISDN-Endgeräten sogenannte TK-Anlagen angeboten, die wiederum den Anschluss analoger Endgeräte ermöglichen. Einige TK-Anlagen bieten auch durchgeschleifte S₀-Anschlüsse an, beispielsweise um ISDN-Geräte an einer praktischeren Stelle anzuschließen.

ISDN anschließen

Natürlich sollten Sie nicht versuchen, ein Analogmodem an eine TK-Anlage anzuschließen – es würde nicht etwa nur den eigentlichen Vorteil der ISDN-Datenübertragung zunichte machen, sondern funktioniert gar nicht. Da die übertragenen Daten nach außen wie ISDN aussehen, würde die Gegenstelle ihre Antworten mit einer Übertragungsgeschwindigkeit übermitteln, die das Modem nicht verarbeiten kann.

Um einen Computer mit ISDN zu verbinden, werden stattdessen verschiedene Formen von ISDN-Adaptern angeboten: als PCI-Steckkarten (früher gab es sogar ISA-Modelle), externe USB-Geräte oder PCMCIA-Geräte für Notebooks. Mittlerweile werden einige externe ISDN-Geräte auch mit integrierter TK-Anlagen-Funktion angeboten. Eine interne

ISDN-Adapter

ISDN-Karte sieht genauso aus wie eine moderne Ethernet-Karte, und es kann leicht passieren, dass man das Twisted-Pair-Netzkabel mit seinem baugleichen Stecker in die ISDN-Karte steckt und umgekehrt (selbstverständlich geschieht in diesem Fall gar nichts). Jedenfalls muss normalerweise ein ISDN-Kabel vom Anschluss des ISDN-Adapters zu einem S₀-Anschluss verlaufen.

Der Unterschied zwischen den Übertragungsraten eines heutigen Modems (56,6 KBit/s) und Ein-Kanal-ISDN (64 KBit/s) mag Ihnen nicht besonders groß erscheinen. Als jedoch das Internet für Privatkunden interessant zu werden begann, lag die Übertragungsrate der meisten Modems bei 9.600 oder 14.400 Bit/s; erst allmählich kamen Geräte mit 28.800 Bit/s hinzu. Abgesehen davon besitzt ISDN noch heute einen weiteren Vorteil gegenüber Modem-Verbindungen: Der Verbindungsaufbau geht fast ohne Verzögerung vonstatten, während es bei Modems zu Wartezeiten von etlichen Sekunden kommen kann, bis die Leitung bereit ist.

Klassische ISDN-Daten- übertragung

Neben der Verbindung zu einem Internetprovider, die mittlerweile wohl die häufigste über ISDN genutzte Dienstleistung ist, war über Jahre hinweg auch die direkte Verbindung zwischen Computern für die ISDN-Datenübertragung üblich. Sehr viele Macintosh-Benutzer verwendeten hierfür regelmäßig die Software Leonardo, während Windows-Benutzern beispielsweise das Programm Fritz!Data zur Verfügung stand, das mit der in Deutschland besonders populären Fritz!Card der Berliner Firma AVM geliefert wurde. Auch viele Mailbox/BBS-Systeme wurden in der zweiten Hälfte der 1990er Jahre auf ISDN umgestellt oder um eine ISDN-Einwahlmöglichkeit erweitert.

Insgesamt lässt sich feststellen, dass ISDN erst mit dem Aufkommen von Internetzugängen in Firmen und Privathaushalten wirklich populär wurde. Zuvor wurde es manchmal als Telefonleitung für Firmen eingesetzt, allerdings nicht annähernd so häufig, wie die Telefongesellschaften sich dies erhofft hatten. Als ISDN dann schließlich immer öfter benutzt wurde, reichte seine Übertragungsrate immer mehr Nutzern nicht mehr aus; das Bedürfnis nach multimediafähigen Breitband-Verbindungen wuchs deutlich. Dies führte zur Einführung der im nächsten Abschnitt vorgestellten DSL-Dienste.

4.5.3 DSL-Dienste

DSL ist die Abkürzung für »Digital Subscriber Line« (etwa »digitale Abonnement-Leitung«). Der Name soll verdeutlichen, dass es sich de facto um eine Standleitung anstelle einer Wählleitung handelt. Zur Einführung von DSL kam es, da es durch die allmähliche Verbesserung der Qualität von Telefonleitungen möglich wurde, Signale hoher Frequenz zu übertragen. Auch die meisten DSL-Dienste verwenden also genau wie Modem- und ISDN-Verbindungen die klassischen Kupferleitungen der Telefongesellschaften, die allerdings immer häufiger durch Glasfaserleitungen ersetzt oder ergänzt werden.

DSL-Varianten

Es existieren zwei grundsätzliche Varianten von DSL: Bei *Symmetric DSL* (SDSL) sind die Übertragungsraten für ankommende und ausgehende Daten identisch, bei *Asymmetric DSL* (ADSL) ist die ankommende Übertragungsrate höher als die ausgehende. SDSL ist eher für mittlere bis große Unternehmen geeignet, die nicht nur permanent auf das Internet zugreifen, um im Web zu recherchieren oder ihre E-Mails zu lesen, sondern bei denen auch eine Menge Datenausgänge stattfinden – beispielsweise für den Vor-Ort-Betrieb eigener Web- oder Mailserver oder für den Direktzugriff auf das Firmennetzwerk durch externe Mitarbeiter. ADSL dagegen wird häufiger von Privatkunden oder kleineren Firmen eingesetzt, die recht hohe Datenmengen aus dem Internet herunterladen, aber nur verhältnismäßig wenige und eher kleine Uploads durchführen.

Übliche ADSL-Angebote wie T-DSL der Deutschen Telekom stellten ursprünglich eine Download-Rate von 1.024 KBit/s und eine Upload-Rate von 128 KBit/s zur Verfügung, teilweise sogar noch weniger. Aktuelle ADSL-Anschlüsse des klassischen Typs sind dagegen mit Download-Geschwindigkeiten von 2 bis 8 MBit/s ausgestattet. Auch die umgekehrte Datenrate wurde entsprechend vervielfacht. Die neueren Typen ADSL2 und ADSL2+, die allerdings eine nicht überall verfügbare hohe Leitungsqualität und spezielle Hardware benötigen, schaffen sogar 16 beziehungsweise 25 MBit/s im Download.

SDSL-Lösungen werden von sehr vielen kommerziellen Providern angeboten und stellen je nach Bedarf viele verschiedene Übertragungsraten von 1.024 KBit/s bis hin zu mehreren MBit/s zur Verfügung. Sie sind deutlich teurer als ADSL-Angebote mit der gleichen oder gar mit einer höheren Übertragungsrate, weil die entsprechende Technik aufwendiger ist.

Anders als bei Modem- oder ISDN-Angeboten werden die Gebühren für DSL-Anschlüsse in der Regel nicht nach der Nutzungsdauer berechnet, sondern als sogenannte Flatrate für beliebig lange Onlinezeiten. Einige Provider verwenden allerdings eine Volumenbeschränkung, das heißt, ohne Aufpreis darf monatlich nur eine bestimmte Datenmenge transferiert werden.

Neben den DSL-Angeboten, die über normale Telefonleitungen laufen, werden seit einiger Zeit auch spezielle Lösungen angeboten. Eine davon ist die Internetverbindung über das Glasfaserkabel des Kabelfernsehens. Da dieses Kabel für das Passivmedium Fernsehen erfunden wurde, besitzt es in seiner ursprünglichen Version keine Rückkanal-Fähigkeit. Es können Daten empfangen, aber nicht gesendet werden; noch nicht einmal die Anforderung einer URL kann abgesetzt werden. Erst allmählich wird der Rückkanal derjenigen Kabelnetze nachgerüstet, die die Deutsche Telekom bereits verkauft hat.

Seltenere
Varianten

Eine andere Lösung ist besonders interessant für kleine Gemeinden, die so weit von der nächsten größeren Stadt entfernt liegen, dass sich eine Nachrüstung der bestehenden Telefonleitungen oder Fernsehkabelnetze nicht lohnt: die Kommunikation mit einem

Satelliten über eine Parabolantenne. Diese Lösung bietet beispielsweise die Firma Teles unter dem Namen skyDSL an. Die Datenübertragungsrate beträgt bis zu 24.000 KBit/s. Über den Satelliten ist allerdings nur der Datenempfang möglich; Anfragen und andere Sendevorgänge erfolgen über Analogmodem, ISDN oder GSM-Mobilfunk.

ADSL anschließen

An den TAE-Anschluss eines ADSL-Kunden wird ein sogenannter *Splitter* angeschlossen – eine Frequenzweiche, die die hochfrequenten DSL-Signale und die niedrigfrequenten normalen Telefonsignale voneinander trennt. Den Ausgang für die Telefonsignale bietet wiederum ein TAE-Anschluss, an den entweder ein Analog-Telefon oder ein NTBA angeschlossen wird, je nachdem, ob ADSL mit einem Analog- oder mit einem ISDN-Telefonanschluss kombiniert wird.

Den Ausgang für die speziellen ADSL-Signale bietet eine Twisted-Pair-Buchse vom Typ RJ-11. An diesen Anschluss wird in der Regel ein ADSL-Modem angeschlossen, das dann über USB oder Twisted-Pair-Ethernet mit dem Computer verbunden wird. Natürlich ist die Bezeichnung »ADSL-Modem« technisch gesehen Unfug. Bei DSL findet keinerlei Analog-Digital-Umwandlung statt. Dennoch ist der Begriff »Modem« für das Gerät weitverbreitet, weil es den Computer mit einer seriellen Fernleitung verbindet. Beim Anschluss über eine Ethernet-Schnittstelle kommt eine spezielle PPP-Variante namens PPPoE (PPP over Ethernet) zum Einsatz.

Statt der reinen ADSL-Modems zum Anschluss eines einzelnen Rechners werden inzwischen meist DSL-Router verwendet, die gleich einem gesamten Netzwerk per Ethernet, WLAN oder beidem den Internetzugang bereitstellen. Inzwischen erlauben auch die meisten Provider den Einsatz solcher Router; früher waren die günstigsten Tarife dagegen vielfach auf einen Einzelrechner beschränkt.

Internetzugänge über Mobilfunk

Neben den hier behandelten stationären DFÜ-Verbindungen werden auch diejenigen über Mobilfunk immer wichtiger. Über die seit den 1990er Jahren errichteten GSM-Netze (in Deutschland beispielsweise D1, D2, E-Plus und so weiter) kam ursprünglich vor allem ein Verfahren namens GPRS (General Packet Radio Service) zum Einsatz; es wurde mehrfach in Details verbessert, bietet aber noch immer keine allzu hohen Datentransferraten – sie liegen bei 53,6 KBit/s im Download und 26,8 KBit/s im Upload. Neuere Verfahren sind zum Teil erheblich schneller:

- ▶ EDGE: Download 217,6 KBit/s, Upload 108,8 KBit/s
- ▶ UMTS: Download und Upload 384 KBit/s
- ▶ HSPA: Download (HSDPA) 7,2 Mbit/s, Upload (HSUPA) 1,4 Mbit/s

Mobilfunkzugänge können entweder für Web-, E-Mail- und andere Netzwerksoftware auf dem Mobiltelefon selbst verwendet werden, oder aber das Handy dient – beispielsweise über Bluetooth – als Mobilfunkmodem für einen Laptop. Speziell für UMTS oder HSPA gibt es auch eigenständige Netzwerkzugangsgeschäfte, die per USB an den Rechner angeschlossen werden und beinahe überall einen Internetzugang mit annehmbarer Geschwindigkeit bieten.

Um die Vorteile eines solchen Anschlusses wirklich zu nutzen, sollte aus Kostengründen ein Datenflatrate-Vertrag mit dem Mobilfunkanbieter abgeschlossen werden. Beachten Sie aber, dass diese Zugänge sehr oft eine Transfervolumenbeschränkung enthalten. Nachdem das Volumen für den entsprechenden Monat aufgebraucht ist, steht in der Regel nur noch eine langsamere Verbindung wie etwa EDGE zur Verfügung. Glücklicherweise verfügen praktisch alle modernen Smartphones auch über WLAN, so dass zu Hause oder am Arbeitsplatz meist keine Datenübertragung über Mobilfunk erforderlich ist.

Für den Urlaub oder ähnliche Gelegenheiten werden auch Prepaid-SIM-Karten oder USB-Sticks angeboten. Auf keinen Fall sollten Sie den Fehler machen, im Ausland ungeprüft das sogenannte Daten-Roaming Ihres Mobiltelefons zu aktivieren, da es geradezu irrational teuer ist (zum Beispiel 1 Euro pro 50 Kilobyte übertragener Daten). Verschiedene Initiativen der EU-Kommission, diese Tarife zu senken, zeigen erst allmählich Wirkung.

4.6 Die TCP/IP-Protokollfamilie

Nach einigen halbherzigen Versuchen, das OSI-Referenzmodell durch konkrete Protokolle tatsächlich zu implementieren, bemerkte man letzten Endes, dass die bereits Jahre zuvor entwickelten Internetprotokolle hervorragend als flexible, skalierbare und universelle Netzwerkprotokollfamilie einsetzbar sind. Die rasante Ausbreitung des Internets und die freie Verfügbarkeit sorgten dafür, dass diese Protokolle heute häufiger als jeder andere Protokollstapel eingesetzt werden.

Abbildung 4.3 zeigt eine konkrete Version des weiter oben bereits vorgestellten TCP/IP-Protokollstapels: Auf jeder Ebene sind einige der Protokolle zu erkennen, die dort arbeiten können. Die meisten davon werden in den folgenden Abschnitten genau erläutert; die Netzzugangsprotokolle der untersten Schicht wurden bereits weiter oben vorgestellt. Ganz unten habe ich zusätzlich einige Beispiele für die Hardware angegeben, auch wenn sie kein Teil des eigentlichen TCP/IP-Stapels ist.

Zwischen der Hardware und dem Netzzugang auf der einen und den anwendungsorientierten Protokollen auf der anderen Seite befinden sich die Protokolle der Vermittlungs- und der Transportschicht. Insgesamt werden alle Protokolle, die auf den verschiedenen Ebenen eines Schichtenmodells zusammenarbeiten, als *Protokollstapel* oder auch *Protokollfamilie* bezeichnet. Allerdings konzentriert sich der Schwerpunkt von TCP/IP auf die beiden mittleren Ebenen des Internet-Protokollstapels. Sie können zum einen auf fast

Der Internet-
Protokoll-
stapel

Index

- !, Operator 465
- !~, Operator 466
- !~, Perl-Operator 560
- #, Perl-Kommentar 504
- #define, Präprozessor-Direktive 482
- #endif, Präprozessor-Direktive 482
- #ifdef, Präprozessor-Direktive 482
- #include, C-Präprozessor-Direktive 459
- #include, Präprozessor-Direktive 482
- \$, Operator in RegExp 559
- \$, Perl-Variable 499, 501
- \$, PHP-Variable 925
- \$/, Perl-Variable 840
- \$_, Perl-Variable 502
- \$0, Unix-Systemvariable 372
- %, Operator 465
- %, Perl-Variable 501
- &&, Operator
 - als Perl-Fallunterscheidung* 507
 - C 465
 - Perl 507
- &, Dereferenzierungs-Operator 475
- &, Operator, C 465
- *, Operator 465
 - RegExp 557
- ++, Operator 467
- +, Operator 465
 - JavaScript-String-Verkettung* 997
 - RegExp 557
 - String-Verkettung, Java* 488
- , Operator 465, 478
- , Operator 467
- ., Perl-Operator 506
- .bashrc, Unix-Konfigurations-datei 373
- /, Operator 465
- //, Java-Kommentar 489
- /etc/exports, NFS-Konfigurations-datei 431
- /etc/passwd, Unix-Konfigurations-datei 369
- /etc/profile, Unix-Konfigurations-datei 373
- /etc/shadow, Unix-Konfigurations-datei 370
- <a>, HTML-Tag 880
- <address>, HTML-Tag 876
- <applet>, HTML-Tag 904
- <area>, HTML-Tag 887
- <article>, HTML5-Tag 874
- <aside>, HTML5-Tag 874
- <audio>, HTML5-Tag 903
- , HTML-Tag 875
- <body>, HTML-Tag 868
-
, HTML-Tag 870
- <caption>, HTML-Tag 888
- <code>, HTML-Tag 876
- <col>, HTML-Tag 892
- <colgroup>, HTML-Tag 892
- <dl>, HTML-Tag 879
- <dt>, HTML-Tag 879
- , HTML-Tag 875
- <embed>, HTML-Tag 902
- <figcaption>, HTML5-Tag 875
- <figure>, HTML5-Tag 874
- <footer>, HTML5-Tag 874
- <form>, HTML-Tag 894
- <h1> bis <h6>, HTML-Tags 873
- <head>, HTML-Tag 868
- <header>, HTML5-Tag 874
- <hgroup>, HTML5-Tag 874
- <html>, HTML-Tag 868
- <i>, HTML-Tag 875
- , HTML-Tag 884
- <input>, HTML-Tag 896
 - neue Typen in HTML5* 900
- , HTML-Tag 876
- <map>, HTML-Tag 887
- <meta>, HTML-Tag 870, 905
- <nav>, HTML5-Tag 874
- <object>, HTML-Tag 904
- , HTML-Tag 877
- <option>, HTML-Tag 897
- <p>, HTML-Tag 872
- <pre>, HTML-Tag 873
- <script>, HTML-Tag 994
- <section>, HTML5-Tag 874
- <select>, HTML-Tag 897
- <strike>, HTML-Tag 875
- , HTML-Tag 875
- <style>, HTML-Tag 910
- <sub>, HTML-Tag 875
- <sup>, HTML-Tag 875
- <table>, HTML-Tag 888
- <tbody>, HTML-Tag 891
- <td>, HTML-Tag 888
- <textarea>, HTML-Tag 898
- <tfoot>, HTML-Tag 891
- <th>, HTML-Tag 888
- <thead>, HTML-Tag 891
- <title>, HTML-Tag 868
- <tr>, HTML-Tag 888
- <tt>, HTML-Tag 875
- <u>, HTML-Tag 875
- , HTML-Tag 876
- <video>, HTML5-Tag 903
- =, Operator 466–467
- ==, Operator 466
- ~=, Operator 560
- ~=, Perl-Operator 510
- ?, Operator 468
- ?, Operator
 - RegExp 557
- @, Perl-Variable 501
- @_, Perl 508
- @font-face (CSS3) 920
- ^, Operator
 - C 466
 - in RegExp* 557, 559
- __call(), magische PHP-Methode 947
- __get(), magische PHP-Methode 946
- __isset(), magische PHP-Methode 947
- __set(), magische PHP-Methode 946
- __toString(), magische PHP-Methode 937
- `, Operatoren 565

|, Operator 466
 in RegExp 559
 ||, Operator 465
 als Perl-Fallunterscheidung 506
 Perl 506
 ~, Operator 466
 0900-Dialer 1065
 1:1-Beziehung, RDBMS 680
 1:n-Beziehung, RDBMS 680
 1000BaseFL, Ethernet-Standard 202
 1000BaseTX, Ethernet-Standard 202
 100BaseT, Ethernet-Standard 202
 10Base2, Ethernet-Standard 199
 10Base5, Ethernet-Standard 200
 10BaseT, Ethernet-Standard 202
 16-Bit-Anwendung
 unter Win32 321
 3D Now! (CPU-Befehlserweiterung) 121
 8.3 (MS-DOS-Dateinamensschema) 311

A

Abakus 35
 Abfrage, RDBMS
 Änderungsabfrage 695
 Auswahlabfrage 682, 694
 Einfügeabfrage 694
 Löschabfrage 695
 Abfrage, RDBMS, SQL 683
 Abgeleitete Klasse 493
 Absatz
 HTML 872
 Absoluter Pfad 307, 310
 Abstract Factory, Entwurfsmuster 663
 accept(), Perl-Funktion 577
 ACCEPT, iptables-Regel 1077
 Access Point (WLAN) 207, 1087
 Access, Datenbank 686
 ACID (Transaktionen) 705
 Action Mailer (Ruby on Rails) 973
 Action Pack (Ruby on Rails) 973
 Action Web Service (Ruby on Rails) 973
 ActionListener 621

actionPerformed(), AWT-Methode 621
 Active Directory 362, 1087
 Active Record (Ruby on Rails) 973
 Active Server Pages → ASP.NET
 Active Support (Ruby on Rails) 973
 ActivePerl 498
 ActiveX 1087
 ActiveX-Objekt 903
 Ada, Programmiersprache 35
 Adapter, Entwurfsmuster 663
 Addierer (Schaltung) 87
 Addiermaschine 35
 Addierwerk (Schaltung) 88
 Addition, Operator 465
 Administrator, Windows-Benutzer 350
 Administratorendokumentation 649
 Admin-Rechte 1063
 Adobe PostScript 165, 849
 Adressbus 117
 Wortbreite 119
 Adressierung, Speicher 78
 ADSL 215
 anschließen 216
 ADSL2(+) 215
 Advanced Data Guarding (RAID) 149
 AdWare 1087
 Adware 1067
 Aggregatfunktion, SQL 701
 Agile Softwareentwicklung 651
 AGP 138, 1087
 AI → Künstliche Intelligenz
 AIFF, Audio-Dateiformat 857
 AirDrop, Mac OS X 452
 AIX, Betriebssystem 285
 Ajax 1035
 Antwort verarbeiten 1038
 Aspekte 1036
 Bibliotheken für 1053
 DOM-Einsatz für 1039
 JSON 1042, 1051
 komplexes Beispiel 1043
 mit jQuery 1055
 Objekt erzeugen 1036

Ajax (Forts.)
 onreadystatechange, Eigenschaft 1037
 open(), Methode 1037
 PHP-Skript (Serverantwort) 1040
 readyState, Eigenschaft 1037
 responseText, Eigenschaft 1039
 responseXML, Eigenschaft 1043, 1045
 send(), Methode 1038
 Serverantwort 1040
 XML 1042, 1044–1045
 XMLHttpRequest 1036
 Akteur (UML) 656
 Aktivitätsdiagramm (UML) 659
 Akustikkoppler 174
 Al Chwarismi (arab. Mathematiker) 33
 Algebra
 boolesche 60, 92
 Definition 91
 lineare 91
 relationale 679
 zur Algorithmandarstellung 91
 Algorithmus 33, 91, 541, 1087
 algebraische Darstellung 91
 anschaulich-sprachliche Darstellung 92
 Berechenbarkeit 93
 binäre Suche 549
 BubbleSort 544
 Diagrammdarstellung 92
 entwickeln 541
 größter gemeinsamer Teiler (GGT) 542
 Komplexität 94
 lineare Suche 94, 548
 O-Notation der Komplexität 94
 Permutationen 95
 Pseudocode-Darstellung 92
 QuickSort 547
 Sortier- 544
 Such- 548
 Alias (Mac-Verknüpfung) 445
 Alias (Unix-Shell) 407
 Alias, Apache-Direktive 746
 alias, Unix-Befehl 407
 Allen, Paul 286
 Allow, Apache-Direktive 746

- AllowOverride, Apache-Direktive 747
- ALOHA^{Net} 198
- Alpha, Prozessor 119, 122
- ALRM, Signal 299
- Altair 8800, früher Mikrocomputer 40
- ALU 116, 1087
 - FPU 116
- Amazon Web Services 989
- AMD 115
- Amiga 43
- Amigos, drei 654
- Amplitude, Audio 54
- Analog
 - Unterschied zu digital 52
- Analyse, Software-Engineering 643
 - Lastenheft 644
 - objektorientierte Analyse 644
 - Pflichtenheft 645
 - strukturierte Analyse 644
- Analytical Engine 35
- Änderungsabfrage 695
- Android (Smartphone-OS) 44
- AND-Schaltung 85
 - Aufbau mit Transistoren 85
 - mit einfachen Mitteln nachbauen 83
- AND-Verknüpfung 61
- Anführungszeichen
 - PHP 928
 - verschiedene, Bedeutung in Perl 505
- Angewandte Informatik 26
- Animation
 - Doble Buffering 615
 - Java, AWT 615
- ANSI 1087
 - C-Standard 479
- ANSI-C 457
- ANSI-Zeichensatz 842
- Antivirenprogramm 1061
- Antville, Blog-Tool 988
- Anweisung
 - C 461
- Anweisungsblock 469
- Anwenderdokumentation 649
- Anwendung
 - DDN-Modell-Schicht 182
- Anwendung (Forts.)
 - OSI-Schicht 179
- Anwendungsfall 650
- Anwendungsfalldiagramm (UML) 656
- Anwendungsserver 194
 - verteilte Anwendung 195
- Apache
 - Installation, Windows 743
 - Xalan 819
 - Xerces 825
- Apache CouchDB 723–724
- Apache HTTP Server 737
 - Alias, Direktive 746
 - Allow, Direktive 746
 - AllowOverride, Direktive 747
 - apachectl, Hilfsprogramm 743
 - AuthBasicProvider, Direktive 748
 - AuthDigestProvider, Direktive 748
 - Authentifizierung 756
 - AuthName, Direktive 748
 - AuthType, Direktive 748
 - AuthUserFile, Direktive 748
 - Deny, Direktive 749
 - Directory, Direktive 749
 - DirectoryIndex, Direktive 749
 - Direktive 745
 - DocumentRoot, Direktive 750
 - Grundlagen 738
 - htpasswd, Hilfsprogramm 748
 - IfModule, Direktive 750
 - Installation 741
 - Konfiguration 744
 - Konfigurationsbeispiele 755
 - Listen, Direktive 750
 - LoadModule, Direktive 750
 - Location, Direktive 751
 - mod_alias, Modul 746, 752–753
 - mod_auth_basic, Modul 748
 - mod_auth_digest, Modul 748
 - mod_authn_file, Modul 748
 - mod_authz_host, Modul 746, 749, 752
 - mod_dir, Modul 749
 - mod_so, Modul 750
 - Modul dynamisch laden 750
 - Module 739
- Apache HTTP Server (Forts.)
 - NameVirtualHost, Direktive 751
 - Neuerungen in 2.4 747, 749, 752–753
 - Options, Direktive 751
 - Order, Direktive 752
 - Redirect, Apache-Direktive 752
 - Require, Direktive 747, 752
 - RequireAll, Direktive 753
 - RequireAny, Direktive 753
 - RequireNone, Direktive 753
 - Satisfy, Direktive 753
 - ScriptAlias, Direktive 753
 - ServerAdmin, Direktive 753
 - ServerName, Direktive 753
 - ServerRoot, Direktive 754
 - ServerSignature, Direktive 754
 - ServerTokens, Direktive 754
 - SSL-Konfiguration 757
 - Startseite festlegen 749
 - VirtualHost, Direktive 754
 - virtueller Host 751, 754–755
- apachectl, Apache-Hilfsprogramm 743
- API 1087
- Apple
 - iCloud 453
 - iPad 45
 - iPhone 44
 - Mac OS X 437
 - Macintosh 287
 - QuickTime 439
- Apple II 41, 286
- Apple Macintosh 43
- Apple-Menü
 - Mac OS X 440, 442
 - wichtige Befehle 442
- Applet, Java 483, 609, 632
- AppleTalk 272, 1087
 - Protokollstapel 274
 - Vergleich zu TCP/IP 275
- Application Gateway Firewall 1074
- Application Server 194, 1087
 - Java 599
- apt, Linux-Paketmanager 401
- Aqua, Mac OS X-Oberfläche 285, 439
- Arabische Zahlen 35

- Arbeitsspeicher 114
 - des virtuellen Prozessors* 100
- Arbeitsverzeichnis 307
 - anzeigen, Unix* 387
 - wechseln, Unix* 387
 - wechseln, Windows* 336
- Archivdatei 859
 - bzip2* 861
 - GNU zip* 861
 - tar* 401, 860
 - ZIP* 859
- ARCNet
 - Netzwerk* 209
- A-Record (DNS) 771
- ArgoUML, Tool 655
- Arithmetic-Logical Unit → ALU
- Arithmetischer Operator
 - C* 465
 - SQL* 701
- Arithmetisch-logische Einheit
 - ALU
- ARP 1087
 - TCP/IP-Netzzugang* 219
- ARPA 171, 1087
- ARPANET 171
 - Anwendungen* 172
 - MilNet* 173
 - technische Grundidee* 172
 - ursprüngliche Aufgabe* 172
- Array 476, 1088
 - Anzahl der Elemente, Perl* 545
 - aus String, Perl* 503
 - C* 476
 - Deklaration, C* 476
 - mehrdimensionales, PHP* 929
 - Perl* 501
 - PHP* 926
 - zur C-String-Darstellung* 477
 - zusammenfassen, Perl* 503
- Artificial Intelligence → Künstliche Intelligenz
- AS, SQL-Klausel 701
- ASCII 1088
 - Zeichensatz* 55
- ASCII-Art 874
- ASCII-Code 841
 - Erweiterungen* 842
 - IBM-Erweiterung* 842
 - Steuerzeichen* 841
 - Tabelle* 841
- ASCII-Modus 1088
- ASP.NET 986, 1088
- Assembler 46, 1088
 - Mnemonics* 46
 - Nicht-x86* 124
 - praktische Anwendung* 46
 - x86-Beispiele* 124
- Asymmetrische Verschlüsselung 1080
- AT&T 49, 284
 - Unix System V* 285
- AT&T Bell Laboratories 49
- Atari 43, 286
- Atari 800XL 43, 286
- Atari ST 43
- AT-Befehlssatz 1088
- Athlon, Prozessor 115, 121
- ATM (Asynchronous Transfer Mode) 208, 1088
- Atom, Newsfeed-Format 988
- Atomar, Information in RDBMS 683
- Attachment (E-Mail) 264
- attrib, Windows-Befehl 337
- Attribut
 - HTML* 867
 - statisches (PHP)* 942
 - XML* 793, 797
- Attribut (Windows-Datei) 310
- Audio, zeit- und wertdiskret 54
- Audio-CD 151
 - über Soundkarte abspielen* 166
- Audio-Dateiformat 857
 - AIFF* 857
 - MP3* 858
 - MP4* 858
 - OGG Vorbis* 858
 - WAV* 857
- Audiodaten 54
 - Sampling* 54
 - Sampling-Rate* 54
 - Sampling-Tiefe* 54
 - Tonkanal* 55
- Auflicht-Scanner 158
- Auflösung
 - Bild* 54
 - Digitalkamera* 159
 - Grafikkarte* 160
- Aufzählung
 - HTML* 876
- Aufzählungszeichen
 - HTML* 877
- Ausbildung 27
 - Fachinformatiker* 28
 - Informatikkaufmann* 29
 - IT-Systemelektroniker* 29
 - IT-Systemkaufmann* 29
 - Prüfung* 30
 - Studienfächer* 31
- Ausdruck 1088
 - Bedingung* 469
 - C* 464
 - Perl* 504
- Ausgabe
 - in Datei, C* 480
 - Konsole, C* 479
 - Perl* 499
- Ausgabeeinheit 111
- Ausgabegerät 142, 160
 - Drucker* 163
 - Grafikkarte* 160
 - Monitor* 161
- Ausgabesteuerung
 - durch das Betriebssystem* 281
- Ausgabeumleitung
 - in Unix-Shells* 381
 - Windows* 335
- Auslagerungsdatei 303
- Ausnahme
 - auslösen* 497
 - FileNotFoundException, Java* 497
 - IOException, Java* 497
 - Java* 487, 496
 - Java, IOException* 487
- Aussage 58, 1088
 - falsche* 58
 - mathematische* 58
 - wahre* 58
- Aussageformen 59
- Aussagenlogik 57
- Auswahlabfrage 682, 694
- AuthBasicProvider, Apache-Direktive 748
- AuthDigestProvider, Apache-Direktive 748
- Authentifizierung (Apache) 756
- Authentifizierung (MySQL) 707
- AuthName, Apache-Direktive 748

