

Grundlagen der Informatik III

Wintersemester 2010/2011

Wolfgang Heenes, Patrik Schmittat



TECHNISCHE
UNIVERSITÄT
DARMSTADT

12. Aufgabenblatt

07.02.2011

Hinweis: Der Schnelltest und die Aufgaben sollen in den Übungsgruppen bearbeitet werden. Die Hausaufgaben sind in der Kalenderwoche 7 (14.02. bis 18.02.) bei den Tutoren in **physikalischer Form** (handschriftlich oder gedruckt) abzugeben. Bei allen Abgaben ist der Name des Tutors und die Übungsgruppe deutlich anzugeben. Bei Teamabgaben wird nur eine Lösung eingereicht, die alle Namen der Teammitglieder enthält.

Aufgabe 1: Schnelltest

Fragen	Antworten
1. Was gehört zu den Aufgaben des virtuellen Speichers?	<input type="checkbox"/> Schutz von Programmen und Daten vor Zugriff durch Prozesse. <input type="checkbox"/> Ermöglicht Programme, deren Größe die des Hauptspeichers überschreiten. <input type="checkbox"/> Unterstützung der sequentiellen Ausführung von Programmen. <input type="checkbox"/> Ausnutzung von zeitlicher und räumlicher Lokalität.
2. Welche der folgenden Aussagen über eine Seitentabelle (page table) sind richtig?	<input type="checkbox"/> Die virtuelle Adresse steht in der Seitentabelle. <input type="checkbox"/> In der Seitentabelle wird für jeden Eintrag ein Valid-Bit gespeichert. <input type="checkbox"/> Die Seitentabelle wird in einem optimierten Speicher abgelegt. <input type="checkbox"/> Die Seitentabelle kann in den Hintergrundspeicher ausgelagert werden.
3. Caches und virtueller Speicher können folgendermaßen verglichen werden:	<input type="checkbox"/> Die Assoziativität in Caches entspricht der Kachelgröße im virtuellen Speicher. <input type="checkbox"/> Blöcke und Blockgröße in Caches entsprechen Seiten und Seitengrößen. <input type="checkbox"/> Der Cache-Index entspricht der Seitennummer. <input type="checkbox"/> Ein Cache-Miss entspricht einem Seitenfehler (page fault). <input type="checkbox"/> Caches und virtueller Speicher können nicht sinnvoll verglichen werden.
4. Welche der Aussagen zu Dateien und Dateisystemen sind richtig?	<input type="checkbox"/> Die Dateierweiterung/Dateiendung besteht immer aus ein bis drei Zeichen. <input type="checkbox"/> Alle Dateisysteme unterscheiden zwischen Groß- und Kleinschreibung. <input type="checkbox"/> Ein Dateisystem eignet sich i.A. für alle Speichermedien (Platte, DVD, Band, ...) gleich gut. <input type="checkbox"/> Freispeicher wird häufig durch verkettete Listen oder Bitmaps verwaltet.

Aufgabe 2: Interrupts

a) Die folgenden Interrupts mit Prioritäten $prio$ laufen zu den angegebenen Zeiten t auf. Der Wert d gibt die Zeiteinheiten an, die jeweils zur Bearbeitung eines Interrupts benötigt werden:

- Clock $prio = 28$, $t = \{2, 11, 20\}$, $d = 1$
- Device 1 $prio = 3$, $t = \{0, 12, 17\}$, $d = 6$
- Device 2 $prio = 4$, $t = \{15\}$, $d = 2$
- Device 3 $prio = 5$, $t = \{4, 13, 21\}$, $d = 3$

Ein Interrupt mit hoher Priorität (= niedriger Wert) kann dabei einen Interrupt mit niedrigerer Priorität (= hoher Wert) unterbrechen.

Erarbeiten Sie eine Grafik aus der ersichtlich ist, welche Interrupts zu welchem Zeitpunkt bearbeitet werden.

- b) Angenommen, ein Rechner kann ein Speicherwort in 10 ns lesen bzw. schreiben. Für jeden Kontextwechsel (z. B. wenn ein Interrupt eintrifft) müssen alle 32 CPU-Register sowie das PC- und PSW-Register¹ auf den Stack gesichert werden. Wie viele Interrupts kann das System pro Sekunde maximal verarbeiten?
- c) Moderne Architekturen besitzen häufig mehrere Registersätze (register windows), typischerweise zwischen 2 und 32 Stück. Angenommen, bei einem Interrupt müssen nur noch PC- und PSW-Register sowie das Register CWP (current window pointer, zeigt auf das aktuelle window), gesichert werden. Wie viele Interrupts kann ein solches System maximal verarbeiten? Alle weiteren Angaben wie in Aufgabenteil b). Benennen Sie eine weitere Situation, bei der register windows eine Effizienzsteigerung ermöglichen.

