

TechGI3: Systemprogrammierung

Klausur März 2008

ACHTUNG

Die vorliegende Beispielklausur wurde im März 2008 geschrieben. Sie ist für eine Bearbeitungszeit von 120 Minuten gedacht. In diesem Semester wurde das Modul TechGI3 nicht als prüfungsäquivalente Studienleistung geprüft, daher gab es keine Zwischentests und die Klausur war länger als 75 Minuten.

Dieses Muster soll den möglichen Aufbau und die Art der Fragestellung der TechGI3-Tests beispielhaft verdeutlichen. Wir erheben deshalb hier keinen Anspruch auf vollständige Stoffüberdeckung. In den zukünftigen Tests können und werden also auch andere Stoffschwerpunkte geprüft werden. Weiterhin kann die Zusammensetzung ('Quickies', Textaufgaben, Handsimulationen, Programmieraufgaben) variieren.

Diese Beispielklausur soll Ihnen helfen, sich mit der Herangehensweise zum Lösen der Aufgaben vertraut zu machen. Es nutzt Ihnen nichts, wenn Sie die Aufgaben oder Lösungen dieser Beispielklausur auswendig lernen. Prüfungsrelevant ist der in der Veranstaltung (Vorlesung und Übungen) gelehrt Stoff.

Name:

Beispielklausur

Aufgabe 1: Quickies

(13 Punkte)

- (a) **(13 Punkte)** Welche der folgenden Aussagen sind wahr bzw. falsch? Eine Begründung ist nicht erforderlich. Falsche Antworten führen zu Punktabzügen.

Technische Gegebenheiten

Bei direktem Speicherzugriff (DMA) laufen alle Daten zwischen Geräten und dem Hauptspeicher durch den Prozessor.

wahr falsch

Nebenläufigkeit

Für die automatische Prozessumschaltung wird immer ein äußeres Ereignis (z.B. Interrupt) benötigt.

wahr falsch

Bei kritischen Abschnitten muss nur bei echten Mehrprozessorsystemen der gegenseitige Ausschluss garantiert werden.

wahr falsch

Wenn das gleiche System mit verschiedenen Schedulingalgorithmen simuliert wird, können sich verschiedene Bedienzeiten ergeben.

wahr falsch

Wenn das gleiche System mit verschiedenen Schedulingalgorithmen simuliert wird, können sich verschiedene Wartezeiten ergeben.

wahr falsch

Betriebsmittelverwaltung

Die Betriebsmittelverwaltung muss immer durch eine zentrale Instanz durchgeführt werden.

wahr falsch

| | |
|--------|--|
| Punkte | |
|--------|--|

Name:

Beispielklausur

Garbage Collection

Erreichbar sind genau die Objekte, auf die eine Referenz von einer Wurzel zeigt.

wahr falsch

Erreichbar sind genau die Objekte, auf die eine Referenz von einer Wurzel oder einem anderen erreichbaren Objekt zeigt.

wahr falsch

Wenn der Referenzzähler eines Objektes Null ist, dann ist es nicht erreichbar.

wahr falsch

Wenn der Referenzzähler eines Objektes größer als Null ist, dann ist es erreichbar.

wahr falsch

Speicherverwaltung

Die Größe aller Seiten in einem System ist immer gleich.

wahr falsch

Die Größe aller Segmente in einem System ist immer gleich.

wahr falsch

Kommunikation

Die Nachricht selbst wird immer vom Sender in den Kanal gelegt und vom Empfänger von dort kopiert.

wahr falsch

| | |
|--------|--|
| Punkte | |
|--------|--|

Name:

Beispielklausur

Aufgabe 2: Assembler

(8 Punkte)

Schreiben sie in MC68000-Assembler die Funktionen `lock` und `unlock` einer Sperre. Die Sperre soll mit aktivem Warten arbeiten.

