

Skript zur Vorlesung **Betriebssysteme**

für das 4. Semester Wirtschaftsinformatik
an der FH Regensburg
gehalten vom Dr. Ulrich Margull
im Sommersemester 2004

Inhaltsverzeichnis

1	Einführung.....	13
1.1	Betriebsarten.....	13
1.1.1	Batch/Dialog(Interaktiv)/Echtzeit.....	13
1.1.2	Ein-Nutzer/Mehr-Nutzer.....	14
1.1.3	Nebenläufigkeit:.....	14
1.2	Anforderungsprofile.....	14
1.3	Historische Entwicklung.....	14
2	Übersicht.....	16
2.1	Wiederholung: Computer Hardware	16
2.1.1	Funktionsweise einer "von Neumann" - Maschine.....	16
2.1.2	Prozessor.....	17
2.1.3	Zentralspeicher (Memory).....	17
2.1.4	I/O Geräte.....	18
2.1.5	Datenbusse.....	18
2.1.6	Multi-Prozessor-Architekturen.....	19
2.2	Aufgaben, Dienste und Komponenten von Betriebssystemen.....	20
2.2.1	Prozeß- und Prozessorverwaltung.....	20
2.2.2	Programmverwaltung.....	21
2.2.3	Zentralspeicherverwaltung.....	22
2.2.4	Ein-/Ausgabe-Steuerung.....	23
2.2.5	Externe Datenverwaltung.....	23
2.2.5.1	Physikalische Speichermedien.....	23
2.2.6	Betriebsmittelverwaltung (Resource Management).....	24
2.2.7	Zugriffsschutz (Security).....	25
2.2.8	Kommando-Interpreter (Shell).....	25
2.3	Betriebssystem Strukturen.....	25
2.3.1	Einfache Strukturen.....	25
2.3.1.1	Beispiel MS-DOS	25
2.3.1.2	Beispiel: Unix	25
2.3.2	Microkernel.....	26
2.3.2.1	Beispiel Windows XP	27
2.3.3	Konzept virtueller Maschinen.....	28
2.3.3.1	Übersicht VM.....	28
2.3.3.2	OS/370	28
2.3.3.3	Java Virtual Machine (JVM).....	29
2.3.3.4	Weitere Virtuelle Maschinen.....	29
3	Prozesse	30
3.1	Was ist ein Prozess?.....	30
3.2	Prozessverwaltung.....	31
3.2.1	Prozesskontrollblöcke.....	31
3.2.2	Verwaltung der PCBs	32
3.2.3	Prozesserzeugung & -vernichtung.....	32
3.2.3.1	Statisch vs. Dynamisch.....	32
3.2.3.2	Prozesserzeugung.....	33

3.2.3.3	Prozessvernichtung.....	33
3.2.3.4	Beispiel Unix: fork().....	33
3.2.3.5	Beispiel Unix: exec().....	34
3.2.3.6	Beispiel Win32: CreateProcess().....	35
3.3	Kontextwechsel.....	36
3.3.1	Funktionsaufruf.....	37
3.3.2	Interrupt.....	37
3.3.3	Prozesswechsel.....	37
3.3.4	Unterbrechungsarten.....	37
3.4	Interprozess-Kommunikation.....	38
3.4.1	Message Passing.....	38
3.4.1.1	Direkte Kommunikation.....	39
3.4.1.2	Indirekte Kommunikation.....	39
3.4.1.3	Synchronisierung.....	39
3.4.1.4	Pufferung.....	40
3.4.1.5	Beispiel: Mach.....	40
3.4.2	Pipes.....	40
3.4.3	Sockets.....	41
3.4.4	Client-Server: RPC.....	42
3.4.5	Verteilte Objekte.....	44
3.4.5.1	Komponenten-Transaktions-Server (CTS).....	44
4	Threads.....	45
4.1	Allgemein.....	45
4.1.1	Anwendungsbeispiel 1: MS Word.....	46
4.1.2	Anwendungsbeispiel 2: Web-Server.....	46
4.2	Threaderzeugung.....	46
4.2.1	Linux: clone()	47
4.2.2	Win32: CreateThread().....	47
4.2.3	POSIX: pthread-create().....	48
4.3	User- und Kernel-Threads.....	49
4.3.1	User-Level Threads.....	49
4.3.2	Kernel-Level Threads.....	50
4.3.3	Many-to-one Modell (User-Level-Threads).....	50
4.3.4	One-to-one Modell (Kernel-Level-Threads).....	51
4.3.5	Many-to-many Modell (gemischt).....	51
4.4	Beispiele.....	52
4.4.1	Pthreads: User-Level Bibliothek.....	52
4.4.2	Win2000 Threads.....	52
4.4.3	Solaris 2 Threads.....	52
4.4.4	Linux-Threads.....	52
4.4.5	Java Threads.....	53
5	CPU Scheduling.....	54
5.1	Grundlagen.....	54
5.1.1	CPU Burst / I/O Burst.....	54
5.1.2	Scheduler.....	55
5.1.3	Dispatcher.....	56
5.2	Ziele des Scheduling.....	56
5.2.1	Allgemeine Kriterien.....	56

