

---

Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer
--------------------	----------------	----------------------

---

Kennzahl: \_\_\_\_\_

Kennwort: \_\_\_\_\_

Arbeitsplatz-Nr.: \_\_\_\_\_

**Frühjahr**  
**2010**

**66114**

---

**Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen**  
**— Prüfungsaufgaben —**

---

Fach: **Informatik (vertieft studiert)**

Einzelprüfung: **Datenbank und Betriebssysteme**

Anzahl der gestellten Themen (Aufgaben): 4 Aufgaben, von denen 2 gemäß untenstehender Auswahlregel zu bearbeiten sind

Anzahl der Druckseiten dieser Vorlage: 17

---

<p>Zu den zwei Themenschwerpunkten A (Datenbanksysteme) und B (Betriebssysteme) ist jeweils entweder die Aufgabe 1 <u>oder</u> 2 zu wählen. Auf der Vorderseite des Kopfbogens sind im Feld „Gewähltes Thema: Nr.“ die Nummern der beiden gewählten Aufgaben anzugeben (z. B. A2, B1)!</p>
--

Bitte wenden!

Themenschwerpunkt A

## (Datenbanksysteme)

## AUFGABE 1

## 1. Relationale Algebra

Die relationale Algebra wird aufgebaut über einer Grundmenge von mengenwertigen Operationen. Diese Grundoperationen können auf Relationen angewendet werden und erzeugen als Ergebnis wieder eine Relation.

**Notation:**  $\pi$  = Projektion;  $\sigma$  = Selektion;  