(a) (6 Punkte) Schreiben sie die folgenden Funktionen in MC68000-Assembler:

```
void lock(short int *lock_var);  
void unlock(short int *lock_var);
```

Kommentieren sie jede Zeile des Programms. Programme ohne Kommentierung werden nicht bewertet.

| | |
|--------|--|
| Punkte | |
|--------|--|

Name:

Beispielklausur

Matr.Nr.:

(b) (2 Punkte) Wie sieht der Stack nach dem Aufruf der Funktion `lock` und dem Retten der Register aus? Kennzeichnen sie das Element auf das der Stackpointer zeigt.

| |
|--|
| |
| |
| |
| |
| |
| |
| |
| |
| |
| |
| |
| |

| | |
|--------|--|
| Punkte | |
|--------|--|

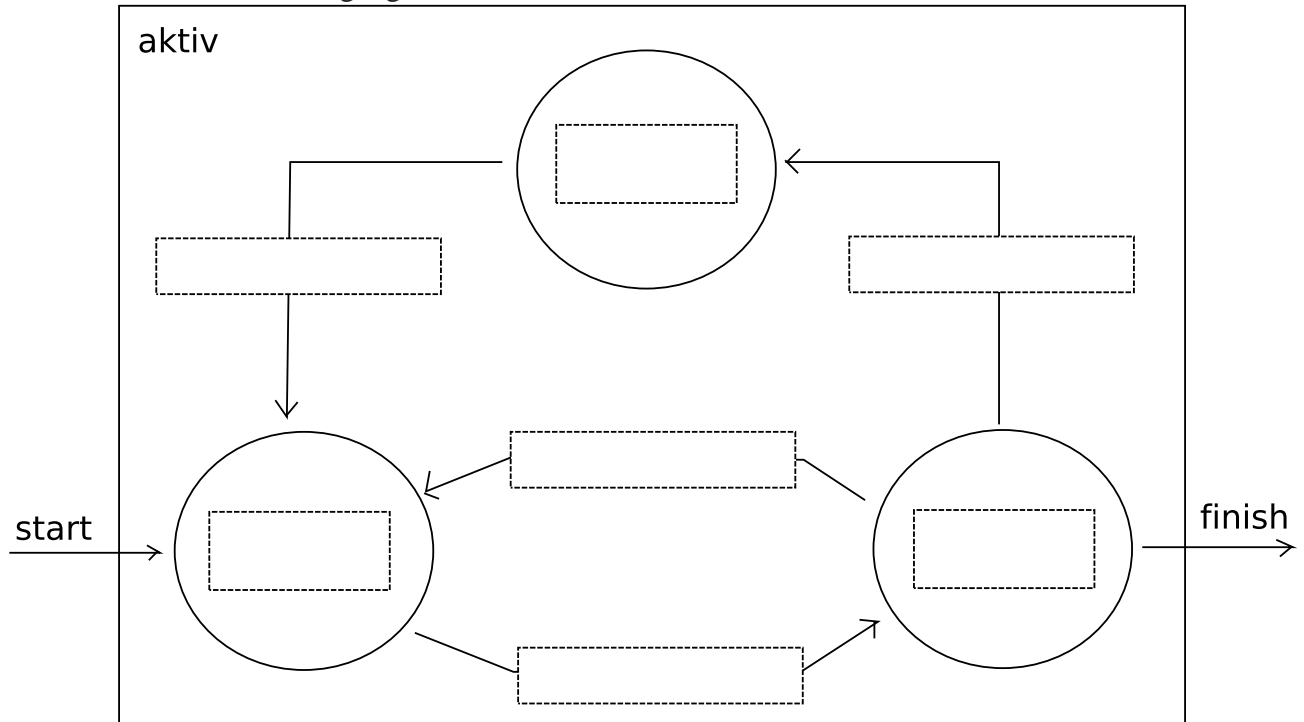
Name:

Beispielklausur

Aufgabe 3: Prozesse

(8 Punkte)

- (a) (4 Punkte) Benennen sie im folgenden Diagramm die drei wichtigsten Zustände eines Prozesses, sowie die Übergänge zwischen ihnen.

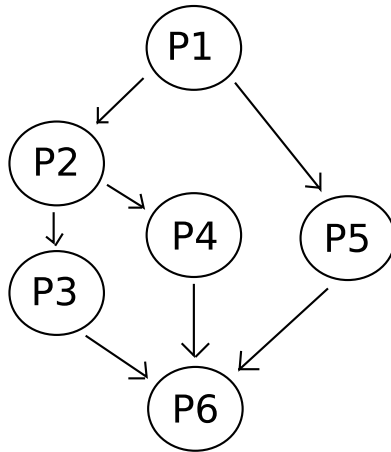


| | |
|--------|--|
| Punkte | |
|--------|--|

Name:

Beispielklausur

- (b) (4 Punkte) Geben sie für den folgenden Prozessfolgegraphen ein Programm in Pseudocode an, dass so viel Nebenläufigkeit wie möglich enthält.



