

Literatur

[Achilles 2005] Achilles, A.: „Betriebssysteme: Eine kompakte Einführung mit Linux“, Springer, Berlin, 2005

Hinweis: Eine Einführung, die sich besonders an Leser mit Interesse an der Arbeitsweise des Linux-Kerns wendet.

[Andrews et al. 1983] Andrews, G. R., Schneider, F. B.: „Concepts and Notations of Concurrent Programming“, Computing Surveys, Vol. 15, No. 1, 1983

Hinweis: Dies ist ein klassischer Übersichtsartikel über die unterschiedlichen Mechanismen der Programmierung von Nebenläufigkeit.

[Boger 1999] Boger, M.: „Java in verteilten Systemen“, dpunkt Verlag, 1999

Hinweis: Das Buch gibt einen Überblick über Grundmechanismen und Frameworks zur Programmierung verteilter Anwendungen in Java.

[Brause 2003] Brause, R.: „Betriebssysteme, Grundlagen und Konzepte“, Springer Verlag, 2003

[Brinch-Hansen 1974] Brinch-Hansen, P.: „The programming language Concurrent Pascal“, IEEE Trans. on Software Engineering, 1, 2 p. 149-557, 1974

Hinweis: Dies kann neben [Hoare 1974] als eine der Grundlagen des Monitor-konzepts angesehen werden.

[Brown 1998] Brown, C.: „Programmieren verteilter UNIX-Anwendungen“, Prentice Hall, 1998

Hinweis: Auf dieses Buch wird sehr häufig Bezug genommen. Es enthält ausführliche und gut dokumentierte Programme zur Interprozess-Kommunikation, und zur Client-Server-Programmierung mit Sockets und Remote Procedure Calls.

[Deitel et al. 2004] Deitel, H. M., Deitel P. J., Choffnes, D. R.: „Operating Systems“, Prentice Hall International, 2004

[Glatz, 2010] Glatz, E.: „Betriebssysteme: Grundlagen, Konzepte, Systemprogrammierung“, dpunkt-Verlag, 2010

Hinweis: Ein sehr gründliches Buch.

[Gulbins et al. 1996] Gulbins, J., Obermayr, K.: „UNIX. System V.4: Begriffe, Konzepte, Kommandos, Schnittstellen“, Springer Verlag, 1996

Hinweis: Ein sehr empfehlenswertes Buch zu UNIX.

- [Haviland et al. 1998] Haviland, H, Gray, D., Salama, B.: „UNIX System Programming“, Addison-Wesley, 1998
Hinweis: Enthält Systemprogramme in C für UNIX, insbesondere zur Signalbehandlung und Interprozess-Kommunikation.
- [Herold et al. 2010] Herold, H., Arndt, J.: „C-Programmierung: Unter Linux, UNIX und Windows“, Millin, März 2010
- [Hoare 1974] Hoare, C. A. R.: „Monitors: An operating systems structuring concept“, Comm. of the ACM, v.17, n.10, p.549-557, 1974
Hinweis: Dies ist neben [Brinch-Hansen 1974] eine der Grundlagen des Monitorkonzepts.
- [Kernighan et al. 1986] Kernighan, B. W., Pilzke, R.: „Der UNIX-Werkzeugkasten: Programmieren mit UNIX“, Carl Hanser Verlag, 1986
Hinweis: Ein exzellentes Buch, insbesondere in Bezug auf die Shell- und Script-Programmierung.
- [Mandl 2009] Mandl, P.: „Grundkurs Betriebssysteme: Architekturen, Betriebsmittelverwaltung, Synchronisation, Prozesskommunikation“, Vieweg+Teubner, 2009
- [Nehmer et al. 2001] Nehmer, J., Sturm, P.: „Systemsoftware: Grundlagen moderner Betriebssysteme“, dpunkt Verlag, 2001
- [RRZN 2010] Rechenzentrum der Universität Hannover: Unix-Benutzung / Eine Einführung, Leibnitz-Universität Hannover, 2010
Hinweis: Eine preiswerte, gerade für Studienanfänger geeignete umfassende Einführung, die über die Hochschulrechenzentren zu beziehen ist.
- [Silberschatz et al. 1998] Silberschatz, A., Galvin, P.: „Operating System Concepts“, Addison-Wesley, 1998
- [Stallings 2003] Stallings, W.: „Betriebssysteme – Prinzipien und Umsetzung“, Pearson Studium, 2003
- [Tanenbaum 2009] Tanenbaum, A. S.: „Moderne Betriebssysteme“, Pearson Studium, 2009
Hinweis: Die Bücher von Tanenbaum gehören zu den Standardwerken der Betriebssystemliteratur und sind unbedingt zu empfehlen. Insbesondere sind daraus Konzepte zum Prozessbegriff in das vorliegende Buch eingeflossen.
- [Vogt, 2001] Vogt, C.: „Betriebssysteme“, Spektrum Akademischer Verlag, 2001
Hinweis: Ein ausgezeichnetes Lehrbuch zu den Konzepten moderner Betriebssysteme mit vielen Beispielen und Übungen.