5.2.2	Großrechner (Batch Systeme).....	56
5.2.3	Interaktive Systeme.....	57
5.2.4	Echtzeit-Systeme.....	57
5.3	Algorithmen.....	57
5.3.1	First Come First Served Scheduling (FCFS).....	57
5.3.2	Shortest Job First Scheduling (SJF).....	59
5.3.3	Round-Robin Scheduling (RR).....	60
5.3.4	Prioritäten-Scheduling.....	61
5.3.5	Multilevel Queue Scheduling.....	62
5.3.6	Realtime Scheduling: Rate-Monotonic Scheduling (RMS).....	63
5.3.6.1	Hard-Realtime.....	63
5.3.6.2	Soft-Realtime.....	63
5.3.6.3	Rate-Monotonic Scheduling.....	63
5.3.6.4	Problem: Prioritäteninversion.....	65
5.3.7	Evaluierung der Algorithmen.....	65
5.4	Thread Scheduling.....	65
5.5	Beispiele.....	66
5.5.1	Solaris 2.....	66
5.5.2	Windows 2000.....	67
5.5.3	Linux (Kernel < 2.6).....	67
6	Prozesssynchronisierung.....	68
6.1	Kritische Bereiche (Critical Section).....	70
6.2	Lösungen.....	71
6.2.1	Abschalten der Interrupts	71
6.2.2	Zwei-Prozess Lösungen.....	72
6.2.2.1	Software-Lösung Versuch 1: Sperr-Variable (lock variable).....	72
6.2.2.2	Lösungsversuch 2: Abwechselnder Zugang (strict alternation).....	72
6.2.2.3	Peterson's Lösung.....	73
6.2.3	Test And Set Lock (TSL).....	75
6.2.4	Sleep And Wakeup.....	75
6.2.5	Semaphoren.....	76
6.2.5.1	Implementierung.....	77
6.2.5.2	Beispiel: Consumer-Producer mit Semaphore.....	77
6.2.6	Monitors.....	78
6.2.7	Weitere Verfahren zur Prozesssynchronisierung.....	80
6.3	Beispiele.....	80
6.3.1	Windows 2000.....	80
6.3.1.1	Interlocked-Funktionen.....	80
6.3.1.2	Critical Section.....	80
6.3.1.3	Dispatcher Objects: Events, Mutex, Semaphore.....	81
6.3.2	Unix.....	81
6.3.2.1	Semaphore.....	81
6.3.3	Java.....	81
6.4	Klassische Probleme der Synchronisierung.....	82
6.4.1	Bounded-Buffer (Producer-Consumer-Problem).....	82
6.4.2	Readers-Writers-Problem.....	82
6.4.3	Fünf speisende Philosophen.....	82
6.5	Prioritäteninversion.....	87