Aufgabe 3: Translation Lookaside Buffer – TLB

Untersucht werden Speicherzugriffe auf einem System mit 256 MByte Hauptspeicher, 1 GByte virtuellem Adressraum, 64 KByte Kachel-/Seitengröße und vollassoziativem Translation Lookaside Buffer mit 4 Einträgen (Ersetzungstrategie LRU).

- a) Wie groß ist der TLB (ohne Verwaltungsinformationen)?
- b) Füllen Sie die folgenden Tabellen aus. Felder, die keinen Eintrag erhalten sollen, sind mit “-“ zu kennzeichnen. Nehmen Sie an, dass zu Beginn sowohl der Translation Lookaside Buffer noch nicht gefüllt ist und dass die angeforderten Seiten sich zunächst nicht im Speicher befinden. Die einzelnen auszufüllenden Tabellen sollen jeweils für sich alleine betrachtet werden.

Zugriff <i>virtl.Adresse</i>	Seitenfehler	virtuelle Seitennummer
0x134F B00F		
0x3D49 1512		
0x2010 5234		
0x4F45 6344		
0x1467 2324		
0x134F 4326		
0x0348 2342		
0x2321 2321		
0x1421 4638		
0x4235 7427		

¹ Prozessorstatuswort

Zugriff <i>virtl.Adresse</i>	hit/miss im TLB		virtuelle Seitennummer (TLB)	
			neu eingelagert	ersetzt
0x1467 4327				
0x3D49 1278				
0x0010 1144				
0x1466 1284				
0x1348 B756				
0x1467 1A34				
0x1466 0012				
0x0010 6B35				
0x2321 34C1				
0x3D49 2138				
0x2235 4327				

Anmerkung: die erste Tabelle enthält virtuelle Adressen, die mit einem virtuellen Adressraum von 1 GByte nicht darstellbar sind. Dies sind diejenigen Adressen, deren vordeste Hex-Ziffer größer als 3 ist.

Aufgabe 4: Seitenflattern

Gegeben ist ein System mit virtuellem Speicher und der Seitenersetzungsstrategie LRU. Die Seitengröße beträgt 4096 Bytes, ein `int` ist 4 Bytes groß. Die Matrix

```
int m[256][256]; /* 256 x 256 Matrix */
```

beginnt mit der zweiten Seite des virtuellen Adressraums (d. h. die virtuelle Adresse von `m[0][0]` ist 4096). Beachten Sie, dass Matrizen in C zeilenweise abgespeichert werden.²

Das Programm liegt in der ersten Seite des virtuellen Adressraums.

1.

```
for(i=0; i<=255; i++){
  for(k=0; k<=255; k++){
    m[i][k] = 0;
  }
}
```

2.

```
for(i=0; i<=255; i++){
  for(k=0; k<=255; k++){
    m[k][i] = 0;
  }
}
```

Einem Prozess stehen zur Ausführung einer dieser beiden Programmvarianten drei physische Speicherseiten zur Verfügung. Das Programm befindet sich bereits in Seite 1, zwei weitere Seiten sind noch frei.

Wieviele Seitenfehler werden von den beiden Varianten (1) und (2) erzeugt?

Aufgabe 5: Virtueller Speicher

Ein Prozess bestehe aus 5 Seiten im virtuellen Adressraum und habe zwei Kacheln mit den Adressen A1 und A2 im Hauptspeicher zur Verfügung. Die Referenzreihenfolge auf die virtuellen Seiten sei wie in nachstehender Tabelle angegeben, wobei n_l einen Lesezugriff und n_s einen Schreibzugriff auf die n -te Seite bedeutet. Es wird LRU als Seitenersetzungsstrategie angenommen.

a) Führen Sie Buch über die Speicherbelegung, indem Sie die leeren Felder in der Tabelle wie folgt ausfüllen:

- Unter **Seitenfehler** sollen Sie eintragen, ob die jeweilige Referenz zu einem Seitenfehler führt (ja) oder nicht (nein).
- Unter **A1** und **A2** sollen Sie die Seite angeben, die sich nach der Referenz in der jeweiligen Speicheradresse befindet.

² Bei der Notation `m[i][k]` bezeichnet also `k` den Zeilen- und `i` den Spaltenindex, wobei alle Elemente einer Zeile aufeinanderfolgend im Speicher liegen.

- Unter **zurückgeschrieben** sollen Sie die Seite angeben, die auf die Platte zurückgeschrieben werden muss. Ist kein Zurückschreiben erforderlich, so tragen sie „-“ ein.