Internet-Links

Linux-Distributionen

| | |
|----------------|---|
| Slackware: | http://www.slackware.com |
| Debian: | http://www.debian.org |
| SuSE: | http://www.suse.de |
| Redhat: | http://www.redhat.de |
| MandrakeLinux: | http://www1.mandrivalinux.com/de/ |
| Knoppix: | http://www.knopper.net |

Sonstige UNIX-Distributionen

Minix: A. Tanenbaum hat Minix entwickelt, um in seiner Vorlesung ein freies und im Quelltext vorliegendes UNIX zur Verfügung zu haben. Es zeichnet sich durch einen leicht verständlichen und gut strukturierten Systemkern aus.

<http://www.cs.vu.nl/~ast/minix.html>

FreeBSD: FreeBSD ist aus der professionellen Entwicklungslinie der BSD-Linie der UNIX-Betriebssysteme hervorgegangen.

<http://www.freebsd.org>

Entwicklungsumgebungen

Java: Die Java-Entwicklungsumgebung ist Voraussetzung für jede Java-Entwicklung.

<http://www.javasoft.com>

<http://www.mingw.org/>

Eclipse: Eclipse ist hervorgegangen aus der Entwicklung der kommerziellen Webplattform Websphere. In der ursprünglichen Form war Eclipse eine freie, erweiterbare Entwicklungsumgebung für Java- und Webanwendungen. Inzwischen ist es gleichzeitig eine generelle Entwicklungsplattform und Framework für Rich-Client-Anwendungen.

<http://www.eclipse.org>

Diskussionsforen und Informationen zu UNIX-Systemen

Prof. J. Plate, Betriebssystem UNIX/Linux

<http://www.netzmafia.de/skripten/unix/>

<http://www.linux.de>

<http://www.linux.com/>

Thread-Programmierung

Java-Tutorial: <http://java.sun.com/docs/books/tutorial/essential/threads/>

Doug Lea (Concurrent Programming in Java): <http://gee.cs.oswego.edu/dl/cpi/index.html>

Installationshinweise für Linux, Eclipse und den C-Compiler

Linux ist heute in hunderten, meist freien Distributionen verfügbar. Eine ziemlich umfassende Übersicht findet sich bei www.linux.org. Einige Distributionen sind auf besondere Bedürfnisse, wie spezielle Hardware, Sicherheits- oder Echtzeitanforderungen usw. zugeschnitten. Ein weiteres Unterscheidungsmerkmal ist die Sprachbindung des Systems. Grundsätzlich bauen alle Distributionen mehr oder weniger auf denselben Grundkomponenten auf. Linux-Distributionen enthalten neben dem eigentlichen Betriebssystem fast immer auch eine Vielzahl von Anwendungen, die fast alle aus dem Bereich der freien Software stammen (meist Open Source), die aber in sehr vielen Fällen nicht spezifisch für Linux sind.

Für den normalen Benutzer sind in erster Linie die Mainstream-Distributionen interessant. Selbst von diesen gibt es aber bereits an die hundert. Die Links zu einigen der verbreiteteren (Slackware, Redhed, Debian und SuSe) sind im Anhang aufgelistet.

