

Inhaltsverzeichnis

1	Einleitung	1
2	Grundlagen der Systemprogrammierung.....	5
2.1	Betriebssysteme	5
2.1.1	Hauptaufgaben von Betriebssystemen.....	5
2.1.2	Schichtenmodelle.....	6
2.2	Systemaufrufe	7
2.2.1	Software-Interrupts	8
2.2.2	Über Calls aufgerufene Prozeduren.....	10
2.2.3	Eine Klassifizierung	11
2.3	Systemprogrammiersprachen	12
2.3.1	Systematik	12
2.3.2	Ein elementares Beispiele in mehreren Sprachen.....	14
2.4	Portabilitätsfragen	22
2.5	Zum Programmierstil im Systembereich	23
3	Ein konkretes System: UNIX	25
3.1	Zur Geschichte von UNIX.....	25
3.2	Der Aufbau von UNIX.....	27
3.3	Vergleichende Bemerkungen bzgl. anderer Betriebssysteme	31
4	Wichtige UNIX-Werkzeuge	35
4.1	UNIX-Shells.....	35
4.1.1	Überblick	35
4.1.2	Die C-Shell	36
4.1.3	Einfache Kommandos, Pipelines, Kommandosubstitution	37
4.1.4	History- und Alias-Mechanismus.....	38
4.1.5	Vordefinierte und benutzerdefinierte Shell-Variablen.....	39
4.1.6	Interne Kommandos der csh.....	40
4.1.7	Ein- und Ausgabeumlenkung.....	42
4.1.8	Programmierung der C-Shell	42
4.1.9	Steuerungsstrukturen der C-Shell.....	44
4.1.10	Beispiele für C-Shell-Skripts	47
4.2	UNIX C-Compiler und verwandte Utilities	50
4.2.1	Was braucht man zum Programmieren unter UNIX?	50
4.2.2	Standard-UNIX C-Compiler	50
4.2.3	GNU C-Compiler.....	52
4.2.4	Bibliotheksverwalter ar.....	55
4.2.5	Binder ld	56
4.3	UNIX Make	57

4.3.1 Überblick	57
4.3.2 Makefiles	58
4.3.3 Benutzung von Make	60
4.4 UNIX Lint	60
4.5 UNIX Debugger	61
4.5.1 Allgemeines	61
4.5.2 Debugger dbx und dbxtool	62
4.5.3 GNU Debugger	67
4.6 UNIX Texteditoren	67
4.6.1 Editor ed	68
4.6.2 Editor vi	69
4.7 Compilerbau-Werkzeuge	72
4.7.1 Scanner-Generator Lex	72
4.7.2 Parser-Generator Yacc	75
4.8 UNIX Dokumentation	78
5 UNIX-kompatible Systemdienste	81
5.1 Portabilitätsgesichtspunkte	81
5.2 Fehlerbehandlung	83
5.2.1 Fehlercodes	83
5.3 Funktionen zum File-Handling	86
5.3.1 Sequentieller Zugriff	86
5.3.2 Random-Zugriff	92
5.3.3 Verweise auf Files	93
5.4 Standard-Files	97
5.5 Files in einer Multi-User-Umgebung	99
5.5.1 Zugriffsrechte	99
5.5.2 File-Attribute	105
5.6 Directories und File-Systeme	111
5.6.1 Directories	111
5.6.2 Systemroutinen zur Directory-Manipulation	116
5.6.3 File-Systeme	131
5.6.4 Special Files	133
5.7 Prozesse	136
5.7.1 Erzeugung von Prozessen	137
5.7.2 Prozeßüberlagerung	140
5.7.3 Vererbung von Filedeskriptoren	148
5.7.4 Prozeß-Attribute	152
5.7.5 Signale	157
5.7.6 Speicherverwaltung	186
5.8 Prozeß-Kommunikation I	188
5.8.1 Lock-Files	190
5.8.2 Pipes	193
5.8.3 Named Pipes (FIFO-Files)	204
5.8.4 File- und Record-Locking	208

5.9	Prozeß-Kommunikation II.....	217
5.9.1	IPC Message-Queues	217
5.9.2	Ein Beispiel für Messages-Queues	221
5.9.3	IPC Semaphore	227
5.9.5	Ein Beispiel für Semaphore und Shared Memory	233
5.9.6	IPC UNIX-Kommandos.....	259
5.9.7	Memory Mapping.....	260
5.10	Terminal-Steuerung	263
5.10.1	Grundlagen	263
5.10.2	Terminal-Übertragungsparameter.....	266
5.10.3	Die termcap- bzw. terminfo-Bibliothek	271
5.10.4	Die curses-Window-Bibliothek	279
6	Portabilität	289
6.1	Entwicklungstools.....	289
6.2	Fehlerbehandlung.....	291
6.3	Arbeit mit Files und Directories	292
6.4	Prozeßbezogene Aufgaben	298
6.5	Synchronisation	299
6.6	Terminalsteuerung.....	300
7	Fallstudien	303
7.1	Ein Wordcount-Programm.....	303
7.2	Ein Undelete-System.....	306
7.3	Eine Minishell	326
7.4	Ein Synchronisationsproblem.....	340
7.4.1	Programm ohne Synchronisation.....	340
7.4.2	Synchronisation mit einem Semaphor	342
7.4.3	Synchronisation durch ein Nachrichtensystem.....	346
7.4.4	Synchronisation durch eine Software-Lösung	349
A	Anhang.....	355
A.1	UNIX-Fehlercodes.....	355
A.2	Systemabhängige Konstanten unter UNIX.....	358
A.3	UNIX-Kommandos im Vergleich zu MS-DOS, OS/2 und Windows NT/95.....	360
	Literaturverzeichnis	363
	Sachwortverzeichnis.....	367