

**Michael Beck, Harald Böhme,  
Mirko Dziadzka, Ulrich Kunitz,  
Robert Magnus, Dirk Verworner,  
Claus Schröter**

# **Linux-Kernel- Programmierung**

**Algorithmen und Strukturen der Version 2.2**

**5., aktualisierte und erweiterte Auflage**



**ADDISON-WESLEY**

---

**An imprint of Pearson Education**

München • Reading, Massachusetts • Menlo Park, California • New York • Harlow, England  
Don Mills, Ontario • Sydney • Mexico City • Madrid • Amsterdam

# Inhaltsverzeichnis

<b>Vorwort zur 5. Auflage</b>	<b>xi</b>
<b>Vorwort der Autoren zur 1. Auflage</b>	<b>xi</b>
<b>Vorwort von Linus Torvalds zur 1. Auflage</b>	<b>xiv</b>
<b>1 Linux – Das Betriebssystem</b>	<b>1</b>
1.1 Wesentliche Eigenschaften . . . . .	3
1.2 Linux-Distributionen . . . . .	5
<b>2 Die Übersetzung des Kerns</b>	<b>7</b>
2.1 Wo finde ich was? . . . . .	7
2.2 Die Übersetzung . . . . .	9
2.3 Zusätzliche Konfigurationsmöglichkeiten . . . . .	11
<b>3 Einführung in den Kern</b>	<b>15</b>
3.1 Wichtige Datenstrukturen . . . . .	20
3.1.1 Die Taskstruktur . . . . .	20
3.1.2 Die Prozeßtablelle . . . . .	28
3.1.3 Files und Inodes . . . . .	29
3.1.4 Dynamische Speicherverwaltung . . . . .	31
3.1.5 Warteschlangen und Semaphore . . . . .	32
3.1.6 Systemzeit und Zeitgeber (Timer) . . . . .	34
3.2 Zentrale Algorithmen . . . . .	36
3.2.1 Signale . . . . .	36
3.2.2 Interrupts . . . . .	37
3.2.3 Booten des Systems . . . . .	38
3.2.4 Timerinterrupt . . . . .	40
3.2.5 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
<b>4 Die Speicherverwaltung</b>	<b>59</b>
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	Das Pagedirectory . . . . .	66
4.1.5	Das mittlere Pagedirectory . . . . .	66
4.1.6	Die Pagetabelle . . . . .	67
4.2	Der virtuelle Adreßraum eines Prozesses . . . . .	70
4.2.1	Das Nutzersegment . . . . .	70
4.2.2	Virtuelle Speicherbereiche . . . . .	71
4.2.3	Der Systemruf brk . . . . .	76
4.2.4	Funktionen für das Mapping . . . . .	76
4.2.5	Das Kernelsegment . . . . .	77
4.2.6	Statische Speicherreservierung im Kernelsegment . . . . .	78
4.2.7	Dynamische Speicherreservierung im Kernelsegment . . . . .	78
4.3	Das Caching der Blockgeräte . . . . .	82
4.3.1	Blockpuffer . . . . .	82
4.3.2	Update und Bdflush . . . . .	84
4.3.3	Die Listenstrukturen des Puffercaches . . . . .	85
4.3.4	Verwendung des Puffercaches . . . . .	86
4.4	Paging unter Linux . . . . .	88
4.4.1	Speicherseitenverwaltung und -cache . . . . .	91
4.4.2	Speicherseitenreservierung . . . . .	92
4.4.3	Seitenfehler und das Zurückladen einer Speicherseite . . . . .	96
<b>5</b>	<b>Interprozeßkommunikation</b> . . . . .	<b>99</b>
5.1	Synchronisation im Kern . . . . .	101
5.2	Kommunikation über Dateien . . . . .	108
5.2.1	Das Sperren ganzer Dateien . . . . .	109
5.2.2	Sperren von Dateibereichen . . . . .	110
5.3	Pipes . . . . .	114
5.4	Debugging mit ptrace . . . . .	116
5.5	System V IPC . . . . .	120
5.5.1	Zugriffsrechte, Nummern und Schlüssel . . . . .	120
5.5.2	Semaphore . . . . .	121
5.5.3	Messagequeues . . . . .	124
5.5.4	Shared Memory . . . . .	128
5.5.5	Die Befehle ipcs und ipcrm . . . . .	132
5.6	IPC mit Sockets . . . . .	134
5.6.1	Ein einfaches Beispiel . . . . .	134
5.6.2	Die Implementierung von Unix-Domain-Sockets . . . . .	138
<b>6</b>	<b>Das LINUX-Dateisystem</b> . . . . .	<b>143</b>
6.1	Grundlagen . . . . .	144
6.2	Die Repräsentation von Dateisystemen im Kern . . . . .	147
6.2.1	Das Mounten . . . . .	148

