

KNr.

MNr.

Zuname, Vorname

Ges.)(100)

1.)(30)

2.)(25)

3.)(25)

4.)(20)

Zusatzblätter:

**Bitte verwenden Sie nur dokumentenechtes Schreibmaterial!**

## 1 Synchronisation (30)

Bei der Studienbeihilfenbehörde: Es gibt **ein Büro mit zwei Sachbearbeitern**. Deshalb können sich im Büro maximal zwei Studenten aufhalten. Der Student kann durch die Glastüre sehen, ob es einen Sachbearbeiter gibt, der keinen Studenten betreut. Ein Sachbearbeiter wartet so lange, bis ein Student eintritt und auf sich aufmerksam macht. Ein freier Sachbearbeiter wird nun aufsehen und den Studenten bitten seine Matrikelnummer zu nennen, um den entsprechenden Akt zu holen.

Der Sachbearbeiter geht daraufhin zum Aktenschrank: Ist der Akt nicht im Schrank bedeutet dies, dass er gerade vom Kollegen bearbeitet wird. In diesem Fall wird dies dem Studenten mitgeteilt und er wird gebeten sich draußen am Ende der Reihe anzustellen (Das Programm Student terminiert und wird neu gestartet). Ist der Akt verfügbar, holt ihn der Sachbearbeiter vom Aktenschrank. **Der Student wartet bis der** seinen Fall bearbeitende **Sachbearbeiter beim Schreibtisch ankommt**. Dann bringt der Student sein Anliegen vor und nach der Klärung des Problems mit dem Sachbearbeiter verabschiedet er sich und verlässt das Büro.

### a) Erstellen und initialisieren Sie alle benötigten Semaphore. (4)

Zum Anlegen eines Semaphores steht Ihnen das Statement `SemInit` zur Verfügung. Mit diesem kann man sowohl einfache Semaphore (`Seminit(SemaphorName, InitWert)`) als auch Semaphorearrays (`Seminit(SemaphorArrayName[Anzahl], InitWert1, InitWert2, ...)`) erzeugen und initialisieren. Der Aufruf `P(SemaphorName)` sperrt eine Semaphore und mit `V(SemaphorName)` gibt man es frei! (**Personalnummern der Sachbearbeiter:**  $\in \{0, 1\}$ .)

```
#define ANZ_SB 2
```

**b) Vervollständigen Sie das Programm für einen Studenten. (14)**

Es stehen Ihnen die Routinen `BueroBetreten` und `BueroVerlassen` zur Verfügung, welche keinerlei Rückgabewerte liefern, sowie die Funktion `SBAuswaehlen` welche die Personalnummer (0 oder 1) des aufsehenden Sachbearbeiters zurückgibt. Um dem Sachbearbeiter seine Matrikelnummer zu nennen und den Verbleib des Aktes zu erfahren verwende man `MatNrNennen(MatNr)`. Diese Funktion gibt einen negativen Wert zurück wenn der Akt gerade bearbeitet wird. (**C Syntax ist nicht erforderlich; es reicht Pseudocode**

**Personalnummern der Sachbearbeiter:**  $\in \{0, 1\}$ .)

```
/* Deklaration von Variablen. */
```

```
/* Warten, bis ein Platz im Buero frei ist. */
```

```
/* Buero betreten */
```

```
/* Begruessen und einen freien Sachbearbeiter *  
 * in seiner Beschaeftigung unterbrechen.      */  
Begruessen
```

```
/* Den SB auswaehlen und die Matrikelnummer    *  
 * nennen. Wenn der entsprechende Ordner        *  
 * nicht frei ist, das Buero wieder verlassen. */
```

```
if( ) {
```

```
    /* Der Akt ist nicht verfuegbar; Buero verlassen */
```

```

} else {
    /* Der Akt ist verfuegbar; warten bis ihn der *
    * Bearbeiter bei sich am Tisch hat.          */

