

Matrikelnummer:



Johann Wolfgang Goethe-Universität  
Frankfurt am Main

FB 15 Institut für Informatik *Praktische Informatik* PD Dr. R. Brause

## Musterlösung zur KLAUSUR

zur Vorlesung Betriebssysteme I WS 2003/04

--	--

Name

Vorname

Zugelassen sind - ausser einem (nicht-programmierbaren) Taschenrechner - keine Hilfsmittel.

Die Klausur dauert 120 Minuten.

Beschriften Sie bitte alle verwendeten Antwortblätter mit Ihrer Matrikelnummer. Blätter ohne Matrikelnummer können nicht in die Bewertung eingehen.

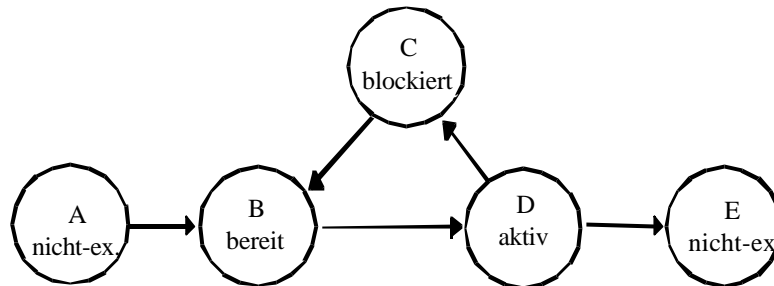
Bitte kontrollieren Sie die Vollständigkeit dieses Aufgabensatzes. Er umfasst 9 Seiten.

**Viel Erfolg!!!**

I. Multiple Choice-Aufgaben (Mehrfachantworten sind möglich !) 11 Punkte

1. Bei welchen Prozesszuständen arbeiten Warteschlangen?

- A
- B
- C
- D
- E



2. Was ist ein Interrupt-Vektor?

- Array, das alle Interrupts enthält
- Adresse der Speicherzellen für PC- und PS-Register bei einem Interrupt
- lineare Mannigfaltigkeit im Zwischenraum

3. Was ist eine Interrupt Service Routine?

- Die Updatefunktion des Betriebssystems
- Für einen Interrupt aufzurufender Betriebssystem-Code
- wird nur von Softwareinterrupts benutzt
- schneller Onlinesupport

4. Zu einer virtuellen Maschine gehören unbedingt

- Schnittstellen zu ihren Funktionen
- Schnittstellen zu einer realen Maschine
- Schnittstellen zum Betriebssystem

5. Kernel-Mode bedeutet

- die Möglichkeit, Arbeitsspeicher zu segmentieren („Kerne“)
- gibt es nur bei Unix
- Modus in dem betriebssystem-immanente Arbeiten durchgeführt werden
- Kernbetrieb bei Notfallsituationen (Stromausfall)

? Betriebsmodus der vom Benutzer nicht manipuliert werden kann

6. Ein Hardlink

- kann immer nur von dessen Besitzer gelöscht werden
- ? ist nur innerhalb eines physischen Datenträgers möglich
- generiert immer einen Zykel

7. Threads

- sind für Multiuser-Betrieb notwendig
- ? ermöglichen parallele Ausführung von Programmabschnitten
- ermöglichen die Portierung von Programmen zwischen Unix und NT
- ? müssen beim Zugriff auf gemeinsame Variablen synchronisiert werden
- ermöglichen Multiprozessorkommunikation

8. Ein blockierter Light-Weight-Thread (LWT) blockiert immer

- alle anderen (aktiven) LWTs des Prozesses
- alle I/O Operationen des Prozesses
- den Prozessor

KEINE ANTWORT RICHTIG!

9. Im allgemeinen Fall ist „ $n$  Tasks auf  $m$  Prozessoren zu verteilen“

- in  $O(n*m)$  zu lösen
- ? einfach, wenn  $n$  kleiner ist als  $m$
- ? ist NP-vollständig für  $m > 2$

10. Bei welcher Seitenersetzungsstrategie kann es zur Belady's Anomalie kommen?

- LRU
- NFU
- ? FIFO

11. Lokalitätsprinzip (bezüglich Seitenreferenzen) bedeutet

- dass alle Seiten gleichwahrscheinlich referenziert werden
- ? dass wenige Seiten häufig und viele Seiten wenig referenziert werden
- es gibt zwei Mengen  $M1$  und  $M2$  von Seiten, die konstante Zugriffswahrscheinlichkeiten haben
- dass die Working-Sets aller Prozesse in den Speicher passen (Normalbetrieb)