6.6	Deadlocks.....	89
6.6.1	Wiederholung: Begrifflichkeiten.....	89
6.6.2	Bedingungen für ein Deadlock.....	89
6.6.3	Ignorieren des Problems (Vogel-Strauss-Strategie).....	90
6.6.4	Entdeckung und Behandlung.....	90
6.6.5	Vermeidung von Deadlocks.....	90
6.6.6	Verhinderung von Deadlocks.....	91
6.6.7	Beispiel (nach [1]).....	93
7	Zentralspeicherverwaltung.....	95
7.1	Grundlagen.....	95
7.1.1	Bindung der Speicheradressen.....	95
7.1.2	Logischer und physikalischer Speicher.....	96
7.1.3	Dynamisches Laden / Overlay Technik.....	97
7.1.4	Dynamisches Linken und gemeinsame Bibliotheken (Shared Libraries).....	97
7.1.5	Echtzeitsysteme.....	98
7.2	Monoprogrammierung.....	100
7.3	Multiprogrammierung-Betrieb mit festen Partitionen.....	101
7.4	Swapping.....	101
7.4.1	Speicherverteilung.....	102
7.4.2	Speicherschutz.....	103
7.4.3	Problem: Fragmentierung und Verschnitt.....	103
7.4.4	Verwaltung von freien Speicherbereichen.....	103
7.4.4.1	Bitmaps: Bitketten für Speicher.....	103
7.4.4.2	Speicherverwaltung über lineare Listen (Freikette).....	104
7.4.5	Algorithmen zur Auswahl von Speicher aus der Freikette.....	105
7.4.5.1	First Fit.....	105
7.4.5.2	Best Fit.....	105
7.4.5.3	Worst Fit:	105
7.4.5.4	Problem: Externer Verschnitt.....	105
7.4.5.5	Buddy-System nach Knuth.....	106
7.5	Virtueller Speicher: Paging.....	109
7.5.1	Übersicht Paging.....	109
7.5.2	Umsetzung mittels Seitentabellen.....	109
7.5.3	Hierarchische Seitentabellen.....	112
7.5.4	Einträge in einer Seitentabelle.....	114
7.5.5	TLBs – Translation Lookaside Buffers.....	114
7.5.6	Invertierte Seitentabellen.....	116
7.5.7	Speicherschutz.....	117
7.5.8	Gemeinsame Seiten.....	118
7.5.9	Demand Paging, Seitentauschstrategien, Seitenflattern.....	118
7.5.10	Anforderungen an den Prozessor.....	118
7.5.11	Nebeneffekte des Paging.....	119
7.6	Segmentierung.....	119
8	I/O Systeme.....	121
8.1	Allgemein.....	121
8.1.1	I/O Geräte.....	121
8.1.2	I/O Gerätecontroller.....	121
8.1.3	Adressierungsarten von I/O Geräten.....	122

8.1.4	Polling.....	124
8.1.5	Interrupts.....	124
8.1.6	Direkter Speicherzugriff (Direct Memory Access, DMA).....	125
8.2	I/O Software Schichten.....	126
8.2.1	Interrupt Handler.....	126
8.2.2	Gerätetreiber.....	127
8.2.3	Geräte-unabhängige Software.....	128
8.2.3.1	Einheitliches Interface für alle Treiber.....	128
8.2.3.2	Datenpufferung.....	128
8.2.3.3	Fehlermeldungen.....	129
8.2.3.4	Belegung und Freigabe von Geräten (Resource-Handling).....	129
8.2.3.5	Geräte-unabhängige Block-Größe.....	129
8.2.4	User-Space I/O Software.....	129
8.3	Plattenlaufwerke.....	129
8.3.1	Hardware.....	130
8.3.2	Zugriffsstrategien.....	130
8.3.2.1	FCFS Scheduling.....	131
8.3.2.2	SSTF Scheduling (auch SST).....	131
8.3.2.3	Aufzug-Algorithmus (SCAN).....	131
8.3.3	Fehlerbehandlung.....	132
8.3.4	Stabilität.....	132
8.4	Serielle Schnittstelle.....	133
8.4.1	Grundlagen der Asynchronen Seriellen Kommunikation (RS232).....	133
8.4.2	Controller-Register.....	133
8.4.3	Ansteuerung auf Interrupt-Ebene.....	133
8.4.4	Ansteuerung auf User-Ebene (Win32).....	134
8.5	Weitere Geräte.....	134
9	Dateisysteme.....	135
9.1	Dateien.....	135
9.1.1	Dateinamen.....	135
9.1.2	Dateistruktur und -zugriff.....	136
9.1.3	Datei-Attribute.....	136
9.1.4	Dateioperationen.....	136
9.1.5	Memory-Mapped Files.....	137
9.2	Verzeichnisse.....	138
9.2.1	Verzeichnisse mit einer Ebene.....	138
9.2.2	Verzeichnisse mit zwei Ebenen.....	138
9.2.3	Hierarchische Verzeichnisstrukturen.....	138
9.2.4	Dateipfade.....	138
9.2.5	Verzeichnis-Operationen	139
9.2.6	Links.....	139
9.2.7	Mount-Punkte.....	140
9.3	Implementierung von Dateisystemen.....	140
9.3.1	Partitionierung von Festplatten.....	140
9.3.2	Implementierung von Dateien.....	141
9.3.2.1	Kontinuierliche Belegung.....	141
9.3.2.2	Verlinkte Liste.....	142
9.3.2.3	FAT.....	143