```

```

AnliegenVortragen
AufSpaeterVertroestenLassen
MitDankVerabschieden
/* Das Buero Verlassen */

```

```

}

```

### c) Vervollständigen Sie das Programm eines Sachbearbeiters (12)

Verwenden Sie die Routinen: `ZumAktenschrackGehen` und `ZumSchreibtischGehen`, sowie `AktHerausnehmen(Matrikelnummer)` und `AktZurueckstellen(Matrikelnummer)`. Weiters existiert eine Funktion mit Namen `WoIstDerAkt(Matrikelnummer)`, welche den Verbleib des Aktes signalisiert (`AKT_IST_BEIM_KOLLEGEN` bzw. `AKT_IST_VERFUEGBAR`). Um die Matrikelnummer des Studenten zu erhalten, verwende man `MatNrFragen`. Die Konstante `PERSONAL_NR` ist ebenfalls vordefiniert.

(Personalnummern der Sachbearbeiter:  $\in \{0, 1\}$ .)

```

/* Deklaration der Variablen. */

```

```

while((Zeit>=0900)&&(Zeit<=1200)){
    /* Warten bis ein Student kommt */

```

```

Begruessen
/* Anhand der Matrikelnummer entscheiden, wo der Akt ist. */

```

```

if(
                                     != AKT_BEIM_KOLLEGEN){

```

```
/* Der Akt ist verfuegbar! Na dann holen wir ihn. */
```

```
/* Dem Studenten signalisieren, dass man bereit zur Arbeit ist. */
```

```
DemAnliegenLauschen  
AufSpaeterVertroesten  
Verabschieden  
/* Den Akt zurueckstellen */
```

```
}  
}
```

## 2 Deadlock (25)

Am Institut für technische Informatik werden über den Sommer Praktika von vier Studenten (**A**ndreas, **B**eatrice, **C**hristine und **D**aniela) absolviert. Zur Abwicklung ihre Praktika brauchen die Leute folgende Ressourcen:

	<b>A</b> ndreas	<b>B</b> eatrice	<b>C</b> hristine	<b>D</b> aniela
<b>L</b> aserdrucker	1	1	1	0
<b>W</b> orkstation	1	4	1	2
<b>K</b> notenrechner	5	3	2	9

Abgesehen von diesen Ressourcen, die sich die Leute teilen müssen, hängen die einzelnen Arbeiten nicht voneinander ab, können also (bzw. sollen) parallel ausgeführt werden. Weiters werden Ressourcen von den Studenten erst dann wieder freigegeben, wenn ihr Praktikum beendet ist.

Insgesamt stehen am Institut 1 **L**aserdrucker, 4 **W**orkstations und 9 **K**notenrechner zur Verfügung.

Im Augenblick (Initialzustand) sieht die aktuelle Zuteilung der Ressourcen zu den Praktikanten folgendermaßen aus:

- **A**ndreas hat bereits den **L**aserdrucker und 3 **K**notenrechner belegt.
- **B**eatrice arbeitet gerade auf 2 **W**orkstations und 1 **K**notenrechner.
- **C**hristine hat zur Zeit 2 **K**notenrechner in Verwendung.
- **D**aniela benützt 1 **W**orkstation und 1 **K**notenrechner.

### a) Initialzustand (5)

Ermitteln Sie *Ressource-Vektor* (RV) und *Available-Vektor* (AV) sowie *Claim-Matrix* (CM) und *Allocation-Matrix* (AM) für den obig beschriebenen Initialzustand.

