

KNr.

MNr.

Zuname, Vorname

Studium:

☐ Informatik (Bakk.)☐ Elektrotechnik (Master)

Ges.)(100)

1.)(30)

2.)(30)

3.)(40)

Zusatzblätter:

Bitte verwenden Sie nur dokumentenechtes Schreibmaterial!

1 Synchronisation mit Semaphoren (30)

Ein Softwarepaket aus vier Prozessen soll eine Anlage simulieren, die Kartons mit Würfeln und Murmeln befüllt. Zwei gleichartige Prozesse simulieren den An- und Abtransport von Kartons zur bzw. von der Befüllanlage. Je ein Prozess kontrolliert das Einfüllen von Murmeln bzw. Würfeln in einen bereitstehenden Karton.

Ergänzen Sie die gegebenen Codestücke mit *Semaphoroperationen*, um die Prozesse entsprechend der unten gegebenen Bedingungen zu synchronisieren. Das Verwenden von globalen Variablen zur Synchronisation oder Kommunikation zwischen Prozessen ist nicht erlaubt.

- Die beiden Prozesse für den An- und Abtransport der Kartons sollen Roboter simulieren. Jeder Roboter hat einen eigenen Vorrat an leeren Kartons und einen eigenen Ablageplatz für volle Kartons und führt wiederholt folgende Aktionen durch:
 - Karton vom Vorrat holen: *karton_aufnehmen()*.
 - Karton zur Befüllanlage transportieren und öffnen: *karton_bereitstellen()*.
 - Karton verschließen und von der Befüllanlage nehmen: *karton_entfernen()*.
 - Karton zum Ablageplatz abtransportieren: *karton_abtransportieren()*.
- Der Prozess Murmelspender ruft in einer Endlosschleife die Funktion *Murmel_liefern()* auf. Jeder Aufruf dieser Funktion lässt eine Murmel in den Karton auf der Befüllanlage fallen.
- Der Prozess Würfelspender ruft in einer Endlosschleife die Funktion *Wuerfel_liefern()* auf. Jeder Aufruf dieser Funktion lässt einen Würfel in den Karton auf der Befüllanlage fallen.
- Die Prozesse sind so zu synchronisieren, dass die Parallelität ihrer Aktionen die folgenden Regeln erfüllt, darüberhinaus aber möglichst wenig eingeschränkt wird.
- Auf dem Befüllplatz der Anlage ist Platz für einen Karton. Synchronisieren Sie die Prozesse so, dass die Roboter die Befüllanlage abwechselnd mit Kartons versorgen.
- Jeder Karton ist mit N Murmeln und K Würfeln zu befüllen. Stellen Sie die korrekte Befüllung der Kartons durch die Synchronisation sicher.

(a) Initialisierungen (vor Start der Prozesse)



In der Folge sind Codestücke für die zu synchronisierenden Prozesse gegeben, die bereits alle notwendigen Funktionsaufrufe enthalten. Ergänzen Sie in diesen Codestücken die fehlenden Semaphoroperationen zur Synchronisation.

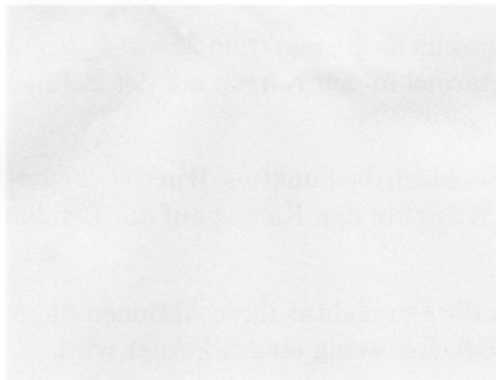
(b) Prozesse für die Befüllung

```
/** Murmelspender **/
```

```
do {
```



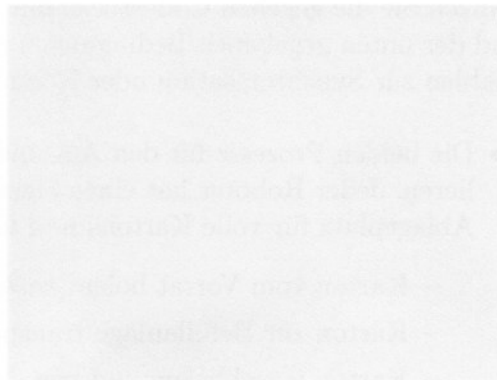
```
    Murrel_liefern();
```



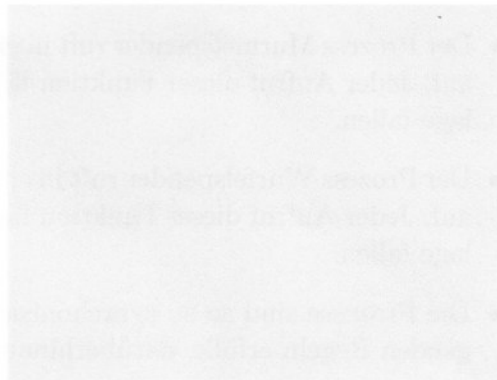
```
} while(1);
```

```
/** Wuerfelspender **/
```

```
do {
```



```
    Wuerfel_liefern();
```



```
} while(1);
```