- AuthType, Apache-Direktive 748
 - AuthUserFile, Apache-Direktive 748
 - AUTO_INCREMENT, SQL-Feld-option 699
 - Autoflush-Modus 576
 - Autoloader, PHP 948
 - Automatentheorie 26, 90
 - Registermaschine* 100
 - Turing-Maschine* 97
 - Automatisch starten
 - Programm unter Unix* 399
 - System V Init* 399
 - Automatische Variable 463
 - Autonomes System 1088
 - Autonomes System (AS), Routing 241
 - Average Case (Komplexität) 94
 - AVI, Video-Dateiformat 858
 - awk 497, 555
 - AWT 1088
 - AWT, Java 609
 - Ereignisbehandlung* 609
- B**
-
- Babbage, Charles 35
 - Back Orifice, Backdoor 1064
 - Backdoor 1064
 - Backend 195
 - background-attachment, CSS-Angabe 916
 - background-color, CSS-Angabe 915
 - background-image, CSS-Angabe 916
 - background-repeat, CSS-Angabe 916
 - Backlog, Scrum 653
 - Backtick, Operator 565
 - Backup 1062
 - Banana Ware 668
 - Band, der Turing-Maschine 97
 - Base, OpenOffice.org-Datenbank 686
 - bash
 - .bashrc, Konfigurationsdatei* 373
 - /etc/profile, Unix-Konfigurations-datei* 373
 - bash (Forts.)
 - alias-Befehl* 407
 - bash (Bourne Again Shell) 372
 - Basic Input/Output System
 - BIOS
 - Basic Service Set (WLAN) 206
 - BASIC, Programmiersprache 40, 48
 - Basis
 - bei Stellenwertsystemen* 73
 - Bastion-Host 1074
 - Batch Processing → Stapelverarbeitung
 - Batch-Datei, Windows 337
 - Baum, Datenstruktur 553
 - Binärbaum* 553
 - Baum-Topologie, Netzwerk 188
 - BCD → Binary Coded Decimal
 - BEA WebLogic-Server 599
 - Beck, Kent 651
 - Bedingter Sprung 123
 - Befehl
 - Dateiverwaltung, Unix* 384
 - des virtuellen Prozessors* 102
 - Systemverwaltung, Unix* 392
 - Textmanipulation, Unix* 388
 - Unix* 383
 - Windows-Konsole* 336
 - Befehlstabelle 116
 - der CPU* 116
 - Befehlszeiger (CPU-Register) 116
 - Behavioral Pattern → Verhaltensmuster
 - Bell Laboratories 284
 - belongs_to, Rails-Relation 977
 - Bemer, Robert 841
 - Benutzer
 - Administrator, Windows* 350
 - an- und abmelden, Windows* 329
 - entfernen, Linux* 395
 - Gruppe hinzufügen, Unix* 395
 - hinzufügen, Linux* 394
 - Home-Verzeichnis* 306, 310
 - Passwort ändern, Unix* 395
 - root (Unix)* 300, 369
 - Verwaltung, Mac OS X* 448
 - Verwaltung, Windows* 350
 - Zugriffsrechte* 308
 - Benutzermodus 292, 299
 - Benutzeroberfläche 282
 - grafische* 282, 287, 297
 - Konsole* 282, 297
 - Benutzerrechte, ändern, Unix 387
 - Berechenbarkeit 26, 1088
 - Halteproblem* 93
 - von Algorithmen* 93
 - Berkeley Socket API 571, 1088
 - Berkeley, Universität 284, 410, 571
 - Unix-Version* 284
 - Berners-Lee, Tim 174
 - Beschaffungsmanagement 642
 - Beschreibbare DVD 155
 - Besitzer, wechseln (Datei, Unix) 388
 - Best Case (Komplexität) 94
 - Betriebssystem
 - Aufbau* 290
 - Aufgaben* 281
 - Ausgabesteuerung* 281
 - Benutzermodus* 292
 - Benutzeroberfläche* 282
 - Bibliothek* 296
 - booten* 292
 - BSD-Unix* 284
 - CP/M* 286
 - Darwin* 285, 439
 - Dateiverwaltung* 282
 - Dialogverarbeitung* 283
 - Eingabesteuerung* 281
 - FreeBSD* 285
 - Gerätetreiber* 294
 - Geschichte* 282
 - herunterfahren, Unix* 397
 - HP UX* 285
 - IBM AIX* 285
 - iOS* 438
 - ITS (Incompatible Timesharing System)* 283
 - Kernel* 291
 - Kernelmodus* 292
 - Konsole* 297
 - Linux* 285, 288
 - Mac OS X* 285, 437
 - Minix* 288
 - MS-DOS* 286
 - MULTICS* 284
 - Multitasking* 293, 298

- Betriebssystem (Forts.)
 - Neustart (Unix)* 397
 - OS/2* 287
 - Prozessmanagement* 281
 - Shell* 297
 - Speichermanagement* 281
 - Stapelverarbeitung* 283
 - Sun Solaris* 285
 - Systemaufruf* 125, 293, 295
 - Systemprogramm* 294
 - Task Scheduler* 292
 - Thread* 291, 302
 - Timesharing* 283
 - Unicode-Unterstützung* 845
 - Unix* 49, 284
 - Unix System V* 285
 - Verwaltungsbefehle, Unix* 392
 - Virtualisierung* 311
 - VMS* 287
 - Win32 API* 296
 - Windows* 319
 - Windows 2000* 288, 320
 - Windows 7* 288, 320
 - Windows 95* 288, 320
 - Windows 98* 288, 320
 - Windows Me* 288, 320
 - Windows NT* 287, 320
 - Windows Server* 320
 - Windows Vista* 288, 320
 - Windows XP* 288, 320
 - Windows, Versionsübersicht* 320
 - Zeichensatzeinstellung* 845
- Bewegungsdaten 676
- Bezeichner 1088
 - C* 462
 - in PHP* 925
- BGP 1088
 - Routing-Protokoll* 243
- Bibliothek des Betriebssystems 296
- Big-Endian-Architektur 1088
- Big-Endian-Plattform 851
- BIGINT, SQL-Datentyp 697
- Bild in HTML einbetten 884
- Bilddateiformat 853
 - BMP* 856
 - GIF* 855
 - JPEG* 855
 - Photoshop* 854
 - PICT* 857
- Bilddateiformat (Forts.)
 - PNG* 856
 - PSD* 854
 - TIFF* 854
- Bilddaten
 - Auflösung* 54
 - digitale* 53
 - Farbkanal* 54
 - Farbtiefe* 54
- Bilddatenbank 677
- Bildwiederholrate (Monitor) 162
- bin, Unix-Verzeichnis 306
- Binärbaum, Datenstruktur 553
 - C* 553
- Binärdaten 52, 72
- Binäre Suche 549, 1088
 - Java* 549
- Binärmodus 1088
- Binary Coded Decimal 82, 1088
- BINARY, SQL-Feldoption 699
- bind(), Perl-Funktion 576
- BIND, DNS-Serversoftware 254
- BIND-Nameserver 767
 - A-Record* 771
 - CNAME-Record* 772
 - Installation* 768
 - Konfiguration* 768
 - MX-Record* 773
 - NS-Record* 772
 - PTR-Record* 772
 - Reverse-Lookup-Zone* 769
 - SOA-Record* 771
 - Zonendaten-Datei* 770
 - Zonendefinition* 769
- Bio-Computer 45
- Bioinformatik 32
- BIOS 114, 127, 1088
 - Aufgaben* 128
 - Betriebssystem starten* 129
 - Bootreihenfolge* 130
 - CMOS löschen* 130
 - CMOS-RAM* 129
 - eingebaute Routinen* 129
 - POST* 129
 - POST beschleunigen* 130
 - Power Management einstellen* 131
 - Setup* 129
 - Update* 132
- BIOS-Setup 129
 - Einstellmöglichkeiten* 130
 - Einstellungen zurücksetzen* 131
- BIOS-Update 132
 - Anleitung* 132
 - Motivation* 132
- Bit 78
- Bit-Komplement, Operator 466
- Bitmap-Grafik 53
- Bit-Operator
 - Vergleich mit logischen Operatoren* 62
- Bit-Operatoren 465
- Bitübertragung, OSI-Schicht 177
- Bit-Verschiebung
 - links* 466
 - rechts* 466
- Bitweises exklusives Oder, Operator 466
- Bitweises Oder, Operator 466
- Bitweises Und, Operator 465
- BLOB, SQL-Datentyp 698
- Block, Anweisungen 469
- Blockgerät (Block Device) 294
- Blocksatz
 - in HTML* 873
- Blog → Weblogs
- Blue Book (Mixed-Mode-CD) 152, 1089
- Bluetooth 142
- Blu-ray Disc 156
- BMP, Bilddateiformat 856
- body, HTML 868
- Booch, Grady 654
- Boole, George 60
- boolean, Java-Datentyp 488
- Boolesche Algebra 60, 92, 1089
 - Java-Datentyp* 488
- Booten 292
 - BSD-Startskript* 401
 - Mac OS X-Startvolume* 447
 - System V Init* 399
- Bootreihenfolge, einstellen 130
- Bootsektor 146
- Bootsektor-Virus 1059
- border, CSS-Angabe 915
- border-image (CSS3) 920
- BorderLayout 620
- border-radius (CSS3) 919
- Bourne Again Shell (bash) 372

Bourne-Shell 372
 box-shadow (CSS3) 921
 Boyce-Codd-Normalform, RDBMS 684
 Branch Prediction → Sprungvorhersage (Prozessor)
 break
 C-Anweisung 471
 break, Schleife abbrechen 548
 Bridge 201, 1089
 Bridge, Entwurfsmuster 663
 Broadcast, IP-Protokoll 222
 Brute-Force-Attacke 370, 1089
 BSD 1089
 BSD-Startskript 401
 BSD-Unix 284
 FreeBSD 285
 BubbleSort, Algorithmus 544
 Java 545
 Buffer im Emacs-Editor 418
 BufferedReader
 Java-Klasse 487, 496
 BufferedReader, *Java-Klasse*, *readLine()*, Methode 487
 Builder, Entwurfsmuster 663
 Bundestrojaner → Online-Durchsuchung
 Buntes Buch (CD-Standards) 151
 bunzip2, Unix-Befehl 862
 BURN-Proof-Technologie 153
 Bus 133
 Bluetooth 142
 Definition 115
 der CPU 117
 drahtloser 141
 EIDE 138
 FireWire 141
 Funk-Schnittstelle 142
 Hot-Plugging 141
 Infrarot 142
 IrDA 142
 Kartensteckplatz 137
 Laufwerksanschluss 138
 Light Peak 141
 paralleler 141
 PS/2 141
 RS-232 141
 SCSI 139
 serieller 141
 Thunderbolt 141

Bus (Forts.)
 USB 140
 Bus Mastering 137, 1089
 Bus-Topologie, Netzwerk 188
 Button, AWT-Klasse 619
 Byron, Ada → Lovelace, Ada
 Byte 79
 byte, *Java*-Datentyp 488
 bzip2, Komprimierung 402
 bzip2, Unix-Befehl 862

C

C#, Programmiersprache 51, 1089
 C, Programmiersprache 49, 51, 284, 456, 1089
 #define, *Präprozessor-Direktive* 482
 #endif, *Präprozessor-Direktive* 482
 #ifdef, *Präprozessor-Direktive* 482
 #include, *Präprozessor-Direktive* 459, 482
 &, *Dereferenzierungs-Operator* 475
 -, *Operator* 478
 ANSI-C 457
 ANSI-Standard 479
 Anweisung 459, 461
 Anweisungsblock 469
 arithmetischer Operator 465
 Array 476
 Ausdrücke 464
 Bezeichner 462
 Binärbaum 553
 Bit-Operatoren 465
 break-Anweisung 471
 char 463
 Compiler 457
 Datentypen 462
 Datentypkonvertierung 464
 Datum und Uhrzeit 481
 difftime()-Funktion 482
 double 463
 else 470
 Escape-Sequenz 460
 exec()-Funktion 564–565
 EXIT_SUCCESS, *Konstante* 460
 C, Programmiersprache (Forts.)
 Exponentialschreibweise 464
 Fallunterscheidungen 469
 fgets()-Funktion 480
 Fließkomma-Datentypen 463
 Fließkomma-Literal 464
 float 463
 Flusskontrolle 466
 fopen()-Funktion 480
 for()-Schleife 473
 fork()-Funktion 563
 fprintf()-Funktion 480
 fscanf()-Funktion 480
 Funktionen 474
 Funktionsaufruf 461, 474
 Funktionsparameter 474
 Funktionsrückgabewert 460
 ganzzahlige Datentypen 463
 Geschichte 457
 getchar() 480
 gets()-Funktion 460, 480
 globale Variable 463
 Header-Datei 479
 Hexadezimalzahl 464
 int 463
 int, *Funktionsdatentyp* 459
 Integer-Literal 464
 Kommandozeilenparameter 475
 Kommentar 461
 kompilieren 458
 Kontrollstrukturen 461, 469
 Liste 550
 Literal 464
 localtime()-Funktion 481
 logische Operatoren 465
 lokale Variable 463
 long 463
 main()-Funktion 459, 474
 malloc()-Funktion 551
 mem.h 551
 NULL 481
 Oktalzahl 464
 Operator 465
 Präprozessor 482
 printf()-Funktion 460, 479
 puts()-Funktion 460, 479
 return-Anweisung 460, 474
 scanf()-Funktion 471, 479
 Schleife 472
 short 463

- C, Programmiersprache (Forts.)
 - Speicher reservieren* 551
 - Standardbibliothek* 296, 479
 - static-Deklaration* 463
 - statische Variable* 463
 - stddef.h* 463
 - stdio.h* 459, 479
 - stdlib.h* 459
 - strcat()*, Funktion 481
 - strcmp()*-Funktion 481
 - strcpy()*-Funktion 481
 - strftime()*-Funktion 481
 - string.h* 481
 - String-Literal* 464
 - struct* 478
 - Strukturen* 478
 - switch/case* 471
 - Syntax* 461
 - sys/types.h* 564
 - time()*-Funktion 481
 - time.h* 481
 - time_t*, Datentyp 481
 - Variable* 462
 - Variable, Gültigkeitsbereich* 463
 - Variablendeklaration* 459, 461–462
 - Vergleichsoperatoren* 466
 - void*, Funktionsdatentyp 474
 - Vorzeichen in Datentypen* 463
 - wchar_t* 463
 - Wertzuweisung* 461
 - while()*-Schleife 472
 - Whitespace, Umgang mit* 462
 - Zeichen-Literal* 464
 - Zeiger* 475
- C64 43, 286
- Cache 117
 - bei Festplatten* 150
 - Level 1* 117
 - Level 2* 117
- Call by Reference 476
 - PHP* 934
- Call by Value 476
- Callback-Methode 569, 1089
- Canvas, AWT-Klasse 610
- Capistrano, Rails-Deployment-Tool 985
- CAPTCHA 1066, 1089
- Carbon 1089
- Carbon, Mac OS X-API 439
- Carry-in, Logikschaltung 87
- Carry-out, Logikschaltung 87
- Cäsar-Code 1079
- Cascading Style Sheets → CSS
- case-Befehl
 - in Shell-Skripten* 405
- CASE-Tools 655
- cat, Unix-Befehl 389
- catch(), Java 487
- CAV → Konstante Winkelgeschwindigkeit
- CCD
 - bei der Digitalkamera* 159
 - beim Scanner* 158
- C-Compiler 457
 - GCC* 457
- CD
 - beschreibbare* 152
 - Brennsoftware* 153
 - BURN-Proof-Technologie* 153
 - Datenformate* 153
 - Disc-at-once* 153
 - Hybrid-CD* 154
 - ISO-9660-Format* 154
 - Joliet-Format* 154
 - Lead-in-Area* 153
 - Lead-out-Area* 153
 - Multisession* 153
 - Track-at-once* 153
- cd, Unix-Befehl 387
- cd, Windows-Befehl 336
- CD-Brenn-Software 153
- CDE, Window-Manager 423
- CDi 152
- CD-R 152
- CD-ROM 151
 - Geschwindigkeit* 152
- CD-RW 152
- CD-Standard 151
 - Blue Book* 152
 - Green Book* 152
 - Orange Book* 152
 - Red Book* 151
 - White Book* 152
 - Yellow Book* 152
- CD-Text 152
- Central Processing Unit
 - Prozessor
- Centronics-Anschluss 141
- CGI 986, 1089
 - Sicherheitsprobleme* 1072
- Chain (iptables) 1076
- Chain of Responsibility, Entwurfsmuster 664
- char, C-Datentyp 463
- CHAR, SQL-Datentyp 698
- Character Devices 294
- charAt(), Java-Methode 489
- Chatbot Eliza 96
- chgrp, Unix-Befehl 388
- Child-Prozess 299, 1089
- Chip 37
- Chipsatz 114
- chmod, Unix-Befehl 387
- chomp, Perl-Funktion 500
- Chomsky, Noam 51
- chown, Unix-Befehl 388
- chroot-Umgebung 1074
- CHS (Festplattenadressierung) 145
- CIDR, IP-Adressierung 224
- CIDR-Adressierung 1089
- CIFS-Protokoll, Microsoft-Netzwerk 277
- Circuit Switching 170
- CISC-Prozessor 121
 - Beispiele* 121
- class, Java-Schlüsselwort 486
- CLASSPATH, Umgebungsvariable 484
- clear, CSS-Angabe 917
- Clear-CMOS-Jumper 130
- Client
 - bei Entwurfsmustern* 663
 - Netzwerk* 190
- Clojure, Programmiersprache 51
- close, Perl-Anweisung 509
- closedir, Perl-Anweisung 510
- Cloud Computing 989, 1089
 - Apple iCloud* 453
 - für Privatanwender* 990
- Cluster (Dateisystem) 304
- CLV → Konstante lineare Geschwindigkeit
- Cmd.exe
 - WinNT-Shell* 334
- Cmdet, PowerShell 338
- CMOS-RAM 129
 - löschen* 130

- CMS → Content-Management
- CMYK-Farbe 54
- cn, LDAP-Attribut 776
- CNAME-Record (DNS) 772
- Cobol, Programmiersprache 47
- Cocoa 1090
- Cocoa, Mac OS X-API 439
- Code-Review 648
- Color, AWT-Klasse 612
- color, CSS-Angabe 915
- column-count (CSS3) 921
- column-width (CSS3) 921
- COM, Microsoft-Komponentenmodell 599
- Command, Entwurfsmuster 664
- COMMAND.COM, MS-DOS-Shell 334
- Commit (Transaktionen) 685
- COMMIT, SQL-Anweisung 705
- Commodore 43, 286
- Commodore Amiga 43
- Common Gateway Interface
 - CGI
- Compact Disc 151
- compareTo(), Java-Methode 489
- Compiler 47, 124
 - Java 486
- Composite, Entwurfsmuster 663
- Computer
 - Definition 33
- Computer Science → Informatik
- Computersystem
 - schematischer Aufbau 109
- Computervirus → Virus
- connect(), Perl-Funktion 575
- Connection, JDBC-Klasse 721
- console, interaktive Rails-Konsole 978
- Constant Angular Velocity → Konstante Winkelgeschwindigkeit
- Constant Linear Velocity → Konstante lineare Geschwindigkeit
- ContentHandler, SAX-Interface 828
- Content-Management 986
 - Drupal 987
 - Joomla! 987
 - papaya CMS 987
 - TYPO3 987
- Controller
 - Ruby on Rails 980
- Convention over Configuration (Rails-Prinzip) 974
- Cookie 1090
 - in PHP 952
- Coprozessor → Koprozessor
- copy, Windows-Befehl 337
- CORBA, Komponentenmodell 599
- CouchDB 723
 - Futon, Administrationsoberfläche 724
- count(), PHP-Funktion 928
- COUNT, SQL-Funktion 702
- cp, Unix-Befehl 385
- CP/M, Betriebssystem 286
- C-Programmiersprache, char, Datentyp 459
- CPU 114
 - alte Bedeutung 115
 - Dualcore 114
- CPU → Prozessor
- CR, Mac-Zeilenumbruch 839
- crack (Passwort-Knackprogramm) 369
- Crackerangriff 1072
- Crackertools 1075
- CREATE DATABASE, SQL-Befehl 695
- CREATE TABLE, SQL-Befehl 695
- CREATE USER, MySQL-Anweisung 708
- create, Rails-Methode 978
- create_table, Rails-Methode 976
- CreateProcess(), Windows-Systemaufruf 300
- Creational Patterns → Erzeugungsmuster
- Creator ID, HFS 448
- CRLF, Windows-Zeilenumbruch 839
- Cronjob 408
- Crosslink-Ethernet-Kabel 201
- Cross-Site-Scripting 1072
- CRT → Röhrenmonitor
- csh (C-Shell) 372
- C-Shell 372
- CSMA/CA 1090
- CSMA/CA, Netzzugangsverfahren 206
- CSMA/CD 1090
- CSMA/CD, Netzzugangsverfahren 198
- CSS 908, 1090
 - <style>, HTML-Tag 910
 - Absatzformatierung 914
 - Abstand vom linken Rand 916
 - Abstand vom oberen Rand 916
 - Anzeigeart 914
 - Aufgabe 908
 - Ausrichtung 914
 - Außenrand 914
 - background-attachment 916
 - background-color 915
 - background-image 916
 - background-repeat 916
 - Bild 915
 - border 915
 - clear 917
 - color 915
 - display 914
 - Einzug 914
 - Element (Tag) formatieren 909
 - externe Datei 910
 - Farbangabe 912
 - Farbe 915
 - feste Werte 911
 - fett 913
 - float 917
 - font-family 913
 - font-size 913
 - font-style 913
 - font-weight 913
 - Format dynamisch ändern, DOM 1031
 - für XML-Dokumente 819
 - Hintergrund befestigen 916
 - Hintergrund kacheln 916
 - Hintergrundbild 916
 - Hintergrundfarbe 915
 - Innenabstand 914
 - Klasse 909
 - kursiv 913
 - Laufweite 913
 - Layer 916
 - Layer, Beispiele 917
 - left 916
 - letter-spacing 913

CSS (Forts.)

line-height 914
Linie 914
margin 914
numerische Werte 911
padding 914
position 916
Positionsart 916
Pseudoformat 909
Rahmen 914–915
Schriftart 913
Schriftgröße 913
Selektor 909
Stapelreihenfolge 917
Struktur 909
style, HTML-Attribut 911
text-align 914
text-decoration 913
Textfarbe 915
Textformatierung 913
text-indent 914
top 916
unabhängiger Stil 909
unterstrichen 913
vertical-align 914
vertikale Ausrichtung 914
Vorteile 908
Wertangaben 911
Zeilenhöhe 914
z-index 917

CSS3 919

border-image 920
border-radius 919
box-shadow 921
browserspezifische Eigenschaften 921
column-count 921
column-width 921
Neue Selektoren 921
opacity 921
Web Fonts 920

C-Standardbibliothek 296, 479, 1089

CUPS 1090

CUPS, Unix-Drucksystem 431
Scheduler 432
starten 432

Cutler, David 287

CygWin 457

D

Daemon 396, 1090
 DARPA 171
 Darstellung, OSI-Schicht 179
 Darwin, Betriebssystem 285, 439
 Data Fork, HFS 448
 Database Management System
 → DMS
 database.yml, Rails-Konfigurationsdatei 975
 Datagramm 1090
 Datagramm-Socket 572
 Date, JavaScript-Klasse 1008
 DATE, SQL-Datentyp 697
 date, Unix-Befehl 393
 Formatierung 394
 Datei 282
 Attribute ändern 337
 Ausgabe in, C 480
 bearbeiten, Perl 509
 Besitzer wechseln, Unix 388
 Eingabe aus, C 480
 Gruppe wechseln, Unix 388
 kopieren unter Windows 331, 337
 kopieren, Mac OS X 444
 kopieren, Unix 385
 löschen unter Unix 386
 löschen unter Windows 337
 öffnen, C 480
 schließen, C 480
 String lesen aus, C 480
 umbenennen unter Windows 337
 umbenennen, Mac OS X 445
 umbenennen, Unix 385
 verarbeiten, Java 496
 verschieben unter Windows 331, 337
 verschieben, Mac OS X 444
 verschieben, Unix 385
 Verwaltung 304
 Zugriffsrechte ändern 387
 Dateiattribut 310
 Windows, ändern 337
 Dateierweiterung 311
 Anzeige einschalten 311
 Dateiformat 837
 AIFF 857

Dateiformat (Forts.)

Audio 857
AVI 858
Bild 853
binäres 850
BMP 856
GIF 855
JPEG 855
MP3 858
MP4 858
MPEG 859
OGG Vorbis 858
PICT 857
PNG 856
PostScript 849
QuickTime 858
Text 837
textbasiertes 847
TIFF 854
Video 858
WAV 857
 Dateihandle
 Perl 509
 Dateiname
 Endung 385
 Endung sichtbar machen 311
 Erweiterung 311, 385
 Groß- und Kleinschreibung 307
 MS-DOS 311
 Unix-Platzhalter 384
 unter Unix 307, 384
 unter Windows 310
 Dateiserver 191
 Dateisystem 304
 Benutzerrechte 308
 CD 153
 erzeugen, Unix 393
 ext3 393
 FAT12 323
 FAT16 323
 FAT32 324
 Fehlerprüfung (Unix) 393
 HFS 448
 HFS+ 448
 inode 307
 Journaling-Funktion 393
 Laufwerksbuchstaben 309
 Link 308
 Linux 305
 Mac OS X 448

- Dateisystem (Forts.)
 - mounten* 308
 - mounten, Unix* 392
 - NTFS* 324
 - reiserfs* 393
 - Unix* 305
 - Unix-Pfadangabe* 306
 - Unix-Verzeichnisbaum* 306
 - virtuelles* 305
 - Windows* 309, 323
 - Windows-Pfadangabe* 309
 - Zuordnungseinheit* 304
- Dateiverwaltung 282, 304
 - unter Unix* 384
- Dateivirus 1060
- Daten
 - Bewegungsdaten* 676
 - Ordnungsdaten* 676
 - Rechendaten* 676
 - Stammdaten* 675
- Datenanalyse 644
- Datenbank 675
 - 1:1-Beziehung* 680
 - 1:n-Beziehung* 680
 - Abfrage, objektorientierte* 689
 - Access* 686
 - atomare Information* 683
 - Auswahlabfrage* 682
 - Bild-* 677
 - Boyce-Codd-Normalform* 684
 - CouchDB* 723
 - Datenarten* 675
 - Datenfeld* 678
 - Datensatz* 678
 - dokumentenbasierte* 723
 - Einzeltabellen-* 676
 - Einzeltabellen-, Definition* 678
 - Einzeltabellen-, Grenzen* 679
 - Entity* 678
 - erzeugen, SQL* 695
 - FileMaker* 686
 - Filterung* 678
 - freier Server* 686
 - Fremdschlüssel* 680
 - Funktionen* 678
 - Grenzen der RDBMS* 687
 - IBM DB2* 686
 - Index* 680
 - Join-Abhängigkeit* 685
 - kommerzieller Server* 686
- Datenbank (Forts.)
 - Konsistenz* 679
 - löschen, SQL* 696
 - m:n-Beziehung* 681
 - Microsoft SQL Server* 686
 - Multimedia-* 677
 - MySQL* 689
 - Normalform* 683
 - Normalisierung* 683
 - NoSQL* 677, 724
 - objektorientierte* 677, 686
 - ODBC* 720
 - ODL* 688
 - OpenOffice.org Base* 686
 - OQL* 689
 - Oracle* 686
 - PostgreSQL* 686
 - Primärschlüssel* 680
 - Programmierung* 720
 - Relation* 680
 - relationale* 676, 679
 - Schlüssel* 680
 - sortieren* 678
 - SQL* 683, 694
 - suchen in* 678
 - Tabelle erzeugen, SQL* 695
 - Tabelle löschen, SQL* 696
 - Transaktion* 685, 705
 - Typen* 676
 - Volltextdatenbank* 677
 - XML* 677
- Datenbus 117
 - Wortbreite* 118
- Datenfeld 678
- Datenfernübertragung → DFÜ
- Datenformat 837
 - Text* 837
- Datenkollision 198
- Datenkomprimierung 852
- Datenpaket 170
 - Frame* 197
- Datensatz 678
- Datensicherung 1062
- Datenstruktur 541, 550
 - Baum* 553
 - Queue* 550
 - Stack* 550
- Datenstrukturen 91
- Datenträger
 - magnetischer* 143
- Datenträger (Forts.)
 - magneto-optischer* 143
 - optischer* 143
- Datenträgeraustausch 176
- Datenträgerverwaltung, Win-
dows 352
- Datentyp
 - boolean, Java* 488
 - ganzzahliger, C* 463
 - in der PowerShell* 343
 - in Java* 488
 - testen, PHP* 936
 - Umwandlung, C* 464
 - Variable, C* 462
- Datenübertragung
 - Geschwindigkeitsmessung* 135
 - Paketvermittlung* 170
 - parallele* 133
 - Schaltkreisvermittlung* 170
 - serielle* 133
- Datenverarbeitung
 - elektrische* 25
 - elektronische* 25
 - manuelle* 25
 - mechanische* 25
- DATETIME, SQL-Datentyp 697
- Datum und Uhrzeit
 - C* 481
 - date-Befehl, Unix* 393
 - Differenz berechnen* 482
 - EPOCH* 296
 - ermitteln, C* 481
 - formatieren (Unix)* 394
 - formatieren, C* 481
 - JavaScript* 1008
 - SQL* 697
- DB2, RDBMS 686
- DBMS (Database Management
System) 675
- dc-Knoten (LDAP) 774
- DDN-Schichtenmodell 179
 - Anwendungsschicht* 182
 - Host-zu-Host-Transportschicht*
181
 - Internetschicht* 181
 - Netzzugangsschicht* 180
- DDR-RAM 126
- de Icaza, Miguel 424
- De Marco, Tom 644
- Deadlock 301, 1090