Zu beachten ist, dass die Installation eines Linux-Systems praktisch immer eine geeignete Partitionierung des Dateisystems erfordert. Dies ist aber im Allgemeinen auch nachträglich möglich, vorausgesetzt das System verfügt über wenigstens ein bis zwei GB freien Speicherplatz.

Daneben gibt es auch die Möglichkeit, auf eine Live-CD zurückzugreifen, die die Installation eines Systems auf der Festplatte vollständig überflüssig macht. Auch dafür gibt es inzwischen einige Anbieter. Es ist dabei aber unbedingt darauf zu achten, dass die Distribution über alle benötigten Werkzeuge verfügt. In der Regel ist es auch möglich, entweder aus der CD eine Festplatten-Installation zu generieren oder selbst eine eigene CD zu konfigurieren.

Unter dem Aspekt der Softwareentwicklung interessieren die folgenden Komponenten:

- Quelltexte des Betriebssystems. Diese sind nötig, wenn das System neu übersetzt werden soll, um es eventuell anzupassen.
- X11-Grafiksystem. Dies ist die Grundlage für alle grafischen Entwicklungen. Eine Distribution stellt in der Regel zusätzlich unterschiedliche Window-Systeme zur Auswahl (z.B. Gnome oder KDE). C-Compiler. Damit ist der GNU-Compiler gcc gemeint, da der Linux-Kernel nur mit diesem Compiler zu übersetzen ist. Fast jede Distribution enthält die Varianten gcc (C-Compiler) und g++ (C++-Compiler). Im Paket des C-Compilers sind in der Regel auch die wichtigsten Bibliotheken enthalten.

- Bibliothekspakete. z.B. zur Graphikprogrammierung.
- Java-Compiler und Ausführungsumgebung. Es stehen verschiedene Systeme zur Auswahl (GNU, Sun, IBM).
- Programmierumgebung. Ein Linux-System bietet mehrere Editoren und häufig auch unterschiedliche Debugger an.
- Integrierte Entwicklungsumgebung. Sowohl für C als auch für Java gibt es freie und kommerzielle integrierte Entwicklungsumgebungen. Diese sind nicht immer fester Bestandteil einer Distribution. Sie können jedoch problemlos nachinstalliert werden.

Fast alle Distributionen haben automatisierte Installationsskripte. Viele Distributionen ermöglichen darüber hinaus die Erweiterung und Aktualisierung des Systems über Online-Updates. Häufig werden sogar Zusatzpakete von Drittanbietern in einer Form angeboten, die eine automatische Installation ermöglichen. Da sich die Installationsverfahren aller Distributionen grundlegend unterscheiden, lassen sich hier aber keine allgemeinen Aussagen treffen.

Meist basieren die Erweiterungsmechanismen auf einem der verbreiteten Paketformate wie entweder den Debian-Paketen oder dem RPM-Format von Redhat. Manchmal werden allerdings nur die Archivformate `.tar`, `.zip` oder `.tar.gz` zur Verfügung gestellt. In der Regel genügt es aber auch, wenn diese Archive in einem passenden Verzeichnis entpackt werden. Hier sollten die Installationsanweisungen der Paketanbieter beachtet werden. Bei der Installation von Software muss man fast immer über Root-Rechte verfügen (bei der normalen Systembenutzung sollte dies nie der Fall sein!).

Eine der mächtigsten Entwicklungsumgebungen für Java, aufgrund seiner leichten Erweiterbarkeit aber zunehmend auch für andere Programmiersprachen und Anwendungen, ist Eclipse. Eclipse wird entwickelt von dem aus einer IBM-Initiative entstandenen Eclipse-Konsortium (www.eclipse.org). Die Installation von Eclipse erfordert noch einige Zusatzbemerkungen. Zunächst ist es so, dass in einigen Linux-Distributionen Eclipse nicht enthalten bzw. veraltet oder (was manchmal auch vorkommt) fehlerhaft ist. Eine Neuinstallation, die auch sehr einfach ist, ist daher fast immer zu empfehlen. Zusätzlich ist es in den allermeisten Fällen sehr sinnvoll, einige zusätzliche Erweiterungen (Plugins) zu installieren.

