

Inhaltsverzeichnis

1	Einführung und Überblick	1
1.1	Aufgaben des Kerns	2
1.2	Implementierungsstrategien	2
1.3	Bestandteile des Kerns	3
1.3.1	Prozesse, Taskswitching und Scheduling	3
1.3.2	Unix-Prozesse	4
1.3.3	Aufbau des virtuellen Adressraums	6
1.3.4	Privilegstufen	7
1.3.5	Virtuelle und physikalische Adressräume	8
1.3.6	Seitentabellen	9
1.3.7	Systemaufrufe	13
1.3.8	Gerätetreiber, Block- und Zeichengeräte	14
1.3.9	Netzwerke	14
1.3.10	Dateisysteme	14
1.3.11	Module	15
1.3.12	Caching	15
1.3.13	Listenverwaltung	16
1.3.14	Datentypen	17
2	Prozessverwaltung	19
2.1	Prozessprioritäten	20
2.2	Lebenszyklus eines Prozesses	22
2.2.1	Präemptives Multitasking	23
2.3	Repräsentation von Prozessen	25
2.3.1	Prozesstypen	29
2.3.2	Prozessidentifikations-Nummern	30
2.3.3	Das Tasknetzwerk	35
2.4	Systemaufrufe zur Prozessverwaltung	36
2.4.1	Prozessverdoppelung	36
2.4.2	Kernel-Threads	47
2.4.3	Starten neuer Programme	49
2.4.4	Prozesse beenden	52
2.5	Implementierung des Schedulers	52
2.5.1	Prozess-Scheduling	52
2.5.2	Datenstrukturen	53
2.5.3	Priority Scheduling	57
2.6	Erweiterungen des Schedulers	65
2.6.1	Echtzeitprozesse	65
2.6.2	SMP-Scheduling	67
2.6.3	Kernel-Präemption	68

3	Speicherverwaltung	71
3.1	Überblick	71
3.2	Organisation nach dem (N)UMA-Modell	73
3.2.1	Überblick	73
3.2.2	Datenstrukturen	75
3.3	Seitentabellen	83
3.3.1	Datenstrukturen	84
3.3.2	Einträge erzeugen und manipulieren	89
3.4	Initialisierung der Speicherverwaltung	90
3.4.1	Aufsetzen der Datenstrukturen	90
3.4.2	Architektur-spezifisches Setup	96
3.4.3	Speicherverwaltung während des Bootens	111
3.5	Verwaltung des physikalischen Speichers	117
3.5.1	Allokator-API	118
3.5.2	Seiten reservieren	124
3.5.3	Seiten zurückgeben	134
3.5.4	Kernallokation unzusammenhängender Seiten	135
3.5.5	Kernelmappings	142
3.6	Der Slab-Allokator	147
3.6.1	Speicherverwaltung im Kern	148
3.6.2	Prinzip der Slab-Allokation	150
3.6.3	Implementierung	154
3.6.4	Allgemeine Caches	168
3.7	Verwaltung des virtuellen Prozessspeichers	170
3.7.1	Der virtuelle Prozess-Adressraum	171
3.7.2	Datenstrukturen	174
3.7.3	Operationen mit Regionen	177
3.7.4	Adressräume	183
3.7.5	Memory Mappings	184
3.7.6	Reverse Mapping	190
3.7.7	Verwaltung des Heaps	195
3.7.8	Behandlung von Seitenfehlern	197
3.7.9	Korrektur von Userspace-Seitenfehlern	202
3.7.10	Kernel-Seitenfehler	207
3.7.11	Daten kopieren zwischen Kernel- und Userspace	209
3.8	Prozessorcache- und TLB-Steuerung	210
4	Interprozesskommunikation und Locking	215
4.1	Steuerungsmechanismen	215
4.1.1	Race Conditions	216
4.1.2	Critical Sections	217
4.2	Locking-Mechanismen des Kerns	219
4.2.1	Atomare Operationen	220
4.2.2	Spinlocks	221
4.2.3	Semaphoren	222
4.2.4	Reader/Writer-Locks	224
4.2.5	Das große Kernel-Lock	224

