

# Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b>	<b>1</b>
1.1	Zweck .....	1
1.2	Definitionen .....	3
1.3	Einordnung im Computersystem .....	5
1.4	Betriebssystemarten .....	6
1.4.1	Klassische Einteilungen .....	7
1.4.2	Moderne Einteilungen .....	7
1.4.3	Geschichte .....	8
1.5	Betriebssystemarchitekturen .....	9
1.5.1	Architekturformen .....	9
1.5.2	Benutzer-/Kernmodus .....	10
1.5.3	Monolithische Systeme .....	12
1.5.4	Geschichtete Systeme .....	13
1.5.5	Mikrokernsysteme (Client/Server-Modell) .....	14
1.5.6	Multiprozessorsysteme .....	15
1.5.7	Verteilte Betriebssysteme .....	16
1.5.8	Beispiele von Systemarchitekturen .....	17
1.5.9	Zukünftige Systemarchitekturen aus Sicht der Forschung ..	20
<b>2</b>	<b>Programmausführung und Hardware</b>	<b>25</b>
2.1	Rechner- und Prozessorgrundlagen .....	26
2.1.1	Grundmodell eines Rechners .....	26
2.1.2	Befehlsverarbeitung in der CPU .....	29
2.1.3	Prozessoraufbau .....	30
2.1.4	Allgemeine Prozessorregister (general purpose registers) ...	31
2.1.5	Steuerregister (control registers) .....	32

2.2	Grundlagen des Adressraums . . . . .	33
2.2.1	Adressraumtypen . . . . .	35
2.2.2	Bytereihenfolge (byte ordering) . . . . .	36
2.2.3	Adressraumbelungsplan (memory map) . . . . .	37
2.2.4	Ausrichtungsregeln im Adressraum . . . . .	39
2.2.5	Adressraumbelung durch Programme . . . . .	40
2.2.6	Adressraumnutzung durch C-Programme . . . . .	41
2.3	Grundlagen der Programmausführung . . . . .	44
2.3.1	Quell- und Binärcode . . . . .	44
2.3.2	Programmausführung und Programmzähler (PC) . . . . .	48
2.3.3	Funktionsweise des Stapels und Stapelzeigers (SP) . . . . .	49
2.3.4	Funktion des Programmstatusworts (PSW) . . . . .	52
2.3.5	Programmunterbrechungen (interrupts) . . . . .	52
2.3.6	Privilegierte Programmausführung (Benutzer-/Kernmodus) . . . . .	55
2.4	Unterprogrammmechanismen . . . . .	57
2.4.1	Unterprogrammaufruf und Kompletierung . . . . .	58
2.4.2	Formen des Unterprogrammaufrufs . . . . .	60
2.4.3	Parameterübergabe beim Unterprogrammaufruf . . . . .	62
2.4.4	Realisierung der Parameterübergabe und lokale Variablen . . . . .	64
<b>3</b>	<b>Systemprogrammierung</b>	<b>71</b>
3.1	Wahl der Systemprogrammiersprache . . . . .	72
3.1.1	Mischsprachenprogrammierung . . . . .	72
3.1.2	Programmiersprache C++ . . . . .	72
3.1.3	Java Native Interface (JNI) . . . . .	73
3.1.4	Microsoft .NET-Sprachen . . . . .	77
3.2	Laufzeitsystem der Programmiersprache C . . . . .	79
3.3	Unterprogrammtechniken . . . . .	80
3.3.1	Formale und aktuelle Parameter . . . . .	80
3.3.2	Idempotente Unterprogramme . . . . .	80
3.4	Grundlagen der Systemprogrammierung . . . . .	81
3.4.1	Dienstanforderung und Erbringung . . . . .	82
3.4.2	Dienstparameter und Resultate . . . . .	83
3.4.3	Umgebungsvariablenliste (environment list) . . . . .	87
3.4.4	Dateideskriptoren & Handles . . . . .	89
3.4.5	Systemdatentypen . . . . .	92
3.4.6	Anfangsparameter für Prozesse . . . . .	94
3.4.7	Beendigungsstatus von Programmen . . . . .	94
3.4.8	Fehlerbehandlung . . . . .	95
3.4.9	Programmierung für 32- und 64-Bit-Systeme . . . . .	99