| | |
|--------|--|
| Punkte | |
|--------|--|

Name:

Beispielklausur

Matrikelnummer:

Aufgabe 4: Koordination

(11 Punkte)

(a) (4 Punkte)

Nennen sie vier Unterschiede zwischen Semaphoren und Sperren die mit aktivem Warten implementiert wurden.

| | |
|--------|--|
| Punkte | |
|--------|--|

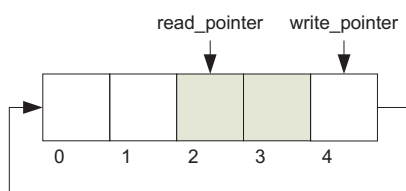
Name:

Beispielklausur

(b) (7 Punkte) Implementieren sie ein Programm in Pseudocode, bei dem verschiedene Prozesse Daten über einen Ringpuffer mit 5 Elementen austauschen.

Es gibt 10 Prozesse: 5 Sender und 5 Empfänger. Die Sender legen jeweils ein Element in dem Puffer ab. Wenn der Puffer voll ist warten sie bis wieder Platz im Puffer ist. Die Empfänger nehmen jeweils ein Element aus dem Puffer. Wenn der Puffer leer ist, warten sie bis wieder ein Element enthalten ist.

Ergänzen sie die beiden Fragmente der Sender- und Empfängerprozesse. Verwenden sie die in der Vorlesung vorgestellten Konstrukte `signal`, `wait`, `lock` und `unlock`



Globale Daten

```
Data buffer[5];  
int read_pointer;  
int write_pointer;  
// definieren sie hier eigene globale Variablen
```

| | |
|--------|--|
| Punkte | |
|--------|--|

Name:

Beispielklausur

Senderprozess

```
...  
// Synchronisation vor dem Ablegen  
  
// Nachricht in Puffer ablegen  
buffer[write_pointer] = message;  
write_pointer = (write_pointer + 1)%5;  
// Synchronisation nach dem Ablegen
```

Empfängerprozess

```
...  
// Synchronisation vor dem Abholen  
  
// Nachricht aus Puffer abholen  
read_pointer = (read_pointer + 1)%5;  
message = buffer[read_pointer];  
// Synchronisation nach dem Abholen
```

| | |
|--------|--|
| Punkte | |
|--------|--|

Name: _____

Beispielklausur

Aufgabe 5: Betriebsmittelverwaltung

(20 Punkte)

- (a) (4 Punkte) Nennen sie zwei Paare von zueinandergehörigen realen und logischen oder virtuellen Betriebsmitteln.

| reales Betriebsmittel | logisches bzw. virtuelles Betriebsmittel |
|-----------------------|------------------------------------------|
| | |

- (b) (2 Punkte) Bei der Kandidatenauswahl mit Best-Fit-Request kann auch eine dynamische Fenstergröße zum Einsatz kommen. Welches Problem wird dadurch gelöst und wie berechnet sich die Fenstergröße?

- (c) (4 Punkte) Nennen sie die drei notwendigen Bedingungen und die hinreichende Bedingung unter denen es zu einer Verklemmung kommt.

- (d) (2 Punkte) Nennen sie zwei Möglichkeiten zur Verklemmungsvorbeugung.

| | |
|--------|--|
| Punkte | |
|--------|--|

Name:

Beispielklausur

- (e) (8 Punkte) Gegeben ist folgende Betriebsmittelsituation mit 4 Prozessen und 2 Betriebsmitteln:

$$\text{Belegungen } B = \begin{pmatrix} 0 & 4 \\ 1 & 0 \\ 3 & 0 \\ 5 & 4 \end{pmatrix} \quad \text{Gesamtanforderungen } G = \begin{pmatrix} 4 & 6 \\ 2 & 4 \\ 3 & 1 \\ 10 & 6 \end{pmatrix}$$

$$\text{freie Betriebsmittel } f = (3 \quad 2)$$

Bestimmen sie mit Hilfe des Banker-Algorithmus, ob die Situation sicher ist. Geben sie alle Zwischenschritte nachvollziehbar an.

| | |
|--------|--|
| Punkte | |
|--------|--|

Name:

Beispielklausur

Aufgabe 6: Speicherverwaltung

(25 Punkte)

(a) **(4 Punkte)** Definieren Sie Verschnitt. Was ist der Unterschied zwischen internem und externem Verschnitt?

