

Inhaltsverzeichnis

Vorwort zur 6. Auflage	xi
Vorwort der Autoren zur 1. Auflage	xi
Vorwort von Linus Torvalds zur 1. Auflage	xiii
Danksagung	xiv
1 Linux – Das Betriebssystem	1
1.1 Wesentliche Eigenschaften	2
1.2 Linux-Distributionen	5
2 Die Übersetzung des Kerns	7
2.1 Wo finde ich was?	7
2.2 Die Übersetzung	10
2.3 Zusätzliche Konfigurationsmöglichkeiten	12
3 Einführung in den Kern	15
3.1 Wichtige Datenstrukturen	19
3.1.1 Die Taskstruktur	19
3.1.2 Die Prozesstabelle	28
3.1.3 Files und Inodes	28
3.1.4 Dynamische Speicherverwaltung	30
3.1.5 Warteschlangen und Semaphore	32
3.1.6 Systemzeit und Zeitgeber (Timer)	34
3.2 Zentrale Algorithmen	34
3.2.1 Signale	34
3.2.2 Hardwareinterrupts	36
3.2.3 Softwareinterrupts	37
3.2.4 Booten des Systems	38
3.2.5 Timerinterrupt	40
3.2.6 Scheduler	43
3.3 Implementierung von Systemrufen	46
3.3.1 Wie funktionieren Systemrufe eigentlich?	46
3.3.2 Beispiele für einfache Systemrufe	48
3.3.3 Beispiele für komplexere Systemrufe	49

4 Die Speicherverwaltung	59
4.1 Das architekturunabhängige Speichermodell	61
4.1.1 Speicherseiten	61
4.1.2 Virtueller Adressraum	61
4.1.3 Übersetzung der linearen Adresse	64
4.1.4 Pagedirectories	65
4.1.5 Die Pagetabelle	68
4.2 Der virtuelle Adressraum eines Prozesses	71
4.2.1 Das Nutzersegment	71
4.2.2 Virtuelle Speicherbereiche	73
4.2.3 Der Systemruf brk	76
4.2.4 Funktionen für das Mapping	77
4.2.5 Das Kernelsegment	77
4.2.6 Speicherreservierung im Kernelsegment während des Bootens	78
4.2.7 Dynamische Speicherreservierung im Kernelsegment	78
4.3 Das Caching der Blockgeräte	82
4.3.1 Blockpuffer	82
4.3.2 Bdflush und Kupdate	84
4.3.3 Die Listenstrukturen des Puffercaches	85
4.3.4 Verwendung des Puffercaches	86
4.4 Paging unter Linux	87
4.4.1 Speicherseitenverwaltung und -cache	90
4.4.2 Speicherseitenreservierung	93
4.4.3 Optimierung der Speicherseitenverwaltung durch Kernelthreads	96
4.4.4 Seitenfehler und das Zurückladen einer Speicherseite	97
5 Interprozesskommunikation	99
5.1 Synchronisation im Kern	101
5.2 Kommunikation über Dateien	107
5.2.1 Das Sperren ganzer Dateien	108
5.2.2 Sperren von Dateibereichen	109
5.3 Pipes	113
5.4 Debugging mit ptrace	116
5.5 System V IPC	119
5.5.1 Zugriffsrechte, Nummern und Schlüssel	119
5.5.2 Semaphore	120
5.5.3 Messagequeues	124
5.5.4 Shared Memory	128

5.5.5	Die Befehle <code>ipcs</code> und <code>ipcrm</code>	131
5.6	IPC mit Sockets	132
5.6.1	Ein einfaches Beispiel	132
5.6.2	Die Implementierung von Unix-Domain-Sockets	137
6	Das LINUX-Dateisystem	141
6.1	Grundlagen	142
6.2	Die Repräsentation von Dateisystemen im Kern	144
6.2.1	Das Mounten	146
6.2.2	Der Superblock	147
6.2.3	Superblock-Operationen	149
6.2.4	Der Verzeichniscache	153
6.2.5	DEntry-Operationen	155
6.2.6	Die Inode	156
6.2.7	Inode-Operationen	158
6.2.8	Die File-Struktur	161
6.2.9	File-Operationen	162
6.2.10	Das Öffnen einer Datei	167
6.3	Das <i>Ext2</i> -Dateisystem	170
6.3.1	Der Aufbau des <i>Ext2</i> -Dateisystems	171
6.3.2	Verzeichnisse im <i>Ext2</i> -Dateisystem	174
6.3.3	Blockallokation im <i>Ext2</i> -Dateisystem	174
6.3.4	Erweiterungen des <i>Ext2</i> -Dateisystems	175
6.4	Das <i>Proc</i> -Dateisystem	177
6.4.1	Strukturen des <i>Proc</i> -Dateisystems	177
6.4.2	Implementierung des <i>Proc</i> -Dateisystems	179
7	Gerätetreiber unter Linux	185
7.1	Zeichen- und Blockgeräte	187
7.2	Hardware	188
7.2.1	Port I/O	188
7.2.2	Der PCI-Bus	189
7.2.3	Der Dinosaurier – ISA Bus	200
7.2.4	ISA-PnP	209
7.3	Polling, Interrupts und Wait Queues	214
7.3.1	Polling	214
7.3.2	Interruptbetrieb	216
7.3.3	Interrupt Sharing	217
7.3.4	Softwareinterrupts	218
7.3.5	Bottom Halfs – die unteren Interrupthälften	219

