

Inhaltsverzeichnis

zum Skript der Vorlesung

Rechnerorganisation

an der Berufsakademie Stuttgart
Fachrichtung Nachrichtentechnik
Studienjahrgang 2005, 5. und 6. Studienhalbjahr

WS 2007 und SS 2008

© 2000 – 2007 Ingo Phleps

Stand 12. Februar 2008

http://home.ntz.de/phleps/vorlesung/rechnerorganisation/skript_inhalt.pdf

Inhaltsverzeichnis

1	Rechner-Architekturen und Rechnerkomponenten	2
1.1	Prozessoren	2
1.1.1	Rechner-Architekturen	2
1.1.1.1	Von-Neumann-Architektur und Harvard-Architektur	3
1.1.1.2	Pipelining	4
1.1.1.3	Parallelverarbeitung bei arithmetischen Operationen	4
1.1.1.4	CISC-, RISC- und EPIC- Architektur	6
1.1.2	Anwendungsorientierte Prozessor-Arten	7
1.2	Speicher	11
1.2.1	RAM	11
1.2.2	Spezielle Speicherarten	13
1.2.3	Massenspeicher	14
1.2.4	Redundanz bei Festplatten mit <i>RAID</i>	18
1.3	Ein- und Ausgabe	22
1.3.1	Direct Memory Access	22
1.3.2	Interrupts	22
1.3.2.1	Software-Interrupts	24
1.3.2.2	Exceptions	24
1.3.3	Timer	24
1.3.4	Serielle Schnittstelle nach RS-232C bzw. V.24	25
1.4	Bussysteme	29
1.4.1	Systembusse	29
1.4.2	Peripheriebusse	30
1.4.3	Feldbusse	34
1.5	Standard-Architekturen	34
1.5.1	PC und Workstation	34
1.5.2	Systeme mit mehreren Prozessoren	35
1.5.3	Virtualisieren von Rechnern	37
1.5.4	Architekturen für Massenspeicher	39
1.5.5	Drei-Schichten-Architektur bei Software-Anwendungen	40
1.5	Standardarchitekturen	35
1.5.1	PC und Workstation	35
1.5.2	Systeme mit mehreren Prozessoren	36
1.5.3	Virtualisieren von Rechnern	37
1.5.4	Architekturen für Massenspeicher	39
1.5.5	Drei-Schichten-Architektur bei Software-Anwendungen	41
2	Ablage von Daten und Programmcode im Speicher	41
2.1	Daten-Kodierung und elementare Datenformate	41
2.1.1	Positive ganze Zahlen	41
2.1.2	Ganze Zahlen	41
2.1.3	Rationale Zahlen	43
2.1.3.1	Grundlagen: Binäre Darstellung rationaler Zahlen	43
2.1.3.2	Darstellung von Realzahlen nach IEEE 754	44
2.1.4	Alphanumerische Zeichen	46