3.5	Systemprogrammierschnittstellen	100
3.5.1	Aufrufverfahren	100
3.5.2	Unix-Programmierschnittstelle	103
3.5.3	Windows-Programmierschnittstelle	103
<b>4</b>	<b>Prozesse und Threads</b>	<b>105</b>
4.1	Parallelverarbeitung	106
4.1.1	Darstellung paralleler Abläufe	106
4.1.2	Hardware-Parallelität	107
4.1.3	Software-Parallelität	107
4.1.4	Begriffe	108
4.2	Prozessmodell	111
4.2.1	Grundprinzip	111
4.2.2	Prozesserzeugung und Terminierung	114
4.2.3	Prozesse unter Unix	118
4.2.4	Funktionsweise der Unix-Shell	123
4.2.5	Prozesse & Jobs unter Windows	126
4.2.6	Vererbung unter Prozessen	128
4.2.7	Systemstart und Prozesshierarchie	128
4.2.8	Ausführungsmodelle für Betriebssysteme	134
4.3	Threads	135
4.3.1	Thread-Modell	136
4.3.2	Vergleich Prozesse zu Threads	136
4.3.3	Implementierung des Multithreading	139
4.3.4	Windows Threads, Fibers und Services	144
4.3.5	Services	149
4.3.6	Threads unter Unix	151
4.3.7	Thread-Pool-Konzept	153
4.3.8	Anwendungsprobleme	154
4.4	Prozessorzuteilungsstrategien	154
4.4.1	Quasiparallelität im Einprozessorsystem	154
4.4.2	Prozess- und Thread-Zustände	155
4.4.3	Konzeptionelle Prozessverwaltung	158
4.4.4	Zuteilungsstrategien	160
4.4.5	Multiprozessor-Scheduling	174
4.4.6	POSIX-Thread-Scheduling	175
4.4.7	Java-Thread-Scheduling	178
4.4.8	Scheduling unter Windows	179
4.4.9	Scheduling unter Unix	186

<b>5</b>	<b>Synchronisation von Prozessen und Threads</b>	<b>195</b>
5.1	Synchronisationsbedarfe und Lösungsansätze	196
5.1.1	Problem der Ressourcenteilung	196
5.1.2	Verlorene Aktualisierung (lost update problem)	197
5.1.3	Inkonsistente Abfrage (inconsistent read)	198
5.1.4	Absicherung mit Selbstverwaltung – naiver Ansatz	199
5.1.5	Absicherung mit Selbstverwaltung – korrekter Ansatz	201
5.1.6	Absicherung mit Systemmitteln	203
5.2	Semaphore	203
5.2.1	Semaphortypen	205
5.2.2	Implementierungsfragen	205
5.3	Anwendung der Semaphore	208
5.3.1	Absicherung kritischer Bereiche (mutual exclusion)	208
5.3.2	Synchronisation von Abläufen (barrier synchronization)	209
5.3.3	Produzenten & Konsumenten (producer and consumer)	211
5.3.4	Leser & Schreiber (readers and writers)	214
5.3.5	Problem der Prioritätsumkehrung (priority inversion)	220
5.3.6	Weitere Anwendungsprobleme	222
5.4	Implementierungen von Semaphoren	222
5.4.1	Semaphore unter Unix	223
5.4.2	Semaphore unter Windows	226
5.5	Unix-Signale	233
5.5.1	Idee & Grundprinzip der Unix-Signale	233
5.5.2	Programmierung der Signale	236
5.5.3	Signale im Multithreading	240
5.5.4	Realtime-Signale	241
5.6	Verklemmungsproblematik (deadlocks)	242
5.6.1	Ursache	242
5.6.2	Deadlock-Bedingungen	246
5.6.3	Lösungsansätze und ihre Beurteilung	246
5.7	Praktische Erwägungen zur Parallelprogrammierung	254
5.7.1	Grenzen der Leistungssteigerung (Amdahl's Law)	254
5.7.2	Korrektheitsbedingungen der Parallelität	257
5.7.3	Vermeidung von Synchronisationsengpässen	257
5.7.4	Speicherkonsistenz (memory consistency)	260