7.3.6	Task Queues	220
7.3.7	Timer	222
7.4	Die Implementierung eines Treibers	223
7.4.1	Beispiel – PC Lautsprechertreiber	223
7.4.2	Ein einfacher Treiber	227
7.4.3	Die Setup-Funktion	228
7.4.4	Init	230
7.4.5	Open und Release	232
7.4.6	Read und Write	233
7.4.7	IOCTL	235
7.4.8	Poll	237
7.4.9	Lseek	240
7.4.10	MMap	240
7.4.11	Fasync	241
7.4.12	Readdir, Fsync	244
7.5	Dynamische und statische Treiber	245
8	Netzwerkimplementierung	247
8.1	Einführung und Überblick	248
8.1.1	Das Schichtenmodell der Netzwerkimplementation	249
8.1.2	Die Reise der Daten	249
8.2	Wichtige Strukturen	254
8.2.1	Die socket-Struktur	254
8.2.2	Die Struktur sk_buff – Pufferverwaltung im Netzwerk	255
8.2.3	Der INET-Socket – spezieller Teil eines Sockets	259
8.2.4	Protokoloperationen in der proto-Struktur	263
8.2.5	Die allgemeine Struktur einer Socketadresse	265
8.3	Netzwerkgeräte unter Linux	265
8.3.1	Ethernet	272
8.3.2	SLIP und PLIP	273
8.3.3	Das Loopback-Gerät	273
8.3.4	Das Dummy-Gerät	273
8.3.5	Ein Beispielgerät	274
9	Module und Debugging	277
9.1	Was sind Module?	277
9.2	Implementierung im Kernel	278
9.2.1	Signatur von Symbolen	280
9.3	Bedeutung der Objektsektionen für Module und Kern	281
9.4	Parameterübergabe und Beispiel	283

9.5	Was kann als Modul implementiert werden?	284
9.6	Der Kernel-Dämon	285
9.7	Einfacher Datenaustausch zwischen Modulen	286
9.8	Ein Modulbeispiel	287
9.9	Debugging	288
9.9.1	Änderungen sind der Anfang vom Ende	289
9.9.2	Der beste Debugger – <code>printk()</code>	290
9.9.3	Debuggen mit GDB	291
10	Multiprocessing	293
10.1	Die Intel-Mehrprozessorspezifikation	293
10.2	Probleme bei Mehrprozessorsystemen	294
10.3	Änderungen am Kern	295
10.3.1	Initialisierung des Kerns	295
10.3.2	Scheduling	296
10.3.3	Interruptbehandlung	296
10.4	Atomare Operationen	297
10.4.1	Der atomare Datentyp	297
10.4.2	Zugriffe auf den atomaren Datentyp	297
10.4.3	Ändern und Testen von atomaren Variablen	297
10.5	Spinlocks	298
10.5.1	Zutrittsfunktionen	298
10.5.2	Read-Write-Spinlocks	299
Anhang		301
A	Systemrufe	301
A.1	Die Prozessverwaltung	302
A.2	Das Dateisystem	350
A.3	Die Kommunikation	385
A.4	Die Speicherverwaltung	388
A.5	Und der ganze Rest	396
B	Kernnahe Kommandos	397
B.1	<code>free</code> – Übersicht über den Systemspeicher	397
B.2	<code>ps</code> – Ausgabe der Prozessstatistik	398
B.3	<code>top</code> – Die CPU-Charts	403
B.4	<code>init</code> – Primus inter pares	405
B.5	<code>shutdown</code> – das Herunterfahren des Systems	413
B.6	<code>strace</code> – Observierung eines Prozesses	415
B.7	Konfiguration des Netzwerk-Interfaces	418

B.8	traceroute — der Ariadnefaden im Internet	419
B.9	Konfiguration einer seriellen Schnittstelle	421
B.10	Konfiguration einer parallelen Schnittstelle	424
B.11	Wir basteln uns einen Verzeichnisbaum	425
C	Das Proc-Dateisystem	433
C.1	Das Verzeichnis <code>/proc/</code>	433
C.2	Das Verzeichnis <code>net/</code>	440
C.3	Das Verzeichnis <code>self/</code>	442
C.4	Das Verzeichnis <code>sys/</code>	446
D	Der Boot-Prozess	449
D.1	Der Ablauf des Bootens	449
D.2	LILO — der Linux-Lader	451
D.2.1	MS-DOS-MBR startet LILO	451
D.2.2	LILO wird von einem Bootmanager gestartet	452
D.2.3	LILO im Master-Boot-Record	452
D.2.4	LILO-Dateien	453
D.2.5	LILO-Boot-Parameter	457
D.2.6	LILO-Startmeldungen	458
D.2.7	Fehlermeldungen	458
E	Nützliche Kernfunktionen	461
	Index	483