4.3	System V-Interprozesskommunikation	225
4.3.1	System V-Mechanismen	225
4.3.2	Semaphoren	226
4.3.3	Message Queues	233
4.3.4	Shared Memory	237
4.4	Andere IPC-Mechanismen	238
4.4.1	Signale	238
4.4.2	Pipes und Sockets	245
5	Gerätetreiber	247
5.1	IO-Architektur	247
5.1.1	Erweiterungshardware	248
5.2	Zugriff auf Erweiterungsgeräte	253
5.2.1	Gerätespezialdateien	253
5.2.2	Zeichen-, Block- und sonstige Geräte	253
5.2.3	Gerätekontrolle mit Ioctls	255
5.3	Treiberregistrierung	257
5.4	Verbindung mit dem Dateisystem	259
5.4.1	Geräte-datei-Elemente in Inoden	259
5.4.2	Repräsentation von Major und Minor Number	261
5.4.3	Standard-Dateioperationen	261
5.4.4	Standardoperationen für Zeichengeräte	262
5.4.5	Standardoperationen für Blockgeräte	262
5.5	Treiberoperationen	263
5.5.1	Zeichengeräte	263
5.5.2	Blockgeräte	265
5.5.3	Implementierung von Ioctls	282
5.6	Ressourcen-Reservierung	283
5.6.1	Ressourcenverwaltung	283
5.6.2	IO-Memory	286
5.6.3	IO-Ports	288
5.7	Bussysteme	289
5.7.1	Das allgemeine Driver Model	290
5.7.2	Der PCI-Bus	294
5.7.3	USB	303
6	Module	313
6.1	Module verwenden	314
6.1.1	Einfügen und Entfernen	314
6.1.2	Abhängigkeiten	317
6.1.3	Automatisches Laden	319
6.1.4	Abfragen von Moduleigenschaften	320
6.2	Module einfügen und löschen	321
6.2.1	Modulrepräsentation	322
6.2.2	Abhängigkeiten und Referenzen	325
6.2.3	Binärer Aufbau von Modulen	328
6.2.4	Module einfügen	333
6.2.5	Module entfernen	340

6.3	Automatisierung und Hotplugging	341
6.3.1	Automatisches Laden mit kmod	341
6.3.2	Hotplug	343
6.4	Versionskontrolle	344
6.4.1	Checksummenverfahren	345
6.4.2	Funktionen zur Versionskontrolle	348
7	Das virtuelle Dateisystem	351
7.1	Dateisystemtypen	352
7.2	Das Common File Model	353
7.2.1	Inoden	354
7.2.2	Verknüpfungen	355
7.2.3	Programmierschnittstelle	356
7.2.4	Dateien als Universalschnittstelle	357
7.3	Aufbau des VFS	358
7.3.1	Strukturüberblick	358
7.3.2	Inoden	360
7.3.3	Prozessspezifische Informationen	365
7.3.4	Dateioperationen	368
7.3.5	Dentry-Cache	373
7.4	Arbeiten mit VFS-Objekten	377
7.4.1	Dateisystemoperationen	377
7.4.2	Dateioperationen	387
7.5	Standardfunktionen	395
7.5.1	Generische Leseroutine	396
7.5.2	Der nopage-Mechanismus	398
8	Dateisystemimplementierungen	401
8.1	Second Extended Filesystem	403
8.1.1	Physikalischer Aufbau	404
8.1.2	Datenstrukturen	411
8.1.3	Anlegen des Dateisystems	425
8.1.4	Dateisystemaktionen	427
8.2	Third Extended Filesystem	446
8.2.1	Konzepte	447
8.2.2	Datenstrukturen	448
8.3	Das proc-Dateisystem	451
8.3.1	Inhalt von /proc	451
8.3.2	Datenstrukturen	458
8.3.3	Initialisierung	462
8.3.4	Einhängen des Dateisystems	464
8.3.5	Verwaltung von /proc-Einträgen	466
8.3.6	Informationen lesen und schreiben	471
8.3.7	Taskbezogene Informationen	473
8.3.8	Der System Control-Mechanismus	479

9	Netzwerke	491
9.1	Verkettete Computer	492
9.2	ISO/OSI- und TCP/IP-Referenzmodell	492
9.3	Kommunikation über Sockets	496
9.3.1	Anlegen eines Sockets	497
9.3.2	Verwendung von Sockets	498
9.3.3	Datagram-Sockets	503
9.4	Das Schichtmodell der Netzwerkimplementierung	504
9.5	Socketpuffer	506
9.5.1	Datenverwaltung mit Socketpuffern	507
9.5.2	Verwaltungsdaten eines Socketpuffers	508
9.6	Datenübertragungsschicht	509
9.6.1	Repräsentation von Netzwerkgeräten	510
9.6.2	Empfangen von Paketen	513
9.6.3	Senden von Paketen	517
9.7	Vermittlungsschicht	517
9.7.1	IPv4	518
9.7.2	Empfangen von Paketen	520
9.7.3	Lokale Auslieferung an die Transportschicht	521
9.7.4	Paketweiterleitung	523
9.7.5	Pakete senden	524
9.7.6	Netfilter	527
9.7.7	IPv6	532
9.8	Transportschicht	534
9.8.1	UDP	534
9.8.2	TCP	536
9.9	Anwendungsschicht	549
9.9.1	Socket-Datenstrukturen	549
9.9.2	Sockets und Dateien	552
9.9.3	Der <code>socketcall</code> -Systemaufruf	553
9.9.4	Sockets erzeugen	554
9.9.5	Daten empfangen	555
9.9.6	Daten versenden	556
10	Systemaufrufe	557
10.1	Grundlagen der Systemprogrammierung	557
10.1.1	Verfolgung von Systemaufrufen	558
10.1.2	Unterstützte Standards	561
10.1.3	Restarting system calls	562
10.2	Vorhandene Systemaufrufe	564
10.3	Realisierung von Systemaufrufen	569
10.3.1	Struktur von Systemcalls	569
10.3.2	Zugriffe auf den Userspace	576
10.3.3	Systemcall-Tracing	577
10.3.4	Systemaufrufe vom Kernel aus	585