$$\begin{aligned}
 RV &= \begin{pmatrix} L & W & K \\ \text{[box]} & \text{[box]} & \text{[box]} \end{pmatrix} & AV &= \begin{pmatrix} L & W & K \\ \text{[box]} & \text{[box]} & \text{[box]} \end{pmatrix} \\
 CM &= \begin{pmatrix} & L & W & K \\ A & \text{[box]} & \text{[box]} & \text{[box]} \\ B & \text{[box]} & \text{[box]} & \text{[box]} \\ C & \text{[box]} & \text{[box]} & \text{[box]} \\ D & \text{[box]} & \text{[box]} & \text{[box]} \end{pmatrix} & AM &= \begin{pmatrix} & L & W & K \\ A & \text{[box]} & \text{[box]} & \text{[box]} \\ B & \text{[box]} & \text{[box]} & \text{[box]} \\ C & \text{[box]} & \text{[box]} & \text{[box]} \\ D & \text{[box]} & \text{[box]} & \text{[box]} \end{pmatrix}
 \end{aligned}$$

## b) Resource Allocation Denial (20)

Benützen Sie zur Abwicklung der weiteren Ressourcenzuteilung die Strategie des *Resource Allocation Denial*. Wenden Sie ausgehend vom Initialzustand den *Banker's Algorithmus* solange an, bis entweder alle Praktika beendet wurden, oder ein Deadlock auftritt.

Notieren Sie hierbei, welcher Praktikant gerade sein Praktikum beenden konnte (also aktiv war) bzw. wenn ein Deadlock entstanden ist, kreuzen Sie an, welche Praktikanten daran beteiligt sind. Es müssen nur die tatsächlich verwendeten Felder ausgefüllt werden.

- ☐ Es liegt ein Deadlock vor, an dem folgende Praktikanten beteiligt sind:

☐ Andreas

☐ Beatrice

☐ Christine

☐ Daniela

- ☐  konnte das Praktikum beenden. Der neue Zustand sieht wie folgt aus:

$$CM = \begin{pmatrix} A & L & W & K \\ B & & & \\ C & & & \\ D & & & \end{pmatrix} \quad AM = \begin{pmatrix} A & L & W & K \\ B & & & \\ C & & & \\ D & & & \end{pmatrix} \quad AV = \begin{pmatrix} L & W & K \\ & & \\ & & \end{pmatrix}$$

- ☐ Es liegt ein Deadlock vor, an dem folgende Praktikanten beteiligt sind:

☐ Andreas

☐ Beatrice

☐ Christine

☐ Daniela

- ☐  konnte das Praktikum beenden. Der neue Zustand sieht wie folgt aus:

$$CM = \begin{pmatrix} A & L & W & K \\ B & & & \\ C & & & \\ D & & & \end{pmatrix} \quad AM = \begin{pmatrix} A & L & W & K \\ B & & & \\ C & & & \\ D & & & \end{pmatrix} \quad AV = \begin{pmatrix} L & W & K \\ & & \\ & & \end{pmatrix}$$

☐ Es liegt ein Deadlock vor, an dem folgende Praktikanten beteiligt sind:

☐ Andreas

☐ Beatrice

☐ Christine

☐ Daniela

☐  konnte das Praktikum beenden. Der neue Zustand sieht wie folgt aus:

$$CM = \begin{pmatrix} & L & W & K \\ A & \text{shaded} & \text{shaded} & \text{shaded} \\ B & & & \\ C & & & \\ D & & & \end{pmatrix} \quad AM = \begin{pmatrix} & L & W & K \\ A & \text{shaded} & \text{shaded} & \text{shaded} \\ B & & & \\ C & & & \\ D & & & \end{pmatrix} \quad AV = \begin{pmatrix} & L & W & K \\ & \text{shaded} & \text{shaded} & \text{shaded} \\ & & & \\ & & & \end{pmatrix}$$

☐ Es liegt ein Deadlock vor, an dem folgende Praktikanten beteiligt sind:

