

# 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 .....	17
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 .....	31
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	Adressraumbelegungsplan (memory map) .....	37
2.2.4	Ausrichtungsregeln im Adressraum .....	39
2.2.5	Adressraumbelegung durch Programme .....	40
2.2.6	Adressraumnutzung durch C-Programme .....	42
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 .....	133
4.3	Threads .....	135
4.3.1	Thread-Modell .....	135
4.3.2	Vergleich Prozesse zu Threads .....	136
4.3.3	Implementierung des Multithreading .....	138
4.3.4	Threads & Fibers unter Windows .....	143
4.3.5	Threads unter Unix .....	148
4.3.6	Anwendungsprobleme .....	150
4.3.7	Task-Konzept .....	150
4.4	Prozessorzuteilungsstrategien .....	151
4.4.1	Quasiparallelität im Einprozessorsystem .....	151
4.4.2	Prozess- und Thread-Zustände .....	152
4.4.3	Konzeptionelle Prozessverwaltung .....	154
4.4.4	Zuteilungsstrategien .....	156
4.4.5	Multiprozessor-Scheduling .....	170
4.4.6	POSIX-Thread-Scheduling .....	171
4.4.7	Java-Thread-Scheduling .....	174
4.4.8	Scheduling unter Windows .....	175
4.4.9	Scheduling unter Unix .....	182

<b>5</b>	<b>Synchronisation von Prozessen und Threads</b>	<b>189</b>
5.1	Synchronisationsbedarfe und Lösungsansätze . . . . .	190
5.1.1	Problem der Ressourcenteilung . . . . .	190
5.1.2	Verlorene Aktualisierung (lost update problem) . . . . .	190
5.1.3	Inkonsistente Abfrage (inconsistent read) . . . . .	192
5.1.4	Absicherung mit Selbstverwaltung – naiver Ansatz . . . . .	193
5.1.5	Absicherung mit Selbstverwaltung – korrekter Ansatz . . . . .	195
5.1.6	Absicherung mit Systemmitteln . . . . .	197
5.2	Semaphore . . . . .	197
5.2.1	Semaphortypen . . . . .	198
5.2.2	Implementierungsfragen . . . . .	199
5.3	Anwendung der Semaphore . . . . .	202
5.3.1	Absicherung kritischer Bereiche (mutual exclusion) . . . . .	202
5.3.2	Synchronisation von Abläufen (barrier synchronization) . . . . .	203
5.3.3	Produzenten & Konsumenten (producer and consumer) . . . . .	205
5.3.4	Leser & Schreiber (readers and writers) . . . . .	208
5.3.5	Problem der Prioritätsumkehrung (priority inversion) . . . . .	213
5.3.6	Weitere Anwendungsprobleme . . . . .	215
5.4	Implementierungen von Semaphoren . . . . .	216
5.4.1	Semaphore unter Unix . . . . .	216
5.4.2	Semaphore unter Windows . . . . .	220
5.5	Unix-Signale . . . . .	226
5.5.1	Idee & Grundprinzip der Unix-Signale . . . . .	226
5.5.2	Programmierung der Signale . . . . .	229
5.5.3	Signale im Multithreading . . . . .	234
5.5.4	Realtime-Signale . . . . .	235
5.6	Verklemmungsproblematik (deadlocks) . . . . .	235
5.6.1	Ursache . . . . .	236
5.6.2	Deadlock-Bedingungen . . . . .	239
5.6.3	Lösungsansätze und ihre Beurteilung . . . . .	240
5.7	Vermeidung von Synchronisationsengpässen . . . . .	247
5.7.1	Granularität der Absicherung . . . . .	248
5.7.2	Replikation der abgesicherten Ressource . . . . .	249

