

# Inhaltsverzeichnis

<b>1 Die Gliederung von Computersystemen</b> . . . . .	<b>1</b>
1.1 Überbrückung der Hardware/Benutzer-Kluft . . . . .	1
1.2 Grenzen der Hardware . . . . .	3
1.2.1 Prozessoren, Speicher und Ein-/Ausgabegeräte . . . . .	3
1.2.2 Die Prozessor-Geräte-Schnittstelle . . . . .	7
1.3 Funktionen eines Betriebssystems . . . . .	13
1.3.1 Ein minimales Betriebssystem . . . . .	13
1.3.2 Prinzipien des Mehrprogramm Betriebs . . . . .	16
1.3.3 Betriebssysteme als virtuelle Maschinen . . . . .	20
1.4 Betriebssystemarten . . . . .	22
1.4.1 Größe der zugrundeliegenden Hardwarekonfiguration . . . . .	22
1.4.2 Grad der gemeinsamen Benutzbarkeit von Betriebsmitteln . . . . .	23
1.4.3 Arten von erlaubtem Dialogbetrieb . . . . .	24
1.5 Struktur von Betriebssystemen . . . . .	25
1.5.1 Eine monolithische Struktur . . . . .	25
1.5.2 Der Systemkern-Ansatz . . . . .	27
1.5.3 Eine Prozeßhierarchie . . . . .	29
1.5.4 Eine funktionale Hierarchie . . . . .	32
1.5.5 Objektorientierte Strukturen . . . . .	33
1.6 Geschichtlicher Ausblick . . . . .	35
1.6.1 Frühe Systeme . . . . .	35
1.6.2 Hardware und Software der zweiten Generation . . . . .	36
1.6.3 Systeme der dritten Generation . . . . .	38
1.6.4 Die vierte Generation und danach . . . . .	38
1.7 Zusammenfassung . . . . .	40
<b>2 Methoden konkurrierender Programmierung</b> . . . . .	<b>41</b>
2.1 Das Prozeßkonzept . . . . .	41
2.2 Konstrukte zur Prozeßerzeugung . . . . .	42
2.2.1 Prozeßflußgraphen . . . . .	42
2.2.2 Das Konstrukt <i>cobegin/coende</i> . . . . .	45
2.2.3 Die Dienste <i>erzeuge, verbinde und verlasse</i> . . . . .	47
2.2.4 Explizite Prozeßdeklarationen . . . . .	50
2.3 Prozeßkommunikation . . . . .	52
2.3.1 Das Problem kritischer Bereiche . . . . .	52
2.3.2 Kooperierende Prozesse . . . . .	57

2.4	Semaphordienste . . . . .	58
2.4.1	Die P und V Operationen . . . . .	58
2.4.2	Wechselseitiger Ausschluß durch Nutzung der Semaphoroperationen . . . . .	59
2.4.3	Semaphore in Erzeuger-/Verbrauchersituationen . . . . .	60
2.4.4	Implementieren von Semaphoroperationen . . . . .	61
2.4.4.1	Implementation mit aktivem Warten . . . . .	62
2.4.4.2	Vermeiden des aktiven Wartens . . . . .	64
2.5	Synchronisationsmethoden auf höheren Ebenen . . . . .	66
2.5.1	Bedingte kritische Bereiche . . . . .	66
2.5.1.1	Definition und Gebrauch . . . . .	66
2.5.1.2	Implementation . . . . .	68
2.5.2	Monitore . . . . .	69
2.5.2.1	Definition und Gebrauch . . . . .	69
2.5.2.2	Implementierung . . . . .	72
2.5.2.3	Andere Monitorvarianten . . . . .	73
2.5.2.4	Warten mit Prioritäten . . . . .	74
2.5.3	Pfadausdrücke . . . . .	75
2.5.3.1	Definition und Gebrauch . . . . .	75
2.5.3.2	Implementierung . . . . .	78
2.6	Nachrichtenbasierte Synchronisationsschemata . . . . .	80
2.6.1	Die Dienste <code>sende</code> und <code>empfang</code> . . . . .	80
2.6.2	Entfernte Prozeduraufrufe . . . . .	87
2.7	Klassische Synchronisationsprobleme . . . . .	91
2.7.1	Das Schreib-/Leseproblem . . . . .	91
2.7.1.1	Lösung unter Verwendung bedingter kritischer Bereiche . . . . .	91
2.7.1.2	Lösung unter Verwendung eines Monitors . . . . .	92
2.7.2	Das Problem der speisenden Philosophen . . . . .	94
2.7.2.1	Lösung unter Verwendung eines Monitors . . . . .	95
2.7.2.2	Lösung unter Verwendung von Pfadausdrücken . . . . .	96
2.7.3	Planung des Zugriffs des Schreib-/Lesekopfs auf Platten . . . . .	97
2.7.3.1	Lösung unter Verwendung eines Monitors . . . . .	97
2.7.3.2	Lösung unter Verwendung entfernter Prozeduraufrufe . . . . .	99
3	Prozeß- und Betriebsmittelsteuerung . . . . .	108
3.1	Der Betriebssystemkern . . . . .	108
3.2	Datenstrukturen für Prozesse und Betriebsmittel . . . . .	111
3.2.1	Prozeßsteuerblöcke . . . . .	111
3.2.2	Beschreibungen für Betriebsmittel . . . . .	115
3.2.3	Das Betriebsmittel Prozessor . . . . .	119
3.2.4	Implementierung von Warteschlangen . . . . .	120

