

<b>Prüfung</b> <b>„Einführung in die technische Informatik“</b> <b>28.06.06</b>	
Name	
Vorname	
Matrikelnummer	
Kennzahl	

**Bitte leserlich schreiben !**  
**Antworten, die wir nicht lesen können, werden**  
**nicht gewertet !**

Arbeitszeit: 90 Minuten

Bsp	Punkte	Bsp	Punkte
1	3	9	4
2	3	10	4
3	8	11	8
4	4	12	10
5	4	13	6
6	8	14	10
7	6	15	6
8	8	16	8
Summe			100

- 1.) Im Fall eines PAGEFAULTs sind welche heuristische Strategien anwendbar: Benennen Sie drei Strategien (Abkürzungen genügen) ( 3 Punkte ).
  
- 2.) Was versteht man unter *atomic actions* ? ( 3 Punkte )
  
- 3.) Entwerfen Sie eine Laufzeitleitung für digitale Signale mit einer Verzögerungszeit von 100 ns. Verwenden Sie dafür NAND-Gatter mit einer typischen Signallaufzeit (engl.: propagation delay) von 25 ns ( 8 Punkte ).
  
- 4.) Was muss das Betriebssystem machen, wenn zwischen zwei Threads, die zum gleichen Prozess gehören, umgeschaltet wird (d.h., der Scheduler führt einen Context Switch zwischen diesen Threads durch)? ( 4 Punkte )

- 5.) Was ist bei Prozessen der Unterschied zwischen den Zuständen "blocked" und "suspended"?  
( 4 Punkte )

- 6.) Das folgende Codefragment ist Teil einer Bankapplikation. Es zeigt eine Prozedur, die zur Überweisung eines Betrags (amount) auf ein Konto mit der Nummer accountID verwendet wird. Die Bankapplikation verwendet zwei Threads, deren threadIDs 0 und 1 lauten. Beide Threads können die Prozedur "einzahlen" aufrufen. Allerdings muss sichergestellt werden, dass sich immer nur ein Thread gleichzeitig in der Critical Section dieser Prozedur befindet. ( 8 Punkte )

```
// initialization
int turn = 0;
boolean lock[2];
lock[0] = lock[1] = false;

0: // threadID ist entweder 0 oder 1
1: public void einzahlen(int accountID, int amount, int threadID)
   {
2:     lock[threadID] = true;
3:
4:     if (threadID == 1)
5:     {
6:         turn = 1;
7:         while (lock[1-threadID]);
8:     }
9:     else /* threadID == 0 */
10:    {
11:        while (lock[1-threadID] && turn == 1);
12:    }
13:
14:    // Critical Section - Geld überweisen
15:
16:    turn = 0;
17:    lock[threadID] = false;
18: }
```

- (a) Stellt das gezeigte Codefragment sicher, dass sicher immer nur ein Thread (entweder der Thread mit threadID 0, oder der Thread mit threadID 1) in der Critical Section befinden kann? Wenn ja, begründen Sie ihre Antwort bitte kurz. Wenn nein, geben Sie eine konkrete Ausführung der beiden Threads an, wo sich beide in der Critical Section befinden.
- (b) Kann es sein, dass ein Deadlock auftritt? Wenn ja, begründen Sie Ihre Antwort bitte kurz. Wenn nein, geben Sie eine konkrete Ausführung der beiden Threads an, wo ein Deadlock auftritt.

Hinweise: Um eine konkrete Ausführung der Threads anzugeben, geben Sie bitte an, in welcher Reihenfolge die beiden Threads die Anweisungen (Zeilennummern) der Prozedur "einzahlen"

ausführen. Berücksichtigen Sie, dass zu einem bestimmten Zeitpunkt immer nur ein Thread eine Anweisung ausführen kann. Nach jeder Anweisung kann der Scheduler zwischen den Threads umschalten.

- 7.) Beschreiben Sie kurz zwei Probleme die auftreten, wenn sich mehrere Prozesse den physikalischen Speicher teilen müssen und keine virtuelle Speicherverwaltung verwendet wird. Erklären Sie dann kurz, wie virtueller Speicher diese Probleme löst. ( 6 Punkte )

- 8.) Nehmen Sie ein Betriebssystem an, welches Paging verwendet und dazu drei physikalische Page Frames im Hauptspeicher zur Verfügung hat. Die Größe jeder Page (und jedes Page Frames) beträgt 200 Bytes. Als Page Replacement Algorithmus wird LRU (least recently used) verwendet ( 8 Punkte ).