- Debian GNU/Linux 366
- DEC → Digital Equipment Corporation (DEC)
- DECIMAL, SQL-Datentyp 697
- DE-CIX 242
- Decorator, Entwurfsmuster 663
- Default Gateway 1090
- DEFAULT, SQL-Feldoption 699
- default, switch/case-Vorgabewert 472
- defined, Perl 507
- Defragmentierung 1090
- Defragmentierung (Festplatten) 146
- Deklaration 1090
- Deklarative Programmiersprache 51
- del, Windows-Befehl 337
- DELETE, SQL-Abfrage 704
- Demilitarisierte Zone → DMZ
- De-Morgan-Theorem 64, 1090
- Denial of Service 1071, 1090
 - Netzwerkangriff* 359
- Deny, Apache-Direktive 749
- Dependency Injection, Entwurfsmuster 966
- Deployment 1090
 - Ruby on Rails* 985
- Dereferenzierung
 - Perl* 504
- Design Pattern
 - Dependency Injection* 966
 - Lazy Initialization* 966
- Design Pattern → Entwurfsmuster
- Desktop-PC 111
- dev, Unix-Verzeichnis 306
- Dezimalsystem 73
 - in dual umrechnen* 75
 - in hexadezimal umrechnen* 76
- DFÜ 210
 - Akustikkoppler* 174
 - Mailbox* 174
 - Onlinedienst* 174
 - PPP-Protokoll* 210
 - Praxis* 210
 - über ISDN* 214
- DHCP 244, 1090
- DHCP-Server, Windows Server 2003 362
- DHTML 1025, 1090
- DHTML-Layer manipulieren, DOM 1032
- Diagrammtypen (UML) 654
- Dialer 1065
- Dialogverarbeitung 283
- Diascanner 159
- Dienst, Windows 352
- diff, Unix-Befehl 391
- Differential Engine 35
- Differenzmenge 70
- difftime(), C-Funktion 482
- Digital Equipment Corporation (DEC) 39
- Digital Versatile Disc → DVD
- Digital, Unterschied zu analog 52
- Digitale Signatur 1081
- Digitalisierung 53
- Digitalkamera 159
 - Auflösung* 159
- Dijkstra, Edsger W. 243, 637
- DIMM-Modul (RAM) 126
- DIN 1090
- dir, Windows-Befehl 336
- DirectoryIndex, Apache-Direktive 749
- Disc-at-once 153, 1090
- Disjunktion, logische 62
- Diskettenlaufwerk 150
- Diskrete Menge 53
- display, CSS-Angabe 914
- Division, Operator 465
- DMA-Kanal 136, 1091
- DMA-Kanal, Direct Memory Access 136
- DMZ 1074
- dn, LDAP-Attribut 776
- DNA-Computer 45
- DNS 1091
 - BIND, Serversoftware* 254
 - Master- und Slave-Nameserver* 255
 - Subdomain* 254
 - Top-Level-Domain* 254
 - Zone* 254
- DNS (Domain Name System) 252
- DNS-Server, Windows Server 2003 362
- do/while()-Schleife 473
- DocBook 792
- Dock
 - Dock-Menü, Mac OS X* 443
 - Mac OS X* 440, 443
- DOCTYPE-Angabe, XML 804
- Document Object Model → DOM
- Document Type Definition → DTD
- document.forms, JavaScript 1001
- document.images, JavaScript 1011
- document.write(), JavaScript-Methode 995
- DocumentRoot, Apache-Direktive 750
- dojo, Ajax-Bibliothek 1053
- Dokumentation 648
 - Administratoren-dokumentation* 649
 - Anwenderdokumentation* 649
 - Entwicklerdokumentation* 649
- Dokumentstruktur
 - HTML* 868
- DOM 834, 1091
 - Ajax-Einsatz von* 1039
 - Baumstruktur anzeigen, JavaScript-Anwendung* 1028
 - CSS-Format ändern* 1031
 - Document-Objekt* 835
 - Dokumenthierarchie ändern* 1033
 - DOMParser* 834
 - für XML* 834
 - getChildNodes(), Methode* 835
 - in JavaScript* 1025
 - jQuery-Selektor* 1053
 - Kindknoten* 835
 - Klasse importieren* 834
 - Knoteneigenschaften* 1026
 - Knotentypen* 835
 - Parser* 834
 - praktische Anwendung, JavaScript* 1030
 - Textknoten* 1026
- Domain Name System (DNS)
 - BIND-Nameserver* 767
 - Round-Robin-Verfahren* 772
- Domain Name System → DNS
- Domainname
 - Schema* 253
- DOMParser, Java-Klasse 834

- Don't Repeat Yourself (Rails-Prinzip) 974
 Doppelte Anführungszeichen
 Perl 505
 DOS 286
 DoS → Denial of Service
 Double Buffering, Animationsstechnik 615
 double, C-Datentyp 463
 DOUBLE, SQL-Datentyp 697
 DoubleWord (DWord) 1091
 DoubleWord, 32 Bit 79
 Download
 HTML-Hyperlink 882
 Drag & Drop 294
 Drahtlose Schnittstelle 141
 Drahtloses Netzwerk 204
 Arten 204
 Gründe für den Einsatz 204
 Wireless LAN 205
 Drain, Stromausgang des Transistors 84
 DRAM 125
 drawLine(), AWT-Methode 612
 drawOval(), AWT-Methode 612
 drawPolygon(), AWT-Methode 612
 drawRect(), AWT-Methode 612
 drawString, AWT-Methode 614
 Drectory, Apache-Direktive 749
 Drei Amigos 654
 Dreisatz 70
 Drei-Wege-Handshake, TCP 249
 DROP DATABASE, SQL-Befehl 696
 DROP TABLE, SQL-Befehl 696
 DROP USER, MySQL-Anweisung 712
 DROP, iptables-Regel 1077
 Drucker 163
 GDI-Drucker 165
 im Netzwerk freigeben, Windows 361
 Kugelpkopfdruker 163
 Laserdrucker 164
 LED-Drucker 165
 Matrixdrucker 163
 Nadeldrucker 163
 Schriftarten 165
 Thermosublimationsdrucker 165
 Drucker (Forts.)
 Thermotransferdrucker 165
 Tintenstrahldrucker 164
 Treiber 165
 Typenraddrucker 163
 Druckserver 192
 Drupal, CMS 987
 DSL 214, 1091
 ADSL 215
 ADSL2(+) 215
 anschließen 216
 einrichten, Mac OS X 450
 SDSL 215
 über Fernsehkabel 215
 über Satellit 215
 DSSS, WLAN-Technik 205
 DTD 804, 1091
 Alternativen angeben 807
 Attributdeklaration 811
 Attributnotwendigkeit 813
 Attributtypen 811
 definieren 805
 Elementdeklaration 807
 Entity deklarieren 814
 Entitys aus externen Dateien 814
 externe Entity-Deklaration 814
 Häufigkeitsangabe 808
 Klammern 807
 du, Unix-Befehl 393
 Dualcore-Prozessor 114
 Dualsystem 52, 73
 in dezimal umrechnen 76
 in hexadezimal umrechnen 77
 in oktal umrechnen 77
 Dualzähle
 mit Vorzeichen speichern 80
 Durchlicht-Scanner 158
 DVD 154
 beschreibbare 155
 DVD+R 156
 DVD+RW 156
 DVD-R 155
 DVD-RAM 156
 DVD-ROM 155
 Dateiformat 155
 Geschwindigkeit 155
 DVD-RW 155
 DVI, Dateiformat 849
 dvips, Dienstprogramm 849
 DWord → DoubleWord, 32 Bit
 Dynamic HTML → DHTML
 Dynamic RAM → DRAM
- ## E
-
- each(), PHP-Funktion 928
 echo, Unix-Befehl 388
 Echte Obermenge 68
 Echte Teilmenge 68
 ECMA 1091
 EDGE 216
 EDO-RAM 126
 EDV → Elektronische Datenverarbeitung
 EEPROM 128
 Effizienz der CPU 120
 EIDE 138, 1091
 anschließen 139
 im Vergleich zu SCSI 139
 Ein-/Ausgabe 282
 C 479
 Datei, Java 496
 Datei, Perl 509
 Dialogverarbeitung 283
 Fehler 487
 Java 486
 Lochkarte 282
 Perl 499
 Stapelverarbeitung 283
 Terminal 283
 Timesharing 283
 Einfache Anführungszeichen
 Perl 505
 Einfügeabfrage 694
 Eingabe
 aus Datei, C 480
 Konsole, C 479
 Perl 499
 String, C 480
 Zeichen, C 480
 Eingabeaufforderung 1091
 Unix 371
 Windows 336
 Eingabeaufforderung → Konsole
 Eingabeeinheit 111
 Eingabegerät 142, 156
 Digitalkamera 159
 Maus 157
 Scanner 158

- Eingabegerät (Forts.)
 - Tastatur* 157
- Eingabesteuerung
 - durch das Betriebssystem* 281
- Eingabeumleitung
 - in Unix-Shells* 381
 - Windows* 335
- Eingabevervollständigung
 - Unix* 375
 - Windows-Eingabeaufforderung* 334
- Einsteckkarte 137
 - einbauen* 137
- Einwegverschlüsselung 1081
- Einzeltabellendatenbank 676
 - Definition* 678
 - Filterung* 678
 - Funktionen* 678
 - Grenzen* 679
 - sortieren* 678
 - suchen in* 678
- EJB → Enterprise Java Beans
- Elektrische Datenverarbeitung 25
- Elektrisches Gerät 36
- Elektrizität 36
- Elektromechanik 36
- Elektronenröhre 36–37
- Elektronik 36
- Elektronische Datenverarbeitung 25
- Elektronisches Gerät 36
- Element
 - einer Menge* 67
 - XML* 795
- Eliza, Chatbot 96
- else, C 470
- elsif, Perl 507
- Elternklasse 493
- Emacs, Texteditor 417
 - Befehlseingabe* 417
 - Befehlsschreibweise* 417
 - Buffer* 418
 - erweiterte Funktionen* 419
 - Fenster wechseln* 418
 - Modi* 418
 - Navigation* 419
 - speichern* 419
 - Suchfunktionen* 419
 - Text ersetzen* 419
 - Text löschen* 419
- Emacs, Texteditor (Forts.)
 - Text markieren* 419
- E-Mail
 - Attachment* 264
 - Entwicklung* 173
 - Hoax* 1070
 - HTML-Hyperlink auf* 883
 - IMAP-Protokoll* 267
 - Kettenmail* 1070
 - MIME-Format* 263
 - Multipart-Nachricht* 264
 - POP3-Protokoll* 266
 - RFC-822-Nachricht* 263
 - SMTP-Protokoll* 262
 - Spam* 1068
 - Verschlüsselung* 1081
- E-Mail-Protokolle 261
- E-Mail-Server 192
- Embedded System 114
- Endlicher Automat 97
- Endlosschleife 548
- Endung (Dateiname) 311
- Endung (Dateiname, Unix) 385
- Engelbart, Douglas 287
- ENIAC, Röhrenrechner 37
- Enterprise Java Beans 599, 1091
 - Client-Interface* 600
 - Deployment* 603
 - Deployment-Deskriptor* 603
 - Entity-Beans* 600
 - Home-Interface* 600
 - Implementierung* 600
 - Message Driven Beans* 600
 - Session-Beans* 600
 - Vorteile* 599
- Enterprise-Anwendung 598, 1091
- Enterprise-Anwendung
 - Verteilte Anwendung
- Entity
 - Datenbank* 678
- Entity-Referenz
 - HTML* 870
 - XML* 799, 814
- Entwicklerdokumentation 649
- Entwicklungsprozess, Software 649
 - agiler Entwicklungsprozess* 651
 - Extreme Programming* 651
 - Scrum* 652
- Entwicklungsprozess, Software (Forts.)
 - Unified Process* 650
- Entwurf, Software-Engineering 646
 - Schnittstelle* 647
 - Standalone-System* 646
 - verteiltes System* 647
- Entwurfsmuster 660
 - Absicht* 661
 - Abstract Factory* 663
 - Adapter* 663
 - Alias* 662
 - Beispiel Singleton* 665
 - Bestandteile* 661
 - Beteiligte* 662
 - Bridge* 663
 - Builder* 663
 - Chain of Responsibility* 664
 - Client* 663
 - Codebeispiele* 662
 - Command* 664
 - Composite* 663
 - Decorator* 663
 - Einordnung* 661
 - Einsatzbeispiele* 662
 - Erzeugungsmuster* 661
 - Facade* 663
 - Factory Method* 663
 - Flyweight* 663
 - Implementierung* 662
 - Interpreter* 664
 - Iterator* 664
 - Katalog* 663
 - Konsequenzen* 661–662
 - Lösung* 661
 - Mediator* 664
 - Memento* 664
 - Motivation* 662
 - MVC* 661, 973
 - Name* 661
 - Observer* 664
 - Problem* 661
 - Prototype* 663
 - Proxy* 664
 - Querverweis* 662
 - Singleton* 663
 - Singleton (als Beispiel)* 665
 - State* 664
 - Strategy* 664

- Entwurfsmuster (Forts.)
 - Struktur* 662
 - Strukturmuster* 661
 - Template Method* 664
 - Verhaltensmuster* 661
 - Verwendungszweck* 662
 - Visitor* 664
 - Zusammenspiel* 662
 - ENUM, SQL-Datentyp 698
 - Environment → Umgebung
 - EPOCH 296
 - EPROM 128
 - EPS, Dateiformat 850
 - Unterschiede zu PostScript* 850
 - eq, Perl-Operator 505
 - equals(), Java-Methode 489
 - Ereignis
 - AWT 609
 - Ereignisanzeige, Windows 352
 - Ereignisbehandlung 609
 - Java, AWT 621
 - Erlang, Programmiersprache 51
 - eRuby (embedded Ruby) 980
 - Erweiterte Partition 147
 - Erweiterung (Dateiname) 311
 - Erweiterung (Dateiname, Unix) 385
 - Erzeugungsmuster 661
 - Escape-Sequenz 391, 1091
 - C 460
 - in RegExp* 558
 - Escaping
 - PHP 938
 - etc, Unix-Verzeichnis 306
 - Ethernet 197, 1091
 - 1000BaseFL-Standard* 202
 - 1000BaseTX-Standard* 202
 - 100BaseT-Standard* 202
 - 10Base2-Standard* 199
 - 10Base5-Standard* 200
 - 10BaseT-Standard* 202
 - Bridge* 201
 - Crosslink-Kabel* 201
 - CSMA/CD-Verfahren* 198
 - Entwicklung* 41
 - Hardware* 199
 - Hardwareadresse* 197
 - Hub* 201
 - Koaxialkabel* 199
 - MAC-Adresse* 197
 - Ethernet (Forts.)
 - Switch* 201
 - Thicknet Coaxial* 200
 - Thinnet Coaxial* 199
 - Twisted-Pair-Kabel* 200
 - Vorläufer ALOHAnet* 198
 - Ethernet-Kabeltypen 199
 - Ettrich, Matthias 424
 - Euklidische Geometrie 91
 - EVA-Prinzip 110
 - Event Handler
 - jQuery* 1054
 - Event Handling 609
 - Exabyte 79
 - Exception
 - auslösen* 497
 - IOException* 497
 - Java 496
 - Exception, Java 487
 - IOException* 487
 - Exceptions
 - FileNotFoundException* 497
 - exec(), C-Funktion 564–565
 - executeQuery(), JDBC-Methode 722
 - exit, Unix-Befehl 377
 - EXIT-SUCCESS, C-Konstante 460
 - Exklusiv-Oder 64
 - explode(), PHP-Funktion 929
 - Exploit 1071
 - Explorer
 - Ordneransicht, Windows* 332
 - Explorer, Windows-Dateimanager 330
 - Exponentialschreibweise 82
 - C 464
 - Exponentielle Komplexität 95
 - Exposé, Mac OS X 445
 - ext3, Linux-Dateisystem 393
 - Extended Service Set (WLAN) 207
 - extends, Java-Vererbung 493
 - extends, PHP-Schlüsselwort 943
 - Extensible Markup Language
 - XML
 - Extreme Programming 651
 - Eigenschaften* 651
 - Programmieren in Paaren* 652
 - Test-first-Verfahren* 652, 671
- ## F
-
- Facade, Entwurfsmuster 663
 - Fachinformatiker 28
 - Anwendungsentwicklung* 28
 - Projektarbeit* 638
 - Prüfung* 30
 - Systemintegration* 28
 - facsimileTelephoneNumber, LDAP-Attribut 776
 - Factory Method, Entwurfsmuster 663
 - Fakultät 546
 - Fallunterscheidung
 - Bedingung* 469
 - C 469
 - durch logische Operatoren, Perl* 506
 - in Shell-Skripten* 404
 - switch/case, C* 471
 - unless, Perl* 507
 - Fallunterscheidungs-Operator 468
 - Falsche Aussage 58
 - false
 - Java 488
 - Farbaddition 54
 - Farbe
 - in HTML* 912
 - Farbkanal, Bild 54
 - Farblaserdrucker 165
 - Farbsubtraktion 54
 - Farbtiefe
 - Grafikkarte* 160
 - Farbtiefe, Bild 54
 - Fast Ethernet 202
 - FAT 1091
 - FAT12, Dateisystem 323
 - FAT16, Dateisystem 323
 - FAT32, Dateisystem 324
 - FAT-Dateisystem
 - FAT12* 323
 - FAT16* 323
 - FAT32* 324
 - FDDI 1091
 - FDDI, Netzwerk 209
 - Fedora, Linux-Distribution 365
 - Feed → Newsfeed
 - Fenster
 - programmieren* 619

- Fensterbedienung
 - Mac OS X* 441
 - Windows* 327
- Festkommazahl 82
- Festplatte 145
 - alternative SSD* 150
 - Anschlüsse* 138
 - belegten Platz ermitteln (Unix)* 393
 - Cache* 150
 - CHS-Adressierung* 145
 - Defragmentierung* 146
 - EIDE* 138
 - formatieren, Unix* 393
 - Geschwindigkeit* 149
 - konstante Winkelgeschwindigkeit* 149
 - LBA-Adressierung* 145
 - mittlere Zugriffszeit* 149
 - Partitionierung* 146
 - Partitionstabelle* 146
 - RAID* 148
 - SCSI* 139
- Festplattengröße 79
- Festwertspeicher → ROM
- fg, Unix-Befehl 373
- fgets(), C-Funktion 480
- FHSS, WLAN-Technik 205
- FIFO 1091, 1095
- FIFO → First In, First Out
- File Type ID, HFS 448
- FileMaker, Datenbank 686
- FileNotFoundException, Java 497
- Fileserver 191
- fillOval(), AWT-Methode 612
- fillRect(), AWT-Methode 612
- filter, iptables-Tabelle 1076
- find, Rails-Methode 978
- Finder
 - Mac OS X* 440, 443
 - Ordneransichten, Mac OS X* 443
- finger, Unix-Dienstprogramm 369
- Firewall 1062, 1073
 - iptables* 1074, 1076
 - netfilter (Linux)* 1076
- FireWire 141, 1092
- First In, First Out 550
- Flachbettscanner 158
- Flag
 - des virtuellen Prozessors* 103
- flash, Rails-Zwischenspeicher 984
- Flash-EEPROM 128
- Fließkomma-Literal 464
- Fließkommazahl 82
 - Exponentialschreibweise* 82
 - Exponentialschreibweise, C* 464
- Flip-Flop (Schaltung) 89
- float, C-Datentyp 463
- float, CSS-Angabe 917
- FLOAT, SQL-Datentyp 697
- Floating Point Number → Fließkommazahl
- Floating Point Unit → FPU
- FLOPS (CPU-Geschwindigkeit) 120
- FLUSH PRIVILEGES, MySQL-Anweisung 712
- FLUSH TABLES, MySQL-Anweisung 714
- Flussdiagramm 92
- Flusskontrolle 466
- Flyweight, Entwurfsmuster 663
- FM-Synthese (MIDI) 166
- font-family, CSS-Angabe 913
- font-size, CSS-Angabe 913
- font-style, CSS-Angabe 913
- font-weight, CSS-Angabe 913
- fopen(), C-Funktion 480
- for()-Schleife 473
- for-Befehl
 - in Shell-Skripten* 405
- foreach, Perl-Schleife 502
- fork(), C-Funktion 563
- fork(), Unix-Systemaufruf 299
- form_tag, Rails-Methode 984
- Formale Logik 57
- Formatieren
 - Datenträger, Unix* 393
- Formular
 - in Ruby on Rails* 984
- Fortran, Programmiersprache 47
- FORWARD, iptables-Chain 1076
- Foto-Multiplier (Trommel-scanner) 159
- FP-RAM 126
- fprintf(), C-Funktion 480
- FPU 116
- Frame, AWT-Klasse 610, 619
- Frame, Datenpaket 197
- Framework 1092
- FreeBSD 285
- Frege, Gottlob 57
- Freie Software 289
- Fremdschlüssel
 - in Ruby on Rails* 977
 - RDBMS* 680
- Frequency Hopping 205
- Frequenz, Audio 54
- Front Side Bus 119
- Frontend 194
- FSB → Front Side Bus
- fscanf(), C-Funktion 480
- fsck, Unix-Befehl 393
- FTP 259, 1092
 - ASCII-/Binärmodus* 261
 - Befehle* 260
 - Clients* 259
 - HTML-Hyperlink auf* 882
- FULLTEXT, SQL-Schlüsselwort 699
- function, JavaScript-Schlüsselwort 1002
- function, PHP-Schlüsselwort 932
- Funk-Schnittstelle 142
- Funktion 48, 1092
 - Argument* 474
 - aufrufen* 474
 - C* 474
 - Call by Reference* 476
 - Call by Value* 476
 - iterative* 546
 - Parameter* 474
 - rekursive* 546
 - Rückgabewert* 474
 - Rückgabewert, C* 460
 - SQL* 701
- Funktionale Programmiersprache 51
- Funktionsaufruf
 - C* 461
- Fußgesteuerte Schleife 473
- Futon, CouchDB-Administration 724
- fvwm2, Window-Manager 423

G

- Galaxy Tab 45
- Gamma, Erich 660
- GAN 1092
- GAN, globales Netz 187
- Gantt-Diagramm 643
- Ganze Zahl 68
- Garret, Jesse James 1035
- gate (Gatter) → Logikschaltung
- Gate, Steuerungseingang des
 - Transistors 84
- Gates, Bill 40, 286
- Gatter → Logikschaltung
- GCC, C-Compiler 457
 - starten* 458
- gcc-Befehl 458
 - Unix-Besonderheiten* 458
- ge, Perl-Operator 506
- Geometrie
 - euklidische* 91
- Gerät
 - elektrisches* 36
 - elektronisches* 36
- Gerätedatei 305–306
- Gerätetreiber 47, 281, 294
 - Blockgerät* 294
 - für Drucker* 165
 - installieren, Windows* 349
 - Netzwerk, Windows* 355
 - Zeichengerät* 294
- GET, HTTP-Methode 270
 - zum HTML-Formularversand* 895
- Get-Alias, PowerShell-Cmdlet 338
- getchar(), C-Funktion 480
- Get-ChildItem, PowerShell-Cmdlet 338
- Get-Command, PowerShell-Cmdlet 338
- getDate(), JavaScript-Methode 1010
- getDay(), JavaScript-Methode 1010
- getElementById(), JavaScript-Methode 1026
- getElementsByTagName(), JavaScript-Methode 1026
- getFullYear(), JavaScript-Methode 1009
- gethostname(), Systemaufruf 573
- getHours(), JavaScript-Methode 1010
- getMinutes(), JavaScript-Methode 1010
- getprotobyname(), Systemaufruf 573
- gets(), C-Funktion 460, 480
- getSeconds(), JavaScript-Methode 1010
- getservbyname(), Systemaufruf 574
- getYear(), JavaScript-Methode 1009
- GGT (größter gemeinsamer Teiler) 542
- GhostScript 849
- GID (Group-ID)
 - von Prozessen* 300
- GID (Group-ID, Unix) 369
- gidNumber, LDAP-Attribut 776
- GIF, Bilddateiformat 855
- Gigabit-Ethernet 202
- Gigabyte 79
- givenName, LDAP-Attribut 776
- Gleichheit 65
- Gleichheit, Operator 466
- Gleichung 58
 - lineare* 59
 - Lösung* 59
- Gleitkommazahl → Fließkommazahl
- Global Area Network 187
- global, PHP-Variablenmodifikation 932
- Globale Variable
 - C 463
- GNOME
 - GtK+-Bibliothek* 424
- GNOME Terminal 371
- GNOME, Desktop 297, 424, 426
 - Desk Guide* 426
 - GNOME-Menü* 427
 - Kontrollzentrum* 427
 - Panel* 427
 - Verknüpfung erstellen* 427
- GNU Emacs, Texteditor 417
- GNU General Public License 289
- GNU zip, Komprimierung 401
- GNU-Projekt 289
- Google
 - Android (Smartphone-OS)* 44
- Gosling, James 483
- GPL 289
- GPRS 216
- Grafikkarte 160
 - AGP* 161
 - Auflösung* 160
 - Farbtiefe* 160
 - Geschwindigkeit* 161
 - mit mehreren Monitoren* 161
 - PCI* 161
 - RAMDAC* 161
- Grafische Benutzeroberfläche
 - 282, 287, 297, 1092
 - Aqua, Mac OS X* 439
 - Drag & Drop* 294
 - Ereignis* 608
 - Fensteranwendung* 619
 - GNOME* 297, 424, 426
 - GtK+, Programmierungsumgebung* 426
 - JFC* 609
 - KDE* 297, 424
 - Menü programmieren* 619
 - Nachricht* 608
 - Programmierung* 608
 - Qt, Programmierungsumgebung* 425
 - Quartz, Grafikbibliothek* 439
 - Terminalfenster* 371
 - Unix* 422–423
 - Widget* 608
 - Window-Manager* 297
 - Windows* 325
 - Windows 95* 325
 - Windows 98* 325
 - Windows XP* 325
 - X Window* 297, 422
- GRANT, MySQL-Anweisung 709
- Graphics, AWT-Klasse 609
- Graphics2D, Java2D-Klasse 609
- Green Book (CDi) 152, 1092
- grep, Unix-Befehl 390
 - Muster* 391
- GridLayout 620
- Groß- und Kleinschreibung
 - ignorieren in RegExp* 559

- Groß- und Kleinschreibung
(Forts.)
 Unix-Dateiname 307
- Größer als, Operator 466
- größer als, Operator 65
- Größer oder gleich, Operator 466
- Großrechner 39
- Größter gemeinsamer Teiler 542
- groupadd, Unix-Befehl 395
- Group-ID
 von Prozessen 300
- Group-ID (Unix) 369
- Grüner Balken (Unit-Test) 670
- Gruppe
 hinzufügen, Linux 395
 wechseln (Datei, Unix) 388
- Gruppenrichtlinienobjekt, Windows 353
- GSM-Mobilfunk 216
- gsub, Ruby-Methode 562
- gt, Perl-Operator 506
- GTK+, Grafikbibliothek 424, 426
- GUI → Grafische Benutzeroberfläche
- Gültigkeitsbereich
 Variable, C 463
 Variable, Java 488
- gunzip, Unix-Befehl 861
- gzip, Dateikomprimierung 401
- gzip, Unix-Befehl 861
- ## H
-
- Halbaddierer (Schaltung) 87
- Halbleiter 38
- Halteproblem (Berechenbarkeit) 93
- Handshake 1092
- Handshake, Modem-Kommunikation 212
- Hard Link 308
- Hardware
 Ausgabegerät 142, 160
 BIOS 114
 Bus 115, 133
 Bus Mastering 137
 Chipsatz 114
 Digitalkamera 159
 DMA-Kanal 136
 Drucker 163
- Hardware (Forts.)
 Eingabegerät 142, 156
 Grafikkarte 160
 I/O-Basisadresse 136
 IRQ 135
 konfigurieren, Windows 349, 351
 Massenspeicher 142
 Maus 157
 Monitor 161
 Netzwerk 196
 Onboard-Peripherie 115
 Peripherie 142
 Plug & Play 137
 Prozessor 114
 RAM 114, 125
 Ressourcen 135
 ROM-Speicher 114
 Scanner 158
 schematischer Aufbau 109
 Schnittstelle 115
 Soundkarte 166
 Steuerung durch Betriebssystem 281
 Tastatur 157
 Zentraleinheit 113
- Hardware-Interrupt 123
- Harvard Mark I 37
- Harvard Mark II 37
- has_and_belongs_to_many, Rails-Relation 977
- has_many, Rails-Relation 977
- Hash
 Perl-Variable 501
 PHP 927
 Schlüssel 503
- Hauptplatine → Mainboard
- Hauptprogramm
 main() als 50
- Hauptspeicher 111
- Hayes, Modem-Befehlssatz 211
- Hayes-Befehlssatz 1088
- HD DVD 156
- head
 HTML 868
- head, Unix-Befehl 389
- Header-Datei
 C 479
- Heinemeier Hansson, David 973
- Hello World 458
- Helm, Richard 660
- help, Windows-Befehl 297
- Helper
 Ruby on Rails 982
- Herunterfahren
 Betriebssystem, Unix 397
 Windows 329
- Hewlett-Packard, HP UX, Betriebssystem 285
- Hexadezimalsystem 74
 in dezimal umrechnen 77
 in dual umrechnen 77
- Hexadezimalzahl
 C 464
- Hex-Editor 46, 850
- HFS+, Dateisystem 448
- HFS, Dateisystem 448
 Data Fork 448
 Resource Fork 448
- HIER-Dokument
 Perl 506
 Variablensubstitution 506
- HIER-Dokument, in Unix-Shell 381
- Hintergrund (Prozess) 373
- History
 Unix-Shell 376
 Windows-Eingabeaufforderung 334
- Hoax 1070
- Hollerith, Hermann 38
- home, Unix-Verzeichnis 306
- Homecomputer 42, 286
- homeDirectory, LDAP-Attribut 776
- Home-Verzeichnis 306, 310
- Host (Netzwerk) 181
- Host-zu-Host-Transport,
 DDN-Modell-Schicht 181
- Hot Spot 887
- Hot-Plugging 141
- HP UX, Betriebssystem 285
- HPGL (Druckersprache) 165
- HR/DSSS, WLAN-Technik 205
- HSPA (HSDPA/HSUPA) 216
- HTML
 <a>-Tag 880
 <address>-Tag 876
 <applet>-Tag 904
 <area>-Tag 887

HTML (Forts.)