3.3	Grundoperationen auf Prozessen und Betriebsmitteln . . . . .	123
3.3.1	Prozeßsteuerung . . . . .	123
3.3.2	Betriebsmitteldienste . . . . .	127
3.4	Organisation der Ablaufsteuerung von Prozessen . . . . .	132
3.4.1	Eigenständige versus gemeinsam benutzte Ablaufsteuerungen . . . . .	132
3.4.2	Ein Beispiel für eine Ablaufsteuerung mit Prioritätsberücksichtigung . . . . .	136
3.5	Methoden der Ablaufsteuerung . . . . .	141
3.5.1	Die universelle Ablaufsteuerung . . . . .	142
3.5.2	Algorithmen zur zeitorientierten Ablaufsteuerung . . . . .	145
3.5.3	Vergleich der Ablaufsteuerungsmethoden . . . . .	151
4	Das Problem der Verklemmungen . . . . .	160
4.1	Beispiele für Verklemmungen in Rechensystemen . . . . .	161
4.2	Das Modell eines Systems . . . . .	166
4.3	Verklemmungen bei seriell wiederverwendbaren Betriebsmitteln . . . . .	169
4.3.1	Graphen wiederverwendbarer Betriebsmittel . . . . .	170
4.3.2	Entdeckung von Verklemmungen . . . . .	173
4.3.3	Spezialfälle von Graphen wiederverwendbarer Betriebsmittel . . . . .	179
4.3.4	Wiederherstellung eines Zustandes ohne Verklemmung . . . . .	183
4.3.5	Methoden zur Verhinderung von Verklemmungen . . . . .	186
4.4	Systeme mit nicht wiederverwendbaren Betriebsmitteln . . . . .	192
4.4.1	Graphen für nicht wiederverwendbare Betriebsmittel bei bekannten Erzeugern . . . . .	193
4.4.2	Graphen nicht wiederverwendbarer Betriebsmittel mit bekannten Erzeugern und Verbrauchern . . . . .	198
4.5	Allgemeine Betriebsmittelgraphen . . . . .	200
4.6	Dynamisches Hinzufügen und Löschen von Prozessen und Betriebsmitteln . . . . .	201
5	Hauptspeicherverwaltung . . . . .	209
5.1	Vorbereitung für Start und Ausführung eines Programms . . . . .	209
5.1.1	Verschiebung . . . . .	210
5.1.1.1	Statische Verschiebung . . . . .	210
5.1.1.2	Dynamische Verschiebung . . . . .	212
5.1.2	Binden . . . . .	215
5.2	Einfache Verfahren zur Hauptspeicherverwaltung . . . . .	217
5.2.1	Feste Partitionen . . . . .	217
5.2.2	Variable Partitionen und Komprimieren des belegten Speichers . . . . .	219
5.2.3	Überlagerungen . . . . .	220

