

Test zur Selbstkontrolle

Hinweise:

- Dieser Test soll Ihrer freiwilligen Selbstkontrolle dienen.
- Bitte bearbeiten Sie ihn ohne Hilfsmittel.
- Die Bearbeitungszeit sollte nicht mehr als 30 Minuten betragen.
- Sie können den Test selbst auswerten. Hinweise dazu am Ende!
- Der Test umfaßt nur den Stoff bis einschließlich Abschnitt 6, Speicherverwaltung!
- Der Test ist von der Frageart nicht typisch für die Klausur! Siehe dazu die Beispielklausur im WWW!

Aufgabe 1: Prozesse und Threads

- a) (8 Punkte) Kreuzen Sie an, welche Ereignisse die angegebenen Übergänge zwischen Prozeß-/Threadzuständen (des aktuellen oder auch anderer Prozesse/Threads) auslösen können!

| | Interrupt | Ausnahme | Seitenfehler | Systemaufruf |
|----------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| rechnerd → bereit | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| rechnerd → blockiert | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| blockiert → bereit | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> |

- b) (5 Punkte) Welche der genannten Elemente gehören (in einem System, das mehrere Kernel-Threads pro Prozeß unterstützt) zum Prozeß und welche zu den einzelnen Threads?

| | Prozeß | Thread |
|---|-------------------------------------|-------------------------------------|
| Seitentabelle | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| Prozessorregister | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| Liste der offenen Dateien | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| Programmzähler | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| Scheduling-Zustand (bereit, blockiert, ...) | <input type="checkbox"/> | <input checked="" type="checkbox"/> |

Aufgabe 2: Synchronisation und Kommunikation

- a) (8 Punkte) Richtig oder falsch? Wechselseitiger Ausschuß ...

| | richtig | falsch |
|---|-------------------------------------|-------------------------------------|
| ... ist nur mit gemeinsamen Variablen und Schreib-/Leseoperationen nicht realisierbar | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| ... kann auf Einprozessorrechnern durch eine Unterbrechungssperre realisiert werden | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| ... ist bei mehreren Prozessen auf einem Prozessor immer gewährleistet | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| ... kann immer durch unteilbare <i>Read-Modify-Write</i> -Befehle realisiert werden | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| ... bedeutet, daß zu einem Zeitpunkt höchstens ein Prozeß rechnet | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| ... kann Verklemmungen zur Folge haben | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| ... ohne aktives Warten wird nur durch Semaphore ermöglicht | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| ... zwischen den Prozeduren eines Monitors ist automatisch sichergestellt | <input checked="" type="checkbox"/> | <input type="checkbox"/> |

b) (4 Punkte) Semaphore: Kreuzen Sie die richtige(n) Alternative(n) an! (Mehrere richtige möglich!)

Der Semaphorzähler ist ein Indikator für die Anzahl der

verfügbaren Ressourcen wartenden Prozesse

Der Semaphorzähler wird hochgezählt in der

P-Operation V-Operation

Ein Prozeß kann blockieren in der

P-Operation V-Operation

c) (8 Punkte) Betrachten Sie folgendes Programmfragment (B1 und B2 sind Bedingungen, die von den Berechnungen im kritischen Abschnitt abhängen):

```
Semaphor b = 0;
Semaphor mutex = 1;
```

Prozeß 1:

```
...
P(mutex);
// kritischer Abschnitt
if (B1) {
    V(b);
}
V(mutex);
...
```

Prozeß 2:

```
...
P(mutex);
// kritischer Abschnitt
if (B2) {
    P(b);
}
V(mutex);
...
```

Es ist nie mehr als ein Prozeß im kritischen Abschnitt
Die Lösung ist verklemmungsfrei

| | | |
|--|-------------------------------------|-------------------------------------|
| | richtig | falsch |
| | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| | <input type="checkbox"/> | <input checked="" type="checkbox"/> |

Falls sich die Lösung verklemmen kann:

| | | | |
|-----------------------|--|--|-----------------------------------|
| Prozeß 1 blockiert in | <input checked="" type="checkbox"/> P(mutex) | <input type="checkbox"/> V(b) | <input type="checkbox"/> V(mutex) |
| Prozeß 2 blockiert in | <input type="checkbox"/> P(mutex) | <input checked="" type="checkbox"/> P(b) | <input type="checkbox"/> V(mutex) |

d) (8 Punkte) Kommunikation: richtig oder falsch?