Das Betriebssystem startet nun ein Programm P, welches das folgende zweidimensionale Array A initialisiert.

```
int A[ ][ ] = new int[100][100];
```

Zur Initialisierung kann das Programm entweder Schleife (a) oder Schleife (b) verwenden.

```
(a) for (int j = 0; j < 100; ++j)
    for (int i = 0; i < 100; ++i)
        A[i][j] = 0;
```

oder

```
(b) for (int i = 0; i < 100; ++i)
    for (int j = 0; j < 100; ++j)
        A[i][j] = 0;
```

Der Code des Programms P benötigt eine der drei verfügbaren Page Frames und liegt in der ersten Page (zwischen der virtuellen Adresse 0 und 199). Das Array A beginnt am Anfang der zweiten Page (d.h., A[0][0] liegt auf Adresse 200 in Page 1).

Wieviele Page Faults verursacht die Ausführung von Schleife (a)?  
 Wieviele Page Faults verursacht die Ausführung von Schleife (b)?  
 Bitte begründen Sie kurz Ihr Ergebnis

( 8 Punkte ).

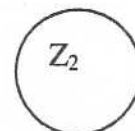
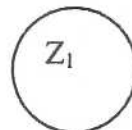
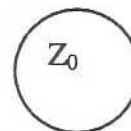
Hinweis: Überlegen Sie sich das Layout von Array A im virtuellen Speicher und die Reihenfolge der Zugriffe der Schleife auf die einzelnen Elemente.

9.) Aus welchen Datenbereichen besteht die Datenstruktur des „Semaphors“ ? ( 4 Punkte )

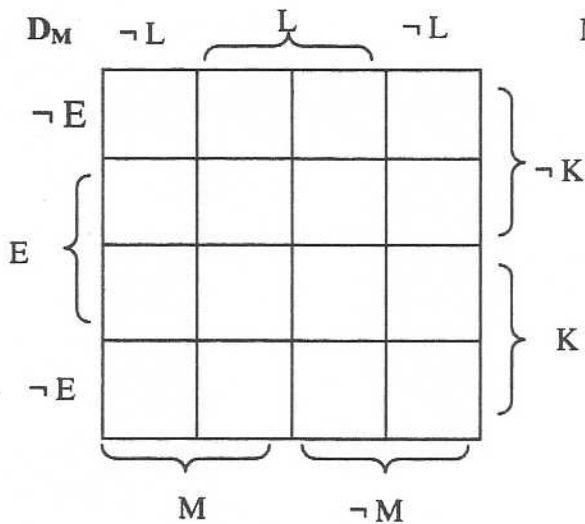
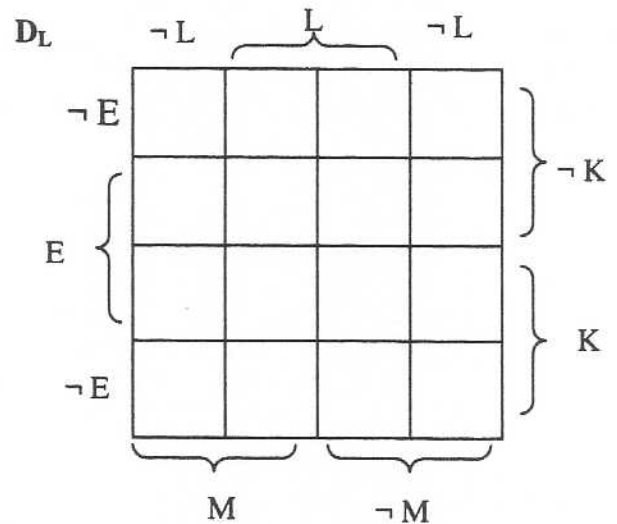
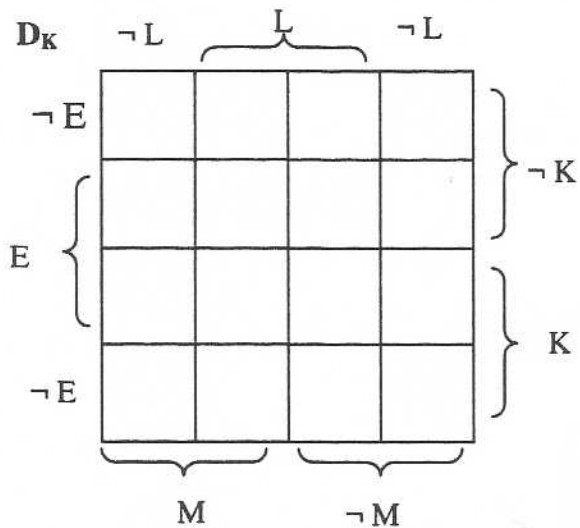
10.) Welche potentielle Gefahr entsteht in einem Rechnersystem, wenn ein direkter Zustandsübergang von BLOCKED nach RUNNING zulässig ist? ( 4 Punkte )