2.2	Ausrichtung von Daten im Speicher	49
2.2.1	<i>Little Endian</i> und <i>Big Endian</i>	49
2.2.2	Alignment	50
2.3	Von der Hochsprache zum ausführbaren Binär-Code	52
3	Grundlagen von Betriebssystemen	56
3.1	Aufgaben von Betriebssystemen	56
3.2	Einteilung von Betriebssystemen	56
3.3	Struktur eines Betriebssystems	58
3.4	Programm-Architekturen von MS-Windows und UNIX	60
3.4.1	Programm-Architektur von MS-Windows	60
3.4.2	Programm-Architektur bei UNIX	60
3.5	Ablauf des Bootvorgangs am Beispiel eines INTEL-PCs	63
3.5.1	MS-DOS	64
3.5.2	MS-Windows9x, Windows-ME	64
3.5.3	Windows XP, Windows-NT	65
3.5.4	Linux	65
4	Mechanismen in Betriebssystemen	67
4.1	Prozessverwaltung	67
4.1.1	Verfahren zur Verwaltung der CPU-Rechenzeit	67
4.1.2	Task-Zustände	67
4.1.3	Speicher für Prozesse	69
4.1.4	Wichtige Systemfunktionen zur Prozessverwaltung	70
4.1.5	Threads	87
4.1.6	UNIX Prozess-Hierarchie	92
4.1.7	Steuern von Prozessen unter UNIX	93
4.2	Speicherverwaltung	95
4.2.1	Virtueller Speicher	95
4.2.2	Mithilfe des Prozessors bei der Speicherverwaltung am Beispiel des 80x86	97
4.2.2.1	Exkurs: Register-Modell und Adressierung des 8086	98
4.2.2.2	Adressierung des 80386 und seiner Nachfolger im Protected Mode	100
4.2.3	Speicher für Prozesse	103
4.3	Datei-Verwaltung	106
4.3.1	Datei-Formate aus Sicht des Betriebssystems	106
4.3.2	Organisation der Dateien in Verzeichnissen	106
4.3.3	UNIX-Dateiverwaltung	108
4.3.3.1	Wichtige UNIX-Verzeichnisse	108
4.3.3.2	UNIX-Dateinamen	108
4.3.3.3	Wildcards für Dateinamen unter UNIX	109
4.3.3.4	UNIX-Dateitypen und -Zugriffsrechte	109
4.3.4	Verwalten von Massenspeichern	110
4.3.4.1	Adressierung	110
4.3.4.2	Partitionen	111
4.3.4.3	Logical Volumes und dynamische Datenträger	113

4.3.5	Verwalten der Dateien in Dateisystemen	114
4.3.5.1	FAT-Dateisystem: MS-DOS, MS-Windows9x und MS-Windows ME	114
4.3.5.2	Dateisystem NTFS: Windows NT, Windows 2000 und Windows XP	116
4.3.5.3	UNIX-Dateisysteme	119
4.3.5.4	Zusammenfassung zu FAT, NTFS und UNIX-Dateisystem	122
4.3.5.5	Journaling File System	123
4.4	Ein- und Ausgabe-Verwaltung	126
4.4.1	Ein- und Ausgabe für Prozesse	126
4.4.1.1	Datenstrom-orientierte Ein- / Ausgabe	126
4.4.1.2	Koordinieren von Datei-Zugriffen	129
4.4.1.3	Nachrichten-orientierte Ein- / Ausgabe	130
4.4.2	Ein- / Ausgabe des Rechners zu Peripheriegeräten	134
4.4.2.1	Architektur von Gerätetreibern	134
4.4.2.2	Block- und Zeichen-orientierte Ein- und Ausgabe	135
4.4.2.3	Koordination von Geräte-Zugriffen: Blocking und Queueing	136
4.5	Interprozess-Kommunikation und Prozess-Synchronisation	137
4.5.1	Signale	137
4.5.1.1	Signal-Behandlung nach POSIX	139
4.5.1.2	Signal-Behandlung nach ANSI-C	143
4.5.1.3	Fallen bei der Signal-Behandlung	144
4.5.2	Pipes	145
4.5.3	FIFO-Dateien	149
4.5.4	Shared Memory	150
4.5.5	Semaphore	158
5	Mechanismen für Client-Server-Systeme	165
5.1	Strukturen verteilter Systeme	165
5.2	Remote Procedure Calls	166
5.2.1	Remote Procedure Calls aus Sicht der Anwendung	166
5.2.2	Remote Procedure Calls aus Sicht des Netzwerks	168
5.3	Sockets	169
5.4	Secure Shell, Remote Shell, Berkeley-Services	169
5.5	Verteilte Dateisysteme	171
5.5.1	Network File System	171
5.5.2	Server Message Blocks	173
5.5.3	Samba	174
5.6	System-Verwaltung bei verteilten Systemen	174
5.6.1	Konfigurationsdaten-Verwaltung mit NIS	174
5.6.2	Abilden von Rechner-Namen auf IP-Adressen mit DNS	177
5.6.3	Zugriff auf Verzeichnis-Dienste mit <i>LDAP</i>	179
5.6.4	Zeit-Synchronisation mit <i>NTP</i>	181
5.6.5	Booten über LAN	183

