

# Inhaltsverzeichnis

1 Einführung .....	1
1.1 Computersysteme .....	1
1.1.1 Einführung .....	2
1.1.2 Aufgabe von Betriebssystemen .....	3
1.1.3 Grundlegende Hardwaremodelle .....	3
1.1.4 CPU-Registersatz .....	7
1.1.5 Multicore-Prozessoren und Hyperthreading-CPU's .....	9
1.1.6 Einfaches Modell der Hardware für die weitere Betrachtung .....	10
1.1.7 Beispiele für Mikroprozessor-Architekturen .....	12
1.2 Entwicklung von Betriebssystemen.....	16
1.2.1 Betriebssystemkategorien.....	16
1.2.2 Universalbetriebssysteme.....	17
1.2.3 Historische Entwicklung .....	17
1.2.4 Geschichte von Microsoft Windows .....	19
1.2.5 Geschichte von Unix .....	21
1.2.6 Geschichte von Linux.....	23
1.3 Übungsaufgaben .....	23
2 Betriebssystemarchitekturen und Betriebsarten .....	25
2.1 Zugriffsschutz in Betriebssystemen.....	25
2.2 Betriebssystemarchitekturen .....	27
2.2.1 Klassische Architekturen .....	27
2.2.2 Mikrokern-Architektur .....	29
2.2.3 Architekturbeispiele .....	30
2.3 Klassische Großrechnerbetriebsarten .....	35
2.3.1 Multiprogramming, Multiprocessing und Multitasking .....	35
2.3.2 Batchverarbeitung und interaktive Verarbeitung .....	36
2.3.3 Teilnehmerbetrieb.....	37
2.3.4 Teilhaberbetrieb .....	38
2.4 Terminalserver-Betrieb .....	40
2.5 Verteilte Verarbeitung .....	42
2.5.1 Echt verteilte Betriebssysteme .....	42
2.5.2 Client-/Server-Systeme.....	42

2.5.3	Peer-to-Peer-Systeme.....	43
2.5.4	Kommunikations-Middleware.....	44
2.5.5	Application-Server-Betrieb.....	45
2.6	Virtualisierung von Betriebs- und Laufzeitsystemen.....	47
2.7	Cloud Computing.....	48
2.8	Übungsaufgaben.....	49
3	Interruptverarbeitung.....	51
3.1	Interrupts.....	52
3.1.1	Überblick.....	52
3.1.2	Interrupt-Bearbeitung.....	53
3.1.3	Interrupt-Verarbeitung bei IA32-Prozessoren.....	58
3.1.4	Interrupt-Bearbeitung unter Windows.....	61
3.1.5	Interruptverarbeitung unter Linux.....	67
3.2	Systemaufrufe.....	70
3.2.1	Systemaufrufe aus Sicht eines Anwendungsprogrammierers.....	70
3.2.2	Software-Interrupts unter Linux.....	71
3.2.3	Software-Interrupts unter Windows.....	74
3.3	Übungsaufgaben.....	74
4	Prozesse und Threads.....	77
4.1	Prozesse.....	78
4.1.1	Prozessmodell.....	78
4.1.2	Prozessverwaltung.....	79
4.1.3	Prozesslebenszyklus.....	82
4.2	Threads.....	83
4.2.1	Threadmodell.....	83
4.2.2	Implementierung von Threads.....	83
4.2.3	Vor-/Nachteile und Einsatzgebiete von Threads.....	86
4.3	Programmierkonzepte für Threads.....	88
4.3.1	Threads in Java.....	88
4.3.2	Threads in C#.....	93
4.4	Prozesse und Threads in konkreten Betriebssystemen.....	96
4.4.1	Prozesse und Threads unter Windows.....	96
4.4.2	Prozesse und Threads unter Unix und Linux.....	104
4.5	Übungsaufgaben.....	109

5 CPU-Scheduling .....	111
5.1 Scheduling-Kriterien.....	112
5.2 Scheduling-Verfahren.....	115
5.2.1 Verdrängende und nicht verdrängende Verfahren .....	115
5.2.2 Überblick über Scheduling-Verfahren.....	116
5.2.3 Multi-Level-Scheduling mit Prioritäten .....	118
5.2.4 Round-Robin-Scheduling mit Prioritäten .....	119
5.3 Vergleich ausgewählter Scheduling-Verfahren .....	121
5.3.1 CPU-Scheduling im ursprünglichen Unix .....	125
5.3.2 CPU-Scheduling unter Linux.....	127
5.3.3 CPU-Scheduling unter Windows .....	136
5.3.4 Scheduling von Threads in Java .....	143
5.3.5 Zusammenfassung .....	144
5.4 Übungsaufgaben .....	146
6 Synchronisation und Kommunikation.....	151
6.1 Grundlegendes zur Synchronisation .....	152
6.1.1 Nebenläufigkeit, atomare Aktionen und Race Conditions.....	152
6.1.2 Kritische Abschnitte und wechselseitiger Ausschluss .....	155
6.1.3 Eigenschaften nebenläufiger Programme .....	158
6.2 Synchronisationskonzepte .....	160
6.2.1 Sperren .....	160
6.2.2 Semaphore .....	162
6.2.3 Monitore.....	167
6.3 Synchronisationstechniken moderner Betriebssysteme.....	173
6.4 Synchronisationsmechanismen in Programmiersprachen .....	175
6.4.1 Die Java-Synchronisationsprimitive „synchronized“ .....	175
6.4.2 Warten auf Bedingungen in Java.....	182
6.4.3 Weitere Synchronisationsmechanismen in Java.....	185
6.4.4 C#-Monitore.....	187
6.4.5 Die C#-Synchronisationsprimitive „lock“ .....	189
6.4.6 C#-Mutex-Objekte.....	190
6.4.7 C#-Lese- und Schreibsperrern .....	192
6.4.8 C#-Interlocked-Klasse .....	193
6.4.9 Warten auf Bedingungen in C# .....	195
6.5 Kommunikation von Prozessen und Threads.....	195
6.5.1 Überblick über Kommunikationsmöglichkeiten.....	196