5.3	Die Grundprinzipien zu virtuellen Speichern . . . . .	222
5.4	Implementierung des virtuellen Speichers . . . . .	225
5.4.1	Einsegmentiger Namensraum . . . . .	225
5.4.1.1	Zusammenhängende Speicherzuweisung . . . . .	225
5.4.1.2	Seitenwechselverfahren . . . . .	226
5.4.2	Mehrsegmentiger Namensbereich . . . . .	232
5.4.2.1	Zuweisung zusammenhängenden Speicherbereichs für die Segmente . . . . .	232
5.4.2.2	Seitenwechselverfahren mit Segmentierung . . . . .	233
5.4.2.3	Seitenwechsel für Systemtabellen . . . . .	235
5.4.2.4	Adreßumsetzungspuffer . . . . .	236
5.5	Speicherzuweisungsstrategien . . . . .	238
5.5.1	Speicherzuweisung in Systemen ohne Seitenwechselverfahren . . . . .	238
5.5.1.1	Erste passende und beste passende Plazierung . . . . .	239
5.5.1.2	Verwaltung des verfügbaren Raums . . . . .	240
5.5.1.3	Analyse der Speicherzerstückelung und der Speichernutzung . . . . .	243
5.5.2	Speicherzuweisung in seitenorientierten Systemen . . . . .	245
5.5.2.1	Statische und dynamische Zuweisung . . . . .	245
5.5.2.2	Austauschverfahren für Seiten mit einer festen Anzahl von Kacheln . . . . .	248
5.5.2.3	Verfahren zum Austausch von einer variablen Anzahl von Seiten . . . . .	256
5.5.2.4	Vorbereitung zum Seitenwechsel . . . . .	261
5.5.3	Steuerung des Ladens und der Überlastung . . . . .	263
5.5.4	Auswertung von Seitenwechselverfahren . . . . .	264
5.6	Binden und gemeinsames Benutzen von Prozeduren und Daten im Arbeitsspeicher . . . . .	268
5.6.1	Warum gemeinsame Benutzung? . . . . .	268
5.6.2	Erfordernisse für die gemeinsame Benutzung von Code . . . . .	269
5.6.3	Statisches Binden und gemeinsame Benutzung eines Moduls . . . . .	271
5.6.3.1	Gemeinsame Benutzung eines Moduls in Systemen ohne Verfahren für Segmentierung oder Seitenwechsel . . . . .	272
5.6.3.2	Gemeinsame Benutzung eines Moduls in Systemen mit Seitenwechselverfahren . . . . .	273
5.6.3.3	Gemeinsame Benutzung eines Moduls in Systemen mit Segmentation . . . . .	275
5.6.4	Dynamisches Binden und gemeinsame Benutzung eines Moduls . . . . .	276
5.6.4.1	Form eines Prozedursegments . . . . .	277
5.6.4.2	Binden von Daten . . . . .	279
5.6.4.3	Referenzen auf Prozeduren . . . . .	281
5.7	Objektorientierte Speicherverwaltung . . . . .	282

<b>6</b>	<b>Dateisysteme und Bearbeitung der Ein-/Ausgabe</b>	<b>294</b>
6.1	Aufbau von Datei- und Ein-/Ausgabesubsystemen	295
6.1.1	Grundfunktionen der Geräteverwaltung	295
6.1.2	Die abstrakte Benutzerschnittstelle	297
6.1.3	Ein hierarchisches Modell für das Datei- und Ein-/Ausgabesubsystem	298
6.2	Verzeichnisverwaltung	303
6.3	Das Dateigrundsystem	308
6.3.1	Dateibeschreibungen	308
6.3.2	Routinen zum Öffnen und Schließen einer Datei	310
6.4	Methoden zur physikalischen Organisation	312
6.4.1	Logische Dateiorganisation	312
6.4.2	Physikalische Dateiorganisation	313
6.4.2.1	Zusammenhängende Organisation	313
6.4.2.2	Verkettete Organisation	316
6.4.2.3	Indexsequentielle Organisation	317
6.4.2.4	Organisation als B-Baum	320
6.4.3	Verwaltung von Speicherbereich auf Hilfsspeichern	323
6.5	Das Ein-/Ausgabesubsystem	325
6.5.1	Programmgesteuerte Ein-/Ausgabe	325
6.5.2	Schnittstelle für den direkten Speicherzugriff	329
6.5.2.1	Direkter Speicherzugriff mit Abfrage des Kanals	331
6.5.2.2	Direkter Speicherzugriff mit Kanalunterbrechungen	338
6.5.3	Ein Prozeßmodell für eine unterbrechungsgetriebene Ein-/Ausgabe	339
6.6	Kommandosprachen und Dienstprogramme des Dateisystems	343
6.7	Wiederherstellen eines arbeitsfähigen Zustandes nach einem Systemausfall	348
<b>7</b>	<b>Datenschutz und Datensicherheit</b>	<b>359</b>
7.1	Einführung	359
7.1.1	Externe Schutzmechanismen	361
7.1.2	Interne Schutzmechanismen	362
7.2	Eindringen in ein Rechensystem	364
7.2.1	Verfahren zur Benutzerauthentifizierung	364
7.2.2	Eindringversuche	366
7.3	Zugriffskontrolle und Steuerung des Informationsflusses	369
7.3.1	Die aktuelle Arbeitsumgebung	369
7.3.2	Schutz im Hauptspeicher	370
7.3.3	Schutz im Hintergrundspeicher	374
7.3.4	Vergleich von Zugriffsausweislisten und Zugriffslisten	376

7.4	Schutzprobleme und einige partielle Lösungen . . . . .	383
7.5	Formale Schutzmodelle . . . . .	391
7.5.1	Das Modell einer Zugriffsmatrix . . . . .	392
7.5.2	Das Nehmen-Gewähren Modell . . . . .	395
7.5.3	Das Bell-La Padula Modell . . . . .	400
7.5.4	Ein Gittermodell für den Informationsfluß . . . . .	403
 <b>Bibliographie</b> . . . . .		 413
 <b>Register</b> . . . . .		 417