☐ Andreas

☐ Beatrice

☐ Christine

☐ Daniela

☐  konnte das Praktikum beenden. Der neue Zustand sieht wie folgt aus:

$$CM = \begin{pmatrix} & L & W & K \\ A & \text{shaded} & \text{shaded} & \text{shaded} \\ B & & & \\ C & & & \\ D & & & \end{pmatrix} \quad AM = \begin{pmatrix} & L & W & K \\ A & \text{shaded} & \text{shaded} & \text{shaded} \\ B & & & \\ C & & & \\ D & & & \end{pmatrix} \quad AV = \begin{pmatrix} & L & W & K \\ & \text{shaded} & \text{shaded} & \text{shaded} \\ & & & \\ & & & \end{pmatrix}$$

c) Erklären Sie den Unterschied zwischen den Begriffen *Deadlock*, *Lifelong* und *Starvation* (5)

### 3 Speicherverwaltung (25)

#### a) Translation Lookaside Buffer (15)

Das gegebene Speicherverwaltungssystem basiert auf der Paging-Technik und verwendet einen *Translation Lookaside Buffer (TLB)* mit drei Einträgen. Der TLB wird mit der *Least Recently Used* Strategie verwaltet, d.h., wenn für eine neu einzutragende Seite kein Eintrag mehr frei ist, wird jener Eintrag ersetzt, der am längsten nicht benutzt wurde. Zu diesem Zweck enthält der TLB für jeden Eintrag ein Feld, das angibt seit wievielen TLB Zugriffen der entsprechende Eintrag nicht benutzt wurde. Bei jedem Zugriff werden die Zähler jener Einträge erhöht, die nicht gebraucht wurden, der Zähler des verwendeten oder neuen Eintrags wird auf 0 gesetzt.

Der TLB und die benötigte Page Table werden mittels **Associative Mapping** angesprochen. Die Pagegröße beträgt 4096 Bytes. Die Page-Table enthält für jeden Eintrag ein Feld *loaded*, das anzeigt, ob sich die entsprechende Seite im Hauptspeicher befindet.

Eine virtuelle Speicheradresse hat folgendes Format:

Page#(4 Bit)	Offset (12 Bit)
--------------	-----------------

Eine physikalische Speicheradresse hat folgendes Format:

Frame#(8 Bit)	Offset (12 Bit)
---------------	-----------------

Die Speicheradressen, Frame- und Pagenummern sind im Hexadezimalsystem (16er-System) angegeben.

- Auf dieser Seite ist ein Translation Lookaside Buffer und eine Page Table vorgegeben. Simulieren Sie ausgehend von diesen Daten den hintereinanderfolgenden Zugriff auf die virtuellen Speicheradressen auf der nächsten Seite.
- Befindet sich eine Seite nicht im Hauptspeicher, so können Sie die Nummer des Page-Frames für die Page aus den nicht belegten frei wählen. Tragen Sie diese in der Page Table auf dieser Seite ein und markieren Sie das entsprechende *Loaded*-Feld.
- Bestimmen Sie, ob es zu einem TLB Hit oder Miss kommt und ob es zu einem Main Memory Page Hit kommt oder nicht (Kreuzen Sie entsprechend **Ja** oder **Nein** an).
- Geben Sie weiters jeweils den Inhalt des TLB **nach** dem Zugriff auf die Page an.

#### Ausgangssituation

Translation Lookaside Buffer

Page#	Frame#	Last Use
2	0x47	0
4	0x00	1

Page Table

Page#	Loaded	Frame#
0	<input checked="" type="checkbox"/>	0x2F
1	<input type="checkbox"/>	
2	<input checked="" type="checkbox"/>	0x47
3	<input type="checkbox"/>	
4	<input checked="" type="checkbox"/>	0x00



