

KNr. MNr. Zuname, Vorname

(Ges.)(100)

1.)(30)

2.)(25)

3.)(25)

4.)(20)

Zusatzblätter:

**Bitte verwenden Sie nur dokumentenechtes Schreibmaterial!**

## 1 Synchronisation (30)

Die Werkstätte der Firma 'Ho Ruck' hat zusätzlich zu ihren Arbeitsgeräten eine neue Drehbank bekommen. Jedes Gerät kann zu jeder beliebigen Zeit gebraucht werden. Unglücklicherweise ist der Stromverbrauch, wenn alle elektrischen Geräte eingeschaltet sind, jetzt grösser als die Zuleitung zur Werkstätte verkraftet. Als Provisorium installiert der Elektriker ein System, in dem Geräte nur eingeschaltet werden können, solange der Strom in der Leitung nicht zu gross wird.

**a) (4)**

Welche Art von Synchronisation wird dafür benötigt?

- ☐ keinerlei Synchronisation
 ☐ wechselseitiger Ausschluss  
☐ Bedingungssynchronization
 ☐ Mutual & Condition Synchronisation

Die Nennleistung der Zubringerleitung ist **16 kW**.

Stk	Maschinenart	Leistungsbedarf / Stk
3	Hebebühnen	3.5kW
1	Bestoßmaschine	4.5kW
1	neue Drehbank	6.0kW

Tabelle 1: Auflistung der Verbraucher

## b) (4)

Der Elektriker installiert folgende Routinen für ein Gerät mit einer Leistung von  $r$  kW.  
**Achtung:** Eine Leistungseinheit stellt **0.5 kW** dar.

```
vor_dem_Einschalten(Leistungsbedarf r){
    for(int i=0;i<(2*r);i++){
        P(A);
    }
}
```

```
nach_dem_Ausschalten(Leistungsbedarf r){
    for(int i=0;i<(2*r);i++){
        V(A);
    }
}
```

Das Semaphor A wurde mit der maximal verfügbaren Anzahl an Leistungseinheiten initialisiert.

```
initS(A,32);
```

Das schlaue Lehrmädchen meint dazu: „Mir scheint, das ist nicht der Weisheit letzter Schluss!“ Welche(s) Problem(e) kann/können bei dieser Lösung auftreten?

- |                                  |   |
|----------------------------------|---|
| <input type="radio"/> Deadlock   | <input type="radio"/> Busy Waiting              |
| <input type="radio"/> Starvation | <input type="radio"/> Überlastung der Zuleitung |

## c) (22)

Glücklicherweise stellt das verwendete System neben Semaphoren auch Routinen für Eventcounter zur Verfügung:

- `initEC(E,Zahl)` legt einen Eventcounter an und initialisiert ihn mit `Zahl`.
- `await(E,Zahl)` wartet solange bis der Eventcounter `E` grösser gleich dem Wert von `Zahl` ist.
- `read(E)` liest den Wert des Eventcounters aus.
- `advance(E,Zahl)` erhöht den Eventcounter um den Wert `Zahl`.

Gemeinsam erstellen die beiden nun ein Programm, welches einen Semaphor und zwei Eventcounter verwendet.

Der erste Eventcounter (`ECused`) entspricht der Anzahl der verbrauchten Ressourcen im System. Der zweite Eventcounter (`ECavail`) entspricht der Anzahl der zurückgegebenen Ressourcen plus der noch verfügbaren Ressourcen im System.