<b>6</b>	<b>Kommunikation von Prozessen und Threads</b>	<b>263</b>
6.1	Überblick über Synchronisation und Kommunikation	264
6.2	Nachrichtenbasierte Verfahren	265
6.2.1	Allgemeine Aspekte	265
6.2.2	Unix-Pipes	271
6.2.3	Windows-Pipes	279
6.2.4	Unix Message Queues	283
6.2.5	Windows-Messages	285
6.2.6	Windows-Mailslots	287
6.3	Speicherbasierte Verfahren	289
6.3.1	Gemeinsamer Speicher unter Windows	290
6.3.2	Gemeinsamer Speicher unter Unix	291
6.4	Monitor	292
6.4.1	Grundprinzip	292
6.4.2	Java-Monitor	295
6.4.3	Monitornachbildung mit Bedingungsvariablen	296
6.5	Rendezvous	302
6.5.1	Grundprinzip	302
6.5.2	Synchronisation in Client/Server-Systemen (barber shop)	303
6.6	Rechnerübergreifende Interprozesskommunikation	305
6.6.1	Netzwerksoftware	305
6.6.2	Berkeley-Sockets	307
6.6.3	Remote Procedure Call (RPC)	314
6.6.4	Überblick über Middleware	320
<b>7</b>	<b>Ein- und Ausgabe</b>	<b>323</b>
7.1	Peripherie	324
7.1.1	Einordnung im Rechnermodell	324
7.1.2	Begriffsdefinitionen	324
7.2	Ein-/Ausgabeabläufe	325
7.2.1	Programmgesteuerte Ein-/Ausgabe	325
7.2.2	Ein-/Ausgabe mittels Programmunterbrechungen	326
7.2.3	Ein-/Ausgabe mittels DMA	327
7.2.4	Ein-/Ausgabearten im Vergleich	331

7.3	Ein-/Ausgabesystem	331
7.3.1	Treiber	332
7.3.2	Geräteverwaltung	333
7.3.3	Treiberschnittstelle	333
7.3.4	Ein-/Ausgabeschnittstelle	335
7.3.5	Ein-/Ausgabepufferung	338
7.3.6	Treibermodell in Linux	340
7.3.7	Treibermodelle in Windows (WDM & WDF)	347
7.4	Massenspeicher	353
7.4.1	Wichtigste Massenspeicher	353
7.4.2	Eigenschaften von Festplattenlaufwerken (HDD)	354
7.4.3	Eigenschaften von Festkörperlaufwerken (SSD)	356
7.4.4	Speicher-Anschlussmöglichkeiten	357
7.4.5	Pufferung von Zugriffsdaten (disk cache)	359
7.4.6	Speicher-Virtualisierung durch RAID	360
7.5	Benutzerinteraktion aus Systemsicht (Benutzeroberflächen)	363
7.5.1	Allgemeines	363
7.5.2	Systemarchitekturen	365
7.5.3	Programmiermodelle	370
7.5.4	Die Unix-Shell als Kommandointerpreter	372
7.5.5	Funktionsweise und Programmierung des X-Window-Systems	374
7.5.6	Funktionsweise und Programmierung des Windows-GUI	388
<b>8</b>	<b>Speicherverwaltung</b>	<b>403</b>
8.1	Speichersystem	404
8.1.1	Einordnung im Rechnermodell	404
8.1.2	Grundlegende Speicherprinzipien	405
8.1.3	Speicherhierarchie & Lokalitätsprinzip	407
8.1.4	Cache-Funktionsweise	410
8.2	Dynamische Speicherbereitstellung (Heap)	416
8.2.1	Verwaltungsalgorithmen	418
8.2.2	Grundprinzip der Speichervorgabe	420
8.2.3	Übersicht Implementierungsvarianten	424
8.2.4	Variante A: Variable Zuordnungsgröße	424
8.2.5	Variante B: Feste Blockgrößen bzw. Größenklassen	426
8.2.6	Variante C: Mehrfache einer festen Blockgröße	428
8.2.7	Variante D: Buddy-System	429
8.2.8	Heap-Erweiterung	433
8.2.9	Heap-Management in Windows	434