- 6.5.2 Verbindungsorientierte versus verbindungslose Kommunikation ...196
- 6.5.3 Speicherbasierte versus nachrichtenbasierte Kommunikation.....197
- 6.5.4 Synchrone versus asynchrone Kommunikation.....201
- 6.5.5 Kommunikationskanäle im Halb- und Vollduplex-Betrieb.....203
- 6.5.6 Varianten der Empfängeradressierung.....204
- 6.5.7 Fallbeispiel: Rechnerinterne Kommunikation über Pipes.....205
- 6.5.8 Rechnerübergreifende Interprozesskommunikation.....209
- 6.6 Übungsaufgaben..... 210
  
- 7 Hauptspeicherverwaltung ..... 213
  - 7.1 Grundlegende Betrachtungen..... 214
    - 7.1.1 Speicherhierarchien .....214
    - 7.1.2 Lokalität .....215
    - 7.1.3 Adressen und Adressräume.....216
    - 7.1.4 Techniken der Speicherverwaltung.....218
  - 7.2 Virtueller Speicher ..... 221
    - 7.2.1 Grundbegriffe und Funktionsweise .....221
    - 7.2.2 Optimierung der Speicherverwaltung.....232
    - 7.2.3 Seitenersetzung und Verdrängung (Replacement) .....236
    - 7.2.4 Vergleich von Seitenersetzungsverfahren .....246
    - 7.2.5 Speicherbelegungs- und Vergabestrategien (Placement).....247
    - 7.2.6 Entladestrategie (Cleaning) .....250
    - 7.2.7 Eine weitere Technik: Segmentadressierung .....251
    - 7.2.8 Shared Memory .....253
  - 7.3 Speicherverwaltung in ausgewählten Systemen..... 254
    - 7.3.1 Linux-Speicherverwaltung .....254
    - 7.3.2 Windows-Speicherverwaltung .....258
    - 7.3.3 Android-Speicherverwaltung .....264
  - 7.4 Übungsaufgaben..... 265
  
- 8 Geräte- und Dateiverwaltung ..... 267
  - 8.1 Aufgaben und Überblick ..... 268
    - 8.1.1 Grundlegendes.....268
    - 8.1.2 Gerätearten .....271
    - 8.1.3 Geräteanbindung unter Unix .....274
    - 8.1.4 Memory-Mapped Ein-/Ausgabe und DMA .....275
  - 8.2 Dateiverwaltung ..... 277

8.2.1	Allgemeines .....	277
8.2.2	Fallbeispiel: Dateisysteme unter Unix .....	279
8.2.3	Fallbeispiel: Dateisysteme unter Linux .....	282
8.2.4	Fallbeispiel: Dateisysteme unter Windows.....	283
8.3	Storage-Systeme .....	288
8.3.1	RAID-Plattensysteme.....	288
8.3.2	NAS und SAN.....	293
8.4	Übungsaufgaben .....	295
9	Betriebssystemvirtualisierung.....	297
9.1	Grundbegriffe und Einsatzgebiete.....	297
9.1.1	Virtualisierungsterminologie.....	297
9.1.2	Einsatz der Virtualisierungstechnik.....	299
9.1.3	Vor- und Nachteile der Virtualisierung .....	300
9.2	Virtualisierbarkeit der Hardware .....	301
9.2.1	Hardware-Voraussetzungen.....	301
9.2.2	Virtualisierung bei x86-Prozessoren .....	303
9.2.3	Virtualisierungstechniken in Prozessoren .....	304
9.3	Varianten der Virtualisierung .....	306
9.3.1	Typ-1-Hypervisor .....	307
9.3.2	Typ-2-Hypervisor .....	307
9.3.3	Paravirtualisierung.....	308
9.3.4	Weitere Virtualisierungsansätze.....	310
9.4	Betriebsmittelverwaltung bei Virtualisierung.....	310
9.4.1	Interruptverarbeitung und Gerätesteuerung.....	310
9.4.2	CPU-Scheduling.....	312
9.4.3	Speicherverwaltung .....	314
9.5	Zusammenfassung.....	317
9.6	Übungsaufgaben .....	318
10	Schlussbemerkung .....	319
11	Lösungen zu den Übungsaufgaben.....	321
11.1	Einführung .....	321
11.2	Betriebssystemarchitekturen und Betriebsarten .....	322
11.3	Interruptverarbeitung.....	325
11.4	Prozesse und Threads.....	329
11.5	CPU-Scheduling .....	333

## Inhaltsverzeichnis

---

11.6 Synchronisation und Kommunikation .....	342
11.7 Hauptspeicherverwaltung .....	349
11.8 Geräte- und Dateiverwaltung .....	356
11.9 Betriebssystemvirtualisierung .....	359
Anhang .....	363
A1 Zahlennamen .....	363
A2 Metrische Grundeinheiten .....	364
A3 Wichtige Datenstrukturen für Betriebssysteme .....	365
A4 Java-Implementierung des Dining-Philosophers-Problems.....	380
A5 C#-Implementierung des Dining-Philosophers-Problems.....	384
A6 C#-Implementierung des Zählerproblems mit Monitoren .....	389
Literaturhinweise .....	395
Sachwortverzeichnis .....	399