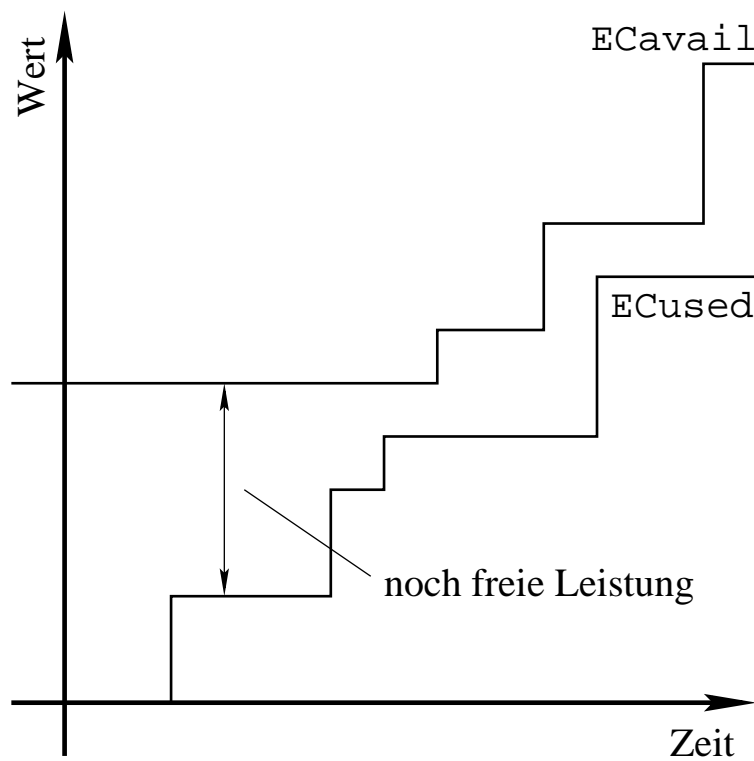


Abbildung 1: Bild eines Leistungsbedarfes

1. Ein Gerät wartet zunächst bis  $EC_{avail}$  mindestens den Wert Leistungsbedarf des Gerätes plus dem Wert von  $EC_{used}$  erreicht hat.
2. Danach testet das Gerät, ob inzwischen ein anderes Gerät möglicherweise weitere Ressourcen verbraucht hat. Dazu greift es auf beide EC zu und prüft, ob die Differenz mindestens dem Leistungsbedarf des Gerätes entspricht. Ist die Differenz kleiner als der Leistungsbedarf des Gerätes, so muss es den Vorgang bei Schritt 1 wiederholen. Ansonsten erhöht das Gerät  $EC_{used}$  um den Wert seines Leistungsbedarfes und schaltet sich ein.
3. Wenn das Gerät ausgeschaltet wird, erhöht es  $EC_{avail}$  um den Wert seines Leistungsbedarfes.

Beachten Sie, dass manche Programmteile möglicherweise exklusiv ausgeführt werden müssen. Weiters darf kein `goto` verwendet werden (umwandeln in eine Schleife). Wie kann eine solche Lösung aussehen?

Initialisierung:

```
vor_dem_Einschalten(Leistungsbedarf r){
```

```
}
```

```
nach_dem_Ausschalten(Leistungsbedarf r){
```

```
}
```

## 2 Speicherverwaltung (25)

### a) Lokalität von Prozessen – Working Sets (11)

Das Working Set Modell beschreibt das Speicherzugriffsverhalten von Prozessen. In diesem Modell ist das Working Set  $W(t, \Delta)$  definiert als die Menge der Speicherseiten, die in den letzten  $\Delta$  Zeiteinheiten vor dem Zeitpunkt  $t$  referenziert wurden.

Die folgende Tabelle beschreibt eine Folge von Speicherseitenzugriffen eines Prozesses. In Spalte 2 jeder Zeile steht die Nummer der Seite, auf die in der Zeitscheibe  $(t - 1, t]$  zugegriffen wird. Geben Sie in den leeren Feldern die Working Sets  $W(t, \Delta)$  für  $\Delta$  gleich 2, bzw. 5 an.

$t$	Seitennr.	$W(t, 2)$	$W(t, 5)$
1	29		
2	12		
3	13		
4	22		
5	29		
6	13		
7	12		
8	45		
9	29		
10	12		
11	12		

## b) Kombination aus Segmentierung und Paging (14)

Es werden folgende Begriffe (englische Notation) aus dem Buch zur Vorlesung verwendet:

Base	Basisadresse Seitentabelle des Segmentes
Length	Länge des Segmentes (Anzahl der Seiten des Segmentes)
Virt.Addr.	Virtuelle Adresse
Frame#	Seitenrahmennummer (im physischen Speicher)
Page#	Seitennummer (im virtuellen Speicher)
Seg#	Segmentnummer

Das in der Folge betrachtete Speicherverwaltungssystem verwendet zur Adressierung 20-bit Adressen. Für das Paging sind alle Seitenrahmen 256 Bytes (sedezimal (hexadezimal) 0x00100) groß. Das verwendete Adressformat ist folgendes:

Seg# (8 bit)	Page# (4 bit)	Offset (8 bit)
--------------	---------------	----------------

Hierbei wird assoziativer Zugriff (associative mapping) auf die Segmenttabelle und direkter Zugriff (direct mapping) auf die Seitentabelle verwendet.

Verwenden Sie für die Adressumsetzung folgende Segmenttabelle und Seitentabelle (alle Werte sind als Sedezimalzahlen (Hexadezimalzahlen) angegeben):

Segmenttabelle		
Seg#	Base	Length
0x03	0x0AFFE	0x3
0xAC	0x7D00F	0x1
0x3D	0x1CE00	0x7
0x00	0x00000	0xD

Seitentabelle	
Address	Frame#
0x00000	0x123
0x00001	0x124
0x00002	0x376
...	...
0x0AFFE	0xEEE
0x0AFFF	0xABC
0x0B000	0x000
0x0B001	0xDDD
...	...
0x1CDFF	0x666
0x1CE00	0x7DA
...	...
0x1CE05	0x5A7
0x1CE06	0x5AC
...	...
0x7D00E	0x471
0x7D00F	0xAAA
0x7D010	0x815
...	...