9.3.2.4	Indizierte Belegung (I-Nodes).....	144
9.3.2.5	Verwaltung von freiem Speicher.....	145
9.3.3	Implementierung von Verzeichnissen.....	146
9.3.3.1	Lineare Liste.....	146
9.3.3.2	Hash-Tabelle.....	147
9.3.4	Freispeicherverwaltung.....	147
9.3.5	Nicht-Funktionale Aspekte: Performance, Zuverlässigkeit, Effizienz.....	148
9.4	Protokollierende Dateisysteme (Journaling File Systems).....	148
9.5	Netzwerk-Dateisysteme: NFS.....	149
9.6	Beispiele.....	149
9.6.1	Win32-Schnittstelle (API).....	149
9.6.2	MS-DOS Dateisystem: FAT.....	152
9.6.3	NTFS.....	153
10	Danksagung.....	155

Abbildungsverzeichnis

Abbildung 1:	Alter IBM Rechner (Folie).....	14
Abbildung 2:	Übersicht Rechneraufbau.....	16
Abbildung 3:	Übersicht "von Neumann" Maschine.....	16
Abbildung 4:	Programmausführung.....	17
Abbildung 5:	Pipeline, Superskalar Pipeline.....	17
Abbildung 6:	Zugriffspyramide.....	17
Abbildung 7:	Pentium Busstruktur (aus [1]).....	18
Abbildung 8:	Schema Rechner.....	19
Abbildung 9:	Mehr-Prozessor-Systeme: Shared Memory, Interconnected, Distributed.....	19
Abbildung 10:	Prozess-Zustände, Prozessorzuteilung.....	20
Abbildung 11:	Interprozess-Kommunikation (a) Message Passing, (b) Shared Memory.....	21
Abbildung 12:	Übersicht Programmverwaltung.....	21
Abbildung 13:	Übersicht virtuelle Speicherverwaltung.....	22
Abbildung 14:	Schema E/A Steuerung.....	23
Abbildung 15:	Deadlock an der Kreuzung.....	24
Abbildung 16:	DOS Struktur.....	25
Abbildung 17:	Unix Struktur (siehe Folie).....	26
Abbildung 18:	Windows Übersicht.....	27
Abbildung 19:	Aufbau BT.....	28
Abbildung 20:	Virtual Machine (VM) OS/370.....	29
Abbildung 21:	Java VM.....	29
Abbildung 22:	Prozess-Zustände.....	30
Abbildung 23:	Organisation von PCBs.....	32
Abbildung 24:	Prozesshierarchie.....	34
Abbildung 25:	Ablauf fork() - exec().....	35
Abbildung 26:	Funktionsaufruf, Interrupt, Prozesswechsel.....	36
Abbildung 27:	Schema Message Passing.....	39
Abbildung 28:	Message Passing.....	39
Abbildung 29:	Schema Pipes.....	40
Abbildung 30:	Schema Sockets.....	41