<b>6</b>	<b>Kommunikation von Prozessen und Threads</b>	<b>251</b>
6.1	Überblick über Synchronisation und Kommunikation	251
6.2	Nachrichtenbasierte Verfahren	253
6.2.1	Allgemeine Aspekte	253
6.2.2	Unix-Pipes	259
6.2.3	Windows-Pipes	267
6.2.4	Unix Message Queues	271
6.2.5	Windows-Messages	273
6.2.6	Windows-Mailslots	275
6.3	Speicherbasierte Verfahren	277
6.3.1	Gemeinsamer Speicher unter Windows	278
6.3.2	Gemeinsamer Speicher unter Unix	279
6.4	Monitor	280
6.4.1	Grundprinzip	280
6.4.2	Java-Monitor	283
6.4.3	Monitornachbildung mit Bedingungsvariablen	284
6.5	Rendezvous	290
6.5.1	Grundprinzip	290
6.5.2	Synchronisation in Client/Server-Systemen (barber shop)	291
6.6	Rechnerübergreifende Interprozesskommunikation	292
6.6.1	Netzwerksoftware	293
6.6.2	Berkeley-Sockets	295
6.6.3	Remote Procedure Call (RPC)	302
6.6.4	Überblick über Middleware	308
<b>7</b>	<b>Ein- und Ausgabe</b>	<b>311</b>
7.1	Peripherie	312
7.1.1	Einordnung im Rechnermodell	312
7.1.2	Begriffsdefinitionen	312
7.2	Ein-/Ausgabeabläufe	313
7.2.1	Programmgesteuerte Ein-/Ausgabe	313
7.2.2	Ein-/Ausgabe mittels Programmunterbrechungen	314
7.2.3	Ein-/Ausgabe mittels DMA	315
7.2.4	Ein-/Ausgabearten im Vergleich	319
7.3	Ein-/Ausgabesystem	319
7.3.1	Treiber	320
7.3.2	Geräteverwaltung	321

7.3.3	Treiberschnittstelle . . . . .	322
7.3.4	Ein-/Ausgabeschnittstelle . . . . .	323
7.3.5	Ein-/Ausgabepufferung . . . . .	327
7.3.6	Treibermodell in Linux . . . . .	329
7.3.7	Treibermodelle in Windows (WDM & WDF) . . . . .	335
7.4	Massenspeicher . . . . .	341
7.4.1	Wichtigste Massenspeicher . . . . .	342
7.4.2	Eigenschaften von Festplattenlaufwerken . . . . .	342
7.4.3	Eigenschaften von Festkörperlaufwerken (SSD) . . . . .	343
7.4.4	Zugriffsplanung für Plattenspeicher (disk I/O scheduling) . . . . .	344
7.4.5	Pufferung von Plattendaten (disk cache) . . . . .	348
7.5	Benutzerinteraktion aus Systemsicht (Benutzeroberflächen) . . . . .	350
7.5.1	Allgemeines . . . . .	350
7.5.2	Systemarchitekturen . . . . .	352
7.5.3	Programmiermodelle . . . . .	357
7.5.4	Die Unix-Shell als Kommandointerpreter . . . . .	359
7.5.5	Funktionsweise und Programmierung des X-Window-Systems . . . . .	361
7.5.6	Funktionsweise und Programmierung des Windows-GUI . . . . .	374
<b>8</b>	<b>Speicherverwaltung</b>	<b>389</b>
8.1	Speichersystem . . . . .	390
8.1.1	Einordnung im Rechnermodell . . . . .	390
8.1.2	Grundlegende Speicherprinzipien . . . . .	391
8.1.3	Speicherhierarchie & Lokalitätsprinzip . . . . .	393
8.1.4	Cache-Funktionsweise . . . . .	396
8.2	Dynamische Speicherbereitstellung (Heap) . . . . .	402
8.2.1	Verwaltungsalgorithmen . . . . .	404
8.2.2	Grundprinzip der Speichertzuteilung . . . . .	405
8.2.3	Übersicht Implementierungsvarianten . . . . .	410
8.2.4	Variante A: Variable Zuordnungsgröße . . . . .	410
8.2.5	Variante B: Feste Blockgrößen bzw. Größenklassen . . . . .	412
8.2.6	Variante C: Mehrfache einer festen Blockgröße . . . . .	413
8.2.7	Variante D: Buddy-System . . . . .	415
8.2.8	Heap-Erweiterung . . . . .	419
8.2.9	Heap-Management in Windows . . . . .	420
8.3	Verwaltung von Prozessadressräumen . . . . .	423
8.3.1	Adressraumnutzung durch Programme . . . . .	423
8.3.2	Adressraumverwaltung durch das Betriebssystem . . . . .	425

