

Name, Vorname: _____

Matrikelnr.: _____

Aufgabe	1	2	3	4	5	6	7	8	Summe
maximale Punktzahl	9	11	14	14	10	14	16	12	100
erreicht									

Organisatorische Hinweise

- Dauer der Klausur: 90 Minuten
- Erlaubte Hilfsmittel:
 - Ein DIN A4-Blatt mit handschriftlichen Notizen (Vorder- und Rückseite beschrieben)
 - Taschenrechner (ohne Mobilfunk- / ohne Internetanschluss, kein Smartphone)

Verhalten während der Klausur

- Beschriften Sie sofort das Deckblatt mit Ihrem Namen und Ihrer Matrikel-Nr.!
- Verwenden Sie die vorgegebenen Klausuraufgabenzettel für Ihre Lösungen (ggf. auch die Rückseite).
- Benutzen Sie keinen Bleistift und keine rote Tinte. Entsprechende Aufzeichnungen werden nicht gewertet.
- Legen Sie Ihren Studierendenausweis gut sichtbar auf den Tisch.
- Sofern Sie ein (max. DinA4-)Hilfsblatt vorgefertigt haben legen Sie es zum Ausweis.
- Melden Sie sich bitte mit Handzeichen, wenn Sie zur Toilette müssen (zeitgleich darf nur ein Klausurteilnehmer den Raum verlassen)
- Vermeiden Sie möglichst alle Störungen!

Aufgabe 1

- a) Kann die Ausführung eines Maschinenbefehls durch eine CPU von einem Hardware-Interrupt unterbrochen und später fortgesetzt werden? Begründen Sie Ihre Antwort!
- b) Welche Informationen müssen für einen laufenden Prozess gesichert werden, damit er nach einem Interrupt fortgesetzt werden kann?
- c) Durch welche Hardware-Mechanismen kann ein Betriebssystem (der „Scheduler“) garantieren, dass jedem Prozess eine zugeteilte CPU irgendwann wieder entzogen wird, also kein Prozess eine CPU „monopolisieren“ kann?

Aufgabe 2

- a) Nennen Sie drei Bereiche, die zum Hauptspeicherlayout eines Prozesses gehören:

- b) Welche Zustände kann ein Prozess in einem Multiprogramming-System annehmen?
Skizzieren Sie ein Zustandsdiagramm und tragen sie insbesondere ein, in welchem Zustand ein Prozess startet und bei welchen Ereignissen ein Zustandswechsel stattfindet!
Das Terminieren des Prozesses muss nicht berücksichtigt werden.

Aufgabe 3

In dieser Aufgabe geht es um das Scheduling der CPU. Betrachtet wird ein Multilevel-Queuing-Algorithmus, der 3 verschiedene Prioritätsklassen umfasst (nummeriert von 1-3).

Es gilt Folgendes:

- Eine Zeitscheibe beträgt 20 ms.
- Ein Prozess darf seine begonnene Zeitscheibe immer voll ausnutzen.
- Eine höhere Prioritätsklasse hat auch eine höhere Dringlichkeit (3 hat also die höchste Dringlichkeit, d.h. die Prozesse in dieser Prioritätsklasse werden zuerst bearbeitet).
- Innerhalb einer Prioritätsklasse wird der Round-Robin-Algorithmus angewendet.
- Prozesswechselzeiten können vernachlässigt werden.

Gegeben sind 4 Prozesse mit Ankunftszeiten (in ms nach Zeitpunkt 0), Ausführungsdauern (in ms) und Prioritätsklassen gemäß der folgenden Tabelle:

Prozessname	Ankunftszeit	Ausführungsdauer	Prioritätsklasse
A	0	30	1
B	15	50	3
C	38	40	3
D	50	10	2

- a) Geben Sie an, wann die CPU welchen Prozess bearbeitet und wann ein Prozesswechsel stattfindet! Füllen Sie dazu die folgende Tabelle aus:

Startzeitpunkt	Endzeitpunkt	Prozessname	Restlaufzeit des Prozesses
0			
	130		

- b) Durch welchen Zusatzmechanismus könnte ein „Verhungern“ eines Prozesses, der eine niedrige Priorität besitzt, verhindert werden?

Aufgabe 4

Wir betrachten die (unvollständige) Java-Klasse **Semaphore**, die Semaphore-Objekte inkl. der Zugriffsmethoden P und V zur Verfügung stellt.

- a) Ergänzen Sie (*auf dem Blatt*) den fehlenden Code für die Methoden P und V, so dass sie die korrekte Funktionalität eines allgemeinen Semaphors implementieren!

Tipp: Verwenden Sie die Java-Monitor-Synchronisationsfunktionen!

```
public final class Semaphore {  
    private int count;  
  
    public Semaphore(int value) {  
        count = value;  
    }  
  
    public synchronized void P() throws InterruptedException {  
        /* TODO */  
  
    }  
  
    public synchronized void V() {  
        /* TODO */  
  
    }  
}
```

- b) Gibt es in diesem Code einen „kritischen Abschnitt“? Wenn ja, geben Sie an, welchen Code er umfasst, wodurch er definiert wird und welchen Zweck er erfüllt.

