

AI-3 WS 07	Klausur Betriebssysteme (BS) 05.02.2008	Hbn
Seite: 1 / 9	Name, Vorname:	Matr.-Nr.:

Punktzahl (max. 60)	0	10	15	20	25	30	33	36	39	42	45	48	51	54	57	59
Leistungspunkte (ab Punktzahl)	0 LP	1 LP	2 LP	3 LP	4 LP	5 LP	6 LP	7 LP	8 LP	9 LP	10 LP	11 LP	12 LP	13 LP	14 LP	15 LP

Aufgabe A1

8 Punkte

- Was ist ein „Interrupt“?
- Kann die Behandlung von Interrupts ausgeschaltet werden? Wenn ja, wie?
- Was ist ein „Interrupthandler“?
- Ergänzen Sie in der folgenden Tabelle, welcher Interrupt nach Eintritt des angegebenen Ereignisses behandelt wird und geben Sie den entsprechenden Zustand des Interrupt-Speicherregisters an! (Höherer Index = höhere Priorität, d.h. $I_3 > I_2 > I_1$)

Ereignis	Behandelter Interrupt	Interruptspeicherregister (I_1, I_2, I_3)
Benutzerprogramm läuft	-	0 0 0
I_2 tritt ein		
I_3 tritt ein		
I_1 tritt ein		
I_3 beendet		
I_1 tritt nochmal ein		
I_2 beendet		
I_1 beendet		
Benutzerprogramm läuft weiter	-	0 0 0

Aufgabe A2

6 Punkte

- Was bewirkt die UNIX-Funktion „fork()“?
- Welche Rückgabewerte gibt es in welchem Fall?

AI-3 WS 07	Klausur Betriebssysteme (BS) 05.02.2008	Hbn
Seite: 2 / 9	Name, Vorname:	Matr.-Nr.:

Aufgabe A3

10 Punkte

In dieser Aufgabe geht es um das Scheduling der CPU. Betrachtet wird ein Multilevel-Feedback-Queuing-Algorithmus, der 8 verschiedene Prioritätsklassen umfasst (nummeriert von 0-7).

Es gilt Folgendes:

- Eine höhere Prioritätsklasse hat auch eine höhere Dringlichkeit (7 hat also die höchste Dringlichkeit, d.h. die Prozesse in dieser Prioritätsklasse werden zuerst bearbeitet).
- Prozesse in niedrigen Prioritätsklassen werden nur bearbeitet, wenn es zur Zeit keinen Prozess in einer höheren Klasse gibt.
- Innerhalb einer Prioritätsklasse wird der Round-Robin-Algorithmus mit einer Zeitscheibe von 30 ms angewendet (Prozesswechselzeiten können vernachlässigt werden).
- Ein Prozess darf seine begonnene Zeitscheibe immer voll ausnutzen.

Gegeben sind 5 Prozesse mit Ankunftszeiten (in ms nach Zeitpunkt 0), Ausführungsdauern (in ms) und Prioritätsklassen gemäß der folgenden Tabelle:

Prozessname	Ankunftszeit	Ausführungsdauer	Prioritätsklasse
A	0	60	1
B	40	100	4
C	60	20	1
D	100	40	4
E	170	50	7

Geben Sie an, wann die CPU welchen Prozess bearbeitet und wann ein Prozesswechsel stattfindet!

Tipp: Erstellen Sie eine CPU-Belegungstabelle mit einem Eintrag pro Prozesswechsel bzw. Zeitscheibenablauf: Prozessname, Startzeitpunkt, Endzeitpunkt, Restlaufzeit

AI-3 WS 07	Klausur Betriebssysteme (BS) 05.02.2008	Hbn
Seite: 3 / 9	Name, Vorname:	Matr.-Nr.:

Aufgabe A4

12 Punkte

Wir betrachten zwei Prozesse A und B, die sich über Semaphore synchronisieren. Beide Prozesse greifen auf denselben Puffer zu, der genau einen Speicherplatz zur Verfügung stellt. Das zu programmierende Verfahren besteht darin, dass Prozess A und B abwechselnd ein Datenobjekt im Puffer speichern, das dann jeweils der andere Prozess entnimmt.

Beispiel: Prozess A beginnt und speichert ein Datenobjekt im leeren Puffer. Danach entnimmt Prozess B dieses Datenobjekt aus dem Puffer und speichert selbst ein neues Objekt. Danach entnimmt Prozess A dieses Datenobjekt aus dem Puffer und speichert wieder ein neues Objekt .. usw.

Definieren Sie geeignete Semaphore und geben Sie den Pseudocode für die Prozesse an!

Aufgabe A5

8 Punkte

Gegeben sei folgendes JAVA-Codefragment:

```
public class Buffer { }

public class Proc1 extends Thread
{
    private Buffer myBuffer;

    public Proc1(Buffer b) {
        myBuffer = b;
    }
    ...
    public void run() {
        ...
        handleBuffer();
    }
    public synchronized void handleBuffer() {
        ...
        // Zugriff auf myBuffer
        ...
    }
}
```

AI-3 WS 07	Klausur Betriebssysteme (BS) 05.02.2008	Hbn
Seite: 4 / 9	Name, Vorname:	Matr.-Nr.:

```

public class Proc2 extends Thread
{
    private Buffer myBuffer;

    public Proc2(Buffer b) {
        myBuffer = b;
    }
    ...
    public void run() {
        ...
        handleBuffer();
    }

    public synchronized void handleBuffer() {
        ...
        // Zugriff auf myBuffer
        ...
    }
}

public static void main(String args[]) {
    Buffer b = new Buffer();

    Proc1 p1 = new Proc1(b);
    Proc2 p2 = new Proc2(b);

    p1.start();
    p2.start();
}

```

Ist mit diesem Code sichergestellt, dass ein Zugriff auf den Buffer b nur von einem Thread gleichzeitig erfolgen kann (exklusiver Zugriff)?
Begründen Sie Ihre Antwort! Wenn nein, geben Sie eine Korrekturidee an!

AI-3 WS 07	Klausur Betriebssysteme (BS) 05.02.2008	Hbn
Seite: 5 / 9	Name, Vorname:	Matr.-Nr.:

Aufgabe A6

10 Punkte

Betrachten Sie einen seitenorientierten virtuellen Speicher mit einer Seitengröße von 1024 Bytes. Gegeben ist der folgende Ausschnitt der Seitentabelle eines Prozesses:

Nummer der virt. Seite (VPN)	0	1	2	3	4	5
Seitenrahmennummer	12	3	0	13	0	8
„Valid“-Flag	true	true	false	true	false	true
„Referenced“-Flag	true	true	false	false	false	false

Gegeben sind die folgenden virtuellen Adressen: 1200, 2000, 3000

- a) Geben Sie für diese Adressen die virtuellen Seitennummern sowie die zugehörigen realen Hauptspeicheradressen an (falls sich die Seite im Hauptspeicher befindet).

- b) Welche der angegebenen virtuellen Adressen löst bei einem Zugriff einen Seitenfehler („Page fault“) aus?

- c) Zur Ermittlung der nächsten zu ersetzenden Seite werde der Clock-Algorithmus verwendet, der eine eigene Liste aller momentan im Hauptspeicher befindlichen Seiten des Prozesses führt. Die Liste sei hier (1,0,5,3), wobei die Nummer der virtuellen Seite, auf die der Uhrzeiger aktuell zeigt, am Anfang steht (also hier 1).
 Betrachten Sie den Zugriff mit der virtuellen Adresse aus Teil b).
 Welche virtuelle Seite wird bei Auslösen des Seitenfehlers aus dem Hauptspeicher verdrängt?
 Wie sieht anschließend die Liste für den Clock-Algorithmus aus?

AI-3 WS 07	Klausur Betriebssysteme (BS) 05.02.2008	Hbn
Seite: 7 / 9	Name, Vorname:	Matr.-Nr.:

Lösungen:

A1)

a) Interrupt = Unterbrechung des Programmablaufs

b) Ja, durch Rücksetzen des PSW Interrupt-Statusbits

c) Ein kurzes Programm, das aufgerufen wird, wenn der entsprechende Interrupt eintritt

d)

Ereignis	Behandelter Interrupt	Interruptspeicherregister (I_1, I_2, I_3)
Benutzerprog. läuft	-	0 0 0
I_2 tritt ein	I_2	0 0 0
I_3 tritt ein	I_3	0 1 0
I_1 tritt ein	I_3	1 1 0
I_3 beendet	I_2	1 0 0
I_1 tritt nochmal ein	I_2	1 0 0
I_2 beendet	I_1	0 0 0
I_1 beendet	-	0 0 0
Benutzerprog. läuft weiter	-	0 0 0

A2)

a) „fork()“ erzeugt einen neuen (Kind-)Prozess als identische Kopie des ausführenden (Eltern-) Prozesses. Beide Prozesse machen mit der Anweisung nach dem fork - Aufruf weiter.

b) Rückgabewert = 0 für den neuen (Kind-)Prozess und Rückgabewert = Kind-PID für den Eltern-Prozess

A3)

Prozessname	Startzeitpunkt	Endzeitpunkt	Restlaufzeit
A	0	30	30
A	30	60	fertig
B	60	90	70
B	90	120	40
D	120	150	10
B	150	180	10
E	180	210	20
E	210	230	fertig
D	230	240	fertig
B	240	250	fertig
C	250	270	fertig

AI-3 WS 07	Klausur Betriebssysteme (BS) 05.02.2008	Hbn
Seite: 8 / 9	Name, Vorname:	Matr.-Nr.:

A4)

Lösung (Einfaches Erzeuger/Verbraucherproblem, bei dem A und B sowohl Erzeuger als auch Verbraucher darstellen):

Binäres Semaphor $S_A = 0$

Binäres Semaphor $S_B = 0$

Prozess A

Loop

Speichern im Puffer

$V(S_B)$

$P(S_A)$

Entnehmen aus dem Puffer

Loop end

Prozess B

Loop

$P(S_B)$

Entnehmen aus dem Puffer

Speichern im Puffer

$V(S_A)$

Loop end

A5)

Nein, es werden hier nämlich in beiden handleBuffer-Methoden verschiedene Monitore verwendet (Proc1 und Proc2), da jede Objektinstanz einen eigenen Monitor besitzt! Vor dem Zugriff auf myBuffer muss daher ein synchronized-Block mit dem Monitor des Buffers eingerichtet werden:

```
public void handleBuffer() {
    ...
    synchronized (myBuffer) {
        // Zugriff auf myBuffer
    }
    ...
}
```

Alternativ könnten zum Zugriff „synchronized“-Methoden in der Buffer-Klasse verwendet werden.

A6) a)

Berechnung der virtuellen Seitennummer VPN (Index für den Seitentabellenzugriff):

$VPN = \text{floor}(\text{virtuelle Adresse} / 1024)$

Berechnung des Offsets (Bytenummer innerhalb der virtuellen Seite):

$\text{Offset} = \text{virtuelle Adresse} \bmod 1024$

Berechnung der realen Hauptspeicheradresse:

$\text{Reale Adresse} = \text{Seitenrahmennummer} * 1024 + \text{Offset}$

Virt. Adresse	1200	2000	3000
VPN	1	1	2
Offset	176	976	
Reale Adresse	3248	4048	-

b) 3000

c) Seite 5 wird durch Seite 2 ersetzt, da Seite 5 in der Clock-Liste die erste Seite mit Referenced-Flag = false ist. Der Uhrzeiger zeigt anschließend auf die nächste zu untersuchende Seite, also: (3,1,0,2)

AI-3 WS 07	Klausur Betriebssysteme (BS) 05.02.2008	Hbn
Seite: 9 / 9	Name, Vorname:	Matr.-Nr.:

- A7) a) Dateiname, Startadresse des ersten Blocks und die Anzahl der kontinuierlich belegten Blöcke
b) 1. Die Dateigröße muss vorher bekannt sein
2. Ein genügend großer Speicherbereich („Lücke“) muss vorhanden sein