

Midterm 29.3.2012

donn, ZHAW Dauer: 45 Minuten

Name:

Klasse:

Aufgabe	Max. Punkte	Punkte
2	1.0	
3	2.0	
4	1.0	
5	2.0	
6	2.0	
7	2.0	
Total	12.0	
	Note	

Hinweise

- Korrektur: Resultate werden nur als richtig oder falsch bewertet - bei Multiple Choice Aufgaben kann pro Teilaufgabe mehr als eine Antwort zutreffen, eine Teilaufgabe gilt als richtig, wenn sämtliche Antworten der Teilaufgabe korrekt sind.

Hilfsmittel

- erlaubt sind Taschenrechner, 2 Blätter selbstverfasste Notizen - nicht erlaubt sind PCs (Laptops), Handhelds/xPads und Handys.

Bedingungen

- kein Bleistift, keine roten Stifte und kein TipEx
- nur die Resultate auf den Aufgabenblättern eintragen
- Zusatzblätter werden nicht eingesammelt oder korrigiert
- Resultate als ganze Zahlen resp. mit 2 Stellen nach dem Komma

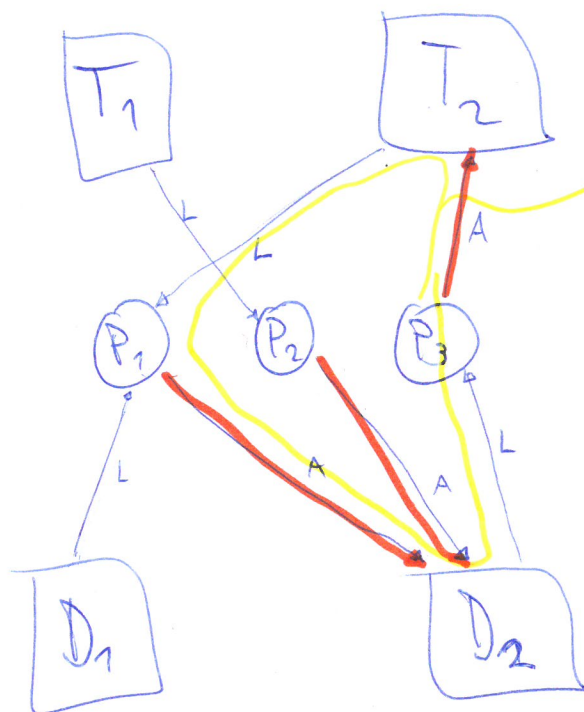
Aufgabe 2 Ressourcengraphen

1.0P

Ein Rechnersystem besitzt zwei Tapestationen (T_1 , T_2) und zwei Disks (D_1 , D_2). Zur Zeit laufen drei Prozesse (P_1 , P_2 , P_3), wobei folgendes gilt:

- Prozess P_1 kopiert Daten von Disk D_1 auf die Tapestation T_2 und möchte Daten auf den Disk D_2 schreiben
- Prozess P_2 hat Tapestation T_1 alloziert und möchte Daten auf Disk D_2 schreiben
- Prozess P_3 hat Disk D_2 alloziert und möchte Daten nach Tapestation T_2 kopieren

Ist das eine Deadlocksituation, wenn die Ressourcen exklusiv alloziert werden und wenn *möchte schreiben* das Gleiche wie anfordern bedeutet? Begründen Sie Ihre Antwort (Ressourcengraphen zeichnen und analysieren).



DL Bedingungen für Deadlock:

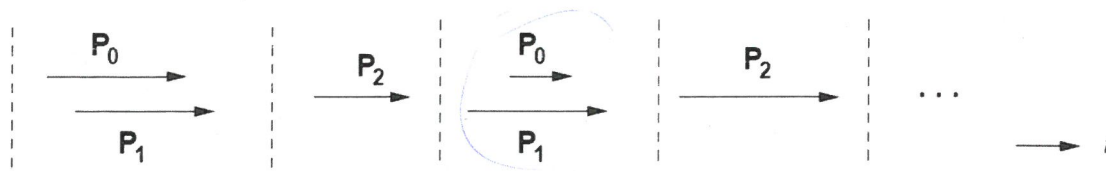
1. Circular Wait
2. Mutual Exclusion
→ nur 1 Prozess kann Ressource allozieren
3. no preemption
4. hold + wait
→ Prozesse warten bis Ressource frei wird und
laufen 2. Bsp nicht in Timeout

→ sind hier alle gegeben
→ haben Deadlock

Aufgabe 3 Semaphore

2.0 P

Gegeben sind drei Prozess P_0 , P_1 , und P_2 die nach folgendem Schema abgearbeitet werden soll:



Die Verarbeitung startet mit den beiden Prozessen P_0 und P_1 , die parallel verarbeitet werden sollen (es spielt keine Rolle, welcher der beiden Prozesse zuerst mit seiner Verarbeitung beginnt oder aufhört). Wenn beide Prozesse eine Iteration ihrer Funktion `working(x)` beendet haben, folgt Prozess P_2 , etc.