Es gibt verschiedene Eclipse-Installationen für Linux. In aller Regel wird es jedoch um eine Installation auf einem 32-Bit-Intel-System gehen. Als Nächstes muss dann die Frage geklärt werden, ob die Grafikbibliothek `motif` oder die Bibliothek `gtk` zur Grundlage genommen wird. Diese Entscheidung führt zu etwas unterschiedlichem Aussehen der Applikation, hat aber sonst keine weitere Auswirkung.

In der Regel wird das anscheinend etwas besser unterstützte `gtk` gewählt. Die Installation selbst besteht in dem Auspacken der Installation in einem passenden

Verzeichnis, z.B. unter `/usr/local`. Nach der Installation wechselt man wieder in den Benutzermodus.

Die Anwendung lässt sich dann durch `/usr/local/eclipse/eclipse` starten. Dies lässt sich vereinfachen, indem man entweder das Eclipse-Verzeichnis in den Systempfad aufnimmt, oder über die Fensterumgebung einen entsprechenden Menüpunkt oder eine passende Verknüpfung definiert. Die Eclipse-Umgebung stellt für diesen Zweck bereits ein Icon bereit.

In der Regel legt Eclipse in dem Benutzerverzeichnis ein Verzeichnis namens `workspace` an, in dem die Eclipse-Projekte gespeichert werden. Es ist aber auch möglich, entweder über den Startdialog von Eclipse oder über den Aufrufparameter `-data Verzeichnisname` ein anderes Verzeichnis zu bestimmen. Falls das Verzeichnis der virtuellen Maschine für Java nicht gefunden wird, kann dieses ebenfalls beim Start mittels der Option `-vm Java-Verzeichnis` angegeben werden.

Die erstmalige Installation von Eclipse bietet bereits eine komplette Entwicklungsumgebung für Java (einschließlich einem eigenen Java-Compiler). Für andere Aufgaben, insbesondere auch für die Entwicklung von C oder C++-Programmen, sind weitere Plugins erforderlich. Dabei sollte angemerkt werden, dass der Entwicklungsstand und die Robustheit dieser Erweiterungen oft stark hinter dem Reifegrad des Kernsystems zurücksteht. Hinzu kommt, dass naturgemäß die Unterstützung der Entwicklungsumgebung für C und C++ aufgrund der nur schwach ausgeprägten statischen Semantik von C und C++ weit hinter den Möglichkeiten von Java zurückbleibt (z.B. kaum Unterstützung für Refactoring, keine dynamische Kompilierung).

Bei der Installation von Erweiterungen ist zu entscheiden, in welcher Form diese angeboten werden. Im einfachsten Fall kann die Erweiterung durch den Eclipse Update-Manager vorgenommen werden. Dies ist z.B. bei der C/C++-Erweiterung der Fall. Hier muss nur beachtet werden, dass man dabei über die Dateirechte für das Installationsverzeichnis verfügen muss (in der Regel Root-Rechte). Nähere Angaben finden sich unter www.eclipse.org.

Einfache Plugins werden dadurch installiert, dass deren Archivdatei unter `eclipse/plugins` entpackt wird. Manche Erweiterungen, die mehrere Plugins kombinieren, verfügen zusätzlich über eine Komplettdescription (Feature), die unter `eclipse/features` abgelegt wird. Auch hier muss man sich also nach der Installationsanleitung richten. Am wichtigsten ist aber, dass man die jeweiligen Voraussetzungen genau beachtet, da viele Plugins die Existenz anderer Plugins voraussetzen.

Die Voraussetzungen beziehen sich manchmal auch auf die passende Eclipse-Version oder auf die Versionsnummer anderer Plugins. Wenn man auf ein problemloses Funktionieren aller Erweiterungen Wert legt, sollte man unbedingt nur offizielle Eclipse-Releases installieren und von den superaktuellen Zwischenreleases (Milestone-Builds u.Ä.) Abstand nehmen!