Aufgabe 5

Wir betrachten eine Sparkassenfiliale, in der für alle Kunden nur eine Warteschlange verwendet wird, auch wenn es mehrere Schalter gibt. Wenn ein Kunde die Sparkassenfiliale betritt und alle Schalter besetzt sind, dann reiht er sich in die Warteschlange ein. Wenn ein Schalter frei wird, verlässt ein Kunde die Warteschlange und lässt sich am Schalter bedienen. Nach der Bedienung verlässt der Kunde die Sparkassenfiliale.

Ergänzen Sie den Pseudocode zur Beschreibung des Kundenverhaltens in einer Sparkassenfiliale mit 4 Schaltern. *Verwenden Sie zur Synchronisation den Semaphore-Mechanismus!*

Globale Initialisierungen:

Prozess KUNDE:

<Kunde betritt die Sparkassenfiliale>

<Kunde wird an einem Schalter bedient>

<Kunde verlässt die Sparkassenfiliale>

Aufgabe 6

Betrachten Sie einen seitenorientierten virtuellen Speicher mit einer Seitengröße von **1000 Bytes** („Paging“-Verfahren). Gegeben ist der folgende Ausschnitt der Seitentabelle eines Prozesses:

Virtuelle Seitennummer	0	1	2	3	4	5
Seitenrahmennummer	5	8	3	24	0	7
„Valid“-Flag	false	true	true	true	true	true
„Referenced“-Flag	false	true	true	false	false	true

- a) Geben Sie für die im folgenden angegebenen virtuellen Adressen die virtuelle Seitennummer sowie die zugehörige reale Hauptspeicheradresse an (falls sich die Seite im Hauptspeicher befindet).

Virtuelle Adresse	3112	5864	768	1024
Virtuelle Seitennummer				
Realen Hauptspeicheradresse				

- 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 (2,1,5,3,4), wobei die Nummer der virtuellen Seite, auf die der Uhrzeiger aktuell zeigt, am Anfang steht (also hier 2).
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?

Aufgabe 7

Gegeben ist ein sehr vereinfachter Ausschnitt eines FAT-basierten Dateiverzeichnisses, bei dem nur die Dateinamen und jeweils rechts davon die Nummern der zugehörigen ersten Plattenblöcke angegeben sind:

A.TXT	8	B.EXE	1	C.DOC	13
-------	---	-------	---	-------	----

Außerdem ist ein Anfangsausschnitt der File Allocation Table (FAT) gegeben (-1 = „End of File“):

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

- a) Geben Sie für die drei Dateien des Verzeichnisses jeweils die zugehörigen Plattenblöcke an, und zwar in der Reihenfolge, in der die Daten der Datei abgespeichert sind.

A.TXT:

B.EXE:

C.DOC:

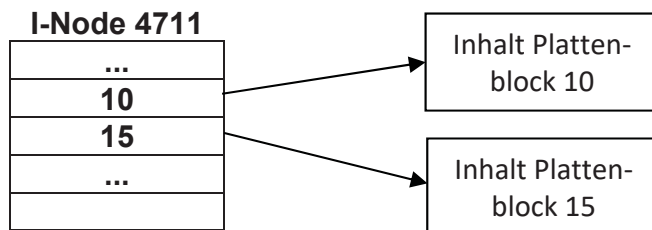
- b) Es finden in dieser Reihenfolge folgende Operationen statt:
1. Die Datei A.TXT wird um zwei Blöcke verkürzt, d.h. ihre beiden letzten Blöcke werden freigegeben.
 2. In die Datei B.EXE wird anschließend als 3. Block der Datei ein neuer Plattenblock eingefügt (umgangssprachliche Ausdrucksweise, Zählung startet daher bei 1).
 3. Anschließend wird die Datei C.DOC um einen Block verlängert.

Wie sieht die FAT nach jeder Operation aus? Geben Sie nur für die veränderten Zeilen FAT-Index und FAT-Eintrag an!

Von den freien Blöcken soll immer der mit der kleinsten Blocknummer belegt werden.

Aufgabe 8

Wir betrachten den folgenden Ausschnitt aus einem I-Node-basierten Dateisystem, welches die „Snapshot“-Technologie unterstützt (z.B. für Datensicherungen im laufenden Betrieb).



Geben Sie an, welche Veränderungen an diesem Dateisystem-Ausschnitt durch die folgenden Aktionen ausgelöst werden (Antwort auch durch Ergänzung der Skizze möglich):

1. „Einfrieren“ des Dateistands durch Erzeugen eines „Snapshots“ :
2. Ein Benutzer verändert Daten, die sich in Plattenblock 15 befinden, und will diese speichern:

Bewertungsschema

Punktzahl (max. 100)	0	10	20	30	40	50	55	60	65	70	75	80	84	87	92	96
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

