
Inhaltsverzeichnis

1	Einführung	1
1.1	Was ist ein Betriebssystem?	1
1.1.1	Betriebssystemkern	2
1.1.2	Systemmodule	2
1.1.3	Dienstprogramme	3
1.1.4	Unterschiedliche Arten von Betriebssystemen	3
1.2	Computer-Hardware	5
1.2.1	CPU	5
1.2.2	Busse	7
1.2.3	Speicher	8
1.2.4	Ein- und Ausgabegeräte	9
1.3	System Calls und Interrupts	11
1.3.1	Der eigentliche Aufruf	11
1.3.2	Interrupts	12
1.4	Ressourcen	13
1.5	Zusammenfassung	14
2	System Calls	15
2.1	Von der Anwendung zum System Call	15
2.2	Der Handler für System Calls	16
2.3	Direkte Aufrufe	17
2.4	Zusammenfassung	18
3	Prozesse und Threads	19
3.1	Grundlagen	19
3.1.1	Linux Process Control Block	20
3.1.2	Weiterführende Konzepte	20
3.1.3	Die Prozess-ID	24
3.1.4	Sessions und Process Groups	25
3.2	Erzeugen von Prozessen	25
3.2.1	Der ursprüngliche <code>fork()</code> -Aufruf	25

3.2.2	Ausführen eines anderen Programms: <code>exec()</code>	26
3.2.3	Prozesse und Vererbung	28
3.3	Beenden eines Prozesses	30
3.3.1	<code>exit()</code> und <code>wait()</code>	30
3.3.2	Synchronisation	32
3.4	„Leichtgewichtige Prozesse“	33
3.4.1	Der neue <code>fork()</code> -System Call	33
3.4.2	Die <code>vfork()</code> -Variante	33
3.4.3	Implementierung mit <code>clone()</code>	34
3.4.4	Threads	35
3.5	Zusammenfassung	38
4	Scheduling	39
4.1	Grundlagen	39
4.1.1	Prioritätsgesteuert	42
4.1.2	Round Robin	42
4.1.3	Prioritätsgesteuert mit Feedback	42
4.1.4	Länge der Zeitscheibe	43
4.1.5	Mehrprozessor-Systeme	43
4.1.6	Realtime	44
4.2	Linux Scheduling	44
4.2.1	Prioritätsarrays und Scheduling-Algorithmus	46
4.2.2	Der Wechsel der Prioritätsarrays	47
4.2.3	Prozesswechsel	47
4.3	Realtime	48
4.3.1	Realtime mit FIFO	49
4.3.2	Realtime mit Round Robin	49
4.3.3	Realtime System Calls	49
4.4	Timesharing	51
4.4.1	Dynamische Prioritäten und Zeitscheibenberechnung	51
4.4.2	Timesharing System Calls	52
4.4.3	Neue Prozesse	52
4.5	Load Balancing	52
4.6	Zusammenfassung	53
5	Speicherverwaltung	55
5.1	Grundlagen	55
5.1.1	Segmentierung	57
5.1.2	Paging	57
5.1.3	Virtueller Speicher	58
5.2	Ziele für Linux	60
5.2.1	Vielzahl von Hardware-Plattformen	60
5.2.2	Inhomogener Speicher	60
5.2.3	NUMA-Architekturen	61
5.2.4	Page Cache	61

5.2.5	pdflush : Zurückschreiben veränderter Pages	62
5.2.6	Slab: Verwaltung von Kernel-Objekten	62
5.2.7	Frame-Allocation zum spätesten Zeitpunkt	62
5.3	Prozess-Adressraum	63
5.3.1	Memory Deskriptor	64
5.3.2	Die Speicherbereiche	64
5.3.3	System Calls	66
5.4	Pagetable	68
5.4.1	Page locking	70
5.4.2	PSE	72
5.5	Paging	72
5.5.1	Pagefaults	72
5.5.2	Zones und NUMA	73
5.5.3	Anforderung zusammenhängender Frames	75
5.6	Page Cache	75
5.7	Swapping	78
5.7.1	Kernel Caches	78
5.7.2	pdflush	78
5.7.3	Auswahlstrategie	79
5.7.4	kswapd	80
5.7.5	Verwaltung des Cache-Bereichs	80
5.8	Slab Layer	82
5.9	Zusammenfassung	83
6	Synchronisation	85
6.1	Grundlagen	85
6.1.1	Race Conditions	85
6.1.2	Ansätze zur Synchronisation	87
6.1.3	Contention und Scalability von Sperren	89
6.2	Deadlock	89
6.3	Kernel Synchronisation	91
6.3.1	Atomare Operationen	91
6.3.2	Spinlocks	91
6.3.3	Semaphoren	92
6.3.4	Reader-/Writer-Locks	93
6.3.5	Big Kernel Lock	94
6.4	Synchronisation in Benutzerprogrammen	94
6.4.1	Signale	95
6.4.2	Semaphoren	99
6.5	Zusammenfassung	105