Index

- absolut 17
- Acknowledgment 215
- adaptive Quantenfestlegung 266
- Addier-Service 243
- Adressraum 2, 3
- Adressumsetzung 280
- Aging 281
- Aktives Warten 92
- Alias 165
- Aliasnamen 185
- Alterung 269
- Antwortzeit 261
- Anweisung 29
- Anwendungsmuster 107, 116
- Array 53, 226
- asynchron 194
- asynchrone Kommunikation 104
- Asynchrone Kommunikationsschemata 151
- asynchrone Signalsicherheit 160
- Asynchronität 156
- At least once 242
- at most once 242
- At most once 242
- atomar 94
- atomaren Anforderns von Schlüsseln 177
- Aufwecken 95
- Ausführungsreihenfolge 107
- Aushungern 126
- Authentifizierung 64, 199
- Authentifizierungsmechanismen 199

- Bankautomat 251
- Barriere 108
- bash 18
- Basisregister 273
- Bedienzeit 261
- Befehlsausführung 1
- Befehlsstrom 1
- Befehlszähler 1
- Benutzeridentifikation 13
- Benutzermodus 12

- Benutzerschnittstelle 13
- Benutzerverzeichnisse 17
- Bereitschaft zum Versenden von Signalen 193
- Best Fit 277
- Betriebssystem 1, 18, 23, 73, 155, 205, 206, 259, 278, 280
- Betriebssysteme 65
- Bibliotheken 16
- Bibliothekspozedur 12
- Bibliothekswerkzeuge 19
- Binärdateien 16
- binäre Semaphore 102
- Binden 224, 228
- Binder 239, 256
- Bitmaps 276
- Blockade 195
- blockierende Warteschlange 119
- blockiert 58
- blockorientiert 170
- blockorientierte Geräte 16
- Botschaftenaustausch 108
- Botschaftenaustausch, Message Passing 103
- Buddy-System 283
- builtins 38

- cachefs-Partitionen 198
- case 251
- case-Anweisung 32
- cat 60
- character special devices 170
- child process 80
- Client-Server 6
- Client-Server-Modell 220
- Clock 281
- Closed Shop 7
- communicating sequential processes 120
- completion mechanism 18
- contiguous space 194
- copy-on-write 283
- CPU-Scheduling 271

- critical section 91
- Csh 18
- CSP 120

- Dämon-Thread 88
- Datagramm-Typ 223
- Dateideskriptortabelle 187
- Dateien 19, 89
- Dateinamen 14, 89
- Dateiorganisation 182
- Dateisuffix 16
- Dateisystem 11, 65, 209, 221
- Dateisystem-Operationen 173
- Dateitypbestimmung 165
- Datenstrom 1, 44
- Datenüberlauf 215
- DCE-RPC 236
- Deadlock 106, 115, 126, 131
- Definitionsdateien 166
- Designkriterien 11
- Diagnose-Ausgabe 14
- Dienst 239
- Directories 53
- Directory 24, 26, 29
- direkte Kommunikation 103
- direkte Kommunikation, Rendezvous-Konzept 103
- dirty Bit 281
- Dispatching 260
- dup2 61
- Durchsatz 1
- dynamische Zuweisung von Prioritäten 264

- echte Parallelität 76
- Echtzeit 122
- Echtzeitanwendungen 157
- Editoren 19
- Effizienz 261
- Ein- und Auslagern 270
- emacs 19
- Ergänzungsmechanismus 18
- errno-Variable 173
- Errorcode 177, 180
- Erzeuger-Verbraucher-Problem 125, 130
- Evolutionsschema für verteilte Systeme 209

- Exactly once 242
- execlp 54
- exec-Programmfamilie 54
- Execute-Bit 23
- exit 56
- exit-Code 33
- Exokernel 5
- Exportieren 28
- Ext2-Dateisystem 184
- External Data Representation 239
- externe Fragmentierung 284

- Factories 221
- Fairness 261
- Fehlerbehandlung 179
- Fehlererkennung 213
- Fehlerindikation 152
- Fehlerzustände 177
- FIFO 281
- File Formats 166
- fileDescriptor 225
- filedescriptor 174
- Filepointer 164
- Filepointer-Struktur 175
- Filter 19
- First Fit 277
- for-Anweisung 29
- fork 54
- Fortführungssemantik 154
- FTP 219
- Funktionsadresse 153