-Tag 875
 <body>-Tag 868

-Tag 870
 <caption>-Tag 888
 <code>-Tag 876
 <col>-Tag 892
 <colgroup>-Tag 892
 <dl>-Tag 879
 <dt>-Tag 879
 -Tag 875
 <embed>-Tag 902
 <form>-Tag 894
 <head>-Tag 868
 <html>-Tag 868
 <i>-Tag 875
 -Tag 884
 <input>-Tag 896
 -Tag 876
 <map>-Tag 887
 <meta>, Tag 905
 <meta>-Tag 870
 <object>-Tag 904
 -Tag 877
 <option>-Tag 897
 <p>-Tag 872
 <pre>-Tag 873
 <script>-Tag 994
 <select>-Tag 897
 <strike>-Tag 875
 -Tag 875
 <style>-Tag 910
 <sub>-Tag 875
 <sup>-Tag 875
 <table>-Tag 888
 <tbody>-Tag 891
 <td>-Tag 888
 <textarea>-Tag 898
 <tfoot>-Tag 891
 <th>-Tag 888
 <thead>-Tag 891
 <title>-Tag 868
 <tr>-Tag 888
 <tt>-Tag 875
 <u>-Tag 875
 -Tag 876
 Absatz 872
 Absatzausrichtung 872
 Absendeknopf 896
 ActiveX-Objekt 903

HTML (Forts.)

Adressangabe 876
 Anker 883
 Attribut 867
 Aufzählung 876
 Aufzählungszeichen wählen 877
 Auswahllisten 897
 Beschreibung für Suchmaschinen 906
 Bild als Hyperlink 886
 Bild einbetten 884
 Blocksatz 873
 Body 868
 Button 897
 Checkbox 896
 clientseitige Image Map 887
 CSS 908
 Datei-Formularfeld 897
 Definitionsliste 879
 Dokumentkopf 868
 Dokumentkörper 868
 Dokumentstruktur 868
 Dokumenttitel 868
 Download-Hyperlink 882
 E-Mail-Hyperlink 883
 Entity-Referenz 870
 Farben 912
 Festbreitenschrift 875
 fett 875
 Formular 894
 Formular, Auswahllisten 897
 Formular, Button 897
 Formular, Checkbox 896
 Formular, Datei-Upload-Feld 897
 Formular, Hidden-Feld 897
 Formular, Löschknopf 897
 Formular, Passwortfeld 896
 Formular, Radiobutton 896
 Formular, Reset-Button 897
 Formular, Schaltfläche 897
 Formular, Submit-Button 896
 Formular, Textbereich 898
 Formular, Textfeld 896
 Formular, Versandmethode 895
 Formulardaten-Codierung 895
 Formulare, Absendeknopf 896
 Formularelemente 896
 Formular-URL 895
 FTP-Hyperlink 882

HTML (Forts.)

geeignete Titel 868
 geschütztes Leerzeichen 872
 GET, Formularversandmethode 895
 Glossarliste 879
 h1 bis h6, Tags 873
 Head 868
 Hidden-Formularfeld 897
 hochgestellter Text 875
 Hyperlink 880
 Hyperlink ins Web 881
 Image Map, client-seitige 887
 Image Map, serverseitige 886
 input-Tag 896
 Java-Applet einbetten 904
 Kommentar 888
 kursiv 875
 Layer 916
 Layout-Tags 875
 Link 880
 Liste 876
 Löschknopf 897
 Meta-Tag 905
 MIME-Typen 902
 Multimedia einbetten 902
 neue Strukturelemente in HTML5 874
 nicht nummerierte Liste 876
 nummerierte Liste 877
 Nummerierungsart wählen 877
 Passwortfeld 896
 Pfadangaben in Links 880
 Plug-in-Formate einbetten 902
 POST, Formularversand-
 methode 895
 Quellcode darstellen 876
 Radiobutton 896
 Refresh 905
 Reset-Button 897
 robots.txt-Datei 907
 Schaltfläche 897
 Schlüsselwörter für Suchmaschinen 906
 Seite neu laden 905
 seiteninterner Link 883
 Server-Side Image Map 886
 Sonderzeichen 870
 Struktur-Tags 875
 Style Sheets 908

- HTML (Forts.)
style-Attribut 911
Submit-Button 896
Suchmaschinen-Informationen 906
Tabelle 888
Tabelle ausrichten 890
Tabelle, Gitternetzlinien steuern 892
Tabelle, Rahmenlinien steuern 892
Tabellen-Attribute 890
Tabellenbereiche 891
Tabellenbeschriftung 888
Tabellenbreite 890
Tabellenhöhe 890
Tabellenrahmen 890
Tabellen-Spaltenbreite 892
Tabellenzeile 888
Tabellen-Zellabstand 890
Tabellenzelle 888
Tabellenzellen verbinden 891
Tabellenzellen-Attribute 890
Tabellenzellen-Ausrichtung 891
Tag 867
Tag-Verschachtelung 875
Text betonen 875
Text durchstreichen 875
Text stark betonen 875
Textbereich 898
Textfeld 896
Textformatierung 869
Textmarke 883
tiefgestellter Text 875
Überschrift 873
unterstrichen 875
verschachtelte Liste 878
vorformatierter Text 873
Web-Palette 912
XHTML 866
XHTML-Besonderheiten 869
Zeichenformatierung 875
Zeichensatz angeben 870
Zeilenumbruch 870
- HTML5 866
 <article>-Tag 874
 <aside>-Tag 874
 <audio>-Tag 903
 <figcaption>-Tag 875
 <figure>-Tag 874
- HTML5 (Forts.)
 <footer>-Tag 874
 <header>-Tag 874
 <hgroup>-Tag 874
 <nav>-Tag 874
 <section>-Tag 874
 <video>-Tag 903
Audio 903
neue Formular-Eingabetypen 900
Strukturelemente 874
Video 903
- HTML-Formular 894
per JavaScript modifizieren 1001
überprüfen per JavaScript 1004
URL 895
- htmlspecialchars(), PHP-Funktion 938
- htpasswd, Apache-Hilfsprogramm 748
- HTTP 269, 727, 1092
Anfrage 729
Antwort 730
Cookie (PHP) 952
Header 271, 734
Kommunikationsablauf 728
Methoden 728
Session-Tracking (PHP) 951
Sitzung 270
Statuscodes 271, 731
- HTTP-Befehle 270
- HTTP-Server 193
- Hub 201
- HUP, Signal 299
- Hybrid-CD 154, 1092
- Hyperlink
HTML 880
- Hypertext 174, 1092
- Hypertext Transfer Protocol
 → HTTP
- I**
-
- I/O → Ein-/Ausgabe
- I/O-Basisadresse 136, 1092
- I/O-Kanäle 381
- IaaS (Infrastructure as a Service) 989
- IANA 222, 1092
- IBM 42, 286
AIX, Betriebssystem 285
DB2 686
OS/2, Betriebssystem 287
- IBM PC 286
- IBM WebSphere-Server 599
- IBM-PC 42
- IC 37
- iCloud 453
- ICMP 1092
- ICMP-Protokoll 247
- IEEE 1092
- IEEE 1394 1092
- IEEE 1394 → FireWire
- IEEE 802.11 → Wireless LAN
- IEEE, Netzzugangsverfahren 802.x 196
- if()
 C 469
Fallunterscheidung, nachgestellte, Perl 507
- if-Befehl
in Shell-Skripten 404
- ifconfig, Unix-Befehl 428
- IfModule, Apache-Direktive 750
- Image Map
clientseitige, in HTML 887
serverseitige, in HTML 886
- Image, AWT-Klasse 615
Bilddatei laden 616
- Image, JavaScript-Klasse 1013
- Imaginäre Zahl 69
- IMAP 267, 1092
- Imperative Programmiersprache 48
- Implementierung, Software-Engineering 647
- implode(), PHP-Funktion 929
- import, Java-Anweisung 486
- include(), PHP-Funktion 947
- include_once(), PHP-Funktion 947
- Index
im RDBMS 680
- INDEX, SQL-Schlüsselwort 699
- indexOf(), Java-Methode 489
- inet_aton, Netzwerkprogrammierung 573
- inetOrgPerson, LDAP-Objekt-klasse 775

- Informatik 26
 - angewandte* 26
 - Bioinformatik* 32
 - Medieninformatik* 32
 - medizinische* 32
 - praktische* 26
 - Studium* 31
 - technische* 26
 - theoretische* 26
 - Wirtschaftsinformatik* 32
- Informatikkaufmann 29
- Information
 - analoge* 52
 - digitale* 52
- Infrarot-Anschluss 142
- Infrastructure as a Service 989
- init-Prozess 299
- Inner Join, SQL 702
 - durch WHERE ausdrücken* 703
- INNER JOIN, SQL-Klausel 702
- InnoDB, MySQL-Tabellentyp 705
- InnoDB-Tabelle (MySQL) 686
- inode 307
- INPUT, iptables-Chain 1076
- Input/Output → Ein-/Ausgabe
- InputStreamReader, Java-Klasse 487
- INSERT, SQL-Abfrage 703
- Instanz 485
 - erzeugen, Java* 488
- Instruction Table → Befehlstabelle
- int
 - C-Datentyp* 463
 - Funktionsdatentyp, C* 459
- INT, Signal 397
- INT, SQL-Datentyp 697
- Integer 1092
 - Java-Klasse* 544
- Integer-Literal 464
- Integrationsmanagement 641
- Integrierter Schaltkreis 37
- Intel 40, 115
- Intel-Assembler 124
- Interface
 - Java* 494
 - Runnable, Java* 495
 - Serializable, Java* 495
- Internet
 - Anwendungsprotokolle* 256
 - Dateiübertragung* 259
- Internet (Forts.)
 - Denial-of-Service-Angriff* 359
 - Dokumentation* 173
 - Geschichte* 171
 - Offizielle Einführung* 173
 - RFC* 173
 - Standards* 173
 - Transportprotokolle* 246
 - Vorläufer ARPANET* 171
 - Zugang per Modem* 211
 - Zugang über DSL* 214
 - Zugang über ISDN* 212
- Internet Explorer
 - ActiveX* 903
- Internet, DDN-Modell-Schicht 181
- Internet-Schichtenmodell 179
 - Anwendungsschicht* 182
 - Host-zu-Host-Transportschicht* 181
 - Internetschicht* 181
 - Netzzugangsschicht* 180
- Interpreter 40, 47, 497
- Interpreter, Entwurfsmuster 664
- Inter-Prozess-Kommunikation 300
 - System V IPC* 301
 - über Pipes* 300
 - über Signale* 300
- Interrupt Request → IRQ
- Interrupt, Hardware- 123
- Intrusion Detection System 1074
- Intrusion Detection Systems
 - Network-IDS* 1074
 - Snort* 1074
- IO::Socket 576–578
- IOException, Java 487, 497
- iOS, Mobilbetriebssystem 438
- iPad 45
- IP-Adresse
 - für Sockets* 573
 - zuweisen, Linux* 429
 - zuweisen, Mac OS X* 449
 - zuweisen, Windows* 356
- IP-Adressen
 - Broadcast-Adresse* 222
 - CIDR-Adressierung* 224
 - CIDR-Berechnungen* 226
 - IPv6* 233
 - Klassen* 220
- IP-Adressen (Forts.)
 - Konzept* 220
 - link local* 223
 - Loopback* 223
 - Netzwerkadresse* 222
 - private* 223
 - spezielle* 223
 - Subnet Mask* 224
 - Subnetting* 225
 - Supernetting* 226
 - Teilnetzmaske* 224
 - VLSM* 229
- IPC → Inter-Prozess-Kommunikation
- iPhone 44
- IP-Masquerading 245
- IP-Protokoll
 - Adressverteilung* 222
 - Datagramme* 230
 - Default Gateway* 237
 - Header* 230
 - IPv6* 233
 - Maximum Transmission Unit (MTU)* 232
 - Multicasting* 222
 - Network Address Translation (NAT)* 245
 - Paketfragmentierung* 232
 - Router* 237
 - Routing* 236
 - Routing-Protokolle* 241
 - spezielle Adressen* 223
 - TTL* 240
- IP-Routing
 - Beispiele* 238
 - Routing-Tabelle* 239
- iptables 1074, 1076
 - Beispiele* 1078
 - Chain* 1076
 - Kommandozeilenoptionen* 1077
 - Regeln* 1077
 - Tabelle* 1076
- IPv4-Adressverteilung 222
- IPv6 233
 - Adressaufbau* 233
 - Datagramm-Header* 234
 - Migration (Umstieg)* 236
 - Motivation* 233
 - Tunnelung* 236
- IPX/SPX, Netzwerkprotokoll 275

- irb
 - console, Rails-Variante* 978
 - IrDA 142, 1093
 - IRQ 135, 1093
 - reservierter* 136
 - IS A-Beziehung, OOP 657
 - is_array(), PHP-Funktion 936
 - is_float(), PHP-Funktion 936
 - is_int(), PHP-Funktion 936
 - is_null(), PHP-Funktion 936
 - is_numeric(), PHP-Funktion 936
 - is_string(), PHP-Funktion 936
 - ISA 138, 1093
 - ISDN 212, 1093
 - anschießen* 213
 - Kanalbündelung* 213
 - Kanäle* 213
 - ISO 1093
 - ISO 9660 (CD-ROM-Format) 154
 - ISO-8859-Zeichensätze 843
 - isset(), PHP-Funktion 936
 - Itanium, Prozessor 119
 - IT-Ausbildung 27
 - Fachinformatiker* 28
 - Informatikkaufmann* 29
 - IT-Systemelektroniker* 29
 - IT-Systemkaufmann* 29
 - Projektarbeit* 638
 - Prüfung* 30
 - Studienfächer* 31
 - IT-Berufe
 - Ausbildungsgänge* 27
 - Iteration 546
 - Iterator, Entwurfsmuster 664
 - ITS, Betriebssystem 283
 - IT-Sicherheit 1057
 - IT-Systemelektroniker 29
 - IT-Systemkaufmann 29
- J**
-
- Jacobson, Ivar 654
 - Java Application-Server 599
 - Java EE 484, 599
 - Enterprise Java Beans* 599
 - Java Enterprise Edition 599
 - Java Foundation Classes 609
 - Java ME 484
 - Java SDK 483
 - Enterprise Edition* 484
 - Java SDK (Forts.)
 - Micro Edition (Java ME)* 484
 - Standard Edition* 483
 - Java Servlet 604
 - Beispiel* 605
 - Java Virtual Machine (JVM) 483
 - java, Programm 486
 - Java, Programmiersprache 51, 483, 570, 1093
 - ActionListener* 621
 - Animation, AWT* 615
 - Applet* 483, 609, 632
 - Ausnahme* 487
 - Ausnahme auslösen* 497
 - Ausnahmen* 496
 - AWT* 609
 - Biddatei laden* 616
 - binäre Suche* 549
 - boolean-Datentyp* 488
 - BorderLayout* 620
 - BubbleSort* 545
 - BufferedReader* 487, 496
 - Button* 619
 - byte-Datentyp* 488
 - Callback-Methode* 569
 - Canvas* 610
 - catch()* 487
 - charAt()-Methode* 489
 - class-Deklaration* 486
 - CLASSPATH* 484
 - Color* 612
 - compareTo()-Methode* 489
 - Connection, JDBC-Klasse* 721
 - Datei verarbeiten* 496
 - Datenbankverbindung herstellen* 721
 - Datentypen* 488
 - drawLine()-Methode* 612
 - drawOval()-Methode* 612
 - drawPolygon()-Methode* 612
 - drawRect()-Methode* 612
 - drawString()-Methode* 614
 - Eigenschaften* 484
 - Ein- und Ausgabe* 486
 - Einsatzgebiete* 483
 - Enterprise Edition* 599
 - equals()-Methode* 489
 - Ereignisbehandlung* 621
 - Exception* 487
 - Exceptions* 496
 - Java, Programmiersprache (Forts.)
 - executeQuery()-Methode* 722
 - extends* 493
 - false* 488
 - fehlender Zeiger* 489
 - FileNotFoundException* 497
 - fillOval()-Methode* 612
 - fillRect()-Methode* 612
 - Frame* 610, 619
 - Graphics2D, Klasse* 609
 - Graphics-Klasse* 609
 - GridLayout* 620
 - Gültigkeitsbereich für Variablen* 488
 - Image-Klasse* 615
 - import-Anweisung* 486
 - indexOf()-Methode* 489
 - init(), Applets* 633
 - InputStreamReader* 487
 - Installation* 484
 - Instanz* 485
 - Instanz erzeugen* 488
 - Integer-Klasse* 544
 - Interface* 494
 - IOException* 487, 497
 - java.awt.** 609
 - java.awt.event.** 609
 - java.lang.** 490
 - java.sql.** 721
 - java.util.** 497
 - Java2D* 609
 - javax.swing.** 609
 - javax.swing.event.** 609
 - javax.swing.table.** 609
 - JDBC-Datenbankschnittstelle* 720
 - JDBC-ODBC-Bridge* 720
 - JDBC-Treiber laden* 721
 - JFC* 609
 - JFrame* 610
 - Kapselung* 485
 - Klasse* 484
 - Kommentar* 489
 - kompilieren* 486
 - Konstruktor* 491
 - Label* 619
 - lastIndexOf()-Methode* 489
 - LayoutManager* 620
 - length()-Methode* 489
 - lineare Suche* 549

- Java, Programmiersprache (Forts.)
- Liste 552
 - Listener 621
 - main()-Methode 486, 610
 - Math-Klasse 490
 - Menu 619
 - MenuBar 619
 - MenuItem 619
 - Methode 484
 - MouseListener 621
 - MouseMotionListener 622
 - new 488
 - Object-Klasse 490
 - Objekt 484
 - Objektorientierung 484, 490
 - org.xml.sax.* 827
 - Package java.io.* 486
 - paint()-Methode 610
 - Panel 610, 619
 - parseInt()-Methode 544
 - print() 487
 - println() 487
 - private (Kapselung) 491
 - Programm starten 486
 - protected 493
 - public (Kapselung) 486, 491
 - readLine()-Methode 487
 - repaint()-Methode 618
 - ResultSet, JDBC-Klasse 722
 - RMI (Remote Method Invocation) 600
 - run()-Methode 568
 - Runnable-Interface 495, 568
 - SAX 827
 - SDK 483
 - SDK, Enterprise Edition 484
 - SDK, Micro Edition (Java ME) 484
 - SDK, Standard Edition 483
 - Serializable-Interface 495
 - Servlet 604
 - setColor()-Methode 612
 - setLayout() 620
 - setVisible()-Methode 610
 - start()-Methode 568
 - Statement, JDBC-Klasse 722
 - static 486
 - String 487
 - String-Methoden 489
 - Strings vergleichen 489
- Java, Programmiersprache (Forts.)
- String-Verkettung 488
 - substring()-Methode 489
 - super 494
 - Swing 609
 - TextArea 619
 - TextField 619
 - this 491
 - Thread 495, 567
 - Thread, Klasse 567
 - throws-Klausel 497
 - true 488
 - try 487
 - try/catch-Block 487
 - Überladung 492
 - Umwandlung von SQL-Datentypen 723
 - Unterschiede zu C 488
 - Unterstützung durch Mac OS X 439
 - update()-Methode 618
 - Variablendeklaration 488
 - Vererbung 493
 - virtuelle Maschine 483
 - WindowListener 622
- java.applet.*, Package 609
- java.awt.*, Package 609
- java.awt.event.*, Package 609
- java.awt.Graphics, Klasse 609
- java.io.*, Package 486
- java.lang.*, Package 490
- java.sql.*, Package 721
- java.util.*, Package 497
- Java2D-API 609
- Java-Applet 609, 632
- in HTML einbetten 904
 - init()-Methode 633
- JavaScript 994, 1093
- +, Operator 997
 - Ajax 1035
 - angepasstes Browserfenster öffnen 1021
 - Ausgabe ins Dokument 995
 - automatisierter Hyperlink 1021
 - Bilder austauschen 1011
 - Bilder vorausladen 1013
 - Bildschirmgröße 1023
 - Browserweiche 1019
 - charAt(), String-Methode 1004
 - CSS-Format ändern, DOM 1031
- JavaScript (Forts.)
- Date, Klasse 1008
 - document.forms, Formulare 1001
 - document.images 1011
 - document.write(), Methode 995
 - Dokumenthierarchie ändern, DOM 1033
 - DOM, Objektmodell 1025
 - DOM-Baumstruktur anzeigen 1028
 - DOM-Knoteneigenschaften 1026
 - DOM-Textknoten 1026
 - Event Handler 999
 - externe Datei einbinden 995
 - Fenstereigenschaften 1021
 - Fenster-Methoden 1024
 - Formular überprüfen 1004
 - Formular, Fokus setzen 1006
 - Formularzugriff 1001
 - function, Schlüsselwort 1002
 - Funktion 1002
 - Geschichte 994
 - getDate(), Methode 1010
 - getDay(), Methode 1010
 - getElementById(), DOM-Methode 1026
 - getElementsByName(), DOM-Methode 1026
 - getFullYear(), Methode 1009
 - getHours(), Methode 1010
 - getMinutes(), Methode 1010
 - getSeconds(), Methode 1010
 - getYear(), Methode 1009
 - history-Objekt 1020
 - Image, Klasse 1013
 - in der Browser-History blättern 1020
 - in HTML einbetten 994
 - indexOf(), String-Methode 1004
 - jQuery 1053
 - JSON 1042, 1051
 - lastIndexOf(), String-Methode 1004
 - length, String-Eigenschaft 1004
 - location-Objekt 1021
 - navigator-Objekt 1019
 - onreadystatechange, Ajax-Eigenschaft 1037

- JavaScript (Forts.)
open(), *Ajax-Methode* 1037
open(), *window-Methode* 1021
parseFloat(), *Methode* 998
parseInt(), *Methode* 998
prompt()-*Methode* 996
readyState, *Ajax-Eigenschaft* 1037
regulärer Ausdruck 1006
responseText, *Ajax-Eigenschaft* 1039
responseXML, *Ajax-Eigenschaft* 1043, 1045
Rollover-Effekt 1013
Rollover-Effekt, *praxistaugliche Lösung* 1014
screen-Objekt 1023
send(), *Ajax-Methode* 1038
setTimeout(), *Methode* 1011
String, *Klasse* 1004
String-Vergleich 999
String-Verkettung 997
substring, *String-Methode* 1004
Timeout 1010
- JavaServer Pages → JSP
 javax.ejb.*, *Java-Package* 600
 javax.swing.*, *Package* 609
 javax.swing.event.*, *Package* 609
 javax.swing.table.*, *Package* 609
 Jaz-Laufwerk 150
 JBoss, *Java Application-Server* 599
 JDBC 1093
 JDBC, *Java-Datenbankschnittstelle* 720
 JDBC-ODBC-Bridge 720
 JDK 483
 JFC 1093
 JFC (*Java Foundation Classes*) 609
 JFrame, *Swing-Klasse* 610
 Jobs, *Steve* 41, 286–287
 Johnson, *Ralph* 660
 join, *Perl-Anweisung* 503
 Join, *SQL* 702
 Join-Abhängigkeit, *RDBMS* 685
 Joliet 1093
 Joliet (*CD-Format*) 154
 Unterstützung auf dem Mac 154
 Joomla!, *CMS* 987
- Journaling-Dateisystem 393
 Joy, *Bill* 410, 483
 JPEG, *Bilddateiformat* 855
 jQuery 1053
 Ajax-Anfrage 1055
 Event Handler 1054
 Funktion 1054
 Selektor 1053
 JSON, *Ajax-Datenaustausch-format* 1042, 1051
 json_encode, *PHP-Funktion* 1051
 JSP 605
 JUnit 668
 Beispiele 668
 grafische Oberfläche 670
- ## K
-
- Kabelanschluss als Internetzugang 215
 Kapselung 50, 485
 Kartensteckplatz
 AGP 138
 ISA 138
 PCI 137
 PCMCIA 138
 KDE 1093
 Konsole (Terminalfenster) 371
 Qt-Bibliothek 424
 KDE, *Desktop* 297, 424
 Kontrollzentrum 426
 Panel 425
 Verknüpfung erstellen 425
 Kernel 291
 mach-Mikrokern 292, 438
 Mikrokern 291
 monolithischer 291
 selbst kompilieren, Linux 402
 Systemaufruf 293, 295
 Task Scheduler 292
 Kernelmodus 292, 299
 Kernighan, *Brian* 49, 284, 457
 Kettenmail 1070
 keys, *Perl* 503
 KI → *Künstliche Intelligenz*
 Kildall, *Gary* 286
 kill(), *Unix-Systemaufruf* 299
 KILL, *Signal* 299, 396
 kill, *Unix-Befehl* 396
 Kilobyte 79
- Kindklasse 493
 Klammer
 Operatoren-Rangfolge verändern 468
 Klammern
 in RegExp 559
 Klasse 50, 484
 abgeleitete 493
 Instanzen erzeugen, Java 488
 Klassen
 Elternklasse 493
 Klassendiagramm (*UML*) 657
 Klassenmethode
 PHP 942
 Klassentest → *Unit-Test*
 Klassenvariable
 PHP 942
 Kleinbildscanner 159
 Kleincomputer 39
 Kleiner als, *Operator* 466
 kleiner als, *Operator* 65
 Kleiner oder gleich, *Operator* 466
 Knuth, *Donald E.* 848
 Koaxialkabel 1093
 KOffice, *KDE-Office-Paket* 424
 Kommandozeile 1093
 Kommandozeile → *Konsole*
 Kommandozeilenparameter
 C 475
 Kommentar
 C 461
 in HTML 888
 Java 489
 Perl 504
 PHP 931
 Kommentare
 in XML 801
 Kommunikation
 zwischen Prozessen 300
 Kommunikationsmanagement 641
 Kommunikationssteuerung,
 OSI-Schicht 179
 Kompakt-Desktop-Rechner 112
 Komplexe Zahl 69
 Komplexität 1093
 exponentielle 95
 logarithmische 95
 O-Notation 94
 polynomielle 95

- Komplexität (Forts.)
 - quadratische* 95
 - von Algorithmen* 94
 - Komplexitätsklasse 94
 - Komponentenmodell 599
 - Komprimierung 852
 - bzip2* 402
 - gzip* 401
 - verlustbehaftete* 853
 - verlustfreie* 852
 - Konfigurationsdatei
 - .bashrc (Unix)* 373
 - /etc/profile (Unix)* 373
 - Konjunktion, logische 61
 - Konqueror, KDE-Browser 424
 - Konsole 282, 297
 - Ausgabe, C* 479
 - Ausgabeumleitung (Unix)* 381
 - Befehl, Windows* 336
 - Eingabe, C* 479
 - Eingabeaufforderung, Unix* 371
 - Eingabeaufforderung, Windows* 336
 - Eingabeumleitung (Unix)* 381
 - Eingabevervollständigung* 375
 - Pipe* 382
 - praktische Verwendung (Unix)* 367
 - starten unter Windows* 336
 - Windows* 333
 - Konsole (KDE-Terminalfenster) 371
 - Konstante
 - symbolische* 482, 1097
 - Konstante lineare Geschwindigkeit 152
 - Konstante Winkelgeschwindigkeit 149
 - Konstruktor
 - Java* 491
 - überladen, Java* 492
 - Kontextmenü
 - Mac OS X* 445
 - Kontrollstruktur 469
 - Fallunterscheidung* 469
 - foreach, Perl* 502
 - in C* 461
 - in der PowerShell* 343
 - Schleife* 472
 - unless, Perl* 507
 - Kooperatives Multitasking 293
 - Kopfgesteuerte Schleife 472
 - Kopieren
 - Datei, Mac OS X* 444
 - Datei, Unix* 385
 - Datei, unter Windows* 331
 - Datei, Windows-Konsole* 337
 - Koprozessor 116
 - Korn Shell (ksh) 372
 - Kosinuskurve
 - zeichnen, AWT* 613
 - Kostenmanagement 641
 - Kritischer Pfad, Netzplan 643
 - Kryptoanalyse 1080
 - Kryptografie 1079
 - asymmetrische Verschlüsselung* 1080
 - Cäsar-Code* 1079
 - digitale Signatur* 1081
 - Einwegverschlüsselung* 1081
 - Grundbegriffe* 1079
 - Message-Digest* 1081
 - ROT13* 1079
 - SSH* 1082
 - SSL/TLS* 1081
 - symmetrische Verschlüsselung* 1080
 - ksh (Korn Shell) 372
 - Kugelpopfdruker 163
 - Künstliche Intelligenz 45
- ## L
-
- l, LDAP-Attribut 776
 - Label, AWT-Klasse 619
 - Lamarr, Hedy 205
 - LAMP-System 924, 1093
 - LAN 186, 1093
 - Technische Lösungen* 187
 - Wireless* 204
 - Laptop 112
 - Laserdrucker 164
 - Farbe* 165
 - Last In, First Out 550
 - Lastenheft 644
 - lastIndexOf(), Java-Methode 489
 - Lasttest 648
 - LaTeX 1093
 - LaTeX, Satzsprache 847
 - Beispieldokument* 848
 - Laufwerk
 - Anschlüsse* 138
 - magnetisches* 143
 - magneto-optisches* 143
 - optisches* 143
 - Laufwerksbuchstabe 309
 - Laufzeit 47
 - Laufzeitbibliothek, C 479
 - Launchpad, Mac OS X 452
 - Lauschender Socket 576
 - LayoutManager, AWT 620
 - BorderLayout* 620
 - GridLayout* 620
 - Lazy Initialization, Entwurfsmuster 966
 - LBA (Festplattenadressierung) 145
 - LCD 1093
 - LCD-Monitor 162
 - Funktionsprinzip* 162
 - Nachteile* 163
 - TFT* 162
 - Vorteile* 162
 - LDAP 773, 1093
 - Active Directory* 773
 - Benutzerkonten abbilden in* 775
 - cn, Attribut* 776
 - dc-Knoten* 774
 - dn, Attribut* 776
 - facsimileTelephoneNumber, Attribut* 776
 - gidNumber, Attribut* 776
 - givenName, Attribut* 776
 - Grundlagen* 774
 - homeDirectory, Attribut* 776
 - inetOrgPerson, Objektklasse* 775
 - l, Attribut* 776
 - loginShell, Attribut* 776
 - mail, Attribut* 776
 - o, Attribut* 776
 - objectClass* 775
 - objectClass, Attribut* 776
 - OpenLDAP* 773
 - Organisationseinheit* 774
 - ou-Knoten* 774
 - people, Organisationseinheit* 774
 - person, Objektklasse* 775
 - posixAccount, Objektklasse* 775
 - Schema* 774

LDAP (Forts.)

sn, Attribut 776
telephoneNumber, Attribut 776
uid, Attribut 776
uidNumber, Attribut 776
userPassword, Attribut 776
 verschiedene Server für 773
 Wurzel 774

le, Perl-Operator 506

Lead-in-Area (CD) 153

Lead-out-Area (CD) 153

LED-Drucker 165, 1093

Leere Menge 70

left, CSS-Angabe 916

Leibniz, Gottfried Wilhelm 35

Leichtgewichtiger Entwicklungs-
 prozess 651

length(), Java-Methode 489

Lerdorf, Rasmus 923

less, Unix-Befehl 389

letter-spacing, CSS-Angabe 913

Level-1-Cache 117

Level-2-Cache 117

LF, Unix-Zeilenumbruch 839

Lichtfarben, RGB 54

Lichtwellenleiter 45

LIFO 1093

LIFO → Last In, First Out

Light Peak 141

Lightweight Directory Access
 Protocol → LDAP

LIKE, SQL-Klausel 700

Lineare Algebra 91

Lineare Geschwindigkeit,
 konstante 152

Lineare Gleichung 59

Lineare Komplexität 94

Lineare Suche 94, 1094

Java 549

line-height, CSS-Angabe 914

Linux 285, 288

\$0, Systemvariable 372

.bashrc, Konfigurationsdatei
 373

/etc/passwd-Datei 369

/etc/profile, Konfigurations-
 datei 373

/etc/shadow, Datei 370

alias-Befehl 407

als Server einrichten 430

Linux (Forts.)

apt, Paketmanager 401

Arbeitsverzeichnis anzeigen 387

auf NFS-Freigaben zugreifen
 431

auf Windows-Server zugreifen
 434

bash 372

Befehle regelmäßig ausführen
 408

Benutzerrechte 308

Bourne-Shell 372

bunzip2-Befehl 862

bzip2-Befehl 862

bzip2-Komprimierung 402

cat-Befehl 389

cd-Befehl 387

chgrp-Befehl 388

Child-Prozess 299

chmod-Befehl 387

chown-Befehl 388

cp-Befehl 385

Cronjob 408

C-Shell 372

CUPS, Drucksystem 431

Daemon 396

date-Befehl 393

Datei kopieren 385

Datei löschen 386

Datei umbenennen 385

Datei verschieben 385

Dateibefehle 384

Dateibesitzer wechseln 388

Dateiendung 385

Dateigruppe wechseln 388

Dateiname 307, 384

Dateinamen-Platzhalter 384

Dateisysteme 305

Datum und Uhrzeit ändern 393

Datum und Uhrzeit formatieren
 394

Debian-Distribution 366

diff-Befehl 391

Distributionen 289

du-Befehl 393

echo-Befehl 388

Emacs, Texteditor 417

Escape-Sequenz 391

exit-Befehl 377

fg-Befehl 373

Linux (Forts.)

finger, Dienstprogramm 369

fork(), Systemaufruf 299

fsck-Befehl 393

Gerätedatei 305–306

GNOME 297, 424, 426

GNU-Projekt 289

grafische Benutzeroberfläche
 422

grep-Befehl 390

groupadd-Befehl 395

Group-ID 300, 369

gunzip-Befehl 861

gzip-Befehl 861

gzip-Komprimierung 401

Hard Link 308

Hardwareplattformen 288

head-Befehl 389

HIER-Dokument 381

Home-Verzeichnis 306

ifconfig-Befehl 428

init-Prozess 299

inode 307

Installation von Software 401

IP-Adresse zuweisen 429

KDE 297, 424

Kernel kompilieren 402

Kernel-Modul laden 402

kill(), Systemaufruf 299

kill-Befehl 396

Korn Shell 372

less-Befehl 389

Link (Dateisystem) 308

logger-Befehl 399

Login 367

ls-Befehl 386

mail-Befehl 409

make-Befehl 402

man-Befehl 297

Mandriva-Distribution 366

mkdir-Befehl 387

mkfs-Befehl 393

modprobe-Befehl 402

more-Befehl 389

mount-Befehl 392

mv-Befehl 385

MySQL-Installation 690

netfilter, Kernel-Firewall 1076

Netzwerkconfiguration 428

Neustart 397

Linux (Forts.)