## II. Textaufgaben

### 12. Multiprozessor-Scheduling

3, 3, 3 Punkte

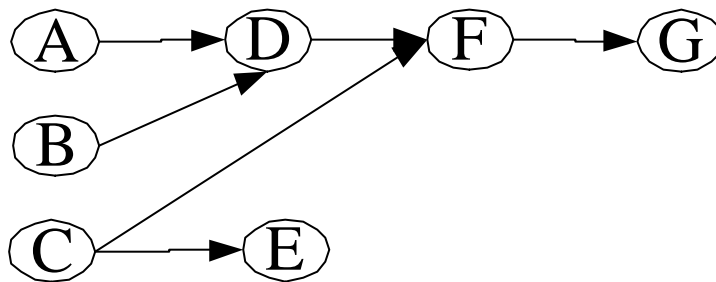
Gegeben ist folgende Präzedenzrelation:

A >> D >> F  
 B >> D >> F >> G  
 C >> E  
 C >> F

mit den Ausführungszeiten:

A – 3, B – 4, C – 6, D – 4, E – 7, F – 5 und G – 2

a) Zeichnen Sie zur angegebenen Präzedenzrelation den zugehörigen Präzedenzgraphen.



b) Zeichnen sie mit obigen Angaben ein Gantt-Diagramm für drei Prozessoren nach der *earliest*-Methode. Schraffieren Sie belegte Blöcke und beschriften diese mit den Prozessnamen.

P1	A		D		F		G									P1
P2	B															P2
P3	C			E												P3

c) Zeichnen sie mit obigen Angaben ein Gantt-Diagramm für drei Prozessoren nach der *latest*-Methode. Schraffieren Sie belegte Blöcke und beschriften diese mit den Prozessnamen.

P1							A	D	F	G						P1
P2								C		E						P2



Matrikelnummer:

<1024>

Ein Eintrag mit  $\diamond$  geklammert, bedeutet: Buddy gefunden,  
 $\langle\langle\rangle\rangle$  bedeutet: Einfach verschachtelter Buddie gefunden.

b) Beschreiben Sie kurz, was passiert, wenn die Seite D 410 kB gross ist

Dann wird Seite D mit Fehlermeldung „Page-Error“ oder ähnlichem abgewiesen.

c) Wie viele Adressen (Einträge) stehen minimal und maximal in einer Freispeicherliste zu „Buddy-Speichern“ mit Speichergrösse  $k$  und minimal allozierbaren Blöcken der Grösse  $m$ ?

Minimal:  $0$

Kein Speicher frei -> Freispeicherliste leer -> 0 Einträge

Maximal:  $m/2k$

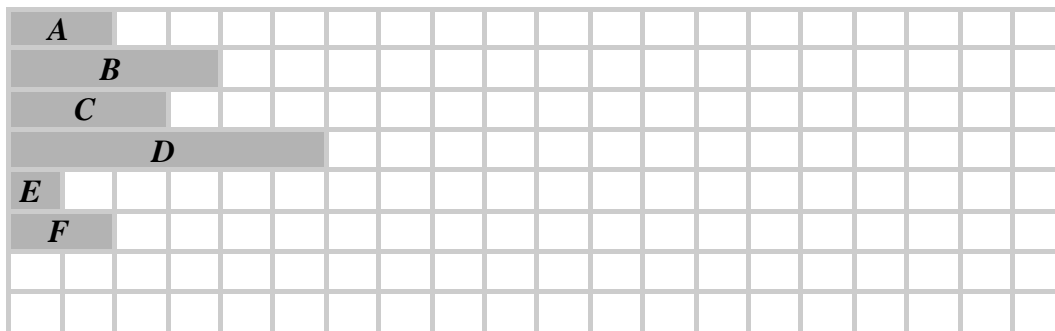
Maximale Granulierung des Speichers und jeder zweite Buddy frei/belegt ergibt maximale Anzahl an Einträgen in der Freispeicherliste.