- Games 166
- Gateway 225
- gegenseitiger Ausschluss 109
- Geldautomaten 231
- gemeinsam benötigtes Betriebsmittel 88
- Gemeinsame Ressourcen-Nutzung 208
- General Commands 166
- Geräte 170
- Gerätetreiber 22
- Geschichtete Systeme 5
- geschützter Bereich 111
- getpid 48
- getpwuid() 250
- GID 171
- grafische Benutzerschnittstelle 13

- Groß- und Kleinschreibung 16
- Gruppenzugehörigkeit 171

- Hardlinks 185
- Hardware-Lösung 92
- Hauptverzeichnis 16
- Heimatverzeichnis 13
- Here-Dokumente 30
- Hilfsprogramme 18
- Hintergrundprozess 15
- history mechanism 18
- HOME 47
- Home-Verzeichniss 17
- Hybrides Threadsystem 83

- idempotent 241
- if-Anweisung 33
- Index-Nodes 187
- indirekte Kommunikation, Mailbox-Kommunikation 103
- indirekten Kommunikation 103
- Inkonsistenz 173
- Inode-Nummer 183
- Inodes 187
- integrierte Schaltkreis 8
- interne Fragmentierung 284
- interne Struktur von Verzeichnissen 183
- Internet 196
- Interprozess-Kommunikation 10, 59, 137
- Intervall-Timer 157
- ipcs 67
- IPv4 218
- IPv6 218
- Iterativer Server 224
- Java 5, 106
- JAVA 209
- job control facilities 18

- Kanäle 48
- Kataloge 165
- Kerberos-Authentifizierung 197
- Kernel Developer 166
- Kernel Interfaces 166
- Kernmodus 12
- kill- Systemaufruf 154
- Kindprozess (Sohnprozess, child process) 80
- KISS-Prinzip 165
- Kommentar 16, 38
- Kommunikationsformen 194
- Kommunikationsprotokolle 213
- Komplexität 123
- Konfigurationsdateien 16
- Konsistenz 170
- Kontrollfluss 3, 81
- Kontrollstrukturen 40
- kooperierende Prozesse 195
- Kreisverkehr-Problem 125, 131
- kritischer Bereich 84, 91, 111

- Laufzeitstack 82
- leasing extension 201
- Least-Recently-Used 281
- Lese-, Schreib- und Ausführungserlaubnis 172
- Leser-Schreiber-Problem 125
- Library Functions 184
- Link 165
- Linux 20, 73, 281
- Listen 276
- lock 111, 112, 116
- Lock 117
- LOCK 94
- login 48

- magic number 165
- Mailbox-Kommunikation 103
- Mainframe 8
- main-thread 82
- main-Thread 87
- Management-Tool 19
- Mapping 260
- marshalling 237
- maximalen Parallelitätsgrad 127
- May be 242
- Mehrbenutzer 8, 11
- Mehrprogrammbetrieb 76
- Mehrprogrammssystem 11, 15
- Mehrprozessorsystem 44, 82
- Memory Management 279
- Message 103
- Message Queues 104, 144
- Metainformationen 175
- Metazeichen 15

- Middleware 210
- Mikroprozessor 9
- MIMD 44
- Minix 21
- Miscellaneous Information 166
- Missbrauch 173, 175
- Monitor 105, 106
- Monitorobjekt 110
- Monolithische Systeme 4
- Mounten 164
- msgget 104
- Multics 21
- Multimedia-Dateien 194
- Multithreading 82, 123
- Mutex 110, 125
- mutual exclusion 91

- N:M-Kommunikation 71
- Nachrichtenaustausch 48, 103
- Nachrichtenwarteschlangen 144
- named pipe 191
- Named Pipe 65
- nd-Robin-Scheduling 263
- netstat 227
- newline 164
- Next Fit 277
- NFS (Network File System) 196
- NFS-Protokoll 199
- nicht-deterministisch 51, 70
- NIS 199
- Non-Preemptive Scheduling 262
- Not-Frequently-Used 281
- Not-Recently-Used 281
- Open-Source 20

- optimale Strategie 268
- OSI (Open Systems Interconnection) 211
- OSI-Modell 206, 211
- OSI-Referenzmodell 206, 210
- OSI-Stack 211

