

# Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b>	<b>1</b>
1.1	Zweck .....	1
1.2	Definitionen .....	3
1.3	Einordnung im Computersystem .....	4
1.4	Betriebssystemarten .....	6
1.4.1	Klassische Einteilungen .....	6
1.5	Betriebssystemarchitekturen .....	7
1.5.1	Architekturformen .....	7
1.5.2	Benutzer-/Kernmodus .....	8
1.5.3	Monolithische Systeme .....	10
1.5.4	Geschichtete Systeme .....	10
1.5.5	Mikrokernsysteme .....	11
1.5.6	Beispiele von Systemarchitekturen .....	12
1.5.7	Abstraktionen aus Benutzer- und Entwicklersicht .....	15
<b>2</b>	<b>Grundlagen der Programmausführung und Systemprogrammierung</b>	<b>21</b>
2.1	Programmausführung und Hardware .....	21
2.1.1	Rechner- und Prozessorgrundlagen .....	23
2.1.2	Grundlagen des Adressraums .....	32
2.1.3	Grundlagen der Programmausführung .....	42
2.1.4	Unterprogrammmechanismen .....	50
2.2	Systemprogrammierung .....	63
2.2.1	Wahl der Systemprogrammiersprache .....	63
2.2.2	Laufzeitsystem der Programmiersprache C .....	70
2.2.3	Unterprogrammtechniken .....	71
2.2.4	Grundlagen der Systemprogrammierung .....	72
2.2.5	Systemprogrammierschnittstellen .....	91

<b>3</b>	<b>Prozesse und Threads</b>	<b>97</b>
3.1	Parallelverarbeitung	97
3.1.1	Darstellung von parallelen Abläufen	97
3.1.2	Hardware-Parallelität	98
3.1.3	Software-Parallelität	99
3.1.4	Begriffe	99
3.2	Prozessmodell	102
3.2.1	Grundprinzip	102
3.2.2	Prozesserzeugung und Terminierung	106
3.2.3	Prozesse unter Unix	109
3.2.4	Funktionsweise der Unix-Shell	115
3.2.5	Prozesse & Jobs unter Windows	117
3.2.6	Vererbung unter Prozessen	119
3.2.7	Prozeshierarchie	120
3.2.8	Systemstart	120
3.3	Threads	124
3.3.1	Thread-Modell	125
3.3.2	Vergleich Prozesse zu Threads	125
3.3.3	Implementierung des Multithreading	128
3.3.4	Threads & Fibers unter Windows	133
3.3.5	Threads unter Unix	138
3.4	Prozessorzuteilungsstrategien	140
3.4.1	Quasiparallelität im Einprozessorsystem	140
3.4.2	Prozess- und Thread-Zustände	142
3.4.3	Zuteilungsstrategien	146
3.4.4	Multiprozessor-Scheduling	159
3.4.5	POSIX-Thread-Scheduling	161
3.4.6	Java-Thread-Scheduling	163
3.4.7	Scheduling unter Windows	165
3.4.8	Scheduling unter Unix	171
<b>4</b>	<b>Synchronisation von Prozessen &amp; Threads</b>	<b>177</b>
4.1	Synchronisationsbedarfe und Lösungsansätze	177
4.1.1	Problem der Ressourcenteilung	177
4.1.2	Verlorene Aktualisierung (lost update problem)	178
4.1.3	Inkonsistente Abfrage (inconsistent read)	180
4.1.4	Absicherung mit Selbstverwaltung – naiver Ansatz	181
4.1.5	Absicherung mit Selbstverwaltung – korrekter Ansatz	182
4.1.6	Absicherung mit Systemmitteln	184

4.2	Semaphore	184
4.2.1	Semaphortypen	186
4.2.2	Implementierungsfragen	186
4.3	Anwendung der Semaphore	190
4.3.1	Absicherung kritischer Bereiche (mutual exclusion)	190
4.3.2	Synchronisation von Abläufen (barrier synchronization)	191
4.3.3	Produzenten & Konsumenten (producer and consumer)	193
4.3.4	Leser & Schreiber (readers and writers)	196
4.3.5	Problem der Prioritätsumkehrung (priority inversion)	201
4.3.6	Weitere Anwendungsprobleme	203
4.4	Implementierungen von Semaphoren	204
4.4.1	Semaphore unter Unix	204
4.4.2	Semaphore unter Windows	208
4.5	Unix-Signale	215
4.5.1	Idee & Grundprinzip der Unix-Signale	215
4.5.2	Programmierung der Signale	218
4.5.3	Signale im Multithreading	222
4.5.4	Realtime-Signale	223
4.6	Verklemmungsproblematik (deadlocks)	223
4.6.1	Ursache	224
4.6.2	Deadlock-Bedingungen	227
4.6.3	Lösungsansätze und ihre Beurteilung	228
4.7	Vermeidung von Synchronisationsengpässen	235
4.7.1	Granularität der Absicherung	236
4.7.2	Replikation der abgesicherten Ressource	237
<b>5</b>	<b>Kommunikation von Prozessen &amp; Threads</b>	<b>239</b>
5.1	Überblick über Synchronisation und Kommunikation	239
5.2	Nachrichtenbasierte Verfahren	241
5.2.1	Allgemeine Aspekte	241
5.2.2	Unix-Pipes	247
5.2.3	Windows-Pipes	255
5.2.4	Unix Message Queues	259
5.2.5	Windows-Messages	261
5.2.6	Windows-Mailslots	263