Schreiben Sie Pseudocode mit maximal 3 Semaphoren S_0 , S_1 und S_3 , der garantiert, dass die oben skizzierte Reihenfolge eingehalten wird. Verwenden Sie dazu **ausschliesslich** Befehle der Form `up(S_0)` und `down(S_0)`, etc. Geben Sie an, wie die Semaphore initialisiert werden müssen.

P_0	P_1	P_2
Sem S_0 <u>1</u> Sem S_1 <u>1</u> Sem S_2 <u>0</u> <pre> while{1} { down(S_0) down(S_2) working(0); up(S_0) up(S_2) } </pre>	<pre> while{1} { down(S_1) down(S_2) working(1); up(S_1) up(S_2) } </pre>	<pre> while{1} { down(S_2) down(S_0) down(S_1) down(S_2); down(S_2); working(2); up(S_2) up(S_0); up(S_1); } </pre>

Aufgabe 4 Buddy System

1.0 P

Ein Betriebssystem-Kernel verwaltet seine Datenbuffer mit einem Buddy System, wobei insgesamt 8MByte Speicher zur Verfügung stehen. Zur Zeit sind folgende Buffer mit 62KByte, 34KByte und 9KByte alloziert worden. Wie viel Speicher geht dabei durch interne Fragmentierung insgesamt verloren, Angabe in KByte:

8M = 8192 KB

interne Fragmentierung KByte:

$$64 - 62 = 2$$

$$64 - 34 = 30$$

$$16 - 9 = 7$$

$$2 + 30 + 7 = 39 \text{ KB}$$

Aufgabe 5 Page Tabellen

2.0 P (je 0.5 P)

Ein Prozessor besitzt eine Wortbreite von 32Bit. Pointer (Adressen) werden in 32Bit Worten gespeichert, aber nur die 24 tieferwertigen Bits werden für die Adressbildung verwendet (Bits 25-31 sind auf 0 gesetzt).

Die logische Adresse ist wie folgt strukturiert:

6-Bit Page Directory	8-Bit Page Nummer	10-Bit Offset
----------------------	-------------------	---------------

- a) Wie gross ist eine Page, Angabe in KBytes?

10 Bit \rightarrow 1024 Adressen
~~32 Bit = 4 Byte pro Adresse~~ \rightarrow 1 KB Page size

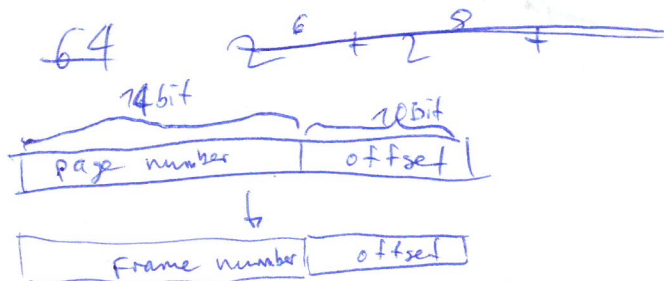
- b) Wie viele Bytes enthält das Page Directory, wenn pro Eintrag ein Pointer (Adresse) auf eine Page Tabelle eingetragen wird, Angabe in KBytes?

6 Bit \rightarrow 64 Adressen
 4 Byte pro Adresse $\rightarrow 64 \cdot 4 = 256 \text{ Bytes}$

- c) Wie viele Page Tabellen kann ein Prozess maximal haben?

6 Bit \rightarrow 64 Page Tabellen

- d) Wie viele Frames kann das System maximal haben?