Abbildung 31: Client-Server mit Sockets.....	42
Abbildung 32: Schema RPC Aufruf.....	43
Abbildung 33: Thread vs. Prozess (nach [2]).....	45
Abbildung 34: Threaderzeugung (einfache Darstellung ohne Kernel).....	46
Abbildung 35: Thread-Erzeugung (Darstellung mit Kernel).....	47
Abbildung 36: Wechsel von User-Thread zu Kernel-Thread.....	49
Abbildung 37: many-to-one Thread Modell (nach [2]).....	50
Abbildung 38: One-to-one Thread Modell (nach [2]).....	51
Abbildung 39: Many-to-many Thread Modell (nach [2]).....	51
Abbildung 40: Solaris 2 Threading Modell (nach [2]).....	52
Abbildung 41: CPU intensiv (a) vs. I/O intensiv (b) (nach [2]).....	54
Abbildung 42: Burst Dauer (Histogramm) (nach [2]).....	54
Abbildung 43: 3-Level Scheduler (nach [2]).....	55
Abbildung 44: Prozess-Zustände.....	55
Abbildung 45: FCFS Warteschlange.....	57
Abbildung 46: FCFS Beispiel.....	58
Abbildung 47: Konvoi Effekt bei FCFS.....	59
Abbildung 48: Round-Robin Beispiel.....	60
Abbildung 49: Multilevel Feedback Queue (nach [2]).....	62
Abbildung 50: RMS Aktivierungsschema (Beispiel).....	64
Abbildung 51: Overview Thread Scheduling (nach [1]).....	66
Abbildung 52: Schema kritischer Bereich.....	71
Abbildung 53: Ein Philosophen-Esstisch.....	83
Abbildung 54: Prioritätsinversion (P2 beendet vor P3).....	87
Abbildung 55: Prioritäten-Vererbung.....	88
Abbildung 56: Prioritäten-Vererbung (P1 erhält dieselbe Priorität wie P3).....	88
Abbildung 57: Priority-Ceiling(P1 erhält kurzzeitig eine hohe Priorität).....	88
Abbildung 58: Betriebsmittelbelegungsgraphen (A, B, C, D: Prozesse, R, S, T, U: Ressourcen).....	89
Abbildung 59: Deadlock-Trajektorie (nach [1]).....	92
Abbildung 60: Deadlock!.....	93
Abbildung 61: Kein Deadlock!.....	94
Abbildung 62: Übersicht Adressenbindung (nach [2]).....	95
Abbildung 63: MMU Architektur.....	96
Abbildung 64: Adressumsetzung mittels Relocation Register.....	97
Abbildung 65: Adressumsetzung mittels Relocation Register.....	97
Abbildung 66: Adress- und Datenbus für ein Echtzeitsystem.....	99
Abbildung 67: Speicherlayout Echtzeitsystem.....	100
Abbildung 68: Speicherverwaltung bei Monoprogrammierung (nach [1]).....	100
Abbildung 69: Speicherverwaltung mit festen Partitionen (nach [1]).....	101
Abbildung 70: Ablauf beim Swapping.....	102
Abbildung 71: Speicherverteilung im Prozess (nach [1]).....	102
Abbildung 72: Adressverschiebung mit Begrenzung.....	103
Abbildung 73: Verschmelzung von freien Bereichen (nachdem Prozess X terminiert).....	105
Abbildung 74: Speicherfragmentierung.....	106
Abbildung 75: Speicheraufteilung nach dem Buddy-System.....	107
Abbildung 76: Buddy-System (Beispiel).....	108
Abbildung 77: Übersicht Paging	109

Abbildung 78: Übersicht Paging MMU.....	110
Abbildung 79: Beispiel Seitentabelle.....	111
Abbildung 80: Zwei-stufige Seitentabelle.....	113
Abbildung 81: Typischer Einträge in einer Seitentabelle.....	114
Abbildung 82: Schema Translation Lookaside Buffer.....	115
Abbildung 83: Seitentabelle a) normal b) invertiert c) mit Hash-Tabelle (nach [1]).....	117
Abbildung 84: Gemeinsamer Speicher durch Paging.....	118
Abbildung 85: Verschiedene Segmente eines Programmes.....	120
Abbildung 86: Segemente.....	120
Abbildung 87: Übersicht Rechneraufbau.....	122
Abbildung 88: Adressierungsarten für I/O Geräte (nach [1]): a) getrennte Befehle; b) Memory-Mapped I/O; c) gemischt;.....	123
Abbildung 89: Busarchitektur für Memory-Mapped-I/O.....	123
Abbildung 90: I/O Adressen eines PCs (nicht vollständig).....	124
Abbildung 91: Vektor-Interrupttabelle des PCs.....	125
Abbildung 92: Schema DMA.....	126
Abbildung 93: (a) Ohne Pufferung (b) User-Space Pufferung (c) Einfach-Pufferung (d) Doppelte Pufferung (nach [1]).....	129
Abbildung 94: Festplatten-Geometrie und Bezeichnungen.....	130
Abbildung 95: Festplattenzugriffe FCFS (links) und SSTF (rechts), nach [2].....	131
Abbildung 96: Festplattenzugriffe SCAN (links) und C-SCAN (rechts), nach [2].....	132
Abbildung 97: Fehlerhafte Sektoren.....	132
Abbildung 98: RS232-Übertragung (7-Bit).....	133
Abbildung 99: Virtueller Speicher und Memory-Mapped Files.....	138
Abbildung 100: Symbolischer Link.....	139
Abbildung 101: Harter Link.....	139
Abbildung 102: Mount-Punkt im Verzeichnis.....	140
Abbildung 103: Partitionen.....	141
Abbildung 104: Kontinuierliche Belegung.....	142
Abbildung 105: Verlinkte Liste.....	142
Abbildung 106: FAT (aus [1]).....	144
Abbildung 107: I-Node (direkt sowie einfache, zweifache und dreifache Indirektion) (aus [1])..	145
Abbildung 108: Liste freier Blöcke.....	146
Abbildung 109: Verzeichnis (lineare Liste).....	146
Abbildung 110: NFS Systemarchitektur.....	149