5.3	Speicherbasierte Verfahren	265
5.3.1	Gemeinsamer Speicher unter Windows	266
5.3.2	Gemeinsamer Speicher unter Unix	266
5.4	Monitor	268
5.4.1	Grundprinzip	268
5.4.2	Java-Monitor	271
5.4.3	Monitornachbildung mit Bedingungsvariablen	272
5.5	Rendezvous	278
5.5.1	Grundprinzip	278
5.5.2	Synchronisation in Client/Server-Systemen (barber shop)	279
5.6	Rechnerübergreifende Interprozesskommunikation	280
5.6.1	Netzwerksoftware	281
5.6.2	Berkeley-Sockets	283
5.6.3	Remote Procedure Call (RPC)	290
5.6.4	Überblick über Middleware	296
<b>6</b>	<b>Ein- und Ausgabe</b>	<b>299</b>
6.1	Peripherie	299
6.1.1	Einordnung im Rechnermodell	299
6.1.2	Begriffsdefinitionen	300
6.2	Ein-/Ausgabeabläufe	300
6.2.1	Programmgesteuerte Ein-/Ausgabe	301
6.2.2	Ein-/Ausgabe mittels Programmunterbrechungen	302
6.2.3	Ein-/Ausgabe mittels DMA	309
6.2.4	Ein-/Ausgabearten im Vergleich	313
6.3	Ein-/Ausgabesystem	314
6.3.1	Treiber	314
6.3.2	Geräteverwaltung	315
6.3.3	Treiberschnittstelle	316
6.3.4	Ein-/Ausgabeschnittstelle	317
6.3.5	Ein-/Ausgabepufferung	321
6.3.6	Treibermodell in Linux	322
6.3.7	Treibermodelle in Windows (WDM & WDF)	329

6.4	Massenspeicher	335
6.4.1	Verfahren der Datenaufzeichnung	335
6.4.2	Wichtigste Massenspeicher	338
6.4.3	Eigenschaften von Plattenspeichern	339
6.4.4	Eigenschaften von Festkörperlaufwerken (SSD)	342
6.4.5	Zugriffsplanung für Plattenspeicher (disk I/O scheduling)	343
6.4.6	Pufferung von Plattendaten (disk cache)	347
6.5	Benutzerinteraktion aus Systemsicht (Benutzeroberflächen)	348
6.5.1	Allgemeines	348
6.5.2	Systemarchitekturen	350
6.5.3	Programmiermodelle	355
6.5.4	Die Unix-Shell als Kommandointerpreter	357
6.5.5	Funktionsweise & Programmierung des X-Window-Systems	359
6.5.6	Funktionsweise und Programmierung des Windows-GUI	373
<b>7</b>	<b>Speicherverwaltung</b>	<b>387</b>
7.1	Speichersystem	388
7.1.1	Einordnung im Rechnermodell	388
7.1.2	Grundlegende Speicherprinzipien	389
7.1.3	Speicherhierarchie & Lokalisierungsprinzip	393
7.1.4	Cache-Funktionsweise	397
7.2	Verwaltungsgrundlagen	402
7.2.1	Abstraktionen mittels Abbildungsfunktionen	403
7.2.2	Aufgaben der Systemsoftware	407
7.2.3	Dynamische Speicherbereitstellung	409
7.3	Verwaltung von Prozessadressräumen	429
7.3.1	Adressraumnutzung durch Programme	429
7.3.2	Adressraumverwaltung durch das Betriebssystem	432
7.4	Realer Speicher	435
7.4.1	Monoprogrammierung	435
7.4.2	Multiprogrammierung mit Partitionen	436
7.4.3	Verfahren für knappen Speicher	440
7.5	Virtueller Speicher	444
7.5.1	Adressumsetzung	445
7.5.2	Seitenwechselfahren (demand paging)	458
7.5.3	Speicherabgebildete Dateien	486
7.5.4	Gemeinsamer Speicher (shared memory)	487