8.4	Realer Speicher	428
8.4.1	Monoprogrammierung	428
8.4.2	Multiprogrammierung mit Partitionen	429
8.4.3	Verfahren für knappen Speicher	433
8.5	Virtueller Speicher	438
8.5.1	Adressumsetzung	439
8.5.2	Seitenwechselferfahren (demand paging)	452
8.5.3	Speicherabgebildete Dateien	481
8.5.4	Gemeinsamer Speicher (shared memory)	482
<b>9</b>	<b>Dateisysteme</b>	<b>485</b>
9.1	Dateisystemkonzepte	486
9.1.1	Logische Organisation	486
9.1.2	Dateisystemfunktionen	497
9.1.3	Gemeinsame Dateinutzung	508
9.1.4	Speicherabgebildete Dateien	512
9.2	Realisierung von Dateisystemen	513
9.2.1	Konzeptionelles Modell	513
9.2.2	Blockspeicher als Grundlage	513
9.2.3	Organisationsprinzipien	514
9.3	UFS – traditionelles Unix-Dateisystem	521
9.3.1	Datenträgeraufteilung	522
9.3.2	Dateihaltung und Verzeichnisorganisation	523
9.3.3	Index Nodes (Inodes)	523
9.4	FAT– traditionelles Windows-Dateisystem	525
9.4.1	Datenträgeraufteilung	527
9.4.2	Aufbau der Belegungstabelle (FAT)	527
9.4.3	Verzeichnisdaten	528
9.5	NTFS – modernes Windows-Dateisystem	531
9.5.1	Entstehung und Eigenschaften	531
9.5.2	Logische Struktur und Inhalt einer NTFS-Partition	531
9.5.3	NTFS-Streams	533
9.5.4	Dateispeicherung	534
9.5.5	Dateiverzeichnisse	535
9.6	ZFS – zukunftsweisendes Dateisystem	535
9.6.1	Datenträgerverwaltung	535
9.6.2	Datenintegrität	536
9.6.3	Pufferung und Deduplizierung	537
9.6.4	Interoperabilität	537

9.7	Netzwerkdateisysteme .....	537
9.7.1	Logische Sicht .....	537
9.7.2	Implementierung .....	539
9.7.3	NFS – Network File System in Unix .....	542
9.7.4	SMB – Netzwerkdateisystem in Windows .....	543
9.8	Spezielle Dateisystemtechnologien .....	544
9.8.1	Protokollierende Dateisysteme .....	544
9.8.2	Schattenkopie .....	546
9.8.3	Disk Scheduling .....	547
9.9	Datenträgerpartitionierung .....	548
9.9.1	Anwendungsbereiche .....	548
9.9.2	Master Boot Record (MBR) .....	549
9.9.3	GUID Partition Table (GPT) .....	550
<b>10</b>	<b>Programmentwicklung</b> .....	<b>553</b>
10.1	Software-Entwicklungswerkzeuge .....	554
10.1.1	Ablauf der Programmübersetzung .....	555
10.1.2	Darstellung von Übersetzungsvorgängen mittels T-Notation .....	560
10.1.3	Automatisierte Übersetzung .....	562
10.1.4	Versionsverwaltung .....	564
10.2	Adressraumbelegung und Relokation .....	567
10.2.1	Storage Class .....	567
10.2.2	Programmorganisation in Sektionen .....	568
10.2.3	Relokation von Programmen .....	569
10.3	Programmbibliotheken .....	576
10.3.1	Grundlagen und Begriffe .....	576
10.3.2	Anwendungsbereiche .....	579
10.3.3	Programmbibliotheken unter Unix .....	579
10.3.4	Programmbibliotheken unter Windows .....	584
10.4	Skriptprogrammierung unter Unix .....	590
10.4.1	Anwendungsbereiche .....	590
10.4.2	Die Shell als Programminterpret .....	591
10.4.3	Portabilität und Kompatibilität .....	592
10.4.4	Erstellung von Skriptprogrammen .....	592
10.4.5	Ausführung von Skriptprogrammen .....	593
10.4.6	Elemente der Skriptsprache .....	594
10.4.7	Shell-Befehle .....	594
10.4.8	Shell-Variablen .....	596