(c) Prozesse für Roboter

```
/** Roboter R1 */
```

```
do {
```

```
    karton_aufnehmen(R1);
```

```
    karton_bereitstellen(R1);
```

```
    karton_entfernen(R1);
```

```
    karton_abtransportieren(R1);
```

```
} while (1)
```

```
/** Roboter R2 */
```

```
do {
```

```
    karton_aufnehmen(R2);
```

```
    karton_bereitstellen(R2);
```

```
    karton_entfernen(R2);
```

```
    karton_abtransportieren(R2);
```

```
} while(1);
```

2 Page Replacement (30)

Gegeben ist ein Arbeitsspeicher mit vier Frames, dessen Seiten mit unterschiedlichen Ersetzungsstrategien ersetzt werden sollen: mit OPT, LRU, bzw. mit dem Clock Algorithmus. Ergänzen Sie in den Tabellen für jeden Algorithmus die Speicherinhalte für jeden Frame nach jedem Zugriff der angegebenen Seitenzugriffsfolge an. Die Seitenzugriffsfolge ist für alle Algorithmen gleich. Sie ist jeweils in der Kopfzeile der Tabelle gegeben. Geben Sie in den Spalten die Speicherbelegung nach dem entsprechenden Seitenzugriff an und kennzeichnen Sie in der letzten, mit *PF* markierten Zeile das Auftreten von Page Faults. Der Arbeitsspeicher ist am Beginn leer. Für die ersten vier Speicherzugriffe sind die Tabellen bereits ausgefüllt.

OPT-Strategie:

	A	B	C	D	C	E	D	B	A	F	E	D	A	F	D	A	E
0	A	A	A	A													
1		B	B	B													
2			C	C													
3				D													
PF	X	X	X	X													

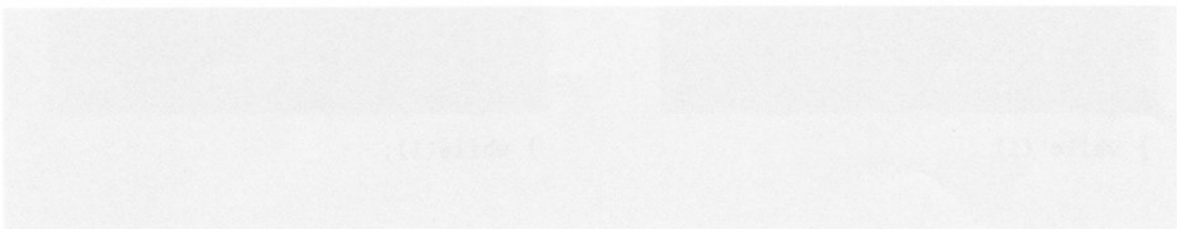
LRU-Strategie:

	A	B	C	D	C	E	D	B	A	F	E	D	A	F	D	A	E
0	A	A	A	A													
1		B	B	B													
2			C	C													
3				D													
PF	X	X	X	X													

Clock-Algorithmus:

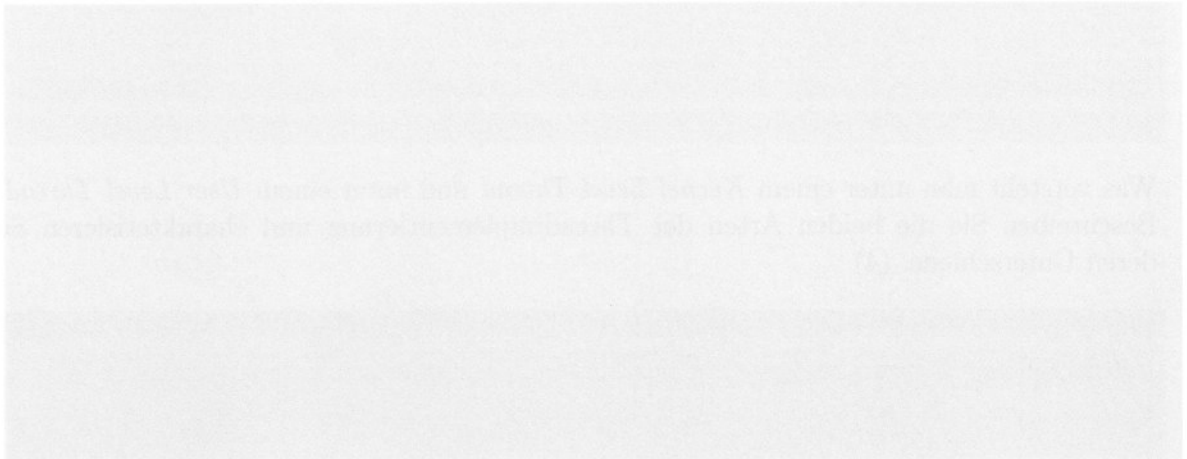
	A	B	C	D	C	E	D	B	A	F	E	D	A	F	D	A	E
0	A	A	A	A													
1		B	B	B													
2			C	C													
3				D													
PF	X	X	X	X													