A	Abkürzungsverzeichnis	I
B	Literatur	III
B.1	Unix, Linux, Programmieren	III
B.2	Rechnerarchitekturen und Rechnerkomponenten	V
B.3	Ablage von Daten und Programmcode im Speicher	XI
B.4	Grundlagen von Betriebssystemen	XII
B.5	Mechanismen in Betriebssystemen	XIV
B.5.1	Verwalten von Massenspeichern	XIV
B.5.2	Verwalten der Dateien in Dateisystemen	XIV
B.5.3	Interprozess-Kommunikation und Prozess-Synchronisation	XV
B.5.4	Mechanismen für Client-Server-Systeme	XV
B.6	Sicherheit bei Rechnersystemen	XVI
B.7	Sonstiges	XVII
C	Index	XVIII

Abbildungsverzeichnis

1	Komponenten eines Rechners	2
2	“von-Neumann“-Architektur (schematisch)	3
3	“Harvard“-Architektur (schematisch)	3
4	Instruction-Pipeline des Prozessors Intel 80486 mit 5 Stufen	4
5	Skalarprozessor	5
6	Vektor-Prozessor	5
7	Ablauf bei einer Pipeline	5
8	Blockschaltbild des Mikroprozessors 8086	8
9	Blockschaltbild des Mikrocontrollers 80515	8
10	Blockschaltbild des Motorola DSP56002	9
11	Speicherpyramide	11
12	Prinzip von Burst-Zugriffen	12
13	Anordnung eines Cache-Speichers	13
14	Kopplung zweier Prozessoren mit Dual Ported RAM	14
15	Festplatte	14
16	Aufbau einer CD-R	17
17	Aufbau von beschreibbaren DVDs	17
18	Aufbau von Blu-ray Disks	17
19	RAID-0: Disk Striping	19
20	RAID-1: Plattenspiegelung	19
21	RAID-5: Rotating Parity Array	20
22	RAID-6	20
23	Beispiele von RAID-Kombinationen	21
24	Beispiel für DMA-Transfer	22
25	Programmablauf bei Polling	23
26	Programmablauf bei Interrupt-Behandlung	23

27	Typische Geräteanordnung zur Datenübertragung	25
28	Beispiele für typische Verbindungen über V.24	26
29	Signalpegel der V.24-Schnittstelle	27
30	Beispiele für die Übertragung der Zeichenfolge "äk" über eine V.24-Schnittstelle	28
31	Verbindungs-Topologien	32
32	Beispiel für Topologie bei USB	33
33	Typische PC-Architektur, Stand 2007	35
34	Strukturen für Rechner-Cluster	36
35	Struktur eines SMP-Rechners	36
36	Prinzip der Rechner-Virtualisierung	38
37	Beispiel für ein Storage Area Network	39
38	Beispiel für Network Attached Storage	40
39	Drei-Schichten-Architektur	40
32	Typische PC-Architektur, Stand 2007	35
33	Strukturen für Rechner-Cluster	36
34	Struktur eines SMP-Rechners	37
35	Prinzip der Rechner-Virtualisierung	38
36	Beispiel für ein Storage Area Network	40
37	Beispiel für Network Attached Storage	40
38	Drei-Schichten-Architektur	41
40	Format einer Realzahl nach IEEE 754 für <i>Single Precision</i> (32 Bit)	44
41	Byte-Anordnung einer Doppelwort-Variablen bei <i>Little Endian</i>	49
42	Byte-Anordnung einer Doppelwort-Variablen bei <i>Big Endian</i>	49
43	Beispiele zur Anordnung von Variablen im Speicher ohne und mit Alignment	50
44	Schritte vom Quellcode zum ausführbaren Binär-Code	52
45	Abstraktionsebenen von Programmiersprachen	52
46	Beispiel zur Umrechnung der Adressen durch den Linker	55
47	Allgemeines Schichtenmodell eines Rechners	56
48	Schichtenmodell eines Betriebssystems	59
49	Standard-Ein-/Ausgabekanäle von UNIX-Programmen	60
50	Verbinden von zwei Programmen durch Pipes	61
51	Befehlskette mit mehreren Programmen	62
52	Beispiel zum zeitlichen Ablauf der Task-Bearbeitung	68
53	Mögliche Zustände und Zustandsübergänge einer Task	68
54	Speicherbereiche unterschiedlicher Prozesse	69
55	Kopieren eines Prozess-Images durch fork()	70
56	Zeitlicher Ablauf bei fork()	70
57	Ablauf beim Aufruf eines Programms mit exec()	75
58	Ablauf beim Start eines anderen Programms in einem eigenen Prozess	78
59	Zeitlicher Ablauf beim Aufruf von ModifyThreadData() als Funktion	87
60	Vergleichbarer Ablauf, wenn ModifyThreadData() als Thread gestartet wird	87
61	Stack nach starten eines zweiten Threads	91
62	Beispiel für Prozess-Hierarchie	93
63	Speicherbereiche und deren verwaltende Instanzen	95
64	Virtuelle Speicher-Verwaltung	96
65	Register des 8086	98
66	Register des 8085 zum Vergleich	98