- page frames 280
- Paging 280, 282
- parallele Server 226
- paralleler Server 123
- Parallelität 76
- Parallelverarbeitung 196

- Parameter für Shell-Skripte 25
- parent process 80
- parent process ID 80
- Parent-Process-ID 47
- Partitionierung 273
- Passing 103
- Passwort-Datei 13
- PATH 53
- PCB 79, 270, 272
- Peer-to-peer-System 205
- pending 157
- periodischer Signal- Interrupt 159
- Petersons Lösung 94
- Peternetz 112
- Pfade 16
- Philosophenproblem 125
- PID 47, 80
- Pipe 14, 59
- Pipe-Kommando 60
- Plattenpartitionen 170
- Pointer 66
- P-Operation 100
- Portabilität 278
- portmap 243
- positionelle Parameter 26
- POSIX 21, 109
- POSIX-Threads 84, 86
- PPID 80
- Preemptive Scheduling 262
- Priorität 14, 48, 77
- Prioritätsfeld 145
- Prioritätsklassen 265
- Prioritäts-Scheduling 264
- process control block (PCB) 79
- process identification (PID) 80
- process table 80
- Process-ID 47
- program counter 1
- Programm 48
- Programm- und eine Versionsnummer 244
- Programm-Modus 47
- Programmstack 1
- Protokollfamilie 216
- Protokollköpfe 213
- Prozess 1, 2, 15, 18, 47
- Prozess des Lesens 73

- Prozess des Notierens 73
- Prozessbegriff 75
- Prozesskontrollblock 270
- Prozessmodell 75, 259
- Prozessnummer 81
- Prozessor 1, 11, 12
- Prozessrechner 8, 21
- Prozess-Scheduling 259
- Prozess-Synchronisationsmechanismen 158
- Prozesstabelle (process table) 80
- Prozesswechsel 1, 51, 79, 263, 266, 271, 278
- Pseudoparallelität 76
- Public-Key-Kryptographie 199
- Punkt 16, 17, 27

- Quick Fit 277

- race condition 107
- Race Condition 88, 90, 97, 140
- rechenbereit 78
- rechnend 78
- referenced Bit 281
- Register 1
- Reihenfolge 215
- relativ 17
- Relokationseigenschaft 273
- remode inode 200
- remote procedure call 199
- Rendezvous-Deskriptor 225
- Rendezvous-Konzept 103, 121
- Reservieren von Speicherplatz 194
- response time 261
- Ressourcen 2, 44, 78, 79, 119
- Ressourcenverwaltung 6, 259
- R-Node 200
- root 166
- Routen 216
- RPC 199, 236
- RPC (remote procedure call) 199
- rpcgen 243
- run-to-completion 285

- Scheduling 10, 77, 83, 89, 259
- Scheduling-Konflikte 261
- Scheduling-Verfahren 262

- Schichtenmodell 210
- Schlafen 95
- Schlafen und Aufwecken 95
- schwebenden 157
- schwieriges Einlagern 272
- Script-Programm 23
- Second-Chance 281
- Seiten 280
- Seitenersetzungsalgorithmus 282
- Seitenersetzungsverfahren 281
- Seitenfehler 281
- Seitenrahmen 280
- Semaphor 98, 109, 119, 125, 126
- Semaphore 209
- semctl 139
- semget 138
- semop 138
- Sequencing 215
- Server 206
- Shared Memory 59, 66, 104, 140
- shell 9
- Shell 4, 12, 13, 53, 80
- Shell-Script 24
- Shell-Variablen 26
- shmat 68
- shmctl 68
- shmdt 68
- shmget 67
- Shortest-Job-first 285
- Signal 48
- Signalaktivierung 154
- Signal-Behandlungsroutine 150
- Signale 49, 152
- Signalhandler 153, 160
- Signalhandler-Adresse 153
- SIMD 44
- situationsangepassten Programmfortführung 154
- Sitzungen 212
- Socket 178, 180, 206, 209, 221
- Socket-Deskriptor 224
- Sockets 224, 226
- Softwareunterbrechungen 48
- Sohnprozess 80
- Speicher 1
- speicherbasierte Kommunikation 66
- Speicherhierarchie 1, 259