10.4.9	Stringoperatoren für Shell-Variable	601
10.4.10	Metazeichen	603
10.4.11	Synonyme und Funktionen	607
10.4.12	Bedingte Tests (conditional tests)	607
10.4.13	Arithmetik	611
10.4.14	Kontrollstrukturen für Skripte	612
<b>11</b>	<b>Sicherheit</b>	<b>619</b>
11.1	Schutzziele	619
11.2	Autorisierung und Zugriffskontrolle	620
11.2.1	Grundlagen und Begriffe	621
11.2.2	Schutzdomänenkonzept	623
11.2.3	Schutzstrategien	631
11.3	Hochsichere Betriebssysteme	633
11.4	Sicherheit unter Unix	634
11.5	Sicherheit unter Windows	638
<b>12</b>	<b>Virtualisierung</b>	<b>641</b>
12.1	Anwendungsbereiche	641
12.2	Virtualisierungstypen	642
12.2.1	Virtuelle Prozessoren	642
12.2.2	Virtuelle Prozessumgebungen	643
12.2.3	Virtuelles Betriebssystem	643
12.2.4	Virtueller Desktop	643
12.2.5	Virtuelle Ressourcen	644
12.2.6	Sandboxing (virtuelles Laufzeitsystem)	645
12.2.7	Virtuelle Computer (Stufe Computerhardware)	646
12.3	Virtual Machine Monitor bzw. Hypervisor	646
12.3.1	Anforderungen	646
12.3.2	VMM-Funktionsweise	647
12.3.3	VMM-Typen	650
12.4	Einsatzgebiete	651

<b>13</b>	<b>Mobile Betriebssysteme</b>	<b>655</b>
13.1	Gemeinsame Eigenschaften .....	655
13.1.1	Anforderungen durch die Plattform .....	655
13.1.2	Middleware als Betriebssystem .....	656
13.2	Google Android .....	658
13.2.1	Überblick .....	658
13.2.2	Architektur .....	659
13.2.3	System- und Applikationsstart .....	660
13.2.4	Lebenszyklus von Applikationen .....	660
13.2.5	Nachrichtensystem .....	661
13.3	Apple iOS .....	662
13.4	Microsoft Windows Phone 8 .....	663
13.5	Mozilla Firefox OS .....	664
<b>A</b>	<b>Anhang</b>	<b>667</b>
A.1	Maßeinheiten und Darstellungen .....	667
A.1.1	Maßeinheiten in der Informatik .....	667
A.1.2	Darstellung von Bitmustern .....	668
A.1.3	Oktal- und Hexadezimalzahlen .....	668
A.1.4	Kennzeichnung der Zahlensysteme .....	669
A.1.5	Rechnerinterne Zahlendarstellungen .....	669
A.1.6	Textzeichensätze .....	673
A.2	Instruktionssatz der Intel x86-Prozessoren .....	678
	<b>Literaturhinweise</b>	<b>683</b>
	<b>Index</b>	<b>689</b>