67	Adressberechnung beim 8086	98
68	Beispiel zur Adressierung von Speichersegmenten	99
69	Register des 80386	100
70	Adressberechnung des 80386 und seiner Nachfolger im Protected Mode	101
71	Ausschnitt des UNIX-Verzeichnisbaums	107
72	Verkettung der Verzeichnis-Dateien	107
73	Festplatte mit 2 Partitionen	111
74	Weitere Beispiele zu partitionierten Festplatten	112
75	Beispiel zu Logical Volumes	113
76	Verwaltung des FAT-Dateisystems	114
77	Beispiel zum Verwenden mehrerer FAT-Dateisysteme	116
78	Verwaltung des NT-Dateisystems	117
79	Beispiel zum Verwenden mehrerer Dateisysteme bei NTFS5	118
80	Verwaltung des UNIX-Dateisystems	119
81	Beispiel einer UNIX-Datei mit 3 Links	120
82	Beispiel eines symbolischen Links	121
83	Beispiel für das Einbinden mehrerer Dateisysteme unter UNIX	122
84	Beispiel zum prinzipiellen Ablauf ohne Journaling	123
85	Beispiel zum prinzipiellen Ablauf mit Journaling	124
86	Ein- / Ausgabe mit Datenströmen	126
87	Ein- / Ausgabe mit und ohne Puffer	127
88	Beispiel für Datei-Zugriffe parallel laufender Prozesse	129
89	Ereignis-gesteuertes System	133
90	Typische Struktur von Programmen mit graphischer Oberfläche	133
91	Einbettung der Gerätetreiber in den Kernel	134
92	Architektur von Gerätetreibern	134
93	Zeichen-orientierte Ein- / Ausgabe	135
94	Unterbrechung des Programmablaufs durch Signal-Behandlung	137
95	Ändern einer Signal-Maske	140
96	Pipe zwischen zwei Prozessen	145
97	Aufbau einer Pipe zur Datenübertragung vom Kind- zum Eltern-Prozess	146
98	Kommunikation über eine FIFO-Datei	149
99	Zwei Prozesse mit gemeinsamem Shared Memory	150
100	Semaphore	159
101	Beispiele zu Peer-to-Peer- und Client-Server-Architektur	165
102	Prinzip von Remote Procedure Calls	166
103	Remote Procedure Calls im OSI Schichtenmodell	168
104	Beispiel für Secure Remote Shell	170
105	Beispiel für Secure Copy	170
106	Remote Shell als Filter	171
107	Network File System	172
108	Beispiel für NIS	175
109	Beispiel für Anordnung von NIS-Servern in LAN-Segmenten	176
110	Beispiel für DNS-Domänen und mögliche Einteilung der DNS-Zonen	177
111	Beispiel zur Organisation der Daten in einem Verzeichnis-Dienst	180
112	Zugriff auf Daten eines Verzeichnis-Dienstes mittels LDAP	180
113	Beispiele für die Zeit-Synchronisation mit dem <i>Network Time Protocol</i>	182