11.) Zeichnen Sie das Zustandsdiagramm, das der folgenden Tabelle entspricht! ( 8 Punkte )

A	0	1	0	1
B	0	0	1	1
Z <sub>0</sub>	Z <sub>0</sub>	Z <sub>1</sub>	Z <sub>2</sub>	Z <sub>1</sub>
Z <sub>1</sub>	Z <sub>2</sub>	Z <sub>2</sub>	Z <sub>2</sub>	Z <sub>2</sub>
Z <sub>2</sub>	Z <sub>0</sub>	Z <sub>1</sub>	Z <sub>1</sub>	Z <sub>2</sub>







Minimierte Terme für  $D_K$ ,  $D_L$  und  $D_M$  angeben:

$D_K =$

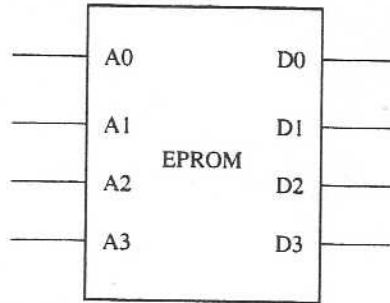
$D_L =$

$D_M =$

- 13.) Angenommen Sie wollen von zu Hause 5 Rechner an das Internet anschließen. Ihr Internet-Provider stellt Ihnen allerdings nur eine Class B Adresse zur Verfügung. Geben Sie eine Netzwerktopologie an, sodass jeder einzelne Rechner mit einem Server im Internet kommunizieren kann. Erklären Sie dazu, welche IP Adressen für die einzelnen Rechner vergeben werden und wie eine Umsetzung dieser Adressen auf die einzige Class B Adresse durchgeführt werden kann. (6 Punkte)

- 14.) Es soll ein Mikro-Programm für den Mikro16 Prozessor entworfen werden, das das Quadrat einer ganzen Zahl  $a$  ( $r = a^2$ ) berechnet. Achten Sie darauf, dass  $a$  auch negativ sein kann. Wenn Ihr Programm gestartet wird, ist  $a$  bereits im Register R3. Das Ergebnis  $r$  soll am Ende des Programms im Register R7 liegen.
- Berücksichtigen Sie beim Erstellen des Mikro-Programms, dass der Instruktionssatz des Mikro16 keine Funktion zur Berechnung des Quadrats einer Zahl und keine Multiplikation enthält. Das Register R0 ist standardmäßig mit der Konstante 0, das Register R1 mit der Konstante 1 belegt. Beschreiben Sie zusätzlich zum Mikro-Programm Ihren Algorithmus auch kurz in (Pseudo-) Code, z.B. in Java oder C und kommentieren Sie Ihr Programm ausführlich. (10 Punkte)

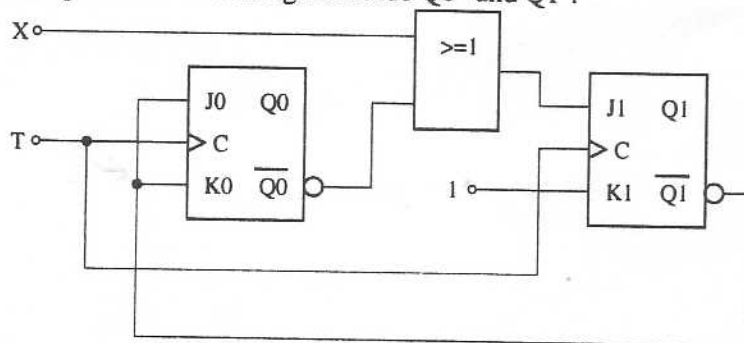
- 15.) Entwerfen Sie einen 4-Bit Synchronzähler (0-15). Zur Verfügung stehen 4 D-Latches und ein EPROM. Zeichnen Sie die zugehörige Schaltung und geben Sie die Programmierung des Speichers in binärer Darstellung an. (6 Punkte)



A3	A2	A1	A0	D3	D2	D1	D0

A3	A2	A1	A0	D3	D2	D1	D0

- 16.) Vervollständigen Sie für folgende Schaltung die Wahrheitstabelle. Die Wahrheitstabelle enthält neben dem Eingang X und den jeweiligen Eingängen der beiden Flip-Flops die aktuellen Zustände Q0 und Q1 und die Nachfolgezustände Q0<sup>+</sup> und Q1<sup>+</sup>. (8 Punkte)



X	J0	K0	J1	K1	Q0	Q1	Q0 <sup>+</sup>	Q1 <sup>+</sup>
0					0	0		
0					0	1		
0					1	0		
0					1	1		
1					0	0		
1					0	1		
1					1	0		
1					1	1		