Bei einer asynchronen Meldung blockiert der Empfänger, bis eine Nachricht vorliegt
Bei einer asynchronen Meldung blockiert der Sender, bis die Nachricht empfangen wurde
Synchrone Aufträge können durch Signale realisiert werden
Synchrone Aufträge können durch RPC realisiert werden
Eine Mailbox hat nur einen möglichen Empfänger
Ein Port hat nur einen möglichen Empfänger
Mehrere Prozesse können an denselben Port senden
Die Ankunft eines Signals bei einem Prozeß führt immer zur Beendigung des Prozesses

| | | |
|--|-------------------------------------|-------------------------------------|
| | richtig | falsch |
| | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| | <input type="checkbox"/> | <input checked="" type="checkbox"/> |

Aufgabe 3: Verklemmungen

a) (7 Punkte) Verklemmungen (durch Ressourcenzuteilung): richtig oder falsch?

Verklemmungen kommen immer nur durch Programmierfehler zustande
Eine notwendige Bedingung für Verklemmungen ist, daß Ressourcen, die einem Prozeß zugeteilt wurden, ihm nicht mehr entzogen werden können
Wenn jeder Prozeß alle von ihm benötigten Ressourcen auf einmal anfordern muß, sind keine Verklemmungen möglich
Verklemmungen können auch dann auftreten, wenn es nur eine einzige exklusiv nutzbare Ressource gibt
Bei einer Verklemmung sind alle Prozesse eines Rechners blockiert
Ein Belegungszustand heißt sicher, genau dann, wenn alle weiteren Abläufe nicht in eine Verklemmung führen
Ein Belegungszustand heißt sicher, genau dann, wenn es einen weiteren Ablauf gibt, der nicht in eine Verklemmung führt

| | | |
|--|-------------------------------------|-------------------------------------|
| | richtig | falsch |
| | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| | <input checked="" type="checkbox"/> | <input type="checkbox"/> |

b) (4 Punkte) Betrachten Sie die folgenden drei Prozesse:

Semaphor $a = 1, b = 1, c = 1;$

Prozeß 1:

...
 $P(a);$
 $P(b);$
 // krit. Abschnitt
 $V(a);$
 $V(b);$
 ...

Prozeß 2:

...
 $P(a);$
 $P(c);$
 // krit. Abschnitt
 $V(c);$
 $V(a);$
 ...

Prozeß 3:

...
 $P(b);$
 $P(c);$
 // krit. Abschnitt
 $V(b);$
 $V(c);$
 ...

Können sich die Prozesse verklemmen? ja nein

c) (6 Punkte) Betrachten Sie den folgenden Belegungszustand:

Ressourcenvektor $E = (3 \ 4)$ Ressourcenrestvektor $A = (1 \ 2)$
 Belegungsmatrix $C = \begin{bmatrix} 1 & 0 \\ 1 & 2 \end{bmatrix}$ Anforderungsmatrix $R = \begin{bmatrix} 2 & 3 \\ 1 & 1 \end{bmatrix}$

Kreuzen Sie die richtigen Aussagen an:

- Der Zustand ist nicht sicher
 Der Zustand ist sicher: zuerst kann Prozeß 1 beendet werden, dann Prozeß 2
 Der Zustand ist sicher: zuerst kann Prozeß 2 beendet werden, dann Prozeß 1

Aufgabe 4: Scheduling

a) (6 Punkte) Richtig oder falsch?

| | richtig | falsch |
|--|-------------------------------------|-------------------------------------|
| Beim präemptiven Scheduling kann ein Prozeß so lange rechnen, bis der die Kontrolle über den Prozessor freiwillig abgibt | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| Round Robin (RR) ist nicht-präemptiv | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| RR basiert auf Zeitscheiben | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| RR basiert auf Prozeß-Prioritäten | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| RR unterbricht rechnende Prozesse nach Ablauf einer vorher festgelegten Zeit | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| Bei RR bekommt der Prozeß als nächstes die CPU, der ganz vorne in der „Bereit“-Warteschlange steht | <input checked="" type="checkbox"/> | <input type="checkbox"/> |

b) (3 Punkte) Kreuzen Sie jeweils die richtige Alternative an!

- Round Robin
 - ist ein faires Schedulingverfahren
 - benachteiligt CPU-lastige Prozesse
 - benachteiligt E/A-lastige Prozesse
- Beim Multilevel Feedback Scheduling
 - bekommt jeder Prozeß eine feste statische Priorität
 - ändern sich die Prioritäten dynamisch, je nach Verhalten des Prozesses
- Die mittlere Durchlaufzeit von Aufträgen wird minimiert durch

| | |
|--|--|
| <input type="checkbox"/> First Come First Served (FCFS) | <input type="checkbox"/> Round Robin |
| <input checked="" type="checkbox"/> Shortest Job First (SJF) | <input type="checkbox"/> Prioritätenbasiertes Scheduling |

Aufgabe 5: Speicherverwaltung

a) (5 Punkte) Richtig oder falsch?

| | richtig | falsch |
|---|-------------------------------------|-------------------------------------|
| Seiten und Kacheln müssen immer die gleiche Größe haben | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| Es muß immer genauviele Seiten wie Kacheln geben | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| Beim Swapping entsteht das Problem der Relokation | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| Swapping löst das Problem der Relokation | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| Paging löst das Problem der Relokation | <input checked="" type="checkbox"/> | <input type="checkbox"/> |

b) (3 Punkte) Kreuzen Sie jeweils die richtige Alternative an!

| | | |
|-----------------------------------|---|---|
| <input type="checkbox"/> Swapping | <input checked="" type="checkbox"/> Paging | erlaubt das Auslagern von Teilen des logischen Adreßraums |
| <input type="checkbox"/> Swapping | <input checked="" type="checkbox"/> Paging | stellt automatisch auch Speicherschutz sicher |
| <input type="checkbox"/> Seiten | <input checked="" type="checkbox"/> Kacheln | sind Einheiten des physischen Adreßraums |

c) (6 Punkte) Kreuzen Sie jeweils die richtigen Aussagen an! (Mehrere richtige möglich!)

- Die Seitentabelle ...
 - wird vom Betriebssystem aufgebaut
 - wird von der MMU (*Memory Management Unit*) des Prozessors aufgebaut
 - enthält einen Eintrag für jede Adresse des logischen Adreßraums
 - gibt für jede Seite an, welcher Prozeß auf sie zugreifen darf
 - gibt für jede Seite an, welche Kachel ihr zugeordnet ist
 - gibt es genau einmal pro Rechner
 - gibt es einmal für jeden Prozeß auf einem Rechner
- Seitenersetzungsalgorithmen bestimmen ...
 - auf welche Kachel eine Seite abgebildet wird
 - wann eine Seite in den Hauptspeicher geladen wird
 - wann eine Seite aus dem Hauptspeicher verdrängt wird
 - welche Seite aus dem Hauptspeicher verdrängt wird
- Thrashing tritt auf, wenn über einen längeren Zeitraum ...
 - das *Working Set* eines Prozesses größer ist als der dem Prozeß zugewiesene Platz im Hauptspeicher
 - das *Working Set* eines Prozesses größer ist als der Hauptspeicher
 - das *Resident Set* eines Prozesses größer ist als der Hauptspeicher

d) (6 Punkte) Geben Sie für die folgenden Aussagen an, für welche der angegebenen Seitenersetzungsstrategien sie zutrifft! „Optimal“ steht hier für die optimale Strategie nach Belady.

| | Optimal | FIFO | LRU |
|--|-------------------------------------|-------------------------------------|-------------------------------------|
| Es wird die Seite verdrängt, die ... | | | |
| ... sich am längsten im Hauptspeicher befindet | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| ... in Zukunft am längsten nicht mehr benötigt wird | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| ... in der Vergangenheit am längsten nicht benutzt wurde | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| Die Strategie kann in der Praxis ... | | | |
| ... sehr einfach in Software realisiert werden | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| ... nicht effizient in Software realisiert werden | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| ... gar nicht realisiert werden | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

Hinweise zur Auswertung:

- Sie finden eine Musterlösung unter <http://www.bs.informatik.uni-siegen.de/www/lehre/ws0506/bs1/testML.pdf>
- Bitte sehen Sie sich die Lösung erst **nach** der Bearbeitung an!
- Punkteschema: pro richtiges Kreuz gibt es einen Punkt (Ausnahmen: Aufgabe 2c: 2 Punkte; 3b: 4 Punkte; 3c: 6 Punkte); für jedes falsche Kreuze werden entsprechend Punkte abgezogen; es gibt für jede Teilaufgabe aber minimal 0 Punkte.
- Insgesamt gibt es damit 87 Punkte, sie sollten mindestens 50% erreichen.