Referenz	Seitenfehler	A1	A2	zurückgeschrieben
1 _l			-	-
2 _s				-
4 _s				
2 _l				
5 _l				
3 _s				
2 _s				
3 _l				
1 _s				

- b) Führen Sie Buch über den Inhalt der Seitentabelle des Prozesses, welche aus einer *P*-Spalte (Present-Bit) und einer *Adr.*-Spalte (physische Speicheradresse) besteht. In den unteren Tabellen ist zunächst der Zustand der Seitentabelle vor der Ausführung angegeben. Dabei steht die *i*-te Zeile jeder Tabelle für die *i*-te virtuelle Speicherseite. Tragen Sie für jeden Zugriff nur die Änderungen gegenüber dem jeweiligen Vorzustand der Tabelle ein.

	1 _l	2 _s	4 _s	2 _l																																																													
	<table border="1" style="display: inline-table;"><tr><th>P</th><th>Adr.</th></tr><tr><td>0</td><td>A2</td></tr><tr><td>0</td><td>A1</td></tr><tr><td>0</td><td>A1</td></tr><tr><td>0</td><td>A2</td></tr><tr><td>0</td><td>A1</td></tr></table>	P	Adr.	0	A2	0	A1	0	A1	0	A2	0	A1	<table border="1" style="display: inline-table;"><tr><th>P</th><th>Adr.</th></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr></table>	P	Adr.											<table border="1" style="display: inline-table;"><tr><th>P</th><th>Adr.</th></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr></table>	P	Adr.											<table border="1" style="display: inline-table;"><tr><th>P</th><th>Adr.</th></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr></table>	P	Adr.																							
P	Adr.																																																																
0	A2																																																																
0	A1																																																																
0	A1																																																																
0	A2																																																																
0	A1																																																																
P	Adr.																																																																
P	Adr.																																																																
P	Adr.																																																																
	5 _l	3 _s	2 _s	3 _l	1 _s																																																												
	<table border="1" style="display: inline-table;"><tr><th>P</th><th>Adr.</th></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr></table>	P	Adr.											<table border="1" style="display: inline-table;"><tr><th>P</th><th>Adr.</th></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr></table>	P	Adr.											<table border="1" style="display: inline-table;"><tr><th>P</th><th>Adr.</th></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr></table>	P	Adr.											<table border="1" style="display: inline-table;"><tr><th>P</th><th>Adr.</th></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr></table>	P	Adr.											<table border="1" style="display: inline-table;"><tr><th>P</th><th>Adr.</th></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr></table>	P	Adr.										
P	Adr.																																																																
P	Adr.																																																																
P	Adr.																																																																
P	Adr.																																																																
P	Adr.																																																																

Hausaufgabe 1: Adressumrechnung (10 Punkte)

Für ein Computersystem mit virtuellem Speicher sind die folgenden Daten bekannt:

- 11 Bit Seitenoffset (Byteadressierung)
- 32 virtuelle Seiten
- 8 physische Seiten

- Wie groß ist der virtuelle Adressraum des Systems?
- Wieviel physischen Speicher hat das System?
- Wieviele Bits werden für die Adressierung der virtuellen Seiten benötigt?
- Wieviele Bits werden für die Adressierung der physischen Seiten benötigt?

e) Betrachten Sie die nachstehende Momentaufnahme der Seitentabelle eines Prozesses. Geben Sie für die folgenden Zugriffe auf virtuelle Speicheradressen die vom Betriebssystem zu ermittelnde physische Speicheradresse an. Führt der Zugriff zu einem Seitenfehler, so schreiben Sie unter Phys. Adresse einfach „Seitenfehler“.

0x42ed, 0xb84b, 0xaf1e, 0xe6d8, 0x1f2c, 0x1348, 0xa324, 0xe5b6
 0xb8ed, 0x4209, 0xa0d3, 0xb81e, 0x5d93, 0x95d7, 0x10db, 0xe11d

Dabei sollen die Zugriffe unabhängig von einander erfolgen. Sie sollen also keine Ein- oder Auslagerung von Seiten betrachten, so dass sich die Seitentabelle nicht ändert.

0	0	7	8	1	2	16	0	2	24	1	7
1	0	5	9	0	4	17	0	2	25	0	1
2	0	6	10	0	7	18	1	1	26	0	6
3	0	3	11	0	6	19	0	5	27	1	6
4	0	3	12	0	5	20	1	0	28	1	4
5	0	4	13	0	2	21	1	5	29	0	6
6	0	4	14	0	1	22	0	2	30	0	4
7	0	6	15	0	6	23	1	3	31	0	4

<u>Virt. Adresse</u>	<u>Seitennr.</u>	<u>Seitenoffset</u>	<u>Phys. Adresse</u>
0x42ed			
0xb84b			
0xaf1e			
0xe6d8			
0x1f2c			
0x1348			
0xa324			
0xe5b6			
0xb8ed			
0x4209			
0xa0d3			
0xb81e			
0x5d93			
0x95d7			
0x10db			
0xe11d			