



# KLAUSUR

## zur Vorlesung Betriebssysteme SS 2002

Vorname

Name

Matrikelnummer

### I) Multiple Choice-Aufgaben (Mehrfachantworten sind möglich !)    10 Pkte

1.        Betriebsmittel sind
  - ausschließlich Hardware
  - ausschließlich Software
  - können sowohl aus Hardware oder Software bestehen.
  
2.        Eine virtuelle Maschine hat
  - nur eine Schnittstelle
  - zwei Schnittstellen
  - drei Schnittstellen
  
3.        Der Banker Algorithmus
  - testet, ob in einem System eine Verklemmung vorliegt.
  - stellt sicher, dass es zu keinen Verklemmungen kommen kann.
  
4.        Wenn das Request-Bit einer Speicherseite ‚0‘ ist, wurde sie bisher noch nie referenziert
  - JA                       NEIN
  
5.        Eine Speicherverwaltung, die Segmentierung unterstützt
  - hat genau einen linearen Adressraum
  - ermöglicht, dass der Adressraum größer ist als der physikalische Hauptspeicher
  - bietet keine automatische Relozierung des Codes
  - teilt den Speicher in Segmente fester Größe.
  - bietet explizite Zugriffskontrolle.

## I) Textaufgaben

### 6. Virtuelle Maschinen, Interfaces

5 Pkte

Skizzieren sie die Unterschiede zwischen einem Interface und einer virtuellen Maschine.

### 7. Prozesse, Threads

10 Pkte

- a) Wodurch unterscheiden sich *lightweight*-Threads von *heavyweight*-Threads? Nennen sie je ein Beispiel in welchem BS (bzw. API oder Technologie) die jeweilige Implementierung verwendet wird.
- b) Nennen sie vier Eigenschaften wodurch sich Prozesse und Threads von einander unterscheiden.

### 8. Scheduling

10 Pkte

Fünf Stapelaufträge treffen in einem Computer fast zur gleichen Zeit ein. Sie besitzen geschätzte Ausführungszeiten von 7, 16, 4, 9 und 2 Minuten und die Prioritäten 5, 2, 4, 1 und 3, wobei 5 die höchste Priorität ist. Geben sie für jeden der folgenden Scheduling-Algorithmen die durchschnittliche Verweilzeit an. Vernachlässigen sie dabei die Kosten für einen Prozesswechsel. Alle Aufträge benutzen als einziges Betriebsmittel die CPU.

- i) Round Robin  
ii) Priority Scheduling  
iii) First-Come-First-Serve (Reihenfolge: 8, 4, 10, 6, 2)  
iv) Shortest-Job-First

**Annahmen:** Nehmen Sie für i) an, dass das System präemptiven Mehrprogrammbetrieb verwendet und jeder Auftrag einen fairen Anteil an Prozessorzeit erhält. Des weiteren sei die Zeitscheibe sehr viel kürzer als die Zeiteinheiten, die für die geschätzten Ausführungszeiten angeführt sind, so dass die Aufträge quasi parallel ausgeführt werden.

Nehmen sie für ii) bis iv) an, dass die Aufträge nacheinander (non-präemptiv) ausgeführt werden.

### 9. Prozesssynchronisation

20 Pkte

- a) Geben sie ein Beispiel für eine *Race Condition*
- b) Die Prozesse  $P_1$  und  $P_2$  benötigen exklusiven Zugriff auf die Ressourcen  $R_1, R_2, R_3$ . Eine Ressource wird mit dem Befehl `open (Rx)` geöffnet und mit `close (Rx)` freigegeben. Wenn ein Prozess versucht eine Ressource zu öffnen, die bereits ein anderer Prozess benutzt, blockiert dieser mindestens so lange, bis die Ressource mittels `close ()` freigegeben wird.

Der Programmcode für die Prozesse  $P_1$  und  $P_2$  ist:

P<sub>1</sub>:

```
open (R1) ;  
if (b) {  
    open (R2) ;  
    close (R1) ;  
    open (R3) ;  
} else {  
    open (R3) ;  
    close (R1) ;  
    open (R2) ;  
}  
close (R2) ;  
close (R3) ;
```

P<sub>2</sub>:

```
open (R3) ;  
if (b) {  
    open (R2) ;  
    close (R3) ;  
    close (R2) ;  
    open (R1) ;  
} else {  
    close (R3) ;  
    open (R1) ;  
    open (R2) ;  
    close (R2) ;  
}  
close (R1) ;
```

Untersuchen Sie unter der Bedingung, dass die boolesche Variable *b* beliebig initialisiert wird und nicht geändert wird, ob es zu Verklemmungen kommen kann.