Vergleichen und diskutieren Sie die Anzahl der bei den Seitenersetzungsstrategien beobachteten Page Faults. Entspricht das Ergebnis dem zu erwartenden Verhalten?

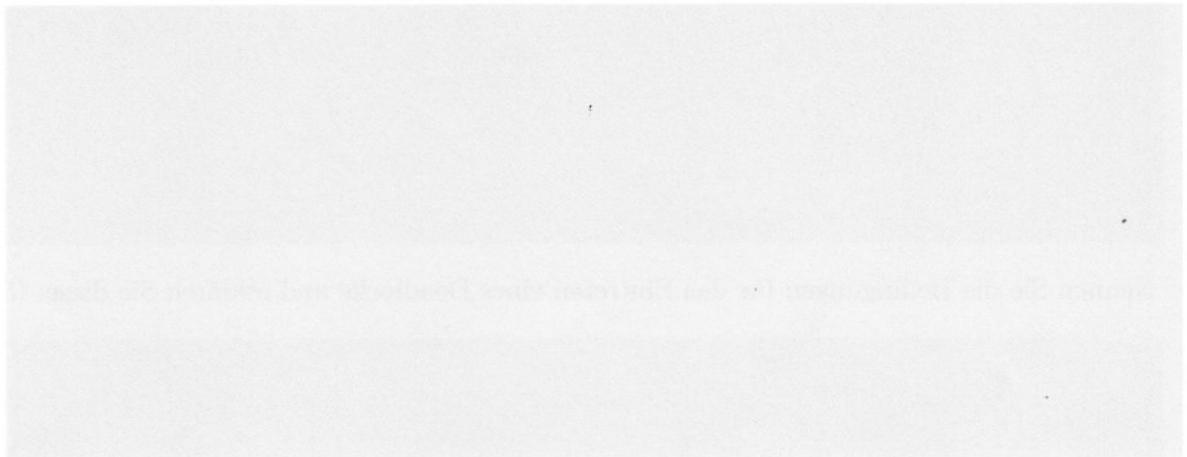


3 Fragen zu Betriebssystemen (40)

Was versteht man unter einem *Process Image*? Erklären Sie, aus welchen Teilen ein Process Image besteht? (4)



Erklären Sie die Aktionen, die vom Betriebssystem bei einem *Process Switch* durchzuführen sind. Welche Arten von Ereignissen können zu einem Process Switch führen? (3)



Nennen Sie Möglichkeiten, um in einem Paging-System Speicherschutz zu realisieren? (3)