NFS 430
openSUSE-Distribution 365
Pager 389
Paketmanager 401
Parent-Prozess 299
passwd-Befehl 395
Passwort ändern 395
patch-Befehl 392
PATH, Umgebungsvariable 374
pause(), Systemaufruf 299
Pfadangabe 306
Pipe 382
Programm automatisch starten 399
Prozessmodell 298
Prozessverwaltung 395
ps-Befehl 300, 395
pstree-Befehl 396
pwd-Befehl 387
Red-Hat-Distribution 365
regulären Ausdruck suchen 390
rm-Befehl 386
rmdir-Befehl 387
root, Benutzer 300, 369
route-Befehl 429
rpm, Paketmanager 401
Runlevel 399
Samba-Server 432
SaX, openSUSE-X-Konfigurationsprogramm 423
Shell 297, 367
Shell-Ausgabeumleitung 381
Shell-Eingabeumleitung 381
Shell-Eingabevervollständigung 375
Shell-History 376
Shell-Skript 403
shutdown-Befehl 397
Software installieren 401
Stand-alone Shell 372
Standardrouter einrichten 429
startx-Befehl 423
su-Befehl 377
Swap-Partition 303
Symbolic Link 308
Syslog 396–397
System herunterfahren 397
System V Init 399
Systemprogramme 288, 383

Linux (Forts.)

tail-Befehl 389
tar-Befehl 402, 860
tar-Datei 401
Textbefehle 388
Textdatei anzeigen 389
Textdateien vergleichen 391
Texteditoren 410
Ubuntu-Distribution 366
Umgebung 372
Umgebungsvariable setzen 375
umount-Befehl 393
unalias-Befehl 408
unzip-Befehl 862
useradd-Befehl 394
userdel-Befehl 395
User-ID 300, 369
Verwaltungsbefehle 392
Verzeichnis anlegen 387
Verzeichnis löschen 387
Verzeichnis wechseln 387
Verzeichnisbaum 306
Verzeichnisbefehle 384
Verzeichnisinhalt anzeigen 386
vi, Editor 410
virtuelles Terminal 370
wc-Befehl 392
Window-Manager 297
Wörter zählen 392
X Window 297, 422
Xconfigurator, X-Konfigurationsprogramm 423
zip-Befehl 862
Zugriffsrechte 308
LISP, Programmiersprache 51, 422
Liste
C 550
HTML 876
Java 552
Perl 501
Listen
qw//, Perl 505
listen(), Perl-Funktion 576
Listen, Apache-Direktive 750
Listener, AWT-Ereignisbehandlung 621
Literal 464, 1094
Fließkommazahlen 464
Integer 464

Literal (Forts.)

Perl 504
String 464
Zeichen 464
Little-Endian-Architektur 1094
Little-Endian-Plattform 851
LOAD DATA INFILE, MySQL-Anweisung 715
LoadModule, Apache-Direktive 750
Local Area Network 186
LocalTalk 273
localtime(), C-Funktion 481
Location, Apache-Direktive 751
Lochkarte 38, 282
Logarithmische Komplexität 95
Log-Datei
MySQL 716
logger, Unix-Befehl 399
Logical Link Control (LLC) 178
Logik
AND 61
Aussage 58
Aussageformen 59
Aussagenlogik 57
Definition 57
Disjunktion 62
formale 57
Gleichung 58
Konjunktion 61
lügende Kreter 93
mathematische Aussage 58
Operator, C 465
OR 61
Prädikatenlogik 51, 57
Schlussfolgerung 60
Term 58
Umkehrschluss 60
Ungleichung 58
Verknüpfung 60
wahre und falsche Aussagen 58
Widerspruch 93
XOR 64
Logikschaltkreis → Logikschaltung
Logikschaltung 83
Addierwerk 88
AND/OR-Aufbau durch Transistoren 85
Carry-in 87
Carry-out 87

- Logikschaltung (Forts.)
 - Flip-Flop* 89
 - Gattersymbole* 85
 - Halbaddierer* 87
 - mit einfachen Mitteln nachbauen* 83
 - Multiplexer* 86
 - NAND-Schaltung* 84
 - NOR-Schaltung* 84
 - NOT-Schaltung* 84
 - RS-Flip-Flop* 89
 - Speicherzelle* 89
 - Übertrag* 87
 - Volladdierer* 87
 - XOR-Schaltung* 86
 - Login 1094
 - Login (Unix-Anmeldung) 367
 - loginShell, LDAP-Attribut 776
 - Logische Partition 148
 - Logische Programmiersprache 51
 - Logische Schlussfolgerung 60
 - Logische Verknüpfung 60
 - Logische Verknüpfungen
 - in Programmiersprachen* 66
 - Logischer Operator
 - C* 465
 - Short-Circuit-Logik* 507
 - Vergleich mit Bit-Operatoren* 62
 - Logisches Laufwerk → Logische Partitionen
 - Logisches Nicht, Operator 465
 - Logisches Oder, Operator 465
 - Logisches Und, Operator 465
 - Logo, Programmiersprache 51
 - Logos 57
 - Lokale Variable
 - in C* 463
 - Lokales Netz
 - Entwicklung* 41
 - Lokalisierung
 - Mac OS X* 448
 - long, C-Datentyp 463
 - LONGBLOB, SQL-Datentyp 698
 - LONGTEXT, SQL-Datentyp 698
 - Loopback, IP-Protokoll 223
 - Löschabfrage 695
 - Löschen
 - Datei, Windows-Konsole* 337
 - Dateien, Unix* 386
 - Verzeichnis, Windows* 337
 - Lösung
 - einer Gleichung oder Ungleichung* 59
 - Lösungsmenge 59
 - Lovelace, Ada 35
 - ls, Unix-Befehl 386
 - LS-120-Laufwerk 150
 - lt, Perl-Operator 506
 - LVALUE 467
 - Liste als* 501
-
- ## M
- m//, Perl-Operator 556, 560
 - m:n-Beziehung, RDBMS 681
 - Mac 43
 - Mac OS X 285, 437
 - 10.7 Lion, Neuerungen* 452
 - AirDrop* 452
 - Alias* 445
 - als Server einrichten* 451
 - Anwendungsmenü* 442
 - APIs* 439
 - Apple-Menü* 440, 442
 - Aqua* 285, 439
 - Aqua, praktische Anwendung* 440
 - Aqua-Fenster* 441
 - Architektur* 438
 - Benutzerverwaltung* 448
 - Carbon* 439
 - Classic-Umgebung* 439
 - Cocoa* 439
 - Darwin* 439
 - Data Fork* 448
 - Datei kopieren* 444
 - Datei umbenennen* 445
 - Datei verschieben* 444
 - Dateisysteme* 448
 - Desktop* 440
 - Dock* 440, 443
 - DSL einrichten* 450
 - Exposé* 445
 - File Sharing* 451
 - Finder* 440, 443
 - Geschichte* 437
 - HFS+-Dateisystem* 448
 - HFS-Dateisystem* 448
 - IP-Adresse zuweisen* 449
 - Java* 439
 - Mac OS X (Forts.)
 - Konfiguration* 447
 - Kontextmenü* 445
 - Landeseinstellungen* 448
 - Launchpad* 452
 - Menüleiste* 440
 - Mission Control* 452
 - Monitor, Systemeinstellungen* 448
 - Netzwerkconfiguration* 449
 - OpenGL, 3D-Grafikbibliothek* 439
 - Ordneransichten* 443
 - Papierkorb* 443
 - Quartz, Grafikbibliothek* 439
 - QuickTime* 439
 - Resource Fork* 448
 - Resume* 452
 - Server nutzen* 452
 - Spotlight* 443
 - Startvolume einstellen* 447
 - Systemeinstellungen* 447
 - Terminal starten* 440
 - Unix-Komponenten* 440
 - Windows File Sharing* 451
 - MAC-Adresse 197, 1094
 - mach-Mikrokern 292
 - Macintosh 43, 287
 - Leonardo, ISDN-Leonardo, Mac-ISDN-DFÜ* 214
 - Mac OS X* 437
 - Serversysteme* 451
 - Magische Methode (PHP) 937, 946
 - Magnetband 144
 - Magnetbandspeicher 40
 - Magnetischer Datenträger 143
 - Magnetbänder* 144
 - Magnetscheibe* 144
 - Magneto-optischer Datenträger 143
 - Magnetscheibe 144
 - mail, LDAP-Attribut 776
 - mail, Unix-Befehl 409
 - Mailserver 192
 - mailto, HTML-Hyperlink 883
 - main(), AWT-Methode 610
 - main(), C-Funktion 50, 459, 474
 - Kommandozeilenparameter* 475
 - main(), Java-Methode 486

- Mainboard 113
 - Chipsatz* 114
 - Kartensteckplatz* 137
 - Onboard-Peripherie* 115
- Mainframe 39
- make, Unix-Befehl 402
- Makefile 402
- Makrovirus 1060
- malloc(), C-Funktion 551
- MAN 1094
- MAN, Stadtnetz 187
- man, Unix-Befehl 297
- Mandriva, Linux-Distribution 366
- mangle, iptables-Tabelle 1077
- Man-in-the-Middle-Angriff 1072
- Manuelle Datenverarbeitung 25
- Marconi, Guglielmo 204
- margin, CSS-Angabe 914
- Mark I, Röhrenrechner 37
- Mark II, Röhrenrechner 37
- Maschinenbefehl 124
- Maschinensprache 45
- Massenspeicher 142
 - CD-ROM 151
 - Diskettenlaufwerk* 150
 - DVD 154
 - Festplatte* 145
 - Jaz-Laufwerk* 150
 - LS-120 150
 - magnetischer* 143
 - magneto-optischer* 143
 - optischer* 143
 - Übersicht* 143
 - Wechseldatenträger* 150
 - Zip-Laufwerk* 150
- Master Boot Record 146
- Master-Nameserver 255
- Matching, Reguläre Ausdrücke 560
- Math, Java-Klasse 490
- Mathematische Aussage 58
- Mathematische Funktion, Undefiniertkeitsstelle 93
- Mathematische Variable 59
- Mathematischer Term 58
- Matrixdrucker 163
- Maus 157
- MAX, SQL-Funktion 702
- Maximum Transmission Unit (MTU) 232
- MBR → Master Boot Record
- md → mkdir, Windows-Befehl
- MDI, Windows-Anwendungen 327
- Mechanische Datenverarbeitung 25
- Media Access Control (MAC) 178
- Median, QuickSort 547
- Mediator, Entwurfsmuster 664
- MediaWiki, Wikipedia-Software 989
- Medieninformatik 32
- MEDIUMBLOB, SQL-Datentyp 698
- MEDIUMINT, SQL-Datentyp 697
- MEDIUMTEXT, SQL-Datentyp 698
- Medizinische Informatik 32
- Megabyte 79
- Megapixel (Digitalkamera) 159
- Mehrfachvererbung 494
- Mehrspaltenlayout (CSS3) 921
- mem.h, C-Bibliothek 551
- Memento, Entwurfsmuster 664
- Memory Management Unit → MMU
- Mena, Federico 424
- Menge
 - diskrete* 53
 - leere* 70
- Mengenoperation 67
 - Differenzmenge* 70
 - echte Obermenge* 68
 - echte Teilmenge* 68
 - Element* 67
 - leere Menge* 70
 - Obermenge* 68
 - Schnittmenge* 70
 - Teilmenge* 67
 - Vereinigungsmenge* 70
 - Verknüpfung* 69
- Menu, AWT-Klasse 619
- MenuBar, AWT-Klasse 619
- MenuItem, AWT-Klasse 619
- Message-Digest 1081
- Message-Passing 609
- METAFONT, TeX-Zeichensätze 848
- Methode 484
 - Callback* 569
 - statische (PHP)* 942
 - überladen* 492
- Metropolitan Area Network 187
- Microsoft 40, 286
 - ASP.NET 986
 - BizTalk Server* 363
 - Exchange Server* 363
 - Management Console (MMC)* 352
 - MS-DOS 286
 - SQL Server 363, 686
 - Systems Management Server* 363
 - Windows 287, 319
 - Windows 2000 288
 - Windows 95 288
 - Windows 98 288
 - Windows Me 288
 - Windows NT 287
 - Windows Phone 45
 - Windows XP 288
 - Windows, Versionsübersicht* 320
- Microsoft Azure 989
- Microsoft Virtual PC 312
- MIDI 166, 1094
 - FM-Synthese* 166
 - Wavetable-Synthese* 167
- Migration
 - Ruby on Rails* 976
- Mikrokern 291
 - mach* 292, 438
- Mikroprozessor 114
- Mikroprozessor → Prozessor
- MilNet 173
- MIME 1094
 - E-Mail-Nachricht* 263
- MIME-Multipart-Nachricht 264
- MIME-Nachrichtenheader 263
- MIME-Type
 - HTML-Plug-in-Formate* 902
 - XML-Dokument* 793
- MIME-Types 264
 - im HTTP-Header* 271
- MIN, SQL-Funktion 702
- Minicomputer 39
- Minix, Lehrbetriebssystem 288
- MIPS (CPU-Geschwindigkeit) 120
- MIPS, Prozessor 122

- Mirroring (RAID) 149
- Mission Control, Mac OS X 452
- MITS 40
- Mixed-Mode-CD 152
- mkdir, Unix-Befehl 387
- mkdir, Windows-Befehl 337
- mkfs, Unix-Befehl 393
- MMC (Microsoft Management Console) 352
- MMU 118, 303, 1094
 - Seitentabelle* 303
- MMX (CPU-Befehlserweiterung) 121
- Mnemonic 101, 1088
- Mnemonics 46
- Mobile Datenübertragung
 - EDGE* 216
 - GPRS* 216
 - HSPA (HSDPA/HSUPA)* 216
 - UMTS* 216
- Mobilfunk
 - Internetzugang über* 216
- Mock-Objekt (Unit-Tests) 969
- mod_alias, Apache-Modul 746, 752
- mod_alias, Apache-Module 753
- mod_auth_basic, Apache-Modul 748
- mod_auth_digest, Apache-Modul 748
- mod_authn_file, Apache-Modul 748
- mod_authz_host, Apache-Modul 746, 749, 752
- mod_dir, Apache-Modul 749
- mod_so, Apache-Modul 750
- Model
 - Ruby on Rails* 976
- Model, View, Controller → MVC-Entwurfsmuster
- Modem 211
 - anschiießen* 211
 - AT-Befehlssatz* 211
 - Handshake* 212
 - Hayes-Befehlssatz* 211
 - Pulswahl* 212
 - Tonwahl* 212
- modprobe, Linux-Befehl 402
- Modularisierung
 - Programme* 474
- Modularisierung, Programme 48
- Modularität, von Unix 284
- Modulo, Operator 465
- Molenaar, Bram 410
- Mongrel, Ruby-Webserver 979
- Mongrel, Webserver 985
- Monitor 161
 - Bildwiederholrate* 162
 - Konfiguration, Mac OS X* 448
 - LCD* 162
 - mehrere verwenden* 161
 - Röhrenmonitor* 161
 - Zeilenfrequenz* 162
- Monolithischer Kernel 291
- Moore, Gordon 118
- Mooresches Gesetz 118
- more, Unix-Befehl 389
- MosTek 43
- MosTek 6502, Prozessor 119
- Motherboard → Mainboard
- Motorola 68000, Prozessorfamilie 119
- mount, Unix-Befehl 392
- Mounten 1094
 - Dateisystem* 308
 - Dateisystem, Unix* 392
 - NFS-Freigabe* 431
- MouseListener 621
- MouseMotionListener 622
- MovableType, Blog-Tool 988
- MOV-Befehl
 - beim virtuellen Prozessor* 102
 - x86-Assembler* 124
- move, Windows-Befehl 337
- MP3, Audio-Dateiformat 858, 1094
- MP4, Audio-Dateiformat 858
- MPEG 1094
 - Video-Dateiformat* 859
- MS Access, Datenbank 686
- MS-DOS 286
 - Kommandozeile* 334
- MS-DOS-Anwendung
 - unter Win32* 321
- MTU 232, 1094
- Multi Document Interface → MDI, Windows-Anwendungen
- Multicasting, IP-Protokoll 222
- MULTICS 284
- Multimedia-Datenbank 677
- Multipart-E-Mail 264
- Multiplexer (Schaltung) 86
- Multiplikation, Operator 465
- Multiplikator
 - bei Rambus-RAM* 126
 - der Taktfrequenz* 119
- Multisession-CD 153
- Multitasking 293, 298
 - kooperatives* 293
 - präemptives* 293
 - Unterstützung durch CPU* 116
- mv, Unix-Befehl 385
- MVC-Entwurfsmuster 661, 973
- MX-Record, BIND-Nameserver 773
- my, Perl-Variablendeklaration 501
- my.cnf, MySQL-Konfigurationsdatei 715
- MyISAM, MySQL-Tabellentyp 705
- MyISAM-Tabelle (MySQL) 686
- MySQL 689
 - Authentifizierung* 707
 - Backup* 712
 - Backups automatisieren* 714
 - Benutzerrechte* 709
 - Benutzerverwaltung* 707
 - CREATE USER, Anweisung* 708
 - Datentypen* 697
 - DROP USER, Anweisung* 712
 - Export* 712
 - Export in Textdateien* 715
 - FLUSH PRIVILEGES, Anweisung* 712
 - FLUSH TABLES, Anweisung* 714
 - GRANT, Anweisung* 709
 - Import* 712
 - Import aus Textdateien* 715
 - InnoDB-Tabelle* 686
 - Installation, Unix* 690
 - Installation, Windows* 691
 - JDBC-Anbindung* 720
 - Konfiguration, Windows* 692
 - Konfigurationsdateien* 715
 - LOAD DATA INFILE, Anweisung* 715
 - Log-Datei lesen* 717
 - Log-Dateien* 716
 - my.cnf* 715

- MySQL (Forts.)
MyISAM-Tabelle 686
MySQL Administrator 691
MySQL Query Browser 691
mysqldadmin, Hilfsprogramm 706
mysqlbinlog, Hilfsprogramm 717
mysqldump, Hilfsprogramm 712
mysqli, PHP-Schnittstelle 959
mysql-Kommandozeilen-Client 693
PHP-Zugriff auf 954
Replikation 718
REVOKE, Anweisung 711
SET PASSWORD, Anweisung 709
Sicherheitshinweise, Unix 691
Sicherheitshinweise, Windows 693
Tabellentypen 705
Testdatenbank 954
Transaktion 686, 705
MySQL Connector/J, JDBC-Schnittstelle 720
MySQL Grant Table 690
mysqldadmin, Hilfsprogramm 706
mysqlbinlog, Hilfsprogramm 717
mysql-Client
nicht-interaktiver Betrieb 714
mysqldump, Hilfsprogramm 712
mysqli
real_escape_string(), PHP-Methode 966
mysqli-Schnittstelle 959
- N**
-
- Nachrichtenübermittlung 609
Nadeldrucker 163
Namensraum
XML 815
Nameserver
BIND 767
Master 255
Slave 255
Namespace, PHP 948
NameVirtualHost, Apache-Direktive 751
- Nassi-Shneiderman-Struktogramm 92
NAT 1094
NAT, IP-Protokoll 245
IP-Masquerading 245
nat, iptables-Tabelle 1076
Natürliche Zahl 68
navigator, JavaScript-Objekt 1019
ne, Perl-Operator 505
Nebenläufigkeit 563
Nessus 1075
.NET 1094
.NET Framework 51
NetBEUI 1094
NetBEUI, Netzwerkprotokoll 277
NetBIOS, Windows-Netzwerk 277
Netbook 112
netfilter, Linux-Kernel-Firewall 1076
Netscape-Palette 912
netstat, TCP/IP-Dienstprogramm 359
Routing-Tabellen anzeigen 239
Network Address Translation (NAT) 245
Network File System → NFS
Netzplan 642
kritischer Pfad 643
Netzwerk 169
Anwendungsgebiete 170
ARCNet 209
ATM 208
Client-Server 189
drahtloses 204
Drucker freigeben, Windows 361
Ethernet 197
FDDI 209
Funktionsebene 176
GAN (Global Area Network) 187
Geschichte 171
Hardware 169, 196
IEEE-802-Standard 196
Klassifizierung 186
Konfiguration, Linux 428
Konfiguration, Mac OS X 449
Konfiguration, Windows 355
LAN (Local Area Network) 186
- Netzwerk (Forts.)
Logical Link Control 178
lokales, Entwicklung 41
MAN (Metropolitan Area Network) 187
Media Access Control 178
Netzwerkprogrammierung 571
OSI-Referenzmodell 177
OSI-Schicht 178
Peer-to-Peer 189
Protokoll 170, 217
Reichweite 186
TCP/IP-Protokoll 217
Token Ring 202
Topologie 188
Verkabelung 169
WAN (Wide Area Network) 187
Zentralisierungsgrad 189
Zugang per Modem 211
Zugang über DSL 214
Zugang über ISDN 212
- Netzwerkclient
Definition 190
Netzwerke
Denial-of-Service-Angriff 359
Schichtenmodell 176
Netzwerkhardware 196
Ethernet 199
ISDN-Adapter 213
Modem 211
Netzwerkprogrammierung 571
Berkeley Socket API 571
IO::Socket 577
sockaddr_in 573
Socket 571
Netzwerkprotokoll 170
AppleTalk 272
IPX/SPX 275
NetBEUI 277
Netzwerkserver
Definition 190
Netzwerktopologie 188
Baum 188
Bus 188
logische 189
physikalische 189
Ring 188
Stern 188
Netzzugang, DDN-Modell-Schicht 180

Netzzugangsverfahren

- CSMA/CA 206
- CSMA/CD 198
- IEEE-802-Standard 196
- Token Passing 202
- Neumann, John von 37, 111
- Neuronales Netz 45
- Neustart
 - Betriebssystem, Unix 397
 - Windows 329
- new, Java-Operator 488
- Newsfeed 988
- Newsgroup 267
 - Hierarchie 268
 - NNTP-Protokoll 267
- NeXT 437
- NFS 430, 1094
 - /etc/exports, Konfigurationsdatei 431
 - auf andere Server zugreifen 431
 - starten 431
 - Verzeichnis freigeben 431
- NNTP 267, 1094
 - Header 268
- Normalform, RDBMS 683
- Normalisierung 1094
- Normalisierung, RDBMS 683
- NOR-Schaltung 84
- NoSQL-Datenbank 677, 724
- NoSQL-Datenbanken 1094
- Notebook 112
- NOT-Schaltung 84
- Novell NetWare 176
- NSFNet 173
- nslookup, TCP/IP-Dienstprogramm 359
- NS-Record (DNS) 772
- NTBA (ISDN-Endgerät) 213
- NTFS, Dateisystem 324
- Nucleus, Blog-Tool und CMS 988
- Null
 - im Stellenwertsystem 35
- NULL, leerer C-Zeiger 481
- NULL, SQL-Feldoption 699
- Null-terminierter String 477
- Nyquist-Theorem 1094, 1096

O

- o, LDAP-Attribut 776
- Obermenge 68
 - echte 68
- Object Database Management Group → ODMG
- Object Definition Language → ODL
- Object Management Group (OMG) 654
- Object Query Language → OQL
- Object, Java-Klasse 490
- objectClass (LDAP) 775
- objectClass, LDAP-Attribut 776
- Objective-C, Programmiersprache 439
- Objekt 50, 484
- Objektorientierte Analyse 644
- Objektorientierte Datenbank 677, 686
 - Abfrage 689
 - ODL 688
 - OQL 689
- Objektorientiertes Datenbankverwaltungssystem → OODBMS
- Objektorientierung 50, 484, 1094
 - Eigenschaften 484
 - Elternklasse 493
 - Ereignis 608
 - im Software-Engineering 637
 - Instanz 485
 - Instanz erzeugen, Java 488
 - Interface, Java 494
 - IS A-Beziehung 657
 - Java 490
 - Kapselung 50, 485
 - Kindklasse 493
 - Klasse 484
 - Klassen 50
 - Konstruktor 491
 - Methode 484
 - Nachricht 608
 - Objekt 484
 - Perl 577
 - PHP 936
 - Überladung 492
 - Vererbung 50, 493
- Observer, Entwurfsmuster 664
- ODBC 1095
- ODBC, Microsoft-Datenbankschnittstelle 720
- Oder-Schaltung → OR-Schaltung
- Oder-Verknüpfung → OR-Verknüpfung
- ODL 688
- ODMG 687
- OFDM, WLAN-Technik 205
- OGG Vorbis, Audio-Dateiformat 858
- Oktalsystem 74
 - in dual umrechnen 77
- Oktalzahl
 - C 464
- OMG → Object Management Group
- Onboard-Hardware 115
- Onlinedienst 174
- Online-Durchsuchung 1065
- O-Notation 1094
- O-Notation (Komplexität) 94
- onreadystatechange, Ajax-Eigenschaft 1037
- OODBMS 687
- OOP → Objektorientierung
- opacity (CSS3) 921
- Open Database Connectivity → ODBC, Microsoft-Datenbankschnittstelle
- open(), Ajax-Methode 1037
- open, Perl-Anweisung 509
- opendir, Perl-Anweisung 510
- OpenGL, 3D-Grafikbibliothek 439
- OpenOffice.org Base, Datenbank 686
- OpenSSH 258
- openSUSE, Linux-Distribution 365
- Operator 466
 - 467
 - ! 465
 - != 466
 - !~ 560
 - \$, RegExp 559
 - & 475
 - &&, C 465
 - &&, Perl 507
 - &, C 465

Operator (Forts.)

*, *RegExp* 557
 ++ 467
 +, *Java-String-Verkettung* 488
 +, *RegExp* 557
 -, *C* 478
 ., *Perl* 506
 = 466–467
 == 466
 =~ 560
 =~, *Perl* 510
 ? 468
 ?, *RegExp* 557
 ^, *C* 466
 ^, *RegExp* 557, 559
 `` 565
 | 466
 |, *RegExp* 559
 || 465
 ~ 466
arithmetischer 465
Backticks 565
Bit-Komplement 466
Bit-Verschiebung, links 466
Bit-Verschiebung, rechts 466
bitweise 465
bitweises exklusives Oder 466
eq, Perl 505
Fallunterscheidungs- 468
ge, Perl 506
Gleichheit 466
größer oder gleich 466
gt, Perl 506
in C 465
in der PowerShell 339
kleiner als 466
kleiner oder gleich 466
le, Perl 506
logischer 465
logisches Oder 465
logisches Und 465
lt, Perl 506
m// 560
Menge 67
ne, Perl 505
Perl 504
PHP 926
Post-Dekrement 468
Post-Inkrement 467
Prä-Dekrement 467

Operator (Forts.)