11 Kernel-Aktivitäten und Zeitfluss	587
11.1 Interrupts	587
11.1.1 Interrupt-Typen	588
11.1.2 Hardware-IRQs	590
11.1.3 Bearbeiten von Interrupts	590
11.1.4 Initialisierung und Reservierung von IRQs	593
11.1.5 Abarbeiten von IRQs	600
11.2 Software-Interrupts	606
11.2.1 Starten der SoftIRQ-Verarbeitung	607
11.2.2 Der SoftIRQ-Daemon	608
11.3 Tasklets und Work Queues	609
11.3.1 Tasklets	610
11.4 Wait Queues und Completions	612
11.4.1 Wait Queues	612
11.4.2 Completions	615
11.4.3 Work Queues	616
11.5 Kerntimer	617
11.5.1 Einsatz von Timern	617
11.5.2 Zeitdomänen	618
11.5.3 Der Timer-Interrupt	619
11.5.4 Datenstrukturen	621
11.5.5 Dynamische Timer	622
11.5.6 Aktivierung neuer Timer	627
11.5.7 Implementierung der timerbezogenen Systemaufrufe	627
11.5.8 Verwaltung der Prozesszeiten	628
12 Page- und Buffer-Cache	629
12.1 Struktur des Page-Caches	630
12.1.1 Verwalten und Finden gecacheter Seiten	631
12.1.2 Zurückschreiben modifizierter Daten	632
12.2 Der Buffer-Cache	633
12.3 Adressräume	635
12.3.1 Datenstrukturen	636
12.3.2 Seitenbäume	637
12.3.3 Operationen auf Adressräumen	640
12.4 Implementierung des Puffer-Caches	643
12.4.1 Datenstrukturen	643
12.4.2 Operationen	645
12.4.3 Zusammenspiel von Page und Buffer Cache	646
12.4.4 Eigenständige Puffer	651
12.4.5 Operationen mit ganzen Seiten	657
13 Datensynchronisation	659
13.1 pfflush	660
13.2 Starten eines neuen Threads	661
13.3 Thread-Initialisierung	661
13.4 Durchführen der Arbeit	663
13.5 Periodisches Zurückschreiben	664

13.6	Assoziierte Datenstrukturen	664
13.6.1	Seitenstatus	664
13.6.2	Writeback-Kontrolle	665
13.6.3	Anpassbare Parameter	666
13.7	Zentrale Steuerung	667
13.8	Superblock-Synchronisation	669
13.9	Inoden-Synchronisation	670
13.10	Verstopfungen	674
13.11	Zurückschreiben unter Druck	677
13.12	Systemaufrufe zur Synchronisationskontrolle	679
13.13	Vollständige Synchronisierung	679
13.13.1	Synchronisieren der Inoden	680
13.14	Synchronisieren einzelner Dateien	681
13.15	Synchronisieren von Memory Mappings	683
14	Swapping	685
14.1	Überblick	685
14.1.1	Auslagerbare Seiten	686
14.1.2	Page Thrashing	687
14.1.3	Algorithmen zur Seitenersetzung	687
14.2	Swapping im Linux-Kernel	689
14.2.1	Organisation des Swap-Bereichs	690
14.2.2	Überprüfen der Speicherauslastung	691
14.2.3	Auswahl auszulagernder Seiten	691
14.2.4	Behandlung von Page Faults	692
14.2.5	Verkleinern von Kernelcaches	692
14.3	Verwaltung von Swap-Bereichen	693
14.3.1	Datenstrukturen	693
14.3.2	Anlegen	698
14.3.3	Aktivieren eines Swap-Bereichs	699
14.4	Der Swap-Cache	703
14.4.1	Identifikation ausgelagerter Seiten	705
14.4.2	Aufbau des Caches	707
14.4.3	Einfügen neuer Seiten	709
14.4.4	Suchen einer Seite	714
14.5	Zurückschreiben der Daten	715
14.6	Seitenauswahl – Swap Policy	716
14.6.1	Überblick	717
14.6.2	Datenstrukturen	719
14.6.3	Verkleinern von Zonen	721
14.6.4	Auffüllen der <i>inactive</i> -Liste	723
14.6.5	Auslagern inaktiver Seiten	728
14.7	Behandlung von Page Faults	732
14.7.1	Einlagern der Seite	732
14.7.2	Lesen der Daten	734
14.7.3	Swap-Readahead	735

14.8	Auslösen des Swappings	736
14.8.1	Periodisches Auslagern mit <code>kswapd</code>	736
14.8.2	Auslagern bei akuter Speicherknappheit	739
14.9	Verkleinern anderer Caches	741
14.9.1	Datenstrukturen	742
14.9.2	Registrieren und Entfernen von Shrinkern	742
14.9.3	Cache-Verkleinerung	743
	Literaturverzeichnis	745
	Index	747
	Symbole	755

Folgende Anhänge sind online auf <http://www.linux-kernel.de> verfügbar:

A Architekturspezifisches

B Arbeiten mit dem Quellcode

C Anmerkungen zu C

D Systemstart

E Das ELF-Binärformat