$$2^6 \cdot 2^{10} = 2^{16}$$

$$64 \cdot 256 = 16 \text{ k}$$

Aufgabe 6 Page Replacement

2.0 P

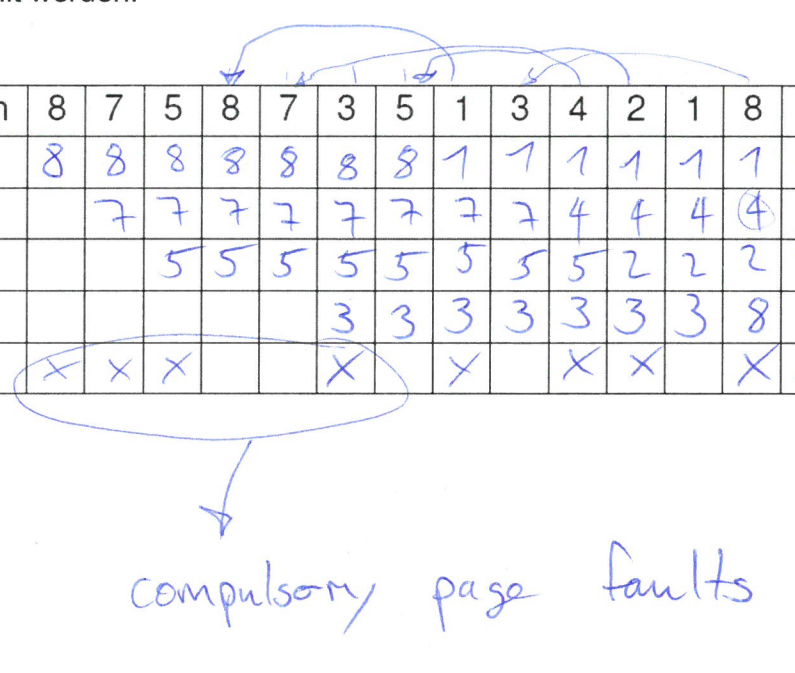
Ein Prozess referenziert der Reihe nach folgende Pages:

8 7 5 8 7 3 5 1 3 4 2 1 8 3 1 2

Gehen Sie davon aus, dass zu Beginn keine Pages im Speicher stehen und dass auch das erstmalige Laden einer Page als Page Fault gezählt wird (demand paging). Pro Prozess stehen 4 Frames zur Verfügung.

Tragen Sie in unten stehender Tabelle die den Frames zugewiesenen Pages für den **Least Recently Used** Algorithmus. Das Page dessen Zugriff am weitesten Zeitlich zurückliegt wird dann mit ein neues Page ersetzt.

Markieren Sie die Spalten mit einem Stern, wo ein Page Fault auftritt. Nehmen Sie an, dass ausschliesslich Demand Paging verwendet wird. Wenn mehrere Frames für das placement resp. replacement in Frage kommen, muss der Frame mit der kleinsten Nummer gewählt werden.



Referenzen	8	7	5	8	7	3	5	1	3	4	2	1	8	3	1	2
frame 1	8	8	8	8	8	8	8	1	1	1	1	1	1	1	1	1
frame 2		7	7	7	7	7	7	7	7	4	4	4	4	3	3	3
frame 3			5	5	5	5	5	5	5	5	2	2	2	2	2	2
frame 4						3	3	3	3	3	3	3	8	8	8	8
page fault	x	x	x			x		x		x	x		x	x		

compulsory page faults

Aufgabe 7 Kurzfragen

0.5P pro Antwort, total 2P

Pro Teilfrage können eine, mehrere oder keine Antworten zutreffen, kreuzen Sie die richtige(n) Antwort(en) an.

Bewertung: wenn Sie eine Teilfrage richtig beantwortet haben (sämtlichen Kästchen korrekt), erhalten Sie 0.5 Punkte sonst 0 Punkte.

a) Welche der folgenden Aussagen treffen zu?

- Alle Mutexes können mit Semaphoren realisiert werden
- Semaphore können in allen Fällen durch Mutexes ersetzt werden
- Mutexes sollten immer anstelle von Semaphoren eingesetzt werden, weil es dann keine Deadlocks geben kann.
- Mit Semaphoren lässt sich die Verarbeitungsgreihenfolge von Prozessen und Threads erzwingen

<input checked="" type="checkbox"/>
<input type="checkbox"/>
<input type="checkbox"/>
<input checked="" type="checkbox"/>

b) Welcher der folgenden Aussagen treffen zu?

- Pages müssen grösser als Frames dimensioniert werden *gleich gross*
- Es müssen mindestens so viele Pages wie Frames in einem System vorhanden sein *falls nicht haben wir einfach nicht benutzte memory*
- Sowohl interne *ja* wie auch externe *nein* Fragmentierung treten bei Paging auf
- Pages und Frames müssen gleich gross dimensioniert werden

<input type="checkbox"/>
<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>

c) Welcher der folgenden Aussagen treffen zu?

- Ein MMU übersetzt Logischen Adressen zu Physikalische Adressen
- Swap in bedeutet dass ein Prozess auf die Hard-Disk verlagert wird
- Ein Prozess der auf der Harddisk verlagert worden ist kann sich im Zustand Running befinden *Teile vom running Prozess können aber ausgelagert sein, der Prozess selber aber nicht*

<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>

d) Welcher der folgenden Aussagen treffen zu?

- Bei Static Partitioning tritt External Fragmentation auf *internal tritt auf*
- Bei Dynamic Partitioning tritt External Fragmentation auf
- Nur bei Best Fit Allocation wird kein Compaction benötigt

<input type="checkbox"/>
<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>

↳ kann man immer brauchen