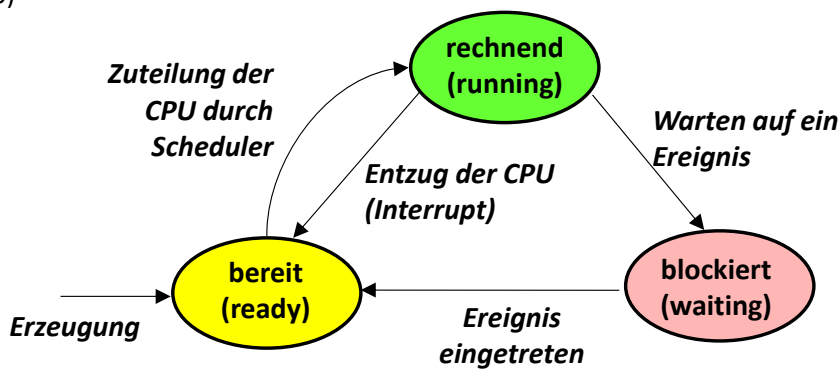
Lösungen

Aufgabe 1 (9 Punkte)

- Nein, weil immer nur nach der Ausführung eines Maschinenbefehls geprüft wird, ob ein Interrupt-Signal an der Signalleitung anliegt!
- Alle Registerinhalte
- Timer + Interrupt: Start eines Timers, der nach Ablauf einer „Zeitscheibe“ einen Hardware-Interrupt auslöst

Aufgabe 2 (11 Punkte)

- Stack, Daten, Programmcode
-



Aufgabe 3 (14 Punkte)

a)

Startzeitpunkt	Endzeitpunkt	Prozessname	Restlaufzeit des Prozesses
0	20	A	10
20	40	B	30
40	60	C	20
60	80	B	10
80	100	C	0
100	110	B	0
110	120	D	0
120	130	A	0

b) Dynamisches Neuberechnen der Prioritäten mit Erhöhung der Priorität, wenn ein Prozess bereits lange wartet!

Aufgabe 4 (14 Punkte)

a)

```
public final class Semaphore {
    private int count;

    public Semaphore(int value) {
        count = value;
    }

    public synchronized void P() throws InterruptedException {
        /* ToDo */
        while (count <= 0){
            wait();
        }
        count--;
    }

    public synchronized void V() {
        /* ToDo */
        count++;
        notify();
    }
}
```

b) Ja, P und V bilden gemeinsam einen kritischen Abschnitt mit Definition durch das Schlüsselwort „synchronized“. Zweck: Wechselseitiger Ausschluss von Threads (nur ein Thread darf sich gleichzeitig im kritischen Abschnitt aufhalten).

Aufgabe 5 (10 Punkte)

Globale Initialisierungen:

Allg. Semaphor S = 4

Prozess KUNDE:

<Kunde betritt die Sparkassenfiliale>

P(S)

<Kunde wird an einem Schalter bedient>

V(S)

<Kunde verlässt die Sparkassenfiliale>

Aufgabe 6 (14 Punkte)

a)

Virtuelle Adresse	3112	5864	768	1024
Virtuelle Seitennummer	3	5	0	1
Realen Hauptspeicheradresse	24112	7864	-	8024

b) 768

c) Seite 3 wird verdrängt, neue Liste: (4,2,1,5,0)

Aufgabe 7 (16 Punkte)

a)

A.TXT: 8,16,17,18,0

B.EXE: 1,7,3,4,19

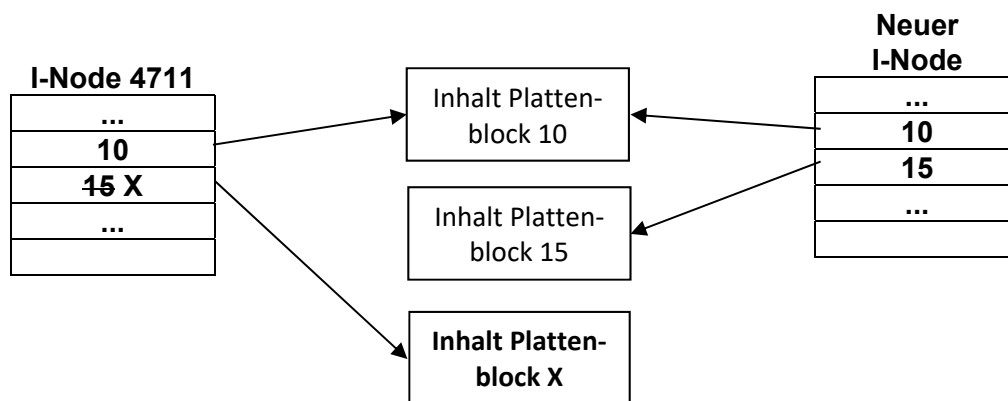
C.DOC: 13,12,2,5,20

b)

0: frei, 18: frei, 17: -1

7: 0, 0: 3

20:6, 6: -1

Aufgabe 8 (12 Punkte)

1. Erzeugen eines neuen I-Nodes für den „Snapshot“ durch Kopieren von I-Node 4711 inkl. aller Zeiger
2. Erzeugen eines neuen Plattenblocks X durch Kopieren von Plattenblock 15 und Ersetzen von Adresse 15 in I-Node 4711 durch die Adresse X des neuen Blocks. Die Daten werden nur im neuen Block verändert.