Prä-Inkrement 467
Rangfolge 468
Rangfolge durch Klammern
 ändern 468
s// 560
s//, Perl 510
Stringvergleiche, Perl 505
String-Verkettung, Java 488
tr// 561
Ungleichheit 466
Vergleichs- 65, 466
Wertzuweisung 467

Operatoren

- 465
 % 465
 * 465
 + 465
 / 465
Addition 465
bitweises Oder 466
bitweises Und 465
Division 465
größer als 466
logisches Nicht 465
Modulo 465
Multiplikation 465
Subtraktion 465

opt, Unix-Verzeichnis 306

Option

bei Systemprogrammen 373

Options, Apache-Direktive 751

Optischer Datenträger 143

OQL 689

Oracle, Datenbank 686

Orange Book 1095

Orange Book (CD-R, CD-RW) 152

ORDER BY, SQL-Klausel 702

Order, Apache-Direktive 752

Ordnungsdaten 676

org.w3c.dom., Java-Package* 834

org.xml.sax., Java-Package* 827

Organisationseinheit (LDAP) 774

OR-Schaltung 85

Aufbau mit Transistoren 85

mit einfachen Mitteln nach-
bauen 84

OR-Verknüpfung 61

OS/2, Betriebssystem 287

OSI-Referenzmodell 177, 1095

Anwendungsschicht 179

Bitübertragungsschicht 177

Darstellungsschicht 179

Kommunikationssteuerungs-
schicht 179

Netzwerkschicht 178

Präsentationsschicht 179

Sicherungsschicht 178

Sitzungsschicht 179

Transportschicht 178

Vergleich mit der Praxis 179

Vergleich zum Internet-Schichten-
modell 180

OSPF 1095

OSPF, Routing-Protokoll 243

ou-Knoten (LDAP) 774

OUTPUT, iptables-Chain 1076

Outsourcing (Computertechnik)
 175

Overclocking → *Übertakten*

P

Paar, Extreme Programming 652

PaaS (Platform as a Service) 989

Packet Switching 170

padding, CSS-Angabe 914

Page Fault (Speicher) 303

Page File (Auslagerungsdatei) 303
Pager (seitenweise anzeigen, Unix)
 389

Paging (Speicher) 118, 303

paint(), AWT-Methode 610

Paketfilter

iptables 1076

netfilter 1076

Paketmanager

Linux 401

Paketvermittlung 170

Panel, AWT-Klasse 610, 619

papaya CMS 987

Papierkorb

Mac OS X 443

Parallele Datenübertragung 133

Parallelport 141

Parallels Desktop 313

Parameter

bei Systemprogrammen 373

von Perl-Subroutinen 508

- PARC, XEROX-Forschungszentrum 287
- parent, PHP-Schlüsselwort 946
- Parent-Prozess 299, 1095
- Parity Bit → Prüfbit
- parseFloat(), JavaScript-Methode 998
- parseInt(), Java-Methode 544
- parseInt(), JavaScript-Methode 998
- Partition
 - erweiterte* 147
 - logische* 148
 - primäre* 146
- Partitionierung 1095
 - praktische Durchführung* 148
- Partitionierung (Festplatte) 146
- Partitionstabelle 146
- Partitionstypen 147
- Pascal, Blaise 35
- Pascal, Programmiersprache 48
- passwd, Unix-Befehl 395
- Passwort 1075
 - Brute-Force-Attacke* 370
 - crack (Knackprogramm)* 369
 - Erzeugungstipps* 370
- Passwort, ändern, Unix 395
- Patch
 - Sicherheit* 1062
- patch, Unix-Befehl 392
- path, Umgebungsvariable
 - Windows* 336
- PATH, Umgebungsvariable (Unix) 374
- pause(), Unix-Systemaufruf 299
- PC 42
 - Aufbau* 113
 - Desktop* 111
 - Geschichte* 286
 - Kompaktrechner* 112
 - Laptop* 112
 - Netbook* 112
 - Notebook* 112
 - Varianten* 111
 - Zentraleinheit* 113
- PC-Card → PCMCIA-Anschluss
- PCDATA, Text in XML 799
- PCI 137, 1095
- PCMCIA-Anschluss 138
- PC-Netzwerk
 - Entwicklung* 176
- PDA (Personal Digital Assistant) 45
- PDF 1095
- PDP, Kleincomputerserie von DEC 284
- PDP, Kleinrechnerserie von DEC 39, 49
- Pentium, Prozessorfamilie 115, 121
- people, LDAP-Organisationseinheit 774
- Peripherie
 - Ausgabegerät* 160
 - Digitalkamera* 159
 - Drucker* 163
 - Eingabegerät* 156
 - Einsteckkarte* 137
 - Grafikkarte* 160
 - Maus* 157
 - Monitor* 161
 - Scanner* 158
 - Tastatur* 157
 - Übersicht* 142
- Perl, Programmiersprache 497, 1095
 - !~-Operator* 560
 - \$* 499, 501
 - \$/-Variable* 840
 - \$_* 502
 - \$_-Operator, RegExp* 559
 - %* 501
 - &&-Operator* 507
 - *-Operator, RegExp* 557
 - +Operator, RegExp* 557
 - .-Operator* 506
 - //Operator* 560
 - =~* 510
 - =~-Operator* 560
 - ?-Operator* 557
 - @* 501
 - @_* 508
 - ^Operator, RegExp* 557, 559
 - || zur Fallunterscheidung* 506
 - |-Operator* 506
 - |-Operator, RegExp* 559
 - accept()-Funktion* 577
 - ActivePerl-Distribution* 498
- Perl, Programmiersprache (Forts.)
 - alternative Zeichen in RegExp* 557
 - Anführungszeichen* 505
 - Array* 501
 - Array-Elementanzahl* 545
 - Arrays zusammenfassen* 503
 - Ausdruck* 504
 - Ausgabe* 499
 - Autoflush-Modus* 576
 - Backticks* 565
 - bind()-Funktion* 576
 - chomp-Funktion* 500
 - close* 509
 - closedir* 510
 - connect()-Funktion* 575
 - Datei bearbeiten* 509
 - Dateihandle* 509
 - defined* 507
 - Dereferenzierung* 504
 - die, Programmabbruch* 507
 - Eingabe* 499
 - elsif* 507
 - eq-Operator* 505
 - ersetzen durch RegExp* 560
 - foreach-Schleife* 502
 - ge-Operator* 506
 - gethostbyname()-Funktion* 573
 - getprotobyname()-Funktion* 573
 - getservbyname()-Funktion* 574
 - gt-Operator* 506
 - Hash* 501
 - HIER-Dokumente* 506
 - if, nachgestelltes* 507
 - IO* 576–577
 - join* 503
 - keys* 503
 - Kommentar* 504
 - le-Operator* 506
 - Liste* 501
 - listen()-Funktion* 576
 - Listenfunktionen* 502
 - Literal* 504
 - lt-Operator* 506
 - m//-Operator* 556
 - my* 501
 - ne-Operator* 505
 - Objektorientierung* 577
 - open* 509
 - opendir* 510

- Perl, Programmiersprache (Forts.)
 - Operator 504
 - Operatoren 499
 - Parameter von Subroutinen 508
 - Pipe verwenden 565
 - pipe() 566
 - pop 502
 - print-Anweisung 499
 - Programm starten 498
 - Programmabbruch 507
 - push 502
 - q// 505
 - qq// 505
 - Queue 550
 - qw// 505
 - readdir 510
 - Referenz 503
 - regulärer Ausdruck 510, 555
 - s///-Operator 510, 556, 560
 - Shebang 498
 - shift 502
 - Skalar 499, 501
 - Socket-Modul 572
 - split 503
 - Stack 550
 - strenge Wertüberprüfung 500
 - strict 500
 - String zerlegen 503
 - String-Vergleich 505
 - String-Verkettung 506
 - sub 508
 - Subroutine 508
 - tr///-Operator 561
 - unless 507
 - unshift 502
 - use-Direktive 500
 - Variable 500
 - Variablenexistenz prüfen 507
 - Variablensubstitution 500
 - Verzeichnis verarbeiten 510
 - w, Modifikator 499
 - Warnungen aktivieren 499
 - Zeichengruppe in RegEx 557
 - Zeilenumbruch 840
 - Zeilenumbrüche entfernen 500
- Perl-Anweisung 507
- Permutation
 - Algorithmus 95
- person, LDAP-Objektklasse 775
- Personal Computer → PC
- Personal Digital Assistant (PDA) 45
- Petabyte 79
- Pfad
 - absoluter 307, 310
 - in HTML-Hyperlink 880
 - relativer 307, 310
 - unter Unix 306
 - unter Windows 309
- PGP 1081, 1095
- Phishing 1067, 1095
- PHP 923, 1095
 - \$, Variablenbeginn 925
 - __call(), magische Methode 947
 - __get(), magische Methode 946
 - __isset(), magische Methode 947
 - __set(), magische Methode 946
 - __toString(), magische Methode 937
 - Ajax-Antwort durch 1040
 - Anführungszeichen 928
 - Array 926
 - Array als Hash 927
 - Array, mehrdimensionales 929
 - Attribut, statisches 942
 - Autoloader 948
 - Bezeichner 925
 - Call by Reference 934
 - Cookie 952
 - count()-Funktion 928
 - Datei-Upload 950
 - Datenbank-Escaping 966
 - Datenbankzugriff 954
 - Datentyp testen 936
 - Docblock-Kommentar 931
 - Dokumentaufbau 924
 - each()-Funktion 928
 - Elternklasse ansprechen 946
 - Escaping der Ausgabe 938
 - explode()-Funktion 929
 - extends, Schlüsselwort 943
 - function, Schlüsselwort 932
 - Funktion 932
 - Funktionsparameter 933
 - Funktionswertrückgabe 934
 - global 932
 - globale Variable 932
 - htmlspecialchars(), Funktion 938
 - implode()-Funktion 929
- PHP (Forts.)
 - include()-Funktion 947
 - include_once()-Funktion 947
 - Include-Datei 947
 - Installation 759
 - is_array()-Funktion 936
 - is_float()-Funktion 936
 - is_int()-Funktion 936
 - is_null()-Funktion 936
 - is_numeric()-Funktion 936
 - is_string()-Funktion 936
 - isset()-Funktion 936
 - json_encode(), Funktion 1051
 - Kommentar 931
 - Konstruktor 938
 - magische Methode 937, 946
 - mehrdimensionales Array 929
 - Mock-Objekt für Unit-Tests 969
 - mysqli-Schnittstelle 959
 - Namespace 948
 - Objektorientierung 936
 - Operator 926
 - parent, Schlüsselwort 946
 - php.ini, Konfigurationsdatei 763
 - preg_match()-Funktion 930
 - preg_replace()-Funktion 930
 - Referenz 934
 - regulärer Ausdruck 930
 - require_once()-Funktion 947
 - return-Anweisung 934
 - Session 951
 - sizeof()-Funktion 928
 - Sprachgrundlagen 925
 - static, Schlüsselwort 942
 - statische Methode 942
 - str_replace(), Funktion 948
 - String zerlegen 929
 - strip_tags(), Funktion 938
 - Type Hint 942
 - Unit-Test 966
 - unset()-Funktion 936
 - Variable 925
 - Vererbung 943
 - webspezifische Funktionen 948
- php.ini, Konfigurationsdatei 763
- PHPDocumentor 931
- PHPUnit 1095
 - Mock-Objekt 969
- PHPUnit, Test-Framework 966

- PICT, Bilddateiformat 857
- PID (Prozess-ID) 299
- ping, TCP/IP-Dienstprogramm 357
 - Ergebnisse auswerten* 358
- Pipe 1095
 - Anwendung* 565
 - in Programmiersprachen* 563
 - in Unix-Shell* 382
 - Perl* 565
 - Windows* 335
 - zur Inter-Prozess-Kommunikation* 300
- pipe(), Perl-Funktion 566
- Pipeline (CPU-Warteschlange) 121
- Pixelgrafik 53
- PKZIP-Dateien → ZIP-Datei
- Plankalkül, Programmiersprache 36
- Planung, Software-Engineering 640
- Platform as a Service 989
- Platzhalter
 - in Unix-Dateinamen* 384
 - Windows-Dateiname* 335
- Plug & Play 137, 1095
- Pluralisierung
 - Ruby on Rails* 976
- PNG, Bilddateiformat 856
- Point 1095
- Polymorpher Virus 1060
- Polynomielle Komplexität 95
- pop, Perl-Anweisung 502
- POP3 266, 1095
 - Befehle* 267
 - Sitzung* 266
- Port
 - TCP* 249
- Port, UDP 251
- position, CSS-Angabe 916
- POSIX 1095
- posixAccount, LDAP-Objekt-klasse 775
- POSIX-Standard 285
- POST 129
- POST (BIOS-Selbsttest) 129
- POST, HTTP-Befehl 271
- POST, HTTP-Methode
 - zum HTML-Formularversand* 895
- Post-Dekrement 468
- PostgreSQL 686
- Post-Inkrement 467
- POSTROUTING, iptables-chain 1076
- PostScript 165, 849, 1095
 - EPS* 850
 - PPD* 849
- Power Management 131
- Power-on Self Test → POST
- PowerPC, Prozessor 119
- PowerShell 337
 - Benutzereingabe* 343
 - Cmdlets* 338
 - Datentypen* 343
 - Fallentscheidung* 343
 - Get-Alias, Cmdlet* 338
 - Get-ChildItem, Cmdlet* 338
 - Get-Command, Cmdlet* 338
 - Kontrollstruktur* 343
 - Operator* 339
 - Read-Host, Cmdlet* 343
 - Schleife* 344
 - Skriptdatei* 346
 - Variable* 342
 - Write-Host, Cmdlet* 342
- PPD, Druckerbeschreibungsdatei 849
- PPP 1095
- PPP-Protokoll, DFÜ 210
- Prä-Dekrement 467
- Prädikatenlogik 51, 57, 1095
- Präemptives Multitasking 293
- Prä-Inkrement 467
- Praktische Informatik 26
- Präprozessor, C 482
 - #define* 482
 - #endif* 482
 - #ifdef* 482
 - #include* 459, 482
- Präsentation, OSI-Schicht 179
- preg_match(), PHP-Funktion 930
- preg_replace(), PHP-Funktion 930
- PREROUTING, iptables-Chain 1076
- Primäre Partition 146
- Primärschlüssel 680, 1095
 - einrichten, SQL* 699
- PRIMARY KEY, SQL-Feldoption 699
- print(), Java-Methode 487
- print, Perl-Anweisung 499
- printf(), C-Funktion 460, 479
 - Formatangabe* 460
- println(), Java-Methode 487
- Printserver 192
- private, Java-Kapselung 491
- Problemorientierte Programmiersprache 47
- Programmablaufplan 92
- Programmgesteuerter Rechen-automat 33
- Programmiersprach
 - PHP* 923
- Programmiersprache 45
 - Ada* 35
 - Algorithmus* 541
 - Anweisungsblock* 469
 - Assembler* 46
 - BASIC* 48
 - Baum, Datenstruktur* 553
 - binäre Suche* 549
 - C* 49, 284, 456
 - C#* 51
 - C++* 51
 - Closure* 51
 - Cobol* 47
 - Compiler* 47, 124
 - Datenstruktur* 541
 - Datenstrukturen* 550
 - deklarative* 51
 - Erlang* 51
 - Fortran* 47
 - Funktion* 48
 - funktionale* 51
 - GUI-Programmierung* 608
 - imperative* 48
 - Interpreter* 47, 497
 - Iteration* 546
 - Java* 51, 483
 - Kontrollstrukturen, C* 469
 - lineare Suche* 548
 - LISP* 51, 422
 - logische* 51
 - Logo* 51
 - Maschinensprache* 45

- Programmiersprache (Forts.)
 - mit Datenbanken arbeiten* 720
 - Modularisierung* 48, 474
 - Objective-C* 439
 - objektorientierte* 50
 - Objektorientierung* 484
 - Pascal* 48
 - Perl* 497
 - Pipe* 563
 - Plankalkül* 36
 - problemorientierte* 47
 - Prolog* 51
 - prozedurale* 48
 - Rekursion* 546
 - Scala* 52
 - Skriptsprachen* 47, 497
 - Smalltalk* 50
 - Sortier-Algorithmus* 544
 - Strukturierung* 48, 474
 - Such-Algorithmus* 548
 - Systemaufruf* 125, 563
 - Thread* 567
 - Turing-Vollständigkeit* 100
 - Unicode-Unterstützung* 845
- Programmiersprachen
 - Prozedur* 48
- Programmstrukturierung 48
- Projektmanagement 640
 - Netzplan* 642
- Prolog, Programmiersprache 51
- PROM 127
- Prompt 1095
 - Unix* 371
- prompt(), JavaScript-Methode 996
- protected, Java-Kapselung 493
- Prototype, Ajax-Bibliothek 1053
- Prototype, Entwurfsmuster 663
- Proxy, Entwurfsmuster 664
- Prozedur 48
- Prozedurale Programmiersprache 48
- Prozess
 - als Baum anzeigen, Unix* 396
 - Benutzermodus* 299
 - Child-Prozess* 299
 - Definition* 298
 - duplizierter* 563
 - im Benutzermodus* 292
 - in den Hintergrund stellen* 373
- Prozess (Forts.)
 - init* 299
 - Kernelmodus* 299
 - Kommunikation* 300
 - Management durch Betriebssystem* 281
 - Multitasking* 298
 - Parent-Prozess* 299
 - Prozess-ID (PID)* 299
 - Race Condition* 301
 - Signalverarbeitung* 299
 - Threads als Alternative* 302
 - unter Unix* 298
 - Unterstützung durch CPU* 116, 123
 - Verwaltung durch Betriebssystem* 298
 - Verwaltung, Unix* 395
 - Windows* 300
- Prozessanalyse 644
- Prozesse 1095
 - Deadlock* 301
 - im Kernelmodus* 292
- Prozess-ID 299
- Prozessmanagement 281
- Prozessor 37, 114
 - 3D Now!* 121
 - Adressbus-Wortbreite* 119
 - Alpha* 119, 122
 - als Bauteil* 115
 - ALU* 116
 - AMD* 115
 - Arbeitsweise* 122
 - Architektur* 121
 - Athlon* 115, 121
 - Aufbau* 115
 - bedingter Sprung* 123
 - Befehlstabelle* 116
 - Befehlszeiger* 116
 - Bestandteile* 116
 - Bus* 117
 - Cache* 117
 - CISC* 121
 - Datenbus-Wortbreite* 118
 - der Grafikkarte* 160
 - Effizienz* 120
 - FLOPS* 120
 - Intel* 115
 - Itanium* 119
 - Maschinenbefehle* 124
- Prozessor (Forts.)
 - MIPS* 122
 - MIPS (Geschwindigkeitsangabe)* 120
 - MMX* 121
 - Mooresches Gesetz* 118
 - MosTek 6502* 119
 - Motorola-68000-Familie* 119
 - Pentium-Familie* 115, 121
 - Pipeline* 121
 - PowerPC* 119
 - Prozesse* 123
 - Register* 116
 - Registerwortbreite* 118
 - RISC* 121
 - Sprungbefehl* 122
 - Stack* 123
 - Steuerbus-Wortbreite* 119
 - Steuerwerk* 116
 - Sun SPARC* 122
 - Taktfrequenz* 119
 - übertakten* 120
 - unbedingter Sprung* 122
 - Unterprogramm-Aufruf* 123
 - virtueller* 100
 - Wortbreite* 118
 - Wortbreiten-Vergleich* 119
 - Z80* 119
- Prozessorarchitektur 121
- Prozessverwaltung 298
- Prüfbit 134
- Prüfung
 - IT-Berufe* 30
- ps, Unix-Befehl 300, 395
- PS/2-Anschluss 141
- PSD, Bilddateiformat 854
- Pseudocode, zur Algorithmen-darstellung 92
- pstree, Unix-Befehl 396
- PTR-Record (DNS) 772
- public, Java-Kapselung 486, 491
- Public-Key-Verschlüsselung 1080
- Pulswahlverfahren 212
- Punkt 1095
- push, Perl-Anweisung 502
- puts(), C-Funktion 460, 479
- pwd, Unix-Befehl 387

Q

q//, Perl-Quoting 505
 QBit 45
 qq//, Perl-Quoting 505
 Qt, Grafikbibliothek 424–425
 Quadratische Komplexität 95
 Qualitätsmanagement 641
 Quantencomputer 45
 Quantifizierer (RegExp) 557
 Quartz, Grafikbibliothek 439
 Queue 1095
 Queue, Datenstruktur 550
 Perl 550
 QUEUE, iptables-Regel 1077
 QuickEdit-Modus, Windows-Eingabeaufforderung 335
 QuickSort 1096
 QuickSort, Algorithmus 547
 Funktionsprinzip 547
 Median 547
 QuickTime, Multimedia-Technologie 439
 QuickTime, Video-Dateiformat 858
 qw//, Perl-Quoting 505

R

Race Condition 301, 1096
 Rackspace Cloud Services 989
 radio_button, Rails-Methode 984
 RAID 148, 1096
 Advanced Data Guarding 149
 Levels 148
 Mirroring 149
 Stripe Set 148
 Stripe Set mit Parity 149
 RAID 0 148
 RAID 01 149
 RAID 1 149
 RAID 10 149
 RAID 5 149
 RAID 6 149
 Rails 1096
 Rails → Ruby on Rails
 rails, Generator 974
 RAILS_ENV, Umgebungsvariable 975
 rake, Ruby-Automatisierung 977

RAM 114, 125
 als Bauteil 125
 Auslagerungsdatei 303
 Bedeutung in der Speicherhierarchie 117
 CMOS 129
 DDR-RAM 126
 der Grafikkarte 160
 DIMM-Module 126
 dynamic 125
 EDO 126
 einbauen 126
 empfohlene Menge 127
 FP 126
 Paging 303
 Rambus 126
 RD-RAM 126
 RIMM-Modul 126
 SD-RAM 126
 Segmentierung 303
 Seitenfehler 303
 SIMM-Module 126
 static 125
 Verwaltung durch das Betriebssystem 302
 virtuelle Adressierung 302
 Rambus-RAM 126
 RAMDAC (Grafikkarte) 161
 Random Access Memory → RAM
 Rangfolge
 durch Klammern ändern 468
 Operatoren 468
 Rational Unified Process 650
 Rationale Zahl 68
 Raumfolgearithmetik 80
 rd → rmdir, Windows-Befehl
 RDBMS 1096
 RDBMS → Relationale Datenbank
 RD-RAM 126
 readdir, Perl-Anweisung 510
 Read-Host, PowerShell-Cmdlet 343
 readLine(), Java-Methode 487
 Read-only Memory → ROM
 readyState, Ajax-Eigenschaft 1037
 REAL, SQL-Datentyp 697
 Rechenautomat 33
 Rechenbefehl
 des virtuellen Prozessors 102

Rechendaten 676
 Rechenmaschine
 mechanische 35
 Rechentafel 35
 Rechenwerk 111
 Record → Datensatz
 Red Book (Audio-CD) 151
 Red, Green, Refactor (TDD) 671
 Redirect, Apache-Direktive 752
 redirect_to, Rails-Methode 981
 Reelle Zahl 69
 Referenz
 auflösen, Perl 504
 Perl 503
 regedit, Windows-Dienstprogramm 354
 RegExp → Regulärer Ausdruck
 Regionale Einstellungen
 Mac OS X 448
 Register 1096
 der CPU 116
 des virtuellen Prozessors 100
 Wortbreite 118
 Register (Schaltung) 90
 Registermaschine 100, 1096
 Registrierdatenbank → Registry, Windows
 Registry, Windows 353
 Regulärer Ausdruck 555, 1096
 !~, Operator 560
 =~, Perl-Operator 560
 Alternative Textteile 559
 alternative Zeichen 557
 beliebig viele Zeichen 557
 ein oder mehr Zeichen 557
 ersetzen 560
 Escape-Sequenz 558
 grep 390
 Groß- und Kleinschreibung ignorieren 559
 in PHP 930
 JavaScript 1006
 Klammern 559
 Leerzeichen 558
 m//, Operator 560
 m//-Operator 556
 Matching 560
 mehrere Treffer finden 559
 Modifikator 559
 Muster 556

- Regulärer Ausdruck (Forts.)
 - optionale Zeichen* 557
 - Perl* 510, 555
 - Quantifizierer* 557
 - s///-Operator* 556, 560
 - Sonderzeichen* 558
 - Teilausdruck* 559
 - Whitespace* 558
 - Wortgrenze* 558
 - Wortzeichen* 558
 - Zeichen ausschließen* 557
 - Zeichenanzahl* 558
 - Zeichengruppe* 557
 - Zeilenanfang* 559
 - Zeilenende* 559
 - Ziffer* 558
- Regulärer Ausdrucke
 - in Ruby* 562
- reiserfs, Linux-Dateisystem 393
- REJECT, iptables-Regel 1077
- Rekursion 546, 1096
- Relais 36
- Relation
 - im RDBMS* 680
- Relationale Algebra 679
- Relationale Datenbank 676, 679, 1096
 - 1:1-Beziehung* 680
 - 1:n-Beziehung* 680
 - Access* 686
 - Änderungsabfrage* 695
 - Arten* 686
 - atomare Information* 683
 - Auswahlabfrage* 682, 694
 - Boyce-Codd-Normalform* 684
 - Desktopdatenbank* 686
 - Einfügeabfrage* 694
 - FileMaker* 686
 - freier Server* 686
 - Fremdschlüssel* 680
 - Grenzen* 687
 - Index* 680
 - Java-Programmierung* 720
 - JDBC* 720
 - kommerzieller Server* 686
 - Konsistenz* 679
 - Löschabfrage* 695
 - m:n-Beziehung* 681
 - MySQL* 689
 - Normalform* 683
- Relationale Datenbank (Forts.)
 - Normalisierung* 683
 - ODBC* 720
 - OpenOffice.org Base* 686
 - PostgreSQL* 686
 - Primärschlüssel* 680
 - Primärschlüssel einrichten, SQL* 699
 - Programmierung* 720
 - Relation* 680
 - Schlüssel* 680
 - SQL* 683, 694
 - Tabelle erzeugen, SQL* 695
 - Tabelle löschen, SQL* 696
- Relationale Datenbank, Join-Abhängigkeit 685
- Relativer Pfad 307, 310
- Remote Method Invocation
 - RMI, Java
- rename, Windows-Befehl 337
- repaint(), AWT-Methode 618
- Replikation 172
 - MySQL* 718
- Request For Comments → RFC
- Require, Apache-Direktive 747, 752
- require_once(), PHP-Funktion 947
- RequireAll, Apache-Direktive 753
- RequireAny, Apache-Direktive 753
- RequireNone, Apache-Direktive 753
- Resource Fork, HFS 448
 - Creator ID* 448
 - File Type ID* 448
- responseText, Ajax-Eigenschaft 1039
- responseXML, Ajax-Eigenschaft 1043, 1045
- Ressource
 - Hardware-* 135
 - Plug & Play* 137
 - Zuweisung* 137
- Ressourcenmanagement 641
- REST 1096
- ResultSet, JDBC-Klasse 722
- Resume, Mac OS X 452
- return, C-Anweisung 460, 474
- RETURN, iptables-Regel 1077
- REVOKE, MySQL-Anweisung 711
- RFC 173, 1096
 - 1034 und 1035, DNS* 253
 - 1300* 173
 - 1723, RIP-2* 243
 - 1738, URL* 270
 - 1918, private IP-Adressen* 224
 - 2045 bis 2049, MIME* 263
 - 2060, IMAP* 267
 - 2131 und 2123, DHCP* 244
 - 2178, OSPF* 243
 - 2324, HTCPCP* 173
 - 2460, IPv6* 233
 - 2616, HTTP* 269
 - 2821, SMTP (Neufassung)* 262
 - 2822, Textnachricht* 263
 - 3330, Spezial-IP-Adressen* 224
 - 768, UDP* 247
 - 791, IP-Protokoll* 220
 - 793, TCP* 247
 - 821, SMTP* 262
 - 822, Textnachricht* 263
 - 854, Telnet* 257
 - 959, FTP* 259
 - 977, NNTP* 267
- RGB-Farbe 54
- Rhapsody (Mac OS X) 437
- RIMM-Modul (RAM) 126
- Ring-Topologie, Netzwerk 188
- RIP 1096
- RIP, Routing-Protokoll 242
- RISC 1096
- RISC-Prozessor 121
 - Beispiele* 122
- Risikomanagement 642
- Ritchie, Dennis 49, 284, 457
- rm, Unix-Befehl 386
- rmdir, Unix-Befehl 387
- rmdir, Windows-Befehl 337
- RMI (Remote Method Invocation) 1096
- RMI, Java 600
- robots.txt, Suchmaschinen-Info 907
- Röhrenmonitor 161
- Röhrenrechner 36
- Rollback (Transaktionen) 685
- ROLLBACK, SQL-Anweisung 705

- Rollover-Effekt
 - JavaScript* 1013
 - praktische JavaScript-Lösung* 1014
- ROM 114, 127
 - Bauarten* 127
 - Bedeutung* 114
 - bei 8-Bit-Homecomputern* 114
 - BIOS* 127
- Römische Zahl 73
- root, Benutzer 300
 - Home-Verzeichnis* 306
 - temporär arbeiten als* 377
- root, Unix-Benutzer 369
- root, Unix-Verzeichnis 306
- Rootkit 1072
- ROT13 1079
- Roter Balken (Unit-Test) 670
- Round-Robin-DNS 772
- route, Unix-Befehl 429
- Router 1096
- Router, IP-Protokoll 237
- Routing
 - autonomes System* 241
 - DE-CIX* 242
 - IP-Protokoll* 236
- Routing-Protokoll 241
 - BGP* 243
 - OSPF* 243
 - RIP* 242
- Routing-Tabelle 239
 - anzeigen* 239
- rpm, Linux-Paketmanager 401
- RS-232 141
- RS-Flip-Flop 89
- RSS, Newsfeed-Format 988
- Ruby 1096
 - gsub, Methode* 562
 - rake, Automatisierungstool* 977
 - regulärer Ausdruck* 562
 - Singleton, Modul* 667
 - sub, Methode* 562
- Ruby on Rails 973, 1096
 - Action Mailer* 973
 - Action Pack* 973
 - Action Web Service* 973
 - Active Record* 973
 - Active Support* 973
 - Anwendung erzeugen* 974
 - Anwendung testen* 979
- Ruby on Rails (Forts.)
 - belongs_to, Relation* 977
 - Capistrano, Deployment-Tool* 985
 - console* 978
 - Controller* 980
 - Convention over Configuration* 974
 - create, Methode* 978
 - create_table, Methode* 976
 - database.yml, Konfigurations-datei* 975
 - Datensätze ändern* 979
 - Deployment* 985
 - Designprinzipien* 974
 - Don't Repeat Yourself* 974
 - eRuby* 980
 - find, Methode* 978
 - flash, Zwischenspeicher* 984
 - form_tag, Methode* 984
 - Formular* 984
 - Fremdschlüssel* 977
 - Generator* 974
 - has_and_belongs_to_many, Relation* 977
 - has_many, Relation* 977
 - Helper* 982
 - Installation* 974
 - interaktive Konsole* 978
 - Komponenten* 973
 - Migration* 976
 - Model erzeugen* 976
 - Mongrel, Webserver* 979, 985
 - Pluralisierung* 976
 - Praxisbeispiel* 974
 - radio_button, Methode* 984
 - rails, Generator* 974
 - RAILS_ENV, Umgebungsvariable* 975
 - rake db:migrate* 977
 - redirect_to, Methode* 981
 - Relation* 977
 - Scaffolding* 979
 - Startseite* 981
 - submit_tag, Methode* 985
 - Verzeichnisstruktur* 974
 - View* 980
- Rumbaugh, James 654
- run(), Java-Methode 568
- Runlevel (Unix) 399
- Runnable, Interface 568
- Runnable, Java-Interface 495