<b>8</b>	<b>Dateisysteme</b>	<b>491</b>
8.1	Aufgaben des Dateisystems	491
8.2	Dateisystemkonzepte	492
8.2.1	Logische Organisation	492
8.2.2	Dateisystemfunktionen	503
8.2.3	Gemeinsame Dateinutzung	518
8.2.4	Speicherabgebildete Dateien	522
8.3	Realisierung von Dateisystemen	522
8.3.1	Konzeptionelles Modell	522
8.3.2	Blockspeicher als Grundlage	523
8.3.3	Organisationsprinzipien	524
8.4	UFS – traditionelles Unix-Dateisystem	531
8.5	FAT– traditionelles Windows-Dateisystem	535
8.5.1	Struktur der Gesamtplatte	536
8.5.2	Aufbau der Belegungstabelle (FAT)	537
8.5.3	Verzeichnisdaten	538
8.5.4	Lange Dateinamen (VFAT)	540
8.6	NTFS – modernes Windows-Dateisystem	542
8.7	Netzwerkdateisysteme	545
8.7.1	Logische Sicht	545
8.7.2	Implementierung	547
8.7.3	NFS – Network File System in Unix	550
8.7.4	SMB – Netzwerkdateisystem in Windows	551
8.8	Spezielle Dateisystemtechnologien	552
8.8.1	Protokollierende Dateisysteme	552
8.8.2	Schattenkopie	554
8.8.3	Disk Scheduling	555
8.9	Festplattenpartitionierung	556
8.9.1	Anwendungsbereiche	556
8.9.2	Festplattenpartitionierung der PC-Systeme	557
<b>9</b>	<b>Programmentwicklung</b>	<b>561</b>
9.1	Software-Entwicklungswerkzeuge	561
9.1.1	Ablauf der Programmübersetzung	562
9.1.2	Darstellung von Übersetzungsvorgängen mittels T-Notation	566
9.1.3	Automatisierte Übersetzung	569
9.1.4	Versionenkontrolle	571

9.2	Adressraumbelegung und Relokation	573
9.2.1	Storage Class	573
9.2.2	Programmorganisation in Sektionen	574
9.2.3	Relokation von Programmen	575
9.3	Programmbibliotheken	582
9.3.1	Grundlagen und Begriffe	582
9.3.2	Programmbibliotheken unter Unix	584
9.3.3	Programmbibliotheken unter Windows	588
9.4	Skriptprogrammierung unter Unix	593
9.4.1	Anwendungsbereiche	594
9.4.2	Die Shell als Programminterpret	595
9.4.3	Portabilität und Kompatibilität	596
9.4.4	Erstellung von Skriptprogrammen	596
9.4.5	Elemente der Skriptsprache	597
9.4.6	Shell-Befehle	597
9.4.7	Shell-Variablen	599
9.4.8	Metazeichen	604
9.4.9	Synonyme und Funktionen	606
9.4.10	Bedingte Tests (conditional tests)	607
9.4.11	Arithmetik	610
9.4.12	Kontrollstrukturen für Skripte	610
<b>10</b>	<b>Sicherheit</b>	<b>617</b>
10.1	Schutzziele	617
10.2	Autorisierung und Zugriffskontrolle	618
10.2.1	Grundlagen und Begriffe	619
10.2.2	Schutzdomänenkonzept	621
10.2.3	Schutzstrategien	629
10.3	Hochsichere Betriebssysteme	631
10.4	Sicherheit unter Unix	632
10.5	Sicherheit unter Windows	636
<b>11</b>	<b>Spezielle Technologien</b>	<b>639</b>
11.1	Multiprozessorsysteme	639
11.1.1	Einprozessorsysteme	640
11.1.2	Multiprozessorsysteme (Klassifikation nach Flynn)	642
11.1.3	Rechenleistung paralleler Prozessoren (Amdahl's Law)	645
11.1.4	Strukturen von Multiprozessorsystemen	648
11.1.5	Leistungsbewertung von Computern	654

11.2	Handheld-Computing	655
11.2.1	Allgemeines	656
11.2.2	Spezielle Anforderungen an das Betriebssystem	657
11.2.3	Windows CE	659
11.2.4	Embedded Linux	663
11.2.5	Android, Maemo und WebOS	663
11.2.6	Symbian OS	665
11.3	Virtualisierungskonzepte und ihre Anwendung	670
11.3.1	Anwendungsbereiche	670
11.3.2	Virtuelle Prozessoren	671
11.3.3	Virtuelle Prozessumgebungen	671
11.3.4	Virtuelle Ressourcen	672
11.3.5	Virtuelle Computer (Stufe Computerhardware)	672
11.3.6	Virtuelle Sekundärspeicher (storage virtualization)	676
11.3.7	Virtuelle Netzwerke (network virtualization)	676
<b>A</b>	<b>Anhang</b>	<b>677</b>
A.1	Maßeinheiten und Darstellungen	677
A.1.1	Maßeinheiten in der Informatik	677
A.1.2	Darstellung von Bitmustern	678
A.1.3	Oktal- und Hexadezimalzahlen	678
A.1.4	Kennzeichnung der Zahlensysteme	679
A.1.5	Rechnerinterne Zahlendarstellungen	679
A.1.6	Textzeichensätze	683
A.2	Instruktionssatz der Intel iA32-Prozessoren	688
	<b>Literaturhinweise</b>	<b>693</b>
	<b>Index</b>	<b>699</b>