### 10. Speicherverwaltung

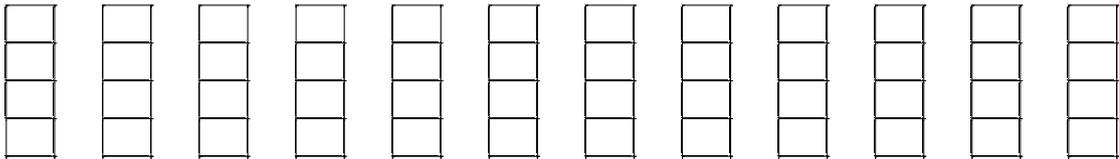
30 Pkte

a) Wir betrachten die folgende Sequenz von Seitenzugriffen bei einer Speichergröße von 4 Seiten: 1, 2, 3, 1, 5, 3, 4, 1, 2, 3, 5, 4 ausgehend von einem vollständig freien Speicher.

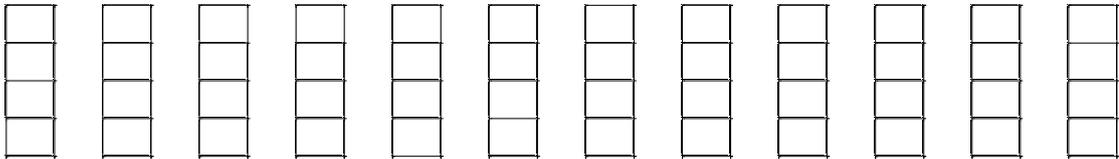
Skizzieren Sie die Auswirkungen bekannter Ersetzungsstrategien

(15 Pkte)

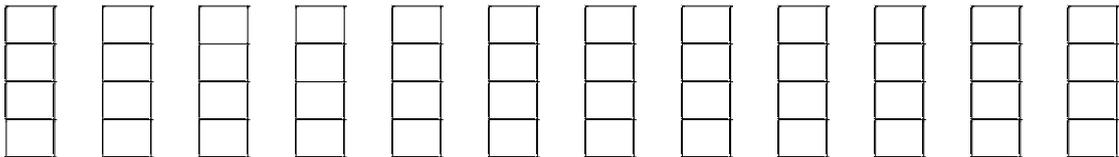
FIFO



LRU



Optimal



b) Ein Computer benutzt das Buddy-System zur Speicherverwaltung. Zu Beginn verfügt er über einen freien Speicherblock von 1 Megabyte, beginnend bei der Adresse 0. Geben Sie jeweils ein Abbild der Speicherbelegung nach jedem Speicherzugriff, wenn die folgenden Speicheranforderungen (*request*) und Freigaben (*release*) in der angeführten Reihenfolge bedient werden:

A: request 64K, B: request 230K, C: request 100K, D: request 129K, release A, release B, E: request 75K, release C, release E, release D  
 Beschriften Sie dabei die zusammenhängenden Speicherblöcke jeweils mit ihrer Größe.  
 (8 Pkte)

- c) Die Erweiterung des Hauptspeicher (eine größere Anzahl von Seitenrahmen) garantiert nicht für alle Strategien, dass auch die Anzahl der Seitenfehler sinkt. Wann kommt es zu diesem Effekt ? (7 Pkte)

**11. Virtueller Arbeitsspeicher 20 Pkte**

Gegeben ist die folgende Seitentabelle für einen Prozeß. Alle Zahlen sind dezimal und starten mit 0. Die Seitengröße beträgt 2024 Bytes.

Seite	Valid	R	M	Seitenrahmen
0	1	1	0	--
1	1	1	1	5
2	0	0	0	17
3	1	0	0	1
4	0	0	0	--
5	1	0	1	0

- a) Beschreiben Sie, wie aus den virtuellen Adressen die physikalischen Hauptspeicheradressen erzeugt werden. Nehmen Sie dazu (unnötigerweise) an, dass eine virtuelle Adresse 16 Bit lang ist.
- b) An welchen physikalischen Adressen, so sie eine haben, finden sich die folgenden virtuellen Adressen? Falls Seitenfehler auftreten geben Sie nur dies an.  
 i) 2052      ii) 6177      iii) 8999
- c) Ein Computer mit 32-bit Adressen benutzt eine zweistufige Adresskonversion. Virtuelle Adressen bestehen aus 9 Bit für die Top-Level-Seite, 11 Bit für die zweite Seitentabelle und dem Offset. Wie groß ist eine Seite und wie viele Seiten können im Adressraum verwaltet werden?

**12. Thrashing 8 Pkte**

In dem folgenden Diagramm ist die Bearbeitungsdauer gegen die Prozesszahl aufgetragen. Beschreiben Sie was an den Knickpunkten geschieht.