Virtuelle Adresse

0x0313

Physikalische Adresse

TLB Hit ☐ Ja ☐ Nein

Page Hit ☐ Ja ☐ Nein

Translation Lookaside Buffer

Page#	Frame#	Last Use

Virtuelle Adresse

0x042F

Physikalische Adresse

TLB Hit ☐ Ja ☐ Nein

Page Hit ☐ Ja ☐ Nein

Translation Lookaside Buffer

Page#	Frame#	Last Use

Virtuelle Adresse

0x1141

Physikalische Adresse

TLB Hit ☐ Ja ☐ Nein

Page Hit ☐ Ja ☐ Nein

Translation Lookaside Buffer

Page#	Frame#	Last Use

Virtuelle Adresse

0x4711

Physikalische Adresse

TLB Hit ☐ Ja ☐ Nein

Page Hit ☐ Ja ☐ Nein

Translation Lookaside Buffer

Page#	Frame#	Last Use

## b) TLB-Zeitverhalten(4)

Man nehme an, ein Zugriff auf den TLB oder ein Zugriff auf den Cache-Speicher dauert 15ns, ein Zugriff auf den Hauptspeicher braucht 60ns und das Laden einer Seite vom Platte dauert 10ms. Gehen Sie davon aus, dass die Pagetable *zur Gänze im Hauptspeicher* gehalten wird und die durch den TLB referenzierten Seiten sowie der TLB vom Cache-Speicher unterstützt werden. Rechnen Sie den Quotienten minimale Zugriffszeit : maximale Zugriffszeit auf zwei signifikante Stellen genau aus.

Quotient  $\frac{t_{min}}{t_{max}} =$

## c) Verständnisfragen (6)

- Eine Austauschstrategie, die auf alle Seiten des Hauptspeichers angewandt wird, nennt man
  - ☐ lokale Ersetzungsstrategie
  - ☐ globale Ersetzungsstrategie
- Beim *Fixed Partitioning* sind die Partionen *immer* von gleicher Größe.
  - ☐ richtig
  - ☐ falsch
- Welche dieser Replacement-Policies ist am einfachsten zu implementieren?
  - ☐ Least recently used
  - ☐ Clock algorithmus
  - ☐ First in - first out
- Die Clock Replacement Strategie liefert weniger Page Faults als die Least Recently Used Replacement Strategy.
  - ☐ richtig
  - ☐ falsch
- Beim *Fixed Partitioning* können sich Partitionen im Hauptspeicher überlappen.
  - ☐ richtig
  - ☐ falsch
- Große Seitengrößen bei reinem Paging führen zu großen Seitentabellen (page tables).
  - ☐ richtig
  - ☐ falsch
- Um die interne Fragmentierung zu reduzieren, muss man die *Page Size*
  - ☐ verringern
  - ☐ vergrößern
- Wieviele Zugriffe auf die Pagetable benötigt ein System mit Translation Lookaside Buffer im Falle eines TLB Hits, um die virtuelle Adresse aufzulösen?

Zugriff(e)

## 4 Security (20)

### a) (3)

Welche drei Sicherheitsanforderungen gibt es?

### b) (3)

Welche Sicherheitsanforderungen sind in den folgenden Beispielen verletzt? (bitte kurze **Begründung** angeben!)

- Ein Druckprogramm erlaubt fälschlicherweise den Ausdruck einer Datei mit geheimen Daten.

- Ein Benutzer ändert unerlaubterweise das Passwort eines autorisierten Benutzers.

### c) (1)

Ein frustrierter Programmierer, der sich aufgrund seiner bevorstehenden Kündigung an seiner Firma rächen will, fügt in das Buchhaltungsprogramm eine Routine ein, die einen Tag nach seiner Kündigung alle Daten löscht. Geben Sie den englischen Fachbegriff für diese Routine an!

### d) (1)

Ein unautorisierter Benutzer kann alle Files in einem Computersystem lesen. Wie bezeichnet man diesen *threat* und gegen welche Sicherheitsanforderung verstößt er?

**e) (4)**

Erklären Sie die *Grundzüge* des Public-Key Verschlüsselungsverfahrens, sowie die *Vor-* und *Nachteile* gegenüber konventionellen Verschlüsselungsverfahren.

Grundzüge:

Vorteile:

Nachteile:

**f) (3)**

Was für potenzielle Schwachstellen(mind. 3) weist das Passwort *password* auf?

**g) (3)**

Welche Regeln(mind. 3) sollten bei der Wahl eines "sicheren" Passwortes angewandt werden?