## 14. Prozessscheduling

6, 6, 2 Punkte

Sechs Stapelaufträge treffen fast zeitgleich in einem Computer mit einer einzigen CPU ein. Auftrag A mit 2 Minuten, B mit 4, C mit 3, D mit 6, E mit 1 und F mit 2 Minuten und werden in dieser Reihenfolge einem Round-Robin Scheduler übergeben. Die Zeitscheibe „trifft“ also zuerst Auftrag A, dann B und so weiter. Vernachlässigen Sie die Zeit für einen Prozesswechsel.

- a) Berechnen Sie die durchschnittliche Verweilzeit der Aufträge, wenn die Zeitscheiben gegenüber den Ausführungszeiten vernachlässigbar klein sind.
- b) Berechnen Sie die durchschnittliche Verweilzeit der Aufträge, wenn die Zeitscheiben eine Minute lang sind.



- a)
- E:  $1 \cdot 6 \rightarrow 6$   
 A:  $1 \cdot 5 \rightarrow 11$   
 F:  $1 \cdot 5 \rightarrow 11$   
 C:  $1 \cdot 3 \rightarrow 14$   
 B:  $1 \cdot 2 \rightarrow 16$   
 D:  $2 \cdot 1 \rightarrow 18$   
 Summe: **76**  
 mittlere Verweilzeit **12,66666**

- b)
- E:  $1 \cdot 5 \rightarrow 5$   
 A:  $2 \cdot 1 \rightarrow 7$   
 F:  $1 \cdot 4 \rightarrow 11$   
 C:  $2 \cdot 1 \rightarrow 13$   
 B:  $2 \cdot 1 \rightarrow 15$   
 D:  $1 \cdot 3 \rightarrow 18$   
 Summe: **69**  
 mittlere Verweilzeit: **11,5**

- c) Woher rührt der Unterschied in den Ergebnissen?

Bei b) hängt die Verweilzeit von der Anordnung der Anfragen (FIFO!) ab, bei a) ist die Anordnung der Anfragen egal, die FIFO-Liste wirkt sich nicht aus.

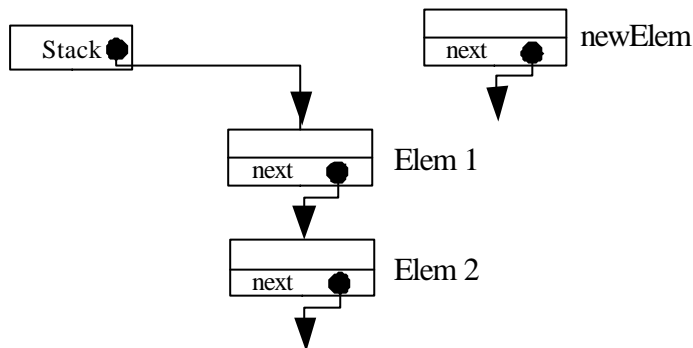
15. Race-Condition und Prozesssynchronisation

6, 4 Punkte

Gegeben sei folgende Implementierung eines Stacks in Pseudocode:

<b>Push(newElem):</b>	<b>Pop:</b>
(1) <code>newElem.next? := Stack?</code> ; // Next-Zeiger des neuen Elementes zeigt nun auf das oberste Element des Stacks	(3) <code>Stack? := Stack?.next;</code> // Der Stackzeiger zeigt nun auf das zweitoberste Element.
(2) <code>Stack? := newElem;</code> //neues Element wird auf den Stack gelegt	

Veranschaulichung der Datenstruktur:



a) Wann und warum treten hier Race-Conditions auf?

Wenn bei Push nach (1) eine Umschaltung auf ein Pop (desselben Stacks!) erfolgt, haben wir eine Race-Condition. Pop hat Push überholt.

Race-Conditions treten bei konkurrierenden Prozessen mit gemeinsamen Variablen auf.

b) Synchronisieren Sie die zwei Prozesse, damit keine Race-Conditions auftreten können mittels des (gegebenen) Semaphors  $s$  und den Operationen  $P(s)$  und  $V(s)$ .

$P(s);$	$P(s);$
Push();	Pop();
$V(s);$	$V(s)$



Matrikelnummer:

## 16. B-Bäume

5 Punkte

Fügen Sie in den gegebenen B-Baum der Ordnung  $m = 5$  die Zahl 4 ein.

Anmerkung: Fertigen Sie gegebenenfalls Zwischenzeichnungen an (optional). Kennzeichnen Sie in diesem Fall die Zeichnung, die das Ergebnis widerspiegelt.