7	Interrupts	107
7.1	Grundlagen	107
7.1.1	Erkennung eines Interrupts	108
7.1.2	Geschwindigkeit versus Umfang	109
7.1.3	Der Interrupt-Handler	109
7.2	Implementierung	110
7.2.1	Datenstrukturen	110
7.2.2	Registrierung	112
7.2.3	Interrupt Requests	113
7.3	Interrupt Control	115
7.4	Bottom Half	117
7.4.1	Veraltete Ansätze	117
7.4.2	Soft-IRQ	117
7.4.3	Tasklets	121
7.4.4	Bearbeitung von Soft-IRQs und Tasklets bei hoher Last	124
7.4.5	Work Queues	125
7.4.6	Warteschlangen	129
7.5	Zusammenfassung	130
8	Dateisysteme und Plattenverwaltung	133
8.1	Grundlagen	133
8.1.1	Unterstützung von Dateitypen	133
8.1.2	Von der Lochkarte zur Platte	134
8.1.3	Directory und Dateioperationen	135
8.1.4	Implementierungen für Directories	137
8.1.5	Zugriffsmethoden	137
8.1.6	Bereitstellung von Speicherplatz	138
8.1.7	Directory	140
8.1.8	Datenschutz und Datensicherheit	142
8.1.9	Unterschiedliche Filesysteme	143
8.1.10	Festplattenzugriffe	143
8.2	Virtual File System (VFS)	144
8.2.1	VFS-Datenstrukturen für Dateien	144
8.2.2	VFS-Datenstrukturen für Partitionen	151
8.2.3	Suche nach einer Datei	154
8.2.4	Filedeskriptoren: Geöffnete Dateien	158
8.2.5	Die Verbindung zum Speicher	160
8.2.6	Filesysteme	161
8.3	System Calls	162
8.4	Beispiel: <code>read()</code>	167
8.4.1	Lesen ohne Readahead	173
8.4.2	Lesen mit Readahead	176
8.5	Elevator	180
8.6	Extended Filesystem 2	183
8.6.1	Der Superblock	186

8.6.2	Die Inode	189
8.6.3	Indirektion und Finden der Blöcke	194
8.6.4	Bereitstellung von Speicherplatz	195
8.7	Zusammenfassung	197
9	Kommunikation zwischen Prozessen	199
9.1	Grundlagen	199
9.2	Pipes	200
9.2.1	Der prinzipielle Aufbau	200
9.2.2	Beispiel zur Benutzung einer Pipe	201
9.2.3	Nicht-blockierender Zugriff	203
9.2.4	Schließen einer Pipe	205
9.2.5	Named Pipes (FIFO)	205
9.3	IPC – Inter Process Communication	206
9.3.1	IPC-Grundlagen	207
9.3.2	Message Queues	208
9.3.3	Benutzung von Message Queues	209
9.3.4	Semaphoren	212
9.3.5	Shared Memory	214
9.3.6	Der System Call <code>sys_ipc()</code>	215
9.4	Sockets	215
9.4.1	Schichtenmodell	216
9.4.2	System Calls	221
9.4.3	Implementierung	225
9.5	Zusammenfassung	235
10	Der Bootvorgang	237
10.1	Grundlagen	237
10.2	Vom BIOS zum Bootmanager	238
10.3	Der Kernel	239
10.4	Die Runlevel	242
10.5	Module	245
10.6	Zusammenfassung	246
A	Kompilieren des Kernels	247
B	Lineare Listen in Linux	251
C	Glossar	257
	Interessante WWW-Adressen	277
	Literaturverzeichnis	279
	Sachverzeichnis	281

Betriebssysteme

Eine kompakte Einführung mit Linux

Achilles, A.

2006, XV, 290 S. 31 Abb., Softcover

ISBN: 978-3-540-23805-8