Ermitteln Sie unter Benützung obiger Tabellen die physikalischen Adressen zu folgenden virtuellen Adressen (ergibt sich bei der Umwandlung eine ungültige Adresse, so schreiben Sie bitte **ungültig** in das entsprechende Feld):

Virtuelle Adresse	Physikalische Adresse (zu ermitteln)		
0xAC2FF			
0x01111			
0x002FF			
0x3D6D7			
0xFF000			
0x0328A			
0x00D33			

### 3 Scheduling (25)

#### a) Rate-Monotonic Scheduling (12)

Task	Laufzeit (ms)	Periode (ms)
T1	2	20
T2	2	10
T3	3	13
T4	2	12
T5	3	23

Folgendes wird angenommen:

- Alle Tasks sind periodisch
- Deadline = Periode
- Scheduling Overhead wird vernachlässigt
- Alle Tasks sind unabhängig

Ist dieses Taskset nach der notwendigen Scheduling-Bedingung unter Verwendung von Rate-Monotonic Scheduling (RMS) schedulbar?

☐ Ja ☐ Nein

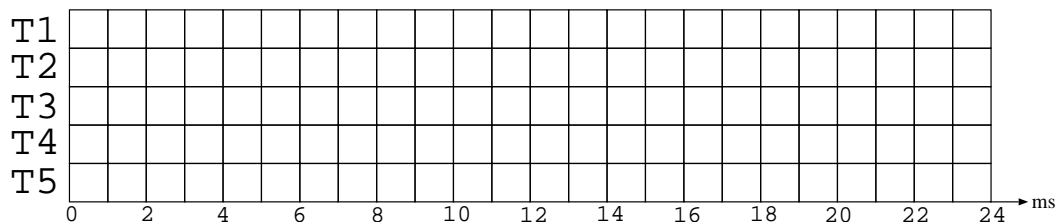
Bitte begründen Sie Ihre Antwort:

Ist dieses Taskset nach der hinreichenden Scheduling-Bedingung unter Verwendung von RMS schedulbar?

☐ Ja ☐ Nein

Bitte begründen Sie Ihre Antwort:

Das oben angegebene Taskset soll mit einem Rate-Monotonic Scheduling Algorithmus für den Worst Case<sup>1</sup> gescheduled werden. Bitte vervollständigen Sie den gesamten Zeitbereich der folgenden Skizze bzw. kennzeichnen Sie eine Verletzung einer Deadline.



---

<sup>1</sup>Im Worst Case beginnen alle Perioden bei 0.

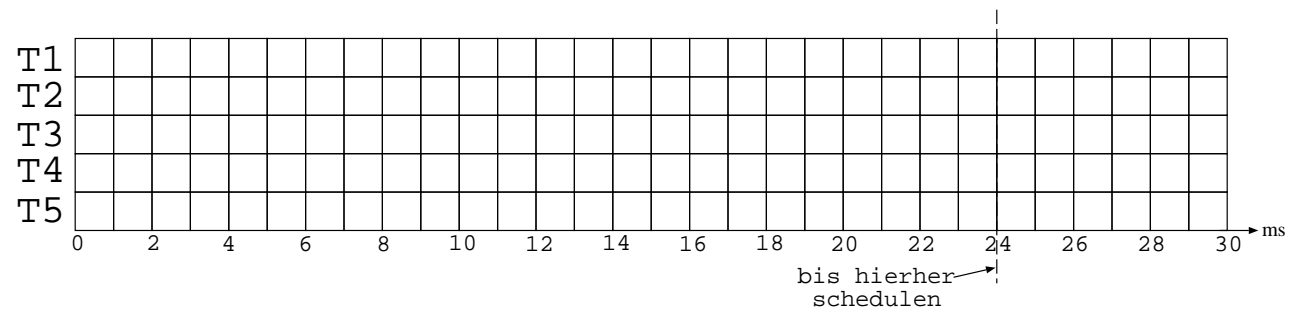


## b) Earliest Deadline First Scheduling (EDF) (8)

Schedulen Sie das gegebene Task Set nach dem Earliest Deadline First Verfahren *mit Pre-emption* bis zu  $t = 24$  ms. Als Arrival Time können Sie für alle Tasks den Zeitpunkt 0 annehmen, der Scheduling Overhead ist wiederum 0. Weiters sind die angegebenen Deadlines zu beachten. Sollte das Taskset nicht gescheduled werden können, so zeichnen Sie bitte die Verletzung der Deadline eindeutig ein.