8.3	Verwaltung von Prozessadressräumen	437
8.3.1	Adressraumnutzung durch Programme	437
8.3.2	Adressraumverwaltung durch das Betriebssystem	439
8.4	Realer Speicher	442
8.4.1	Monoprogrammierung	442
8.4.2	Multiprogrammierung mit Partitionen	443
8.4.3	Verfahren für knappen Speicher	447
8.5	Virtueller Speicher	452
8.5.1	Adressumsetzung	453
8.5.2	Seitenwechselverfahren (demand paging)	466
8.5.3	Speicherabgebildete Dateien	496
8.5.4	Gemeinsamer Speicher (shared memory)	496
<b>9</b>	<b>Dateisysteme</b>	<b>499</b>
9.1	Dateisystemkonzepte	500
9.1.1	Logische Organisation	500
9.1.2	Dateisystemfunktionen	511
9.1.3	Gemeinsame Dateinutzung	522
9.1.4	Speicherabgebildete Dateien	526
9.2	Realisierung von Dateisystemen	527
9.2.1	Konzeptionelles Modell	527
9.2.2	Blockspeicher als Grundlage	527
9.2.3	Organisationsprinzipien	528
9.3	UFS – traditionelles Unix-Dateisystem	535
9.3.1	Datenträgeraufteilung	536
9.3.2	Dateihaltung und Verzeichnisorganisation	537
9.3.3	Index Nodes (Inodes)	537
9.4	FAT– traditionelles Windows-Dateisystem	539
9.4.1	Datenträgeraufteilung	541
9.4.2	Aufbau der Belegungstabelle (FAT)	541
9.4.3	Verzeichnisdaten	542
9.5	NTFS – modernes Windows-Dateisystem	545
9.5.1	Entstehung und Eigenschaften	545
9.5.2	Logische Struktur und Inhalt einer NTFS-Partition	545
9.5.3	NTFS-Streams	547
9.5.4	Dateispeicherung	548
9.5.5	Dateiverzeichnisse	549

9.6	ZFS – zukunftsweisendes Dateisystem . . . . .	549
9.6.1	Datenträgerverwaltung . . . . .	549
9.6.2	Datenintegrität . . . . .	550
9.6.3	Pufferung und Deduplizierung . . . . .	551
9.6.4	Interoperabilität . . . . .	551
9.7	Netzwerkdateisysteme . . . . .	551
9.7.1	Logische Sicht . . . . .	551
9.7.2	Implementierung . . . . .	553
9.7.3	NFS – Network File System in Unix . . . . .	556
9.7.4	SMB – Netzwerkdateisystem in Windows . . . . .	557
9.8	Spezielle Dateisystemtechnologien . . . . .	558
9.8.1	Protokollierende Dateisysteme . . . . .	558
9.8.2	Schattenkopie . . . . .	560
9.8.3	Disk Scheduling . . . . .	561
9.9	Datenträgerpartitionierung . . . . .	562
9.9.1	Anwendungsbereiche . . . . .	562
9.9.2	Master Boot Record (MBR) . . . . .	563
9.9.3	GUID Partition Table (GPT) . . . . .	564
<b>10</b>	<b>Programmentwicklung</b>	<b>567</b>
10.1	Software-Entwicklungswerkzeuge . . . . .	568
10.1.1	Ablauf der Programmübersetzung . . . . .	569
10.1.2	Darstellung von Übersetzungsvorgängen mittels T-Notation . . . . .	574
10.1.3	Automatisierte Übersetzung . . . . .	576
10.1.4	Versionsverwaltung . . . . .	578
10.2	Adressraumbelugung und Relokation . . . . .	581
10.2.1	Storage Class . . . . .	581
10.2.2	Programmorganisation in Sektionen . . . . .	582
10.2.3	Relokation von Programmen . . . . .	583
10.3	Programmbibliotheken . . . . .	590
10.3.1	Grundlagen und Begriffe . . . . .	590
10.3.2	Anwendungsbereiche . . . . .	593
10.3.3	Programmbibliotheken unter Unix . . . . .	594
10.3.4	Programmbibliotheken unter Windows . . . . .	598