S

- s///, Perl-Operator* 510, 556, 560
- SaaS (Software as a Service) 989
- Samba 432, 1096
 - als Client für Windows-Server* 434
 - Drucker freigeben* 434
 - globale Parameter* 433
 - Konfiguration* 433
 - smb.conf, Konfigurationsdatei* 433
 - starten* 432
 - Verzeichnis freigeben* 434
 - Windows-Freigabeart* 434
- Sampling 1096
- Sampling, Audio 54
- Sampling-Rate 54
- Sampling-Tiefe, Audio 54
- Samsung
 - Galaxy Tab* 45
- SAS (Serial Attached SCSI) 140
- sash (Stand-alone Shell) 372
- Satellit, DSL-Verbindung 215
- Satisfy, Apache-Direktive 753
- SAX 1096
 - Beispielprogramm* 834
 - ContentHandler, Interface* 828
 - ContentHandler-Callback* 828
 - ContentHandler-Methode* 828
 - Dokument parsen* 827
 - DTDHandler, Interface* 828
 - EntityResolver, Interface* 828
 - ErrorHandler, Interface* 828
 - Event Handler* 828
 - InputSource, Klasse* 827
 - Parser-Instanz erzeugen* 827
 - XMLReader-Interface* 827
- SAX (Simple API for XML) 827
- SaX (X-Server-Konfigurationsprogramm) 423
- sbin, Unix-Verzeichnis 306
- Scaffolding
 - Ruby on Rails* 979
- Scala, Programmiersprache 52
- scanf(), C-Funktion 471, 479

- Scanner 158
 - Auflichtscanner* 158
 - Diascanner* 159
 - Durchlichtscanner* 158
 - Flachbettscanner* 158
 - Kleinbildscanner* 159
 - Trommelscanner* 158
- Schaltalgebra 26, 83
- Schaltkreisvermittlung 170
- Schaltung, Register 90
- Schema (LDAP) 774
- Schichtenmodell 176, 1096
 - Alltagsbeispiel* 182
 - Mail-Beispiel* 184
 - OSI-Referenzmodell* 177
 - Praxis* 182
 - TCP/IP* 179
- Schleife
 - C* 472
 - do/while()* 473
 - Endlosschleife* 548
 - for()* 473
 - foreach, Perl* 502
 - fußgesteuerte* 473
 - in der PowerShell* 344
 - in Shell-Skripten* 405
 - kopfgesteuerte* 472
 - mit break abbrechen* 548
- Schleifenrumpf 472
- Schleifenzähler, Variable 472
- Schlüssel
 - eines Hashes* 503
 - im RDBMS* 680
- Schlussfolgerung
 - logische* 60
 - Umkehrschluss* 60
- Schnittmenge 70
- Schnittstelle
 - Hardware* 115
 - Softwareentwurf* 647
- Schreib-Lese-Kopf, der Turing-Maschine 97
- Schriftart
 - im Drucker* 165
- screen, JavaScript-Objekt 1023
- script.aculo.us, Ajax-Bibliothek 1053
- ScriptAlias, Apache-Direktive 753
- Scrum 652
 - Backlog* 653
- Scrum (Forts.)
 - Rollen* 652
 - Sprint* 653
- SCSI 139, 1096
 - anschließen* 139
 - ID* 140
 - serielles* 140
 - Terminator* 139
- SCSI-ID 140
- SDI, Windows-Anwendungen 327
- SD-RAM 126
- SDSL 215
- sed 497, 555
- Sedezimalsystem → Hexadezimalsystem
- Segmentierung (Speicher) 303
- Seite (Speicher) 303
- Seitenfehler (Speicher) 303
- Seitentabelle (Speicher) 303
- SELECT, SQL-Abfrage 699
- Selektor
 - jQuery* 1053
- Semikolon
 - Abschluss von Anweisungen* 459
- send(), Ajax-Methode 1038
- Sequenzdiagramm (UML) 659
- Serial Attached SCSI 140
- Serialisierung 495
- Serializable, Java-Interface 495
- Serielle Datenübertragung 133
 - Bedeutung* 134
 - Kontrollbit* 133
 - Leitungskonventionen* 134
 - Prüfbit* 134
 - Startbit* 133
 - Stoppbit* 134
- Server
 - Netzwerk* 190
- Server, Windows-Betriebssysteme 320
- ServerAdmin, Apache-Direktive 753
- Serverdienst
 - Anwendungsserver* 194
 - Application Server* 194
 - Dateiserver* 191
 - Druckserver* 192
 - einrichten, Linux* 430
 - einrichten, Mac OS X* 451
- Serverdienst (Forts.)
 - einrichten, Unix* 430
 - einrichten, Windows* 360
 - Fileserver* 191
 - HTTP-Server* 193
 - Mailserver* 192
 - Printserver* 192
 - Samba* 432
 - Übersicht* 190
 - Webserver* 193
- Servergefahren 1071
- ServerName, Apache-Direktive 753
- ServerRoot, Apache-Direktive 754
- ServerSignature, Apache-Direktive 754
- Serversystem
 - Macintosh* 451
- ServerTokens, Apache-Direktive 754
- Servlet, Java 604
- Session
 - PHP* 951
- Session-Hijacking 1072
- SET PASSWORD, MySQL-Anweisung 709
- SET, SQL-Befehl 704
- SET, SQL-Datentyp 699
- setColor(), AWT-Methode 612
- setLayout(), AWT-Methode 620
- setTimeout(), JavaScript-Methode 1011
- setVisible(), AWT-Methode 610
- SGML 1096
 - HTML-DTD* 866
 - XML als moderne Version* 791
- sh (Bourne-Shell) 372
- Shannon-Theorem 1096
- Share Level Security, Samba 434
- Share Level Security, Windows-Freigabeart 361
- Shebang
 - bei Shell-Skripten* 404
 - Perl* 498
- Shell 297, 1096
 - /etc/profile, Unix-Konfigurationsdatei* 373
 - Ausgabeumleitung (Unix)* 381
 - bash* 372

- Shell (Forts.)
 - Befehl, Windows* 336
 - Bourne-Shell* 372
 - Cmd.exe*
 - WinNT 334
 - COMMAND.COM, MS-DOS 334
 - C-Shell* 372
 - Eingabeaufforderung, Unix* 371
 - Eingabeumleitung (Unix)* 381
 - Eingabevervollständigung* 375
 - ermitteln, welche läuft* 372
 - Escape-Sequenz* 391
 - HIER-Dokument* 381
 - History* 376
 - Korn Shell* 372
 - Pipe* 382
 - Prozess in den Hintergrund stellen* 373
 - Shell-Skript* 403
 - Stand-alone Shell* 372
 - Umgebung* 372
 - Umgebungsvariable* 373
 - Unix* 297
 - unter Mac OS X* 440
 - Windows* 333
- shell
 - Befehl als root ausführen* 377
- Shell-Skript 403
 - Beispiele* 406
 - case-Befehl* 405
 - Fallunterscheidung* 404
 - for-Befehl* 405
 - if-Befehl* 404
 - Schleife* 405
 - Shebang* 404
 - Variable* 405
 - while-Befehl* 405
- shift, Perl-Anweisung 502
- short, C-Datentyp 463
- Short-Circuit-Logik 507
- shutdown, Unix-Befehl 397
- Sicherheit 1057
 - Admin-Rechte* 1063
 - Adware* 1067
 - Backdoor* 1064
 - Backup* 1062
 - CGI* 1072
 - chroot-Umgebung* 1074
 - Crackerangriff* 1072
 - Sicherheit (Forts.)*
 - Crackertools* 1075
 - Cross-Site-Scripting (XSS)* 1072
 - Dialer* 1065
 - Exploit* 1071
 - Firewall* 1062, 1073
 - Hoax* 1070
 - Intrusion Detection System* 1074
 - keine absolute* 1057
 - Kettenmail* 1070
 - Kryptografie* 1079
 - Man-in-the-Middle-Angriff* 1072
 - menschliches Versagen* 1075
 - MySQL, Unix* 691
 - MySQL, Windows* 693
 - Passwort* 1075
 - Patch installieren* 1062
 - PC-Gefahren* 1058
 - Phishing* 1067
 - Rootkit* 1072
 - Servergefahren* 1071
 - Session-Hijacking* 1072
 - Social Engineering* 1075
 - Spam* 1068
 - Spyware* 1067
 - SQL-Injection* 1073
 - Virus* 1058
 - Webanwendungen* 1072
 - Wurm* 1060
 - Sicherung, OSI-Schicht* 178
 - SIGALRM, Signal* 299
 - SIGHUP, Signal* 299
 - SIGINT, Signal* 397
 - SIGKILL, Signal* 299, 396
 - Signal* 1096
 - an Prozesse senden, Unix* 396
 - SIGALRM* 299
 - SIGHUP* 299
 - SIGINT* 397
 - SIGKILL* 299, 396
 - SIGTERM* 299, 396
 - Verarbeitung durch Prozess* 299
 - zur Inter-Prozess-Kommunikation* 300
- Signatur
 - digitale* 1081
- signed, C-Datentyp 463
- SIGTERM, Signal 299, 396
- Silicon Valley 38
- Silizium 38
- SIMM-Modul (RAM) 126
- Simple API for XML → SAX
- Simulation
 - eines Prozessors* 100
- Sinclair ZX Spectrum 286
- Sinclair ZX81 42, 286
- Sinclair, Clive 42
- Single Document Interface → SDI, Windows-Anwendu
- Singleton, Entwurfsmuster 663, 665
 - Implementierung (Java)* 666
 - Implementierung (Ruby)* 666
 - Ruby-Modul* 667
- Sinuskurve
 - zeichnen, AWT* 613
- Sitzung, OSI-Schicht 179
- sizeof(), PHP-Funktion 928
- Skalar, Perl-Variable 499, 501
- Skriptsprache 47, 497
- Slash, Blog-Tool 988
- Slave-Nameserver 255
- Slot
 - Prozessor* 115
- SMALLINT, SQL-Datentyp 697
- Smalltalk 638
- Smalltalk, Programmiersprache 50
- Smartphone 44
- smbclient, Samba-Dienst 434
- SMB-Protokoll, Windows-Netzwerk 277
- SMTP 262
 - Befehle* 263
 - Sitzung* 262
- sn, LDAP-Attribut 776
- Snort 1074
- SOAP 1096
 - Client* 607
- SOAP, Webservices 606
- SOA-Record (DNS) 771
- Social Engineering 1075
- sockaddr_in, Netzwerkprogrammierung 573
- Sockel
 - für Prozessoren* 115
- Socket 571, 1097
 - accept(), Funktion* 577
 - Adresse* 573

- Socket (Forts.)
 - bind()*, Funktion 576
 - connect()*, Funktion 575
 - Datagramme senden und empfangen 574
 - Domain 572
 - erzeugen 572
 - IO 577
 - IP-Adresse 573
 - lauschender 576
 - listen()*, Funktion 576
 - Protokoll 573
 - TCP 575
 - TCP-Client 575
 - TCP-Port 573
 - TCP-Server 576
 - Typ 572
 - UDP 574
 - Verbindung aufnehmen 577
- Socket, Perl-Modul 572
- Software
 - freie 289
 - installieren unter Unix 401
 - installieren, Windows 350
- Software as a Service 989
- Software-Engineering 637
 - agiler Entwicklungsprozess 651
 - Analyse 643
 - CASE-Tools 655
 - Code-Review 648
 - Dokumentation 648
 - Entwicklungsprozess 649
 - Entwicklungszyklus 639
 - Entwurfsmuster 660
 - Entwurfsphase 646
 - Extreme Programming 651
 - Implementierungsphase 647
 - in der IT-Ausbildung 638
 - Lastenheft 644
 - objektorientierte Analyse 644
 - Objektorientierung 637
 - Pflichtenheft 645
 - Planungsphase 640
 - Projektmanagement 640
 - Projektphasen 639
 - Schnittstelle 647
 - Scrum 652
 - Spiralmodell 639
 - strukturierte Analyse 644
 - Test-first-Verfahren 652, 671
- Software-Engineering (Forts.)
 - Testphase 648
 - UML 654
 - Unified Process 650
 - Unit-Test 648
 - Unit-Tests 668
 - Wasserfallmodell 639
- Softwareentwicklung → Software-Engineering
- Softwarekrise 637
- Softwaretechnik → Software-Engineering
- Solaris 1097
- Solaris, Betriebssystem 285
- Solid State Disk 1097
- Solid State Disk (SSD) 150
- Sortier-Algorithmus 544
 - BubbleSort 544
 - QuickSort 547
- Soundkarte 166
 - Anschlüsse 166
 - Audio-CD abspielen 166
 - MIDI 166
 - SP-DIF-Anschluss 166
- Source, Stromeingang des Transistors 84
- Spam 1068
- SpamAssassin 1069
- SP-DIF-Anschluss 166
- Special File → Gerätedatei
- Spectrum, Homecomputer 286
- Speicher
 - Management durch Betriebssystem 281
 - RAM 114
 - reservieren, C 551
 - ROM 114
 - virtueller 118, 302
- Speicheradressierung 78
- Speichermanagement 281
- Speicherseite 303
- Speicherverwaltung 302
 - x86-System 303
- Speicherzelle (Schaltung) 89
- Spiralmodell 639
- split, Perl-Anweisung 503
- Spotlight (Mac OS X) 443
- Sprache (Umgangssprache), zur Algorithmendarstellung 92
- Sprint, Scrum 653
- Sprungbefehl
 - bedingter 123
 - beim virtuellen Prozessor 103
 - der CPU 122
 - unbedingter 122
- Sprungvorhersage (Prozessor) 118
- Spyware 1067
- SQL 683, 694, 1097
 - Aggregatfunktion 701
 - Änderungsabfrage 695, 704
 - arithmetische Operatoren 701
 - AS-Klausel 701
 - Auswahlabfrage 694, 699
 - AUTO_INCREMENT, Feldoption 699
 - BIGINT, Datentyp 697
 - BINARY, Feldoption 699
 - BLOB, Datentyp 698
 - CHAR, Datentyp 698
 - COMMIT, Anweisung 705
 - COUNT-Funktion 702
 - CREATE DATABASE, Befehl 695
 - CREATE TABLE, Befehl 695
 - DATE, Datentyp 697
 - Datentypen 697
 - Datentypen in Java 723
 - Datentypen, Aufzählung 698
 - Datentypen, Binärobjekte 698
 - Datentypen, Datum und Uhrzeit 697
 - Datentypen, Fließkomma 697
 - Datentypen, ganzzahlige 697
 - Datentypen, Text 698
 - DATETIME, Datentyp 697
 - DECIMAL, Datentyp 697
 - DEFAULT, Feldoption 699
 - DELETE-Abfrage 704
 - DOUBLE, Datentyp 697
 - DROP DATABASE, Befehl 696
 - DROP TABLE, Befehl 696
 - Einfügeabfrage 694, 703
 - ENUM, Datentyp 698
 - Felder mit Nullen füllen 699
 - Feldoptionen 699
 - Feldwert einmalig machen 699
 - FLOAT, Datentyp 697
 - Funktionen 701
 - Index erstellen 699
 - Inner Join 702

- SQL (Forts.)
 - Inner Join durch WHERE ausdrücken* 703
 - INNER JOIN-Klausel 702
 - INSERT-Abfrage 703
 - INT, Datentyp 697
 - Join 702
 - LIKE-Klausel 700
 - LONGBLOB, Datentyp 698
 - LONGTEXT, Datentyp 698
 - Löschabfrage 695, 704
 - MAX-Funktion 702
 - MDEIUMBLOB, Datentyp 698
 - MEDIUMTEXT, Datentyp 698
 - MIN-Funktion 702
 - Mustervergleich 700
 - NULL, Feldoption 699
 - ORDER BY-Klausel 702
 - Primärschlüssel einrichten 699
 - PRIMARY KEY, Feldoption 699
 - REAL, Datentyp 697
 - ROLLBACK, Anweisung 705
 - SELECT-Abfrage 699
 - SET, Datentyp 699
 - SET-Befehl 704
 - SMALLINT, Datentyp 697
 - sortieren 702
 - Standardwert angeben 699
 - START TRANSACTION, Anweisung 705
 - SUM-Funktion 701
 - TEXT, Datentyp 698
 - TIME, Datentyp 697
 - TIMESTAMP, Datentyp 698
 - TINYBLOB, Datentyp 698
 - TINYINT, Datentyp 697
 - TINYTEXT, Datentyp 698
 - Transaktion beginnen 705
 - UNSIGNED, Feldoption 699
 - UPDATE-Abfrage 704
 - VARCHAR, Datentyp 698
 - Vergleichsoperatoren 701
 - Volltextindex 699
 - WHERE-Klausel 700
 - YEAR, Datentyp 697
 - ZEROFILL, Feldoption 699
- SQL Server, Microsoft 686
- SQL-Injection 1073
- SRAM 125
- SSD 1097
- SSD (Solid State Disk) 150
- SSH 1082, 1097
- SSH (Secure Shell) 257
 - OpenSSH 258
- SSL 757, 1081
- Stack
 - der CPU 116, 123
 - des virtuellen Prozessors 104
 - Java-Klasse 497
- Stack Overflow 123
- Stack Pointer → Stack-Zeiger
- Stack, Datenstruktur 550
 - Perl 550
- Stack-Overflow 1097
- Stack-Zeiger 116, 123
- Stallman, Richard 289
- Stammdaten 675
- Stand-alone Shell (sash) 372
- Standardausgabe
 - Java 487
- Standardausgabe (stdout) 381
- Standardeingabe (stdin) 381
- Standardfehlerausgabe 570
- Standard-Fehlerausgabe (stderr) 381
- Stapelverarbeitung 283
- START TRANSACTION, SQL-Anweisung 705
- Startbit 133
- Startmenü
 - Windows 328
- Startvolume, Mac OS X-Einstellung 447
- startx, X Window starten 423
- State, Entwurfsmuster 664
- Statement, JDBC-Klasse 722
- static
 - Java-Methoden 486
 - Variableneigenschaft, C 463
- Static RAM → SRAM
- static, PHP-Schlüsselwort 942
- Statische Methode
 - PHP 942
- Statische Variable 463
- Statisches Attribut
 - PHP 942
- stddef.h, C-Bibliothek 463
- stderr 570
- stderr, Standard-Fehlerausgabe 381
- stdin, Standardeingabe 381
- stdio.h, C-Bibliothek 479
- stdio.h, C-Header-Datei 459
- stdlib.h, C-Header-Datei 459
- stdout, Standardausgabe 381
- Stealth-Virus 1060
- Stellenwertsystem 35, 72
 - Basis 73
- Stern-Topologie, Netzwerk 188
- Steueranweisung
 - XML 795
- Steuerbus 117
 - Wortbreite 119
- Steuerwerk 111
 - der CPU 116
- Stoppbit 134
- str_replace(), PHP-Funktion 948
- Strategy, Entwurfsmuster 664
- strcat(), C-Funktion 481
- strcmp(), C-Funktion 481
- strcpy(), C-Funktion 481
- Stream-Socket 572
- strftime(), C-Funktion 481
- strict, Perl 500
- String 459, 1097
 - aufteilen, Java 489
 - aus Array, Perl 503
 - aus Datei lesen, C 480
 - Darstellung in C 477
 - Darstellung, C 459
 - Eingabe, C 480
 - einlesen, C 460
 - einzelne Zeichen lesen, Java 489
 - Funktionen in C 481
 - in GUI schreiben 614
 - in JavaScript 1004
 - Java 487
 - kopieren, C 481
 - Länge ermitteln, Java 489
 - null-terminierter 477
 - Operationen in Java 489
 - Position ermitteln, Java 489
 - vergleichen, C 481
 - vergleichen, JavaScript 999
 - vergleichen, Perl 505
 - verketteten, C 481
 - verketteten, Java 488
 - verketteten, JavaScript 997
 - verketteten, Perl 506
 - zerlegen, Perl 503

- String (Forts.)
 - zerlegen, PHP* 929
 - String, Java-Datentyp 487
 - string.h, C-Bibliothek 481
 - String-Literal 464
 - String-Verkettung, Perl 506
 - strip_tags(), PHP-Funktion 938
 - Stripe Set (RAID) 148
 - Stripe Set mit Parity (RAID) 149
 - Stroustrup, Bjarne 51
 - struct, C 478
 - Structural Pattern → Struktur-
muster
 - Structured Query Language
→ SQL
 - Struktur
 - in C* 478
 - Strukturierte Analyse 644
 - Strukturierung
 - Programme* 474
 - von Programmen* 48
 - Strukturmuster 661
 - Studiengänge
 - Informatik* 31
 - Style Sheets → CSS
 - su, Unix-Befehl 377
 - sub, Perl-Subroutine 508
 - sub, Ruby-Methode 562
 - Subdomain (DNS) 254
 - submit_tag, Rails-Methode 985
 - Subnet Mask 1097
 - Subnet Mask, IP-Adresse 224
 - Subnetting 1097
 - Subnetting, IP-Netze teilen 225
 - Subroutine
 - Parameter* 508
 - Perl* 508
 - Werrückgabe* 509
 - SubSeven, Backdoor 1064
 - substring(), Java-Methode 489
 - Subtraktion, Operator 465
 - Such-Algorithmus 548
 - binäre Suche* 549
 - lineare Suche* 548
 - Suche
 - binäre* 549, 1088
 - lineare* 94, 1094
 - nach Permutationen* 95
 - Suchmaschine
 - Anmeldung bei* 907
 - Suchmaschine (Forts.)
 - HTML aufbereiten für* 906
 - robots.txt-Datei* 907
 - SUM, SQL-Funktion 701
 - Sun Microsystems
 - Java* 483
 - Solaris, Betriebssystem* 285
 - Sun SPARC, Prozessor 122
 - super, Java 494
 - Supernetting 1097
 - Supernetting, IP-Netze zusammen-
fassen 226
 - Superuser 300, 369
 - SVG 1097
 - SVG (Scalable Vector Graphics)
792
 - SVG → Scalable Vector Graphics
 - Swap-Partition 303
 - Swapping (Speicher) 118
 - Swing, Java 609
 - Ereignisbehandlung* 609
 - Tabellen* 609
 - Switch 201
 - switch/case-Fallunterscheidung
471
 - default-Wert* 472
 - Symbolic Link 308
 - Symbolische Konstante 482,
1097
 - Symmetrische Verschlüsselung
1080
 - sys/types.h, C-Bibliothek 564
 - Syslog 397, 1097
 - Syslog, Unix 396
 - System
 - autonomes* 1088
 - System V 1097
 - System V Init 399
 - System V IPC 301
 - System V, Unix 285
 - System.err, Java 570
 - System.out, Java 487
 - Systemanalyse 643
 - Systemaufruf 125, 293, 295,
1097
 - CreateProcess()* 300
 - fork()* 299
 - kill()* 299
 - pause()* 299
 - programmieren* 563
 - Systemaufruf (Forts.)
 - Win32 API* 296
 - Systemaufrufe
 - Unix* 295
 - Systembefehl
 - Unix* 383
 - Unix-Dateimanipulation* 384
 - Unix-Systemverwaltung* 392
 - Unix-Textmanipulation* 388
 - Systemkonfiguration
 - Mac OS X* 447
 - Windows* 348
 - Systemprogramm 294
 - Linux* 288
 - Optionen* 373
 - Parameter* 373
 - Unix* 383
 - Unix-Dateimanipulation* 384
 - Unix-Systemverwaltung* 392
 - Unix-Textmanipulation* 388
 - Systemsteuerung, Windows 348
 - Systemvariable 1097
 - PATH (Unix)* 374
 - setzen (Unix)* 375
 - Systray
 - Windows* 330
- ## T
-
- Tabelle
 - erzeugen, SQL* 695
 - HTML* 888
 - löschen, SQL* 696
 - Tabelle (iptables) 1076
 - Tablet-PC 45
 - Tag
 - HTML* 867
 - Name, XML* 796
 - verschachteltes, XML* 795
 - XML* 793, 795
 - tail, Unix-Befehl 389
 - Taktfrequenz
 - der CPU* 119
 - des Mainboards* 119
 - Multiplikator* 119
 - praktische Bedeutung* 120
 - Tanenbaum, Andrew 292
 - tar, Unix-Befehl 402, 860
 - tar-Datei 401
 - Task Scheduler 292, 1097

- Taskleiste, Windows 328
 - Systray* 330
- Tastatur 157
 - Zeichensatzeinstellung* 845
- Tastenkürzel
 - Windows-Eingabeaufforderung* 334
- TCP 247, 1097
 - Drei-Wege-Handshake* 249
 - Funktionsweise* 247
 - im Vergleich zu UDP* 247
 - Paket-Header* 247
 - Port* 249
 - Urgent Data* 250
 - Verbindungsaufbau* 249
- TCP/IP 217, 1097
 - Adressierung* 220
 - Anwendungsprotokolle* 256
 - ARP, Netzzugang* 219
 - DHCP* 244
 - Dienstprogramme* 357
 - DNS* 252
 - Domain Name System* 252
 - FTP, Anwendungsprotokoll* 259
 - HTTP, Anwendungsprotokoll* 269
 - ICMP-Protokoll* 247
 - IMAP, Anwendungsprotokoll* 267
 - Loopback-Interface* 223
 - Nameserver* 252
 - Netzzugang* 219
 - NNTP, Anwendungsprotokoll* 267
 - POP3, Anwendungsprotokoll* 266
 - Routing* 236
 - Routing-Protokolle* 241
 - SMTP, Anwendungsprotokoll* 262
 - Telnet, Anwendungsprotokoll* 257
 - Transportprotokolle* 246
- TCP/IP-Dienstprogramm 357
 - netstat* 359
 - nslookup* 359
 - tracert* 358
- TCP/IP-Dienstprogramme
 - ping* 357
- TCP/IP-Schichtenmodell 179
- TCP-Client-Socket 575
 - mit IO* 578
- TCP-Header 247
- TCP-Port 249
 - für Sockets* 573
 - Well-known Port* 250
- TCP-Server-Socket 576
 - mit IO* 578
- tcsh (erweiterte C-Shell) 372
- TDD → Test-driven Development
- Technische Informatik 26
- Teilerfremd 542
- Teilmenge 67
 - echte* 68
- Teilnetzmaske, IP-Adresse 224
- Telefongespräch
 - als Schichtenmodell* 182
- Telefonleitung
 - Pulswahl* 212
- Telefonverbindung 170
- telephoneNumber, LDAP-Attribut 776
- Telnet 257, 1097
- Template Method, Entwurfsmuster 664
- Terabyte 79
- Term 58
- TERM, Signal 299, 396
- Terminal 39, 283, 1097
 - unter grafischer Oberfläche* 371
 - virtuelles* 370
- Terminator 1097
 - SCSI* 139
- Terminator (UNIX-Terminalfenster) 371
- Test, Software-Engineering 648
 - Code-Review* 648
 - Lasttest* 648
 - Unit-Test* 648
 - Unit-Tests* 668
- Test-driven Development 671, 1097
- Test-first-Verfahren 652, 671
- Testgetriebene Entwicklung
 - Test-driven Development
- TeX, Textsatzsystem 848
- Text
 - ausgeben (Unix)* 388
 - Dateianzeigen, Windows-Konsole* 337
- Text (Forts.)
 - Dateien vergleichen, Unix* 391
 - Dateiinhalt anzeigen (Unix)* 389
 - Editoren, Unix* 410
 - Emacs, Editor* 417
 - Manipulationsbefehle, Unix* 388
 - vi, Editor* 410
 - Wörter zählen, Unix* 392
- TEXT, SQL-Datentyp 698
- text-align, CSS-Angabe 914
- TextArea, AWT-Klasse 619
- Textdatei
 - anzeigen, Windows-Konsole* 337
 - Inhalt anzeigen (Unix)* 389
- Textdateiformat
 - Vorteile* 791
 - YAML* 975
- text-decoration, CSS-Angabe 913
- Texteditor
 - Emacs* 417
 - für XML verwenden* 792
 - unter Unix* 410
 - vi* 410
 - vim* 410
- TextField, AWT-Klasse 619
- text-indent, CSS-Angabe 914
- TFT 1098
- TFT-Monitor 162
- Theoretische Informatik 26
- Thermosublimationsdrucker 165
- Thermotransferdrucker 165
- Thicknet Coaxial, Ethernet 200
- Thinnet Coaxial, Ethernet 199
- this, Java 491
- Thompson, Ken 284
- Thread 291, 302, 1098
 - in Programmiersprachen* 567
 - Java* 495, 567
 - run()* 568
 - Runnable-Interface* 568
- Thread, Java-Klasse 567
- throws-Klausel, Java 497
- Thunderbolt 141
- TIFF, Bilddateiformat 854
- time(), C-Funktion 481
- TIME, SQL-Datentyp 697
- time.t, C-Bibliothek 481
- time_t, C-Datentyp 481
- Timesharing 283, 1098
- TIMESTAMP, SQL-Datentyp 698

- Tintenstrahldrucker 164
 - Bubble-Technik* 164
 - Piezo-Technik* 164
 - TINYBLOB, SQL-Datentyp 698
 - TINYINT, SQL-Datentyp 697
 - TINYTEXT, SQL-Datentyp 698
 - Titel
 - HTML-Dokument* 868
 - TLS → SSL
 - Token Passing, Netzzugangsverfahren 202
 - Token-Passing 1098
 - Token-Ring-Netzwerk 202, 1098
 - Tomlinson, Ray 173
 - Tonkanal, Audio 55
 - Tonwahlverfahren 212
 - Top Level Domain 1098
 - Länder* 254
 - top, CSS-Angabe 916
 - Top-Level-Domain
 - generic* 254
 - Top-Level-Domain (DNS) 254
 - Topologie (Netzwerk) 188
 - Torvalds, Linus 285, 288
 - tr//, Perl-Operator 561
 - tracert, TCP/IP-Dienstprogramm 358
 - Track-at-once 153, 1098
 - Transaktion (Datenbank) 705
 - Commit* 705
 - in MySQL* 705
 - Rollback* 705
 - Transaktion (RDBMS) 685
 - in MySQL* 686
 - Transistor 36, 38, 84
 - TFT* 162
 - Transistorrechner 36
 - Transport, OSI-Schicht 178
 - Treiber → Gerätetreiber
 - Triode 38
 - tripwire 1074
 - Trojaner → Trojanisches Pferd
 - Trojanisches Pferd 1064
 - Trolltech 425
 - Trommelscanner 158
 - Foto-Multiplier* 159
 - true
 - Java* 488
 - TrueType 1098
 - try
 - Java* 487
 - try/catch-Block
 - Java* 487
 - TTL 1098
 - TTL, IP-Datagramm 240
 - Turing, Alan 96, 1098
 - Turing-Maschine 97, 1098
 - Band* 97
 - Beispiele* 98
 - einfaches Beispiel* 98
 - komplexeres Beispiel* 98
 - Schreib-Lese-Kopf* 97
 - Zeichenvorrat* 97
 - Zustände* 97
 - Turing-Test 96, 1098
 - Turing-Vollständigkeit 100, 1098
 - Tweening 1098
 - Twisted-Pair-Kabel 200
 - Kategorien* 200
 - Type Hint (PHP) 942
 - type, Windows-Befehl 337
 - Typecasting
 - C* 465
 - Typenraddrucker 163
 - TYPO3, CMS 987
- ## U
-
- Überladung 492
 - Übertakten 120
 - Übertrag (Logikschaltung) 87
 - Ubuntu Linux 366
 - UDDI 1098
 - UDDI, Webservices 606
 - UDF 154–155
 - UDP 250, 1098
 - Anwendungsbeispiel* 251
 - im Vergleich zu TCP* 247
 - Paketheader* 251
 - Port* 251
 - UDP-Header 251
 - UDP-Port 251
 - UID (User-ID)
 - von Prozessen* 300
 - UID (User-ID, Unix) 369
 - uid, LDAP-Attribut 776
 - uidNumber, LDAP-Attribut 776
 - Umbenennen
 - Datei, Mac OS X* 445
 - Umbenennen (Forts.)
 - Datei, Unix* 385
 - Umfangsmanagement 641
 - Umgebung
 - Unix* 372
 - Umgebungsvariable
 - CLASSPATH* 484
 - PATH (Unix)* 374
 - RAILS_ENV* 975
 - setzen (Unix)* 375
 - setzen unter Windows* 336
 - Umgebungsvariablen 373
 - Umkehrschluss 60
 - UML 92, 654, 1098
 - Akteur* 656
 - Aktivitätsdiagramm* 659
 - Anwendungsfalldiagramm* 656
 - ArgoUML, Tool* 655
 - Diagrammtypen* 654
 - Klassendiagramm* 657
 - praktischer Einsatz* 655
 - Sequenzdiagramm* 659
 - Version 2.0* 654
 - umount, Unix-Befehl 393
 - Umrechnung
 - dezimal nach dual* 75
 - dezimal nach hexadezimal* 76
 - dual nach dezimal* 76
 - dual nach hexadezimal* 77
 - dual nach oktal* 77
 - hexadezimal nach dezimal* 77
 - hexadezimal nach dual* 77
 - oktal nach dual* 77
 - Zahlensysteme* 75
 - UMTS 216
 - unalias, Unix-Befehl 408
 - Unbedingter Sprung 122
 - Undefiniertheitsstelle (Funktion) 93
 - Und-Schaltung → AND-Schaltung
 - Und-Verknüpfung → AND-Verknüpfung
 - Ungleichheit 65
 - Ungleichheit, Operator 466
 - Ungleichung 58
 - Lösung* 59
 - Unicode 844, 1098
 - BMP-Teilmenge* 844
 - Tabelle wichtiger Teilzeichensätze* 844