Tabellenverzeichnis

1	Häufig verwendete Signale bei asynchroner Übertragung mit RS-232	25
2	Parallele SCSI-Varianten	31
3	Kodierung von ganzen Zahlen mit Zweier-Komplement für negative Werte	41
4	Beispiele für <i>Signed Integer</i> -Werte mit 8-Bit	42
5	Kodierung nach IEEE 754 für <i>Single Precision</i> (32 Bit)	44
6	Formeln zum Dekodieren einer im <i>Single Precision</i> Format nach IEEE 754 dargestellten Zahl	45
7	Parameter der Datenformate nach IEEE 754	45
8	Standard US ASCII-Zeichensatz	47
9	Einige nationale 7-Bit-Zeichensätze nach ISO	47
10	Zeichensatz ISO 8859-15 = Latin9	47
11	Legende der Steuerzeichen	48
12	Run-Levels bei Linux entsprechend Linux Standard Base	65
13	Adressierungs-Eigenschaften von Prozessoren der Familie Intel 80x86	97
14	Default-Segmentregister bei Speicherzugriffen des 8086	100
15	Kennnummern einiger Partitions-Typen	113
16	Datei-Attribute des FAT-Dateisystems	115
17	Kennzahlen der verschiedenen FAT-Dateisysteme	116
18	Übersicht zur Verwaltung der Dateisysteme	122
19	Korrespondierende High-Level- und Low-Level-I/O-Funktionen	127
20	Namen und Bedeutung einiger Signale	138
21	Funktionen zur Manipulation von Variablen vom Typ <code>sigset_t</code>	140
22	Funktionen, die laut POSIX.1-Standard reentrant sind	144
23	Auswahl wichtiger Befehle für NIS	174
24	Hierarchie der Datenbank-Struktur eines Verzeichnis-Dienstes	179

Verzeichnis der Programmbeispiele

1	Ausrichtung von Variablen im Arbeitsspeicher: align_bsp.c	51
2	mittelwert	54
3	Eltern- und Kind-Prozess in der einfachsten Form: fork_simple.c	71
4	Typischer Ablauf zum Starten eines Kind-Prozesses	72
5	Eltern- und Kind-Prozess mit unterschiedlichem Ablauf: childproc.c	72
6	Programmaufruf mit exec(): exec_simple.c	76
7	Typischer Ablauf zum Starten eines Programms in einem eigenen Prozess	78
8	Start eines Programms in einem eigenen Prozess: startprog.c	79
9	Exit-Status eines Kind-Prozesses auswerten: exitstatus.c	85
10	Threads: pthread.c	88
11	Verarbeiten des Ergebnisses von realloc()	104
12	Index-Verwaltung mit dynamischem Speicher	105
13	Typische Struktur von Programmen mit graphischer Oberfläche: Auszug aus gui_bsp.c .	131
14	Ändern der Signal-Behandlung für SIGTERM nach POSIX: Ausschnitt aus sig_posix.c .	141
15	POSIX-Signalbehandlung: sig_posix.c	141
16	Erzeugen einer Pipe vom Kind- zum Eltern-Prozess: pipe.c	147
17	Shared Memory, Teil 1: shm_common.h	152
18	Shared Memory, Teil 2: shm_send.c	153
19	Shared Memory, Teil 3: shm_recv.c	156
20	Auszug aus sem_demo.c	160