10.4	Skriptprogrammierung unter Unix	604
10.4.1	Anwendungsbereiche	604
10.4.2	Die Shell als Programminterpret	605
10.4.3	Portabilität und Kompatibilität	606
10.4.4	Erstellung von Skriptprogrammen	607
10.4.5	Ausführung von Skriptprogrammen	607
10.4.6	Elemente der Skriptsprache	608
10.4.7	Shell-Befehle	608
10.4.8	Shell-Variablen	610
10.4.9	Stringoperatoren für Shell-Variable	615
10.4.10	Metazeichen	617
10.4.11	Synonyme und Funktionen	621
10.4.12	Bedingte Tests (conditional tests)	621
10.4.13	Arithmetik	625
10.4.14	Kontrollstrukturen für Skripte	626
10.5	Anwendungs- und Systemkonfiguration	632
10.5.1	Konfiguration mit Textdateien	632
10.5.2	Konfiguration mit Registrierungsdatenbank	633
<b>11</b>	<b>Sicherheit</b>	<b>637</b>
11.1	Schutzziele	637
11.2	Autorisierung und Zugriffskontrolle	639
11.2.1	Grundlagen und Begriffe	639
11.2.2	Schutzdomänenkonzept	641
11.2.3	Schutzstrategien	649
11.3	Hochsichere Betriebssysteme	651
11.4	Sicherheit unter Unix	652
11.5	Sicherheit unter Windows	656
<b>12</b>	<b>Virtualisierung</b>	<b>659</b>
12.1	Anwendungsbereiche	659
12.2	Virtualisierungstypen	660
12.2.1	Virtuelle Prozessoren	660
12.2.2	Virtuelle Prozessumgebungen	661
12.2.3	Virtuelles Betriebssystem	661
12.2.4	Virtueller Desktop	662
12.2.5	Virtuelle Ressourcen	662
12.2.6	Sandboxing (virtuelles Laufzeitsystem)	663
12.2.7	Virtuelle Computer (Stufe Computerhardware)	664

12.3	Virtual Machine Monitor bzw. Hypervisor .....	665
12.3.1	Anforderungen .....	665
12.3.2	VMM-Funktionsweise .....	665
12.3.3	VMM-Typen .....	668
12.3.4	Unikernel .....	670
12.4	Einsatzgebiete .....	670
<b>13</b>	<b>Mobile Betriebssysteme</b>	<b>675</b>
13.1	Gemeinsame Eigenschaften .....	675
13.1.1	Anforderungen durch die Plattform .....	675
13.1.2	Middleware als Betriebssystem .....	676
13.2	Google Android .....	678
13.2.1	Überblick .....	678
13.2.2	Architektur .....	679
13.2.3	System- und Applikationsstart .....	679
13.2.4	Lebenszyklus von Applikationen .....	680
13.2.5	Nachrichtensystem .....	681
13.3	Apple iOS .....	682
<b>A</b>	<b>Anhang</b>	<b>685</b>
A.1	Maßeinheiten und Darstellungen .....	685
A.1.1	Maßeinheiten in der Informatik .....	685
A.1.2	Darstellung von Bitmustern .....	686
A.1.3	Oktal- und Hexadezimalzahlen .....	686
A.1.4	Kennzeichnung der Zahlensysteme .....	687
A.1.5	Rechnerinterne Zahlendarstellungen .....	687
A.1.6	Textzeichensätze .....	691
	<b>Literaturhinweise</b>	<b>697</b>
	<b>Index</b>	<b>703</b>