- Unicode (Forts.)
 - Unterstützung durch Software* 845
 - UTF-8-Codierung* 844
- Unicode, Zeichensatz 55
- Unified Modeling Language
 - UML
- Unified Process 650
 - Aktivitäten* 650
 - Anwendungsfall* 650
 - Artefakt* 650
 - Phasen* 651
 - Rollen* 650
 - Vorgehen* 650
- UNIQUE, SQL-Schlüsselwort 699
- Unit-Test 648
 - Mock-Objekt* 969
 - PHP* 966
- Unit-Tests 668
 - grüner Balken* 670
 - JUnit-Framework* 668
 - Motivation* 668
 - roter Balken* 670
- Universal Disk Format → UDF
- University of California, Berkeley
 - 284, 410, 571
- Unix 49, 284
 - \$0, Systemvariable* 372
 - .bashrc, Konfigurationsdatei* 373
 - /etc/passwd-Datei* 369
 - /etc/profile, Konfigurationsdatei* 373
 - /etc/shadow, Datei* 370
 - alias-Befehl* 407
 - als Server einrichten* 430
 - Arbeitsverzeichnis anzeigen* 387
 - auf NFS-Freigabe zugreifen* 431
 - auf Windows-Server zugreifen* 434
 - bash* 372
 - Befehle regelmäßig ausführen* 408
 - Benutzerrechte* 308
 - Berkeley System Distribution (BSD)* 284
 - Bourne-Shell* 372
 - BSD* 284
 - BSD-Startskript* 401
 - bunzip2-Befehl* 862
 - Unix (Forts.)
 - bzip2-Befehl* 862
 - bzip2-Komprimierung* 402
 - cat-Befehl* 389
 - cd-Befehl* 387
 - chgrp-Befehl* 388
 - Child-Prozess* 299
 - chmod-Befehl* 387
 - chown-Befehl* 388
 - cp-Befehl* 385
 - Cronjob* 408
 - C-Shell* 372
 - CUPS, Drucksystem* 431
 - Daemon* 396
 - Darwin* 285
 - date-Befehl* 393
 - Datei kopieren* 385
 - Datei löschen* 386
 - Datei umbenennen* 385
 - Datei verschieben* 385
 - Dateibefehle* 384
 - Dateibesitzer wechseln* 388
 - Dateiendung* 385
 - Dateigruppe wechseln* 388
 - Dateiname* 307, 384
 - Dateinamen-Platzhalter* 384
 - Datensysteme* 305
 - Datum und Uhrzeit ändern* 393
 - Datum und Uhrzeit formatieren* 394
 - diff-Befehl* 391
 - du-Befehl* 393
 - echo-Befehl* 388
 - Emacs, Texteditor* 417
 - Escape-Sequenz* 391
 - exit-Befehl* 377
 - fg-Befehl* 373
 - finger, Dienstprogramm* 369
 - fork(), Systemaufruf* 299
 - fsck-Befehl* 393
 - Gerätedatei* 305–306
 - GNOME* 297, 424, 426
 - grafische Benutzeroberfläche* 422
 - grep-Befehl* 390
 - groupadd-Befehl* 395
 - Group-ID* 300, 369
 - gunzip-Befehl* 861
 - gzip-Befehl* 861
 - gzip-Komprimierung* 401
 - Unix (Forts.)
 - Hard Link* 308
 - head-Befehl* 389
 - HIER-Dokument* 381
 - Home-Verzeichnis* 306
 - HP UX* 285
 - IBM AIX* 285
 - ifconfig-Befehl* 428
 - init-Prozess* 299
 - inode* 307
 - Installation von Software* 401
 - IP-Adresse zuweisen* 429
 - KDE* 297, 424
 - kill(), Systemaufruf* 299
 - kill-Befehl* 396
 - Korn Shell* 372
 - less-Befehl* 389
 - Link (Dateisystem)* 308
 - Linux* 285, 288
 - logger-Befehl* 399
 - Login* 367
 - ls-Befehl* 386
 - Mac OS X* 285, 440
 - mail-Befehl* 409
 - make-Befehl* 402
 - man-Befehl* 297
 - Minix* 288
 - mkdir-Befehl* 387
 - mkfs-Befehl* 393
 - Modularität* 284
 - more-Befehl* 389
 - mount-Befehl* 392
 - mv-Befehl* 385
 - MySQL-Installation* 690
 - Netzwerkconfiguration* 428
 - Neustart* 397
 - NFS* 430
 - Pager* 389
 - Parent-Prozess* 299
 - passwd-Befehl* 395
 - Passwort ändern* 395
 - patch-Befehl* 392
 - PATH, Umgebungsvariable* 374
 - pause(), Systemaufruf* 299
 - Pfadangabe* 306
 - Pipe* 382
 - POSIX-Standard* 285
 - Programm automatisch starten* 399
 - Prozessmodell* 298

Unix (Forts.)

Prozessverwaltung 395
ps-Befehl 300, 395
pstree-Befehl 396
pwd-Befehl 387
regulären Ausdruck suchen 390
rm-Befehl 386
rmdir-Befehl 387
root, Benutzer 300, 369
route-Befehl 429
Runlevel 399
Samba-Server 432
Shell 297, 367
Shell-Ausgabeumleitung 381
Shell-Eingabeumleitung 381
Shell-Eingabevervollständigung 375
Shell-History 376
Shell-Skript 403
shutdown-Befehl 397
Software installieren 401
Stand-alone Shell 372
Standardrouter einrichten 429
startx-Befehl 423
su-Befehl 377
Sun Solaris 285
Swap-Partition 303
Symbolic Link 308
Syslog 396–397
System herunterfahren 397
System V 285
Systemaufruf 295
Systemprogramme 383
tail-Befehl 389
tar-Befehl 402, 860
tar-Datei 401
Textbefehle 388
Textdatei anzeigen 389
Textdateien vergleichen 391
Texteditoren 410
Umgebung 372
Umgebungsvariable setzen 375
umount-Befehl 393
unalias-Befehl 408
unzip-Befehl 862
useradd-Befehl 394
userdel-Befehl 395
User-ID 300, 369
Verwaltungsbefehle 392
Verzeichnis anlegen 387

Unix (Forts.)

Verzeichnis löschen 387
Verzeichnis wechseln 387
Verzeichnisbaum 306
Verzeichnisbefehle 384
Verzeichnisinhalt anzeigen 386
vi, Editor 410
virtuelles Terminal 370
wc-Befehl 392
Window-Manager 297
Wörter zählen 392
X Window 297, 422
zip-Befehl 862
Zugriffsrechte 308
 Unix System V Init 399
 Unix-Benutzerkonto
 in LDAP abbilden 775
unless, Perl-Fallunterscheidung 507
unset(), PHP-Funktion 936
unshift, Perl-Anweisung 502
unsigned, C-Datentyp 463
 UNSIGNED, SQL-Feldoption 699
 Unterprogramm
 Aufruf durch CPU 123
unzip, Unix-Befehl 862
update(), AWT-Methode 618
 UPDATE, SQL-Abfrage 704
 Urgent Data (TCP) 250
 URL 270, 1099
 Schema 272
 USB 140, 1099
 USB-Stick 150
Use Case → Anwendungsfall
use, Perl-Direktive 500
Usenet 267, 1099
User Level Security, Samba 434
User Level Security, Windows-
 Freigabeart 361
useradd, Unix-Befehl 394
userdel, Unix-Befehl 395
 User-ID
 von Prozessen 300
 User-ID (Unix) 369
userPasssword, LDAP-Attribut 776
Users, Mac OS X-Verzeichnis 306
usr, Unix-Verzeichnis 306
 UTF-8 1099
 UTF-8, Unicode-Codierung 844

V

V.24-Schnittstelle 141
var, Unix-Verzeichnis 306
 VARCHAR, SQL-Datentyp 698
 Variable
 Array, Perl 501
 automatische (lokale) 463
 Datentypen, C 462
 Deklaration, C 459, 461–462
 Deklaration, Java 488
 deklarieren, Perl 501
 Existenz testen, Perl 507
 globale 463
 Gültigkeitsbereich, C 463
 Hash, Perl 501
 in C 462
 in der PowerShell 342
 in Programmiersprachen 66
 in Shell-Skripten 405
 lokale 463
 mathematische 59
 Perl 500
 PHP 925
 Schleifenzähler 472
 Skalar, Perl 499, 501
 statische 463
 Substitution, Perl 500
 Substitution, Shell-Skript 406
 Typecasting 465
 Variablensubstitution
 in Perl-HIER-Dokumenten 506
 in Shell-Skripten 406
 Perl 500
 VAX, Minicomputer-Serie von
 DEC 40
 Vektorgrafik 53
 Vektorrechnung 91
 Vereinigungsmenge 70
 Vererbung 50, 493, 1099
 Interface, Java 494
 PHP 943
 Vergleichsoperation 65
 beim virtuellen Prozessor 103
 Umkehrung 65
 Vergleichsoperator 466
 Vergleichsoperatoren
 SQL 701
 Verhaltensmuster 661

- Verknüpfung
 - logische* 60
- Verschieben
 - Datei, Mac OS X* 444
 - Datei, Unix* 385
 - Datei, unter Windows* 331
- Verschlüsselung 1079
 - asymmetrische* 1080
 - Einweg-* 1081
 - symmetrische* 1080
- Verteilte Anwendung 195, 598
 - Aufgaben* 598
 - Backend* 195
 - Frameworks im Vergleich* 598
 - Frontend* 194
 - Java EE-Plattform* 599
- vertical-align, CSS 914
- Verzeichnis
 - anlegen, Unix* 387
 - anlegen, Windows* 337
 - Arbeitsverzeichnis anzeigen, Unix* 387
 - Inhalt anzeigen, Unix* 386
 - löschen, Unix* 387
 - löschen, Windows* 337
 - verarbeiten, Perl* 510
 - wechseln, Unix* 387
 - wechseln, Windows* 336
- Verzeichnisdienst 1099
 - Active Directory* 362
 - LDAP* 773
- vi, Texteditor 410
 - Befehlsmodus* 410
 - Dateibefehle* 413
 - Editiermodus* 410
 - Navigation* 411
 - Suchfunktionen* 412
 - Text kopieren* 412
 - Text löschen* 412
- Video
 - Dateiformate* 858
- Video-Dateiformat 858
 - AVI* 858
 - MPEG* 859
 - QuickTime* 858
- Video-DVD
 - auf dem PC abspielen* 155
 - Region-Code* 155
- View
 - Ruby on Rails* 980
- Vim, Texteditor 410
- Virtual Box 313
- Virtual PC 312
- Virtual Private Network → VPN
- VirtualHost, Apache-Direktive 754
- Virtualisierung 311
 - Microsoft Virtual PC* 312
 - Parallels Desktop* 313
 - Virtual Box* 313
 - VMware* 312
 - VMware Workstation* 313
 - Xen* 312
- Virtuelle Maschine
 - Java* 483
- Virtueller Host (Apache) 751, 754–755
- Virtueller Prozessor 100
 - Arbeitsspeicher* 100
 - Aufbau* 100
 - Befehle* 102
 - Beispielprogramme* 105
 - Flag* 103
 - Rechenbefehl* 102
 - Register* 100
 - Sprungbefehl* 103
 - Stack* 104
 - Vergleichsoperation* 103
- Virtueller Speicher 118, 302
- Virus 47, 1058
 - Antivirenprogramm* 1061
 - Aufbau* 1059
 - Bootsektor-* 1059
 - Dateivirus* 1060
 - Makrovirus* 1060
 - polymorpher* 1060
 - Schutzmaßnahmen* 1061
 - Stealth-* 1060
- Visitor, Entwurfsmuster 664
- Vista → Windows Vista
- Vlissides, John 661
- VLSM 1099
- VLSM, variables IP-Teilnetz 229
- VMS, Betriebssystem 287
- VMware 312
- VMware Workstation 313
- void, Datentyp
 - Zeiger auf* 551
- void, Funktionsdatentyp, C 474
- Volladdierer (Schaltung) 87
- Volltextdatenbank 677
- von Neumann, John 37, 111
- Von-Neumann-Rechner 100, 111, 1099
- VPN 1082

W

- w, Perl-Modifikator 499
- W3C, DOM 834
- Wahre Aussage 58
- Wahrheitswertetabelle 62
 - NAND-Verknüpfung* 85
 - XOR-Verknüpfung* 64
- Wall, Larry 498
- WAMP-System 924
- WAN 187, 1099
 - technische Lösungen* 187
- Warnock, John 41
- Warnung
 - in Perl aktivieren* 499
- Wasserfallmodell 639
- WAV, Audio-Dateiformat 857
- Wavetable-Synthese (MIDI) 167
- wc, Unix-Befehl 392
- wchar_t, C-Datentyp 463
- Wearable Computer 45
- Web 2.0 993
- Web Fonts (CSS3) 920
- Webanwendung
 - Ajax* 1035
 - CGI* 986
 - Grundprinzip* 923
 - Sicherheitsprobleme* 1072
- Webbrowser
 - KDE Konqueror* 424
- Weblog 988
 - Blogging-Communitys* 988
 - Software für* 988
- WebLogic, Java Application Server 599
- Web-safe Colors 912
- Webserver 193
 - Mongrel* 985
 - Programmierung* 923
- Webserver (Apache) 737
- Webservice 1099
- Webservices 605
 - Beschreibungsdatei* 606
 - Client* 607

- Webservices (Forts.)
 - Definition* 606
 - registrieren* 606
 - SOAP* 606
 - Sprachen* 606
 - UDDI* 606
 - WSDL* 606
- Websichere Farben 912
- Website
 - robots.txt-Datei* 907
- Wechseldatenträger 150
 - CD-ROM* 151
 - Diskettenlaufwerk* 150
 - DVD* 154
 - Jaz-Laufwerk* 150
 - LS-120* 150
 - Zip-Laufwerk* 150
- Weizenbaum, Joseph 96
- Well-known Port 250
- Well-known Ports
 - Tabelle einiger wichtiger* 252
- WEP (Wired Equivalent Privacy) 208
- Wertdiskret 54
- Wertetabelle
 - NOR-Verknüpfung* 85
- Wertrückgabe
 - Perl-Subroutine* 509
- Wertzuweisung
 - C* 461
 - Operator* 467
- What You See Is What You Get
 - WYSIWYG
- WHERE-Klausel, SQL 700
- while()-Schleife 472
- while-Befehl
 - in Shell-Skripten* 405
- White Book 1099
- White Book (Video-CD) 152
- Whitespace
 - in C-Programmen* 462
 - in RegExp* 558
- Wide Area Network 187
- Widerspruch, logischer 93
- Widget 608
- WiFi Protected Access 208
- WiFi → Wireless LAN
- Wiki 988
- Wikipedia 988
- Wildcard
 - in Unix-Dateinamen* 384
 - Windows-Dateiname* 335
- Win32 1099
 - Umgang mit 16-Bit-Anwendungen* 321
- Win32 API 296, 1099
- WindowListener, AWT-Klasse 622
- Window-Manager 423
 - CDE* 423
 - fwmm2* 423
- Windows 287, 319
 - 16-Bit-Versionen* 319
 - 2000* 288, 320
 - 2000 Server* 320
 - 7* 288
 - 95* 288, 320
 - 98* 288, 320
 - Active Directory* 362
 - als Server einrichten* 360
 - Anzeige-Konfiguration* 349
 - Apache-Installation* 743
 - attrib-Befehl* 337
 - auf Netzwerkrechner zugreifen* 361
 - Ausgabeumleitung* 335
 - Batch-Datei* 337
 - Benutzer-An- und Abmeldung* 329
 - Benutzerverwaltung* 350
 - cd-Befehl* 336
 - copy-Befehl* 337
 - CreateProcess(), Systemaufruf* 300
 - CygWin* 457
 - Datei kopieren* 331
 - Datei löschen, Konsole* 337
 - Datei umbenennen* 337
 - Datei verschieben* 331, 337
 - Dateiattribut* 310
 - Dateiattribute ändern* 337
 - Dateiname* 310
 - Dateinamen-Platzhalter* 335
 - Dateisysteme* 309, 323
 - Datenträgerverwaltung* 352
 - del-Befehl* 337
 - Desktopkonfiguration* 349
 - Dienst* 352
 - dir-Befehl* 336
- Windows (Forts.)
 - Drucker freigeben* 361
 - durch Unix-Server bedienen* 432
 - Eingabeaufforderung* 336
 - Eingabeumleitung* 335
 - Entwicklung* 319
 - Ereignisanzeige* 352
 - Explorer* 330
 - Fenster* 327
 - Freigabeart* 361
 - Gruppenrichtlinienobjekt* 353
 - Hardwarekonfiguration* 349, 351
 - help-Befehl* 297
 - herunterfahren* 329
 - Home-Verzeichnis* 310
 - IP-Adresse zuweisen* 356
 - Konsole* 333
 - Konsole starten* 336
 - Konsolenbefehl* 336
 - Laufwerksbuchstaben* 309
 - MDI-Anwendung* 327
 - Me* 288, 320
 - Microsoft Management Console (MMC)* 352
 - mkdir-Befehl* 337
 - MMC (Microsoft Management Console)* 352
 - move-Befehl* 337
 - MySQL-Installation* 691
 - Netzwerkdrucker nutzen* 361
 - Netzwerkkonfiguration* 355
 - Netzwerktreiber* 355
 - Netzwerkverbindung einrichten* 355
 - neu starten* 329
 - NT* 287, 320
 - NT Server* 320
 - NT-Familie* 320
 - Oberfläche* 325
 - Ordner freigeben* 360
 - Ordneransicht* 332
 - path, Umgebungsvariable* 336
 - Pfadangabe* 309
 - Pipe* 335
 - Prompt* 336
 - Prozessmodell* 300
 - regedit, Dienstprogramm* 354
 - Registry* 353
 - rename-Befehl* 337

- Windows (Forts.)
 - rmdir-Befehl* 337
 - SDI-Anwendung* 327
 - Server 2003* 320
 - Server 2008* 320
 - Server-Pakete* 363
 - Serversystem* 361
 - Server-Versionen* 320
 - Share Level Security* 361
 - Software installieren* 350
 - Startmenü* 328
 - Systemkonfiguration* 348
 - Systemsteuerung* 348
 - Systray* 330
 - Taskleiste* 328
 - Textdatei anzeigen, Konsole* 337
 - type-Befehl* 337
 - Umgebungsvariable setzen* 336
 - User Level Security* 361
 - Versionsübersicht* 320
 - Verzeichnis anlegen* 337
 - Verzeichnis löschen* 337
 - Verzeichnis wechseln* 336
 - Verzeichnisinhalt, Konsole* 336
 - Vista* 288
 - Win32 API* 296
 - XP* 288
 - Windows 2000 288, 320
 - Server* 320
 - Shell* 334
 - Windows 2000 Server 362
 - Windows 7 288, 320
 - Windows 95 288, 320
 - Oberfläche* 325
 - Windows 98 288, 320
 - Oberfläche* 325
 - Windows Me 288, 320
 - Windows NT 287, 320
 - Server* 320
 - Shell* 334
 - Windows Phone 45
 - Windows PowerShell → Power-Shell
 - Windows Server 2003 320, 362
 - als DHCP-Server* 362
 - als DNS-Server* 362
 - als Webserver* 362
 - Server-Rollen* 362
 - Windows Server 2008 320
 - Windows Server 2008 R2
 - Varianten* 363
 - Windows Vista 288, 320
 - Windows XP 288, 320
 - Oberfläche* 325
 - Shell* 334
 - Windows-Eingabeaufforderung 333
 - Eingabevervollständigung* 334
 - History* 334
 - QuickEdit-Modus* 335
 - Tastenkürzel* 334
 - WinGate, Backdoor 1064
 - Winkelgeschwindigkeit, konstante 149
 - WinTel-PC 42
 - Wired Equivalent Privacy 208
 - Wireless LAN 204–205
 - Access Point* 207
 - Basic Service Set (BSS)* 206
 - CSMA/CA-Verfahren* 206
 - Extended Service Set (ESS)* 207
 - Frequency Hopping* 205
 - Frequenzbereiche* 205
 - Hardware* 206
 - Sicherheit* 207
 - Übertragungstechnik* 205
 - WiFi Protected Access (WPA)* 208
 - Wired Equivalent Privacy* 208
 - Wirth, Niklaus 48
 - Wirtschaftsinformatik 32
 - WLAN → Wireless LAN
 - Wohlgeformtheit 1099
 - Wohlgeformtheit, XML-Dokument 801
 - Word, 16 Bit 79
 - WordPress, Blog-Tool 988
 - World Wide Web 269
 - Geschichte* 174
 - HTTP-Protokoll* 269
 - Komponenten* 174
 - URL* 270
 - Worst Case (Komplexität) 94
 - Wortbreite 40, 78, 1100
 - der CPU* 118
 - verschiedener CPUs* 119
 - Wörter zählen (Textdateien, Unix) 392
 - Wozniak, Steve 41, 286
 - WPA 208
 - Write-Host, PowerShell-Cmdlet 342
 - WSDL 1100
 - WSDL, Webservices 606
 - Wurm 1060
 - Schutzmaßnahmen* 1061
 - WYSIWYG 41, 1100
-
- ## X
- X Window 297, 422, 1100
 - Konfiguration* 422
 - SaX, openSUSE-X-Konfigurationsprogramm* 423
 - starten* 423
 - Xconfigurator, Fedora-Konfigurationsprogramm* 423
 - X11R7 422
 - x86-Assembler, Beispiele 124
 - Xalan, XSLT-Prozessor 819
 - XAMPP 727
 - Xconfigurator (X-Server-Konfigurationsprogramm) 423
 - Xen 312
 - Xerces, XML-Parser 825
 - XEROX PARC 287
 - Xerox PARC 40
 - XFree86 422
 - XHTML 866, 1100
 - Besonderheiten* 869
 - XML 791, 1100
 - Ajax-Datenaustausch durch* 1042, 1044–1045
 - Attribut* 793, 797
 - Attribut, falsche Verwendung* 798
 - Attribut, Verwendung* 798
 - Attributdeklaration (DTD)* 811
 - Beispieldokument* 794, 826
 - CDATA-Block* 799–800
 - CSS verwenden* 819
 - Dateiendung* 793
 - Datenbank* 677
 - DocBook* 792
 - DOCTYPE-Angabe* 804
 - Dokument parsen, SAX* 827
 - Dokumenteingabe* 792
 - Dokumentfragment* 802
 - Dokumentstruktur* 793

- XML (Forts.)
- DTD 804
 - DTD definieren 805
 - DTD, Attribut deklarieren 811
 - DTD, Element deklarieren 807
 - DTD, Entity deklarieren 814
 - Editor 792
 - EJB-Deployment-Deskriptor 603
 - Element (Tag) 795
 - Elementdeklaration (DTD) 807
 - Entity deklarieren (DTD) 814
 - Entity-Referenz 799, 814
 - Entity-Referenz, numerische 799
 - Entity-Referenzen vermeiden 800
 - Hierarchie 796
 - in Datenbank schreiben, Beispiel-programm 834
 - Kommentar 801
 - leeres Tag 797
 - leeres Tag, Kurzfassung 797
 - mehrere Namensräume 815
 - MIME-Type 793
 - Namensraum 815
 - Namensräume, mehrere im Dokument 815
 - Parser 825
 - Parser, Xerces 825
 - PCDATA 799
 - Processing Instruction (PI) 795
 - Programmierung 825
 - Programmierung, SAX 827
 - PUBLIC-ID 804
 - SAX 827
 - Schema 816
 - Schema-Beispiel 816
 - SGML-Erbe 791
 - Sonderzeichen 799
 - spezielle Editoren 792
 - Standalone-Dokument 795
 - Standard-Namensraum 815
 - Steueranweisung 795
 - SVG 792
 - SYSTEM-ID 804
 - Tag 793
 - Tag-Name 796
 - Tag-Verschachtelung 795
 - Universalität 792
 - verschachteltes Tag 795, 803
 - Verschachtelungsfehler 803
- XML (Forts.)
- Webservices 605
 - wichtige Dokumentformate 792
 - Wohlgeformtheit 801
 - Wurzelement 802
 - Xerces, Parser 825
 - XHTML 792
 - xmlns-Angabe 816
 - xml-Steueranweisung 795
 - XPath 1100
 - XSL 819
 - XSL-FO 819
 - XSLT 819
 - Zeichensatz 795
- XML Schema 816, 1100
- Attribut deklarieren 817
 - Element deklarieren 817
 - verschachteltes Element 817
- XML-Datenbank 677
- XML-Editor 792
- XMLHttpRequest 1036
- XMLReader, Java-Interface 827
- XOR-Schaltung 86
- XOR-Verknüpfung 64
- XP → Extreme Programming
- XPath 1100
- Überblick 824
- X-Server 297, 422
- Konfiguration 422
 - über das Netzwerk betreiben 258
- XSL 819
- XSL Formatting Objects → XSL-FO
- XSL-FO 819, 1100
- XSLT 819, 1100
- div, Operator 825
 - Funktion 824
 - position(), Funktion 825
 - Prozessor 819
 - round(), Funktion 825
 - text(), Funktion 825
 - wichtige Elemente 822
 - Wurzelement 822
 - Xalan, Prozessor 819
 - XPath-Ausdruck 824
 - xsl:attribute, Tag 823
 - xsl:choose, Tag 824
 - xsl:copy-of, Tag 823
 - xsl:for-each, Tag 823
 - xsl:if, Tag 823
 - xsl:otherwise, Tag 824
- XSLT (Forts.)
- xsl:stylesheet, Tag 822
 - xsl:template, Tag 822
 - xsl:value-of, Tag 823
 - xsl:variable, Tag 824
 - xsl:when, Tag 824
- XSLT-Prozessor 819
- XSS → Cross-Site-Scripting
-
- ## Y
-
- YAML, Dateiformat 975
- YEAR, SQL-Datentyp 697
- Yellow Book (CD-ROM) 152
-
- ## Z
-
- Z3, erster funktionierender Computer 36
- Z80, Prozessor 119
- Zahl
- Festkomma- 82
 - Fließkomma- 82
 - ganze 68
 - imaginäre 69
 - komplexe 69
 - natürliche 68
 - rationale 68
 - reelle 69
 - römische 73
- Zahlenmengen
- ganze Zahlen 68
 - imaginäre Zahlen 69
 - komplexe Zahlen 69
 - natürliche Zahlen 68
 - rationale Zahlen 68
 - reelle Zahlen 69
- Zahlensysteme 72
- Dezimalsystem 73
 - Dualsystem 73
 - Hexadezimalsystem 74
 - Oktalsystem 74
 - römische Zahlen 73
 - Schreibweise 77
 - Umrechnung 75
- Zeichen
- alternatives in RegExp 557
 - aus String lesen, Java 489
 - ausschließen in RegExp 557
 - beliebig viele, in RegExp 557

- Zeichen (Forts.)
 - Darstellung, C* 463
 - eines oder mehr, in RegExp* 557
 - Eingabe, C* 480
 - genaue Anzahl, in RegExp* 558
 - Gruppe in RegExp* 557
 - optionales, in RegExp* 557
 - Wortbestandteil in RegExp* 558
- Zeichengerät (Char Device) 294
- Zeichenkette → String
- Zeichen-Literal 464
- Zeichensatz 55, 840
 - ANSI* 842
 - ASCII* 55
 - Codepage* 843
 - Eingabe chinesischer Zeichen* 847
 - in HTML angeben* 870
 - Unicode* 55, 844
 - XML* 795
- Zeichenvorrat
 - der Turing-Maschine* 97
- Zeiger 475, 1100
 - auf beliebigen Datentyp* 551
 - auf nichts (NULL)* 481
 - fehlender in Java* 489
 - für Call by Reference* 476
- Zeilenfrequenz (Monitor) 162
- Zeilenumbruch
 - auf verschiedenen Plattformen* 838
 - HTML* 870
 - konvertieren* 839
 - Perl-Programme* 840
- Zeilenumbrüche
 - entfernen, Perl* 500
- Zeitdiskret 54
- Zeitmanagement 641
 - Netzplan* 642
- Zentraleinheit 113
 - alte Bedeutung* 115
- ZEROFILL, SQL-Feldoption 699
- Ziffer
 - in RegExp* 558
- z-index, CSS-Angabe 917
- zip, Unix-Befehl 862
- ZIP-Datei 859
- Zip-Laufwerk 150
- Zugriffsrechte 308
 - ändern, Unix* 387
- Zuordnungseinheit (Datei-system) 304
- Zuse, Konrad 36
- Zustand
 - der Turing-Maschine* 97
- Zweierkomplement 80
- ZX Spectrum 286
- ZX81 42, 286