Die Deadlines für die Tasks sind:

Task	Laufzeit (ms)	Periode (ms)	Deadline (ms)
T1	2	20	12
T2	2	10	10
T3	3	13	13
T4	2	12	7
T5	3	23	8



## c) Fragen zu Scheduling (5)

Wenn ein Test unter Verwendung von Rate-Monotonic Scheduling aufgrund der notwendigen Scheduling-Bedingung ergibt, dass ein Taskset schedulbar ist, aber ein Test aufgrund der hinreichenden Scheduling-Bedingung ergibt, dass ein Taskset *nicht* schedulbar ist, könnte es trotzdem sein, dass das Taskset schedulbar ist oder ist das unmöglich?

- ☐ Ja, ein solches Taskset könnte schedulbar sein.
- ☐ Nein, es ist unmöglich, dass ein solches Taskset schedulbar ist

Kreuzen Sie bei der folgenden Aussage bitte an, ob diese richtig oder falsch ist.

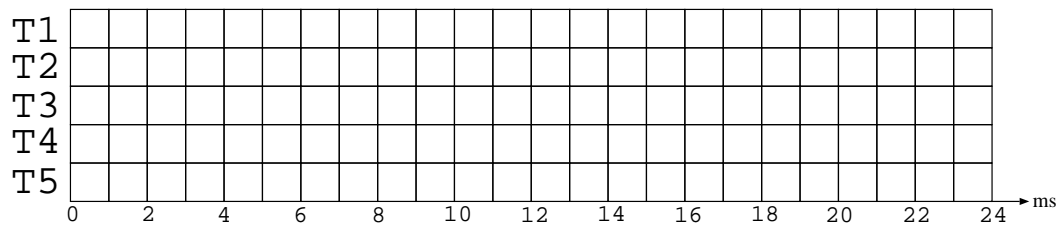
Bei Scheduling nach dem Round-Robin-Verfahren werden I/O-intensive Prozesse gegenüber rechenzeit-intensiven (mit wenig I/O) benachteiligt.

- ☐ richtig
- ☐ falsch

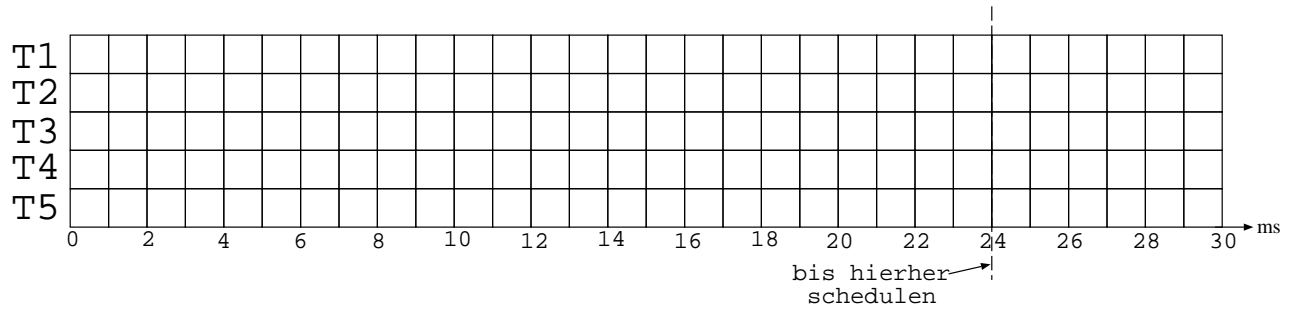
## Ersatzvorlagen

Bitte streichen Sie die anderen Vorlagen durch, wenn Sie diese Vorlagen verwenden.

Ersatz a)



Ersatz b)



## 4 Security (20)

a) (10)

Was versteht man unter einem *aktiven* Security Threat?

Welche *aktiven* Security Threats gibt es?

Was versteht man unter einem *passiven* Security Threat?

Welche *passiven* Security Threats gibt es?

**b) (4)**

Was versteht man unter *Audit Records*?

Wozu dienen *Audit Records*?

**c) (6)**

Wann nennt man einen Verschlüsselungsalgorithmus *symmetrisch*?

Wann nennt man einen Verschlüsselungsalgorithmus *asymmetrisch*?

Welchen Vorteil hat die symmetrische gegenüber der asymmetrischen Verschlüsselung?

Welchen Vorteil hat die asymmetrische gegenüber der symmetrischen Verschlüsselung?