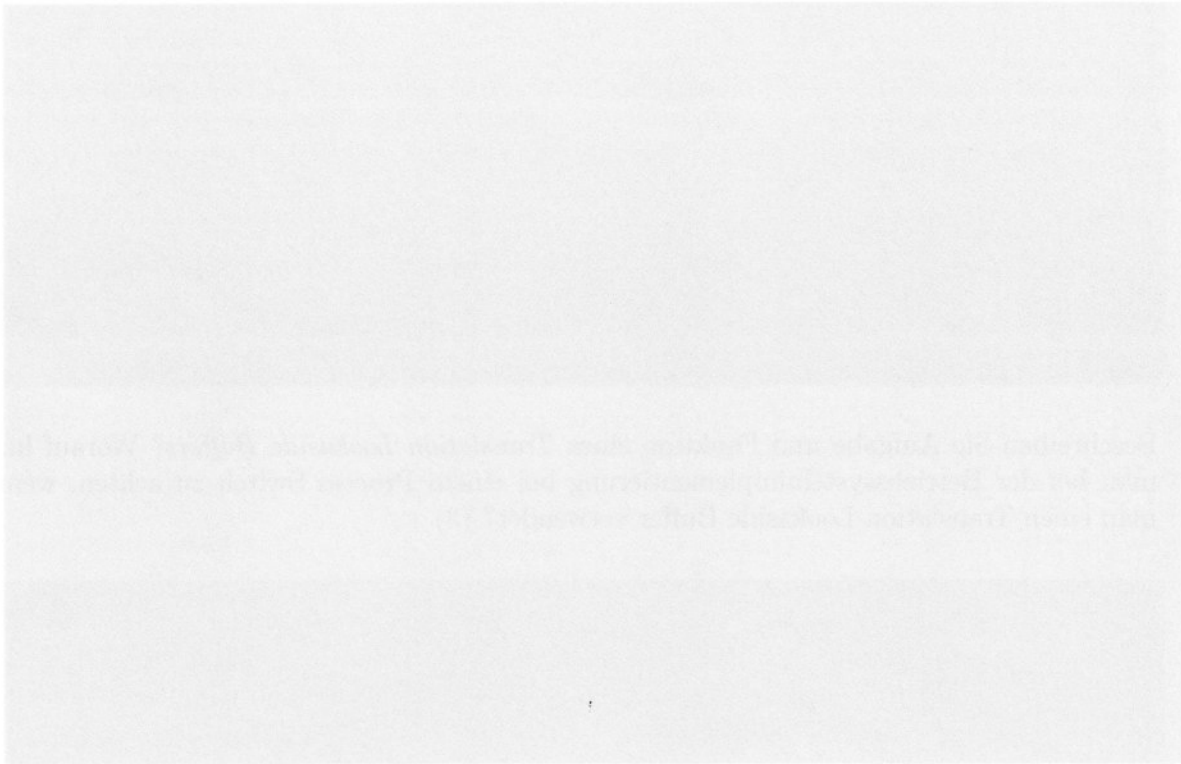


Erklären Sie die Begriffe *absoluter Pfadname* und *relativer Pfadname* und geben Sie jeweils ein Beispiel an. (2)

Was versteht man unter einem *Kernel Level Thread* und unter einem *User Level Thread*? Beschreiben Sie die beiden Arten der Threadimplementierung und charakterisieren Sie deren Unterschiede. (4)

Nennen Sie die Bedingungen für das Eintreten eines Deadlocks und erklären Sie diese. (5)

Bei der Realisierung von Dateisystemen gibt es verschiedene Möglichkeiten, um die zu einer Datei gehörenden Datenblöcke zu organisieren bzw. auffindbar zu machen (Block-Allokierung). Nennen Sie vier verschiedene Strategien zur Block-Allokierung von Dateien und beschreiben Sie diese mit ihren Vor- und Nachteilen. (5)



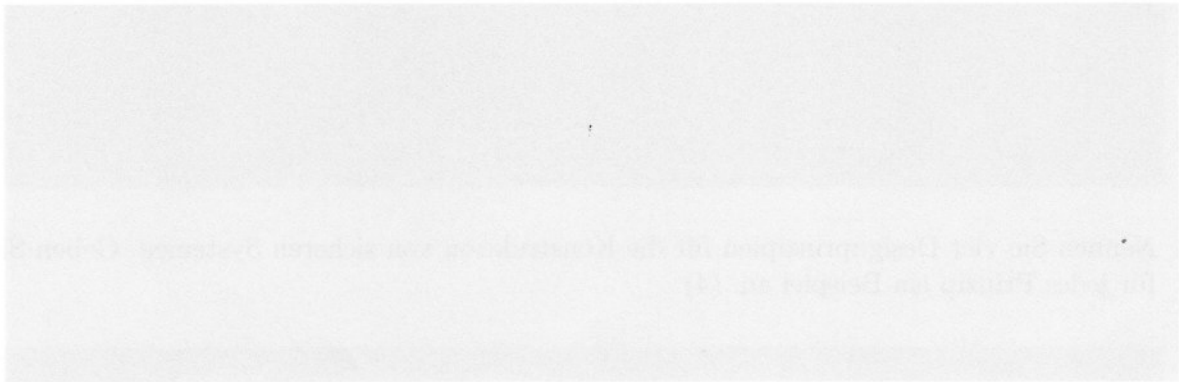
Nennen Sie vier Designprinzipien für die Konstruktion von sicheren Systemen. Geben Sie für jedes Prinzip ein Beispiel an. (4)



Nennen Sie die Arten von Optimierungszielen, die ein Scheduler beim Prozess-Scheduling verfolgen kann und geben Sie jeweils Beispiele an. (4)



Beschreiben Sie Aufgabe und Funktion eines *Translation Lookaside Buffers*? Worauf hat man bei der Betriebssystemimplementierung bei einem Process Switch zu achten, wenn man einen Translation Lookaside Buffer verwendet? (3)



Erklären Sie die Begriffe *Access Control List* und *Capability List*. Wozu und wie werden diese verwendet? (3)