- Speicherplatz 167
- speicherprogrammierbarer Computer 7
- Sperre 92, 95, 111
- Sperrmechanismus 99, 117, 143
- Spezialdateien 16
- stack 210
- Stack 275
- stack pointer 1
- Standard 21
- Standardausgabe 14, 61
- Standardeingabe 14, 61
- Standardfehlerausgabe 14, 61
- Standardisierte Parameterübergabe 36
- Standards 206
- Stapelverarbeitung 8
- Startup- und Shutdownscripte 38
- Starvation 126
- sticky bit 189
- Strategiekriterien 260
- Streams 222
- Stubs 237
- Sub-Shell 24
- Suchmechanismus 20
- SUN-RPC 236, 242
- Supervisor-Modus 47
- Swapper 271
- Swapping 274
- symbolische Links 186
- synchrone Kommunikation 104
- Synchronisation 9, 93, 109, 156, 181
- Synchronisationsprimitive 52, 92, 98, 108, 109, 205
- Synchronisationsprobleme 125
- System Binaries 19
- System Calls 166
- system level thread 83
- Systemaufruf 3, 4, 12
- Systemaufrufsschnittstelle 12
- System-Definitionsdateien 166
- Systemstillstand 106

- Tag 224
- Translation Lookaside Buffer 281
- TCP/IP 10, 21
- TCP/IP (Transmission Control Protocol/Internet Protocol) 206
- TCP/IP-Protokoll 216

- TCP/IP-Stacks 217
- tcsh 18
- tee 15
- telnet 209
- Telnet 217
- Thread 1, 3, 82, 156, 226
- Threaderzeugung 84
- Threading 22
- Thread-Programmierung 107
- threadsicher 118, 120
- Threadterminierung 84
- throughput 261
- Tilde 17
- time-out 158
- Timesharing 8, 21
- Transportebene 217
- Transportsystem 212
- Trap 46
- TRAP 4
- TRAP Anweisung 47
- TRAP-Instruktion 4
- TSL-Instruktion 94
- turn-around time 261

- überladen 251
- Überladen 250
- Übertragungsfenster 215
- UDP 217
- UID 171
- Umlenken der Ein- und Ausgabe 28
- Umlenkung 14
- Umlenkungsoperatoren 28
- UNIX 9, 15, 16
- UNIX-Ordnungsstruktur 165
- UNIX-Semaphore 137
- until-Anweisung 35
- usage 32
- user level thread 83
- Useridentität 171
- Utilities 19

- Variablendeklarationen 29
- Vaterprozess 79, 143, 187
- Vaterprozess (parent process) 80
- Vaterprozesses (PPID, parent process ID) 80
- Verarbeitungsebene 212

- Verbindungsdeskriptor 225
- verbindungsorientiertes Protokoll 213
- Verbindungsorientierung 213
- Verdoppelte Pakete 215
- Verklebung 107, 134
- Verlorene Pakete 215
- verschachtelter Monitor 115
- Versetzen des Filepointers 180
- verteilte Anwendungen 235
- Verteilte Anwendungen 43
- verteilte Systeme 235
- verteilt System 205
- Verweilzeit 261
- Verweis 165
- Verzeichnisbaum 17
- Verzeichnisnamen 16
- Verzeichnisse 18, 165, 182
- virtuelle Maschine 5
- Virtueller Speicher 278
- V-Operation 100
- Vordefinierte Shell-Variablen 33

- wahlfreier Zugriff 181
- wait 52, 152

- Web-Services 206
- Wechselseitiger Ausschluss 90
- Wettlaufbedingung 107, 108, 120
- while 93, 110, 114
- while-Anweisung 35
- Wichtigkeit 264
- Workgroup Computing 221
- Working-Set 281
- Wrapper 47, 237
- WSClock 281

- X11 20
- XDR 239, 243

- zeichenorientierte Geräte 16
- zeitkritische Abläufe 90
- Zeitquantum 263
- Zugangskontrolle 13
- Zugriffsrechte 171
- Zustände 78
- Zustandslisten 80
- Zustandsübergänge 77
- Zweistufiges Scheduling 270
- Zwei-Zeiger-Clock-Algorithmus 282