6.2.2	Superblockoperationen . . . . .	151
6.2.3	Die Inode . . . . .	154
6.2.4	Inode-Operationen . . . . .	156
6.2.5	Die File-Struktur . . . . .	160
6.2.6	File-Operationen . . . . .	160
6.2.7	Das Öffnen einer Datei . . . . .	165
6.3	Das sysctl-Interface . . . . .	168
6.4	Das Ext2-Dateisystem . . . . .	170
6.4.1	Der Aufbau des Ext2-Dateisystems . . . . .	170
6.4.2	Verzeichnisse im Ext2-Dateisystem . . . . .	172
6.4.3	Blockallokation im Ext2-Dateisystem . . . . .	173
6.4.4	Erweiterungen des Ext2-Dateisystems . . . . .	174
<b>7</b>	<b>Gerätetreiber unter Linux</b> . . . . .	<b>177</b>
7.1	Zeichen- und Blockgeräte . . . . .	178
7.2	Hardware . . . . .	180
7.2.1	Port I/O . . . . .	180
7.2.2	Der Dinosaurier - ISA Bus . . . . .	181
7.2.3	PCI-Bus . . . . .	190
7.3	Polling, Interrupts und Wait Queues . . . . .	196
7.3.1	Polling . . . . .	196
7.3.2	Interruptbetrieb . . . . .	197
7.3.3	Interrupt Sharing . . . . .	199
7.3.4	Bottom Halfs – die unteren Interrupthälften . . . . .	200
7.3.5	Task Queues . . . . .	201
7.3.6	Timer . . . . .	204
7.4	Die Implementierung eines Treibers . . . . .	206
7.4.1	Beispiel – PC Lautsprechertreiber . . . . .	206
7.4.2	Ein einfacher Treiber . . . . .	209
7.4.3	Die Setup-Funktion . . . . .	211
7.4.4	Die Request-Funktion . . . . .	212
7.4.5	Init . . . . .	215
7.4.6	Open und Release . . . . .	217
7.4.7	Read und Write . . . . .	218
7.4.8	IOCTL . . . . .	221
7.4.9	Poll . . . . .	224
7.4.10	Lseek . . . . .	226
7.4.11	MMap . . . . .	226
7.4.12	Fasync . . . . .	228
7.4.13	Readdir, Fsync . . . . .	230
7.4.14	Check_media_change und Revalidate . . . . .	231
7.5	Dynamische und statische Treiber . . . . .	231
<b>8</b>	<b>Netzwerkimplementierung</b> . . . . .	<b>233</b>
8.1	Einführung und Überblick . . . . .	234

8.1.1	Das Schichtenmodell der Netzwerkimplementation . . . . .	235
8.1.2	Die Reise der Daten . . . . .	236
8.2	Wichtige Strukturen . . . . .	241
8.2.1	Die <code>socket</code> -Struktur . . . . .	241
8.2.2	Die Struktur <code>sk_buff</code> – Pufferverwaltung im Netzwerk .	242
8.2.3	Der INET-Socket – spezieller Teil eines Sockets . . . . .	245
8.2.4	Protokoloperationen in der <code>proto</code> -Struktur . . . . .	248
8.2.5	Die allgemeine Struktur einer Socketadresse . . . . .	251
8.3	Netzwerkgeräte unter Linux . . . . .	252
8.3.1	Ethernet . . . . .	258
8.3.2	SLIP und PLIP . . . . .	259
8.3.3	Das Loopback-Gerät . . . . .	259
8.3.4	Das Dummy-Gerät . . . . .	259
8.3.5	Ein Beispielgerät . . . . .	260
<b>9</b>	<b>Module und Debugging</b>	<b>265</b>
9.1	Was sind Module? . . . . .	265
9.2	Implementierung im Kernel . . . . .	266
9.2.1	Signatur von Symbolen . . . . .	269
9.3	Bedeutung der Objektsektionen für Module und Kern . . . . .	269
9.4	Parameterübergabe . . . . .	271
9.5	Was kann als Modul implementiert werden? . . . . .	273
9.6	Der Kernel-Dämon . . . . .	273
9.7	Ein Beispielmodul . . . . .	275
9.8	Debugging . . . . .	276
9.8.1	Änderungen sind der Anfang vom Ende . . . . .	277
9.8.2	Der beste Debugger – <code>printk()</code> . . . . .	278
9.8.3	Debuggen mit GDB . . . . .	279
<b>10</b>	<b>Multiprocessing</b>	<b>281</b>
10.1	Die Intel-Mehrprozessorspezifikation . . . . .	281
10.2	Probleme bei Mehrprozessorsystemen . . . . .	283
10.3	Änderungen am Kern . . . . .	284
10.3.1	Initialisierung des Kerns . . . . .	284
10.3.2	Scheduling . . . . .	285
10.3.3	Interruptbehandlung . . . . .	285
10.4	Atomare Operationen . . . . .	285
10.4.1	Der atomare Datentyp . . . . .	285
10.4.2	Zugriffe auf den atomaren Datentyp . . . . .	286
10.4.3	Ändern und Test von atomaren Variablen . . . . .	286
10.5	Spinlocks . . . . .	287
10.5.1	Zutrittsfunktionen . . . . .	287
10.5.2	Read-Write-Spinlocks . . . . .	288
<b>A</b>	<b>Systemrufe</b>	<b>289</b>