(b) **(5 Punkte)** Erläutern sie den Ablauf bei der zweistufigen Adressumsetzung mit Segmenten und Seiten. Wie wird zu einer virtuellen Adresse die physikalische Adresse bestimmt?

Wenn sie eine Skizze verwenden, denken sie an eine nachvollziehbare Beschriftung.

| | |
|--------|--|
| Punkte | |
|--------|--|

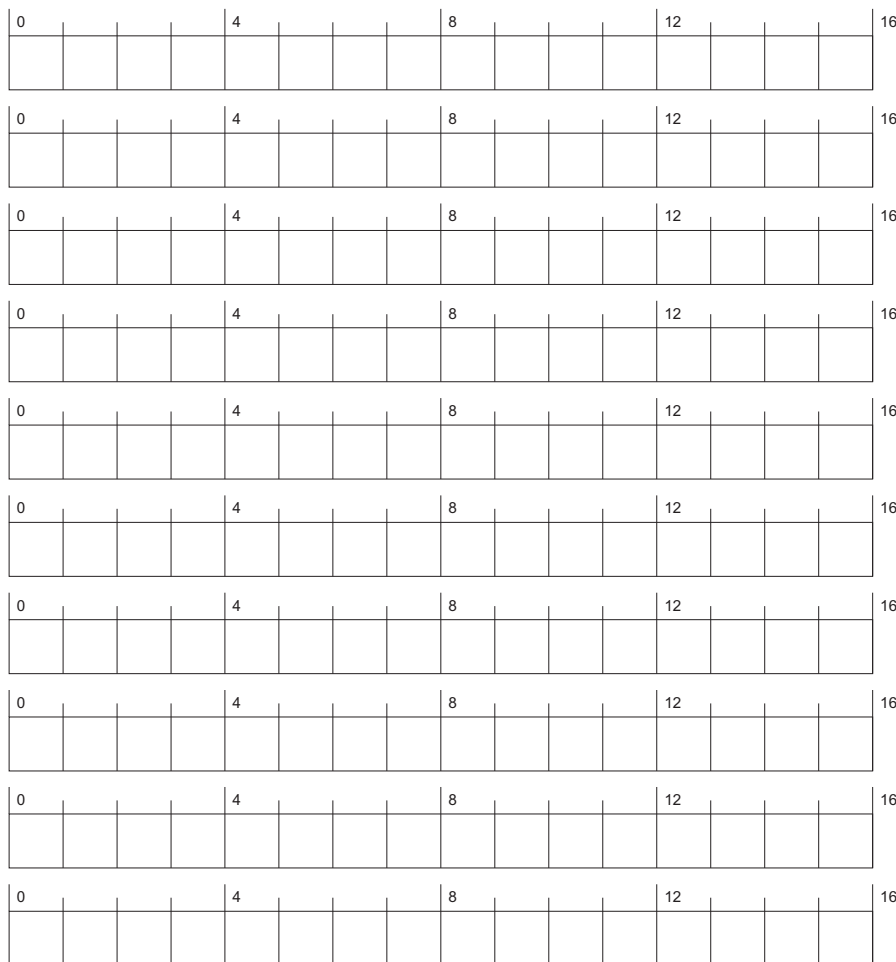
Name: _____

Beispielklausur

(d) (4 Punkte) Die folgenden Anforderungen und Freigaben treffen nacheinander ein und sollen jetzt in einem bisher leeren Speicherstück der Größe 16MB nach dem Buddy-Verfahren erfüllt werden.

```
s1=malloc(3MB)
s2=malloc(4MB)
s3=malloc(2MB)
s4=malloc(4MB)
free(s2)
s5=malloc(2MB)
```

Kennzeichnen sie den belegten Speicher nach jeder Zeile in dem folgenden Diagramm. Geben sie zu jeder Belegung an, zu welcher Anforderung (s1, s2, ...) sie gehört.



| | |
|--------|--|
| Punkte | |
|--------|--|

Name: _____

Beispielklausur

- (e) (4 Punkte) Der Hauptspeicher besteht aus 3 Kacheln. Geben sie für jeden Zeitschritt an, welche Seiten auf welchen Kacheln liegen, wenn **Second Chance** als Seitentauschstrategie verwendet wird.

Kennzeichnen sie Seitenfehler durch Unterstreichen der neuen Seite. Geben sie zusätzliche Datenstrukturen oder Informationen, die von der Seitentauschstrategie benötigt werden, an.

| Zugegriffene Seite | 1 | 2 | 3 | 4 | 2 | 1 | 3 |
|--------------------|---|---|---|---|---|---|---|
| Kachel 1 | | | | | | | |
| Kachel 2 | | | | | | | |
| Kachel 3 | | | | | | | |
| Datenstrukturen | | | | | | | |

- (f) (4 Punkte) Der Hauptspeicher besteht aus 3 Kacheln. Geben sie für jeden Zeitschritt an, welche Seiten auf welchen Kacheln liegen, wenn **LRU** als Seitentauschstrategie verwendet wird.

Kennzeichnen sie Seitenfehler durch Unterstreichen der neuen Seite. Geben sie zusätzliche Datenstrukturen oder Informationen, die von der Seitentauschstrategie benötigt werden, an.

| Zugegriffene Seite | 1 | 2 | 3 | 4 | 2 | 1 | 3 |
|--------------------|---|---|---|---|---|---|---|
| Kachel 1 | | | | | | | |
| Kachel 2 | | | | | | | |
| Kachel 3 | | | | | | | |
| Datenstrukturen | | | | | | | |

| | |
|--------|--|
| Punkte | |
|--------|--|

Name: _____

Beispielklausur

Aufgabe 7: Scheduling

(9 Punkte)

(a) (5 Punkte)

Ordnen sie die folgenden Umschaltstrategien in die Tabelle ein: first come first served (FCFS), shortest job next (SJN), Round Robin (RR), shortest remaining time next (SRTN), last come first served - preemptive resume (LCFS-PR).

| | Bedienzeit aller Prozesse bekannt | Bedienzeit nicht bekannt |
|------------------|-----------------------------------|--------------------------|
| mit Verdrängung | | |
| ohne Verdrängung | | |

(b) (4 Punkte) Die folgenden Prozesse werden in einem Einprozessorsystem gestartet:

| Prozess | A | B | C |
|-------------|---|---|---|
| Ankuftszeit | 0 | 1 | 3 |
| Laufzeit | 8 | 2 | 3 |

Simulieren sie die Prozessorzuteilung mit Hilfe der Round-Robin-Strategie mit $\tau = 2$. Geben sie ggf. benötigte Datenstrukturen und deren Inhalt in jedem Zeitschritt an.

| Zeit | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
|-------------------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|
| Prozessorbelegung | | | | | | | | | | | | | | | | | | |
| Datenstrukturen | | | | | | | | | | | | | | | | | | |

| | |
|--------|--|
| Punkte | |
|--------|--|

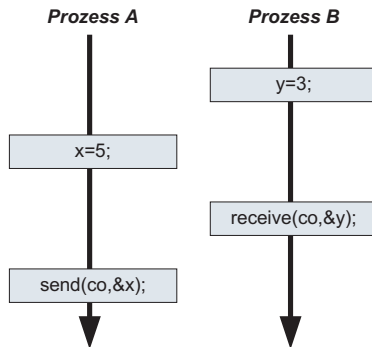
Name: _____

Beispielklausur

Aufgabe 8: Kommunikation

(6 Punkte)

(a) (2 Punkte) Gegeben ist die folgende Verzahnung von zwei Prozessen. Der Kanal c_0 ist zu Beginn leer.



Welchen Wert hat die Variable y am Ende, wenn asynchrones Senden und versuchendes Empfangen verwendet wird?

y=

Welchen Wert hat die Variable y am Ende, wenn synchrones Senden und synchrones Empfangen verwendet wird?

y=

| | |
|--------|--|
| Punkte | |
|--------|--|

Name:

Beispielklausur

(b) (4 Punkte) Gegeben ist folgendes Dienstprogramm:

```
void server() {  
    while (true) {  
        receive(input, jobdata);  
        temp = step1(jobdata);  
        temp2 = step2(temp);  
        temp3 = step3(temp2);  
        send(output, temp3);  
    }  
}
```

Die Funktionen step1 und step3 benötigen jeweils 2 Zeiteinheiten, die Funktion step2 benötigt 6 Zeiteinheiten.

Wie kann das Programm auf 5 Rechner verteilt werden um den Durchsatz zu maximieren? Nennen sie die verwendeten Konzepte und skizzieren sie die Verteilung als Pseudocode oder graphisch.

| | |
|--------|--|
| Punkte | |
|--------|--|