

## ***Aufgabensammlung Betriebssysteme***

Die folgenden Aufgaben sollen einen Eindruck für die bei der Klausur zu erwartenden Aufgaben geben. Die Anzahl der Aufgaben in der Klausur wird etwa drei betragen.

### **Aufgabe 1:**

Drei Prozesse treffen zu den in der Tabelle angegebenen Zeiten in der Warteschlange eines Schedulers ein. Von jedem Prozess ist die Rechenzeit bekannt. Zusätzlich hat jeder Prozess eine Priorität (0 stellt die höchste Priorität dar).

Prozess	Ankunftszeit [100 ms]	Rechenzeit [100 ms]	Priorität
P <sub>1</sub>	0	6	1
P <sub>2</sub>	1	5	1
P <sub>3</sub>	2	3	0

Die Prozesse benutzen nur die CPU und werden nie durch E/A oder andere Gründe blockiert. Der Scheduler entscheidet nur aufgrund der zum Scheduling-Zeitpunkt in der Warteschlange vorliegenden Prozesse.

Betrachten Sie die folgenden Scheduling-Strategien und zeichnen Sie für jede der Strategien den zeitlichen Ablauf der Prozessausführung als Balken-Diagramm (Gantt-Chart):

- Shortest Job First (SJF, nicht-präemptiv).
- Round Robin (RR) mit einer Zeitscheibenlänge von 3 [x 100 ms]. Die Zeit für den Prozesswechsel kann vernachlässigt werden, die Zeitscheiben sind immer vollständig zu nutzen.
- Betrachten Sie nun eine prioritätsbasiertes Scheduling nach dem RR-Verfahren bei einer Zeitscheibenlänge von 1 [x 100 ms]. Zeichnen Sie den zeitlichen Ablauf der Prozessausführung als Balken-Diagramm sofern für jede Prioritätsklasse eine Warteschlange verwaltet wird. (Die Priorität 0 ist die "höchste".)

## Aufgabe 2:

In einem System gibt es 3 gleiche Drucker, auf denen per `dump()` -Befehl Dateien ausgegeben werden können. Allerdings müssen `dump()` -Aufrufe pro Drucker unter gegenseitigem Ausschluss stattfinden. Um nebenläufige Zugriffe zu synchronisieren, wird ein Semaphor "printer" eingeführt, auf den per `down()` -Operation und `up()` -Operation zugegriffen werden kann.

```
class PrintService {
    semaphor printer(COUNTER);
    boolean is_free[COUNTER];
public:
    PrintService(void);
    void print(string fname);
};

PrintService::PrintService() {
    for(int i=0; i< COUNTER; i++){
        is_free[i] = TRUE;
    }
}

void PrintService::print(string fname) {
    int d = -1;
    printer.down();

    while(! is_free[++d])
        ;
    is_free[d]=FALSE;
    dump(d, fname);

    is_free[d]=TRUE;
    printer.up();
}
```

- Mit welchem Wert muss der Semaphor `printer` initialisiert werden?
- In der Lösung kann es zu Wettbewerbsbedingungen kommen. Was kann passieren?
- Ergänzen Sie die Lösung so, dass durch den Einsatz eines mutex-Objekts die Wettbewerbsbedingung vermieden werden kann.

**Aufgabe 3:**

In einem System laufen 4 Prozesse  $P_1 - P_4$ . Es existieren 4 Ressourcenklassen  $R_1 - R_4$  mit jeweils 4, 5, 3, 5 Instanzen.

Die zurzeit von den Prozessen belegten Ressourcen und die angeforderten Ressourcen können aus den folgenden Tabellen entnommen werden.

**Tabelle 1: Belegte Ressourcen**

	P1	P2	P3	P4
R1	0	2	0	0
R2	1	0	1	0
R3	1	2	0	0
R4	0	1	1	1

**Tabelle 2: Angeforderte Ressourcen**

	P1	P2	P3	P4
R1	2	1	0	1
R2	1	0	1	1
R3	1	0	1	0
R4	0	3	1	1

Liegt eine **Verklemmung** vor? Begründen Sie Ihre Entscheidung!

**Aufgabe 4:**

Ein Rechner soll einen 32 Bit Adressbus besitzen. Die Seitengröße des virtuellen Speichers betrage 16 KiByte. Sein physikalischer Speicher ist 512 MiByte groß. Ein Eintrag in der Seitentabelle ist 8 Byte groß.

Im System seien 4 Prozesse. Wie viel Prozent des physikalischen Speichers wird von den Seitentabellen der Prozesse belegt, wenn jeder Prozess seine vollständige Tabelle geladen hat?

**Aufgabe 5:**

Die Seitentabelle enthält folgende Einträge: Seitenrahmen, Ladezeit der Seite, Zeit des letzten Zugriffs auf die Seite sowie R-, M- und Present- Bits.

Die Seitentabelle sieht wie folgt aus:

Index (Seitennummer)	Seitenrahmen	Ladezeit	Zugriffs- zeit	R	M	P
7	0	80	100	1	0	0
6	2	90	125	0	0	0
5	3	126	280	1	0	1
4	1	100	105	0	0	0
3	1	230	270	0	0	1
2	2	140	265	0	1	1
1	0	110	285	1	1	1
0	1	113	130	1	0	0

- a) Welche Seiten sind zurzeit im Speicher und auf welche Seitenrahmen sind sie abgebildet?
- b) Welche Seite wird bei einem Seitenfehler entfernt
- beim FIFO-Algorithmus?
  - beim NRU-Algorithmus?
  - beim Second-Chance Algorithmus?
  - beim LRU-Algorithmus?

**Aufgabe 6:**

Für ein FAT 12 Dateisystem mit einer Clustergröße von 512 Byte ist folgender Auszug aus der Konfiguration gegeben:

**Wurzelverzeichnis:**

Name	Erweiterung	Flags	Erster Cluster	Größe/Byte
DAT1	DAT	HS	7	2500
DIR		Verzeichnis	3	
Weitere ...	Einträge ...			

**Cluster4:**

Name	Erweiterung	Flags	Erster Cluster	Größe/Byte
PIC	1		10	1024
Frei				
....				

**File Allocation Table:**

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
x	x	-1	4	-1	0	0	8	9	12	11	-1	17	0	15	16	17	-1	-1	0	-1

In das Verzeichnis DIR wird eine Datei FILE.TXT kopiert, die 1500 Byte groß ist.

Wie viel Platz belegt die Datei auf dem Datenträger? Wie sieht der Verzeichniseintrag für FILE.TXT aus und wie verändert sich die File Allocation Table?

**Aufgabe 7:**

Gegeben ist ein Unix Dateisystem mit einer Blockgröße von 1024 Byte. Blockadressen in Indirektionsblöcken haben eine Länge von 4 Byte.

Auf dieses Dateisystem soll eine Datei von 4208 KiByte gespeichert werden. Wie viel Platz belegt die Datei auf dem Datenträger?