

Name: _____

Matrikelnummer: _____

Studiengang/Semester: _____

Unterschrift: _____

Hinweis: Dieses Aufgabenblatt sollte genügend Platz für alle Lösungen beinhalten. Es steht ihnen aber frei, zusätzliches Hilfspapier zu verwenden. Vergessen Sie in diesem Fall nicht, dieses zu beschriften.

Es sind keine weiteren Hilfsmittel zugelassen.

Aufgabe	1	2	3	4	Σ
Mögl. Punkte	15	16	15	14	60
Erreichte Punkte					

**Bitte erst nach Freigabe
umdrehen !**

Viel Erfolg !

Aufgabe 1 (15 Punkte):

- a) Welche Ziele hat ein Scheduler? Geben Sie 6 der in der Vorlesung besprochenen Ziele an. (3 Punkte)
- b) Geben Sie ein Beispiel für einen Zielkonflikt und begründen sie ihn. (2 Punkte)
- c) Was versteht man beim Scheduling unter "Verdrängung" ? (1 Punkt)
- d) Wann findet Verdrängung statt? Beschreiben Sie 3 unterschiedliche Ereignisse, in denen ein Scheduler sich entscheidet, einen Prozess zu verdrängen. (3 Punkte)

- e) Benennen Sie 4 der in der Vorlesung vorgestellten Schedulingverfahren und beschreiben sie jedes der Verfahren mit jeweils 1-2 Sätzen. (6 Punkte)

Aufgabe 2 (16 Punkte):

Drei Prozesse P1, P2 und P3 verwenden 3 binäre Semaphore S1, S2 und S3. Alle binären Semaphore seien ordnungsgemäß initialisiert (d.h. mit dem Wert 1). Nachfolgend ist ein Ausschnitt des Rumpfes der 3 Prozesse beschrieben:

P1:

```
Wait (S1);  
Wait (S2);  
/*  
    Kritischer Abschnitt  
*/  
Signal (S1);  
Signal (S2);
```

P2:

```
Wait (S1);  
Wait (S3);  
/*  
    Kritischer Abschnitt  
*/  
Signal (S3);  
Signal (S1);
```

P3:

```
Wait (S2);  
Wait (S3);  
/*  
    Kritischer Abschnitt  
*/  
Signal (S2);  
Signal (S3);
```

- a) Schützt die dargestellte Verwendung von Semaphoren tatsächlich den kritischen Abschnitt, oder ist es möglich, dass sich 2 der Prozesse gleichzeitig im kritischen Abschnitt befinden? Begründen Sie ihre Antwort. (2 Punkte)
- b) Angenommen P2 hat gerade den kritischen Abschnitt, während P1 und P3 noch vor dem kritischen Abschnitt stehen, aber soweit möglich vorangeschritten sind. Zeichnen Sie mit dem in der Vorlesung verwendeten Instrumentarium (Kreise = Prozesse, Quadrate = Ressourcen) den aktuellen Wartezustand (Hinweis: Semaphore als Ressourcen betrachten). (4 Punkte)
- c) Ist dies eine Deadlock-Situation oder nicht? Begründen Sie ihre Aussage. (1 Punkt)
- d) Kann es bei beliebigem Ablauf der 3 Prozesse (d.h. auch in anderen Situationen als der unter b) beschriebenen) irgendwann zum Deadlock kommen? Beschreiben Sie diesen oder geben Sie eine klare Begründung, warum nicht. (3 Punkte)

- e) Angenommen in einem Betriebssystem sind Semaphore über die folgenden Code-Sequenzen realisiert (vereinfachter Pseudocode):

```
typedef int  bin_sem;    // Binäre Semaphore als Integer definiert
static bin_sem my_sems[3] = { 0, 0, 0}; // 3 Semaphore S1, S2, S3

void Wait (bin_sem sem) {
    while (my_sems[sem] == 1) { } ;
    my_sems[sem] = 1;
}

void Signal (bin_sem sem) {
    my_sems[sem] = 0;
}
```

Warum ist diese Implementierung nicht geeignet, um einen kritischen Abschnitt zu sichern? Geben Sie eine genaue Begründung. (4 Punkte)

- f) Welche Methoden haben Sie in der Vorlesung kennengelernt, um kritische Abschnitte zu sichern (mindestens 4)? (2 Punkte)

Aufgabe 3 (16 Punkte):

Gegeben sei ein Rechner mit 32-Bit-Adressierung und einer Rahmengröße von 64 KB. Hauptspeichergöße ist 256 MB.

- a) Wieviel Rahmen hat der physikalische Speicher? (1 Punkt)

- b) Wieviele Einträge hat die Seitentabelle (normales nicht-mehrstufiges Verfahren)? (1 Punkt)

- c) Wieviele physikalische Rahmen brauchen Sie minimal und maximal, um ein File der Größe 340 KB im Hauptspeicher abzulegen (mit Begründung/Rechnung). (4 Punkte)

- d) Das folgende sei ein Ausschnitt aus der Seitentabelle eines Prozesses:

Seitennummer	P-Bit	Rahmennummer
0x4708:	1	0x31A
0x4709:	0	-
0x470A:	1	0x1B3

Auf welcher physikalischen Adresse liegt die virtuelle Adresse 0x470A78BB ? (2 Punkte)

- e) Welche virtuelle Adresse des Prozesses liegt im Hauptspeicher unter der Adresse 0x31A9175 ? (2 Punkte)

- f) Angenommen ihr Programm greift auf eine Heap-Variable mit der Adresse 0x4709FFFF zu. Erklären Sie schrittweise, was passiert. (5 Punkte)

Aufgabe 4 (14 Punkte):

- a) Erklären Sie wie Thrashing zustandekommt. (2 Punkt)

- b) Fördert das "Lokalitätsprinzip" von Programmen das Thrashing oder hilft es Thrashing zu vermeiden? Begründen Sie ihre Antwort. (2 Punkte)

- c) Angenommen der Hauptspeicher hat nur 3 Rahmen, R1, R2 und R3, mit der unten beschriebenen Anfangsbelegung. Auf Seite 4 wurde zuletzt zugegriffen. Stellen Sie fest, wieviel Seitenfehler selbst ein optimaler Seitenersetzungsalgorithmus in jedem Fall generiert, wenn ihr Prozess auf die folgende Sequenz von Seiten zugreift:

8, 7, 8, 3, 2, 8, 3, 7, 3

Verwenden Sie die folgende Tabelle zur Analyse (4 Punkte):

	4	8	7	8	3	2	8	3	7	3
R1:	8									
R2:	4									
R3:	2									
Page Fault ?										

- d) Verwenden Sie jetzt das Second-Chance (Clock)-Verfahren. Der Zeiger kreist R1->R2->R3->R1. Der aktuelle Zustand der Use-Bits ist als kleine Ziffer (0/1) dargestellt. Der Zeiger steht aktuell auf Rahmen 3 (kleines X) . Wenden Sie den Algorithmus an und füllen Sie die Zwischenzustände ein. (Notation: 0/1 für Use-Bit, X für Zeigerposition). (6 Punkte)

	4	8	7	8	3	2	8	3	7	3
R1:	8 ₀									
R2:	4 ₀									
R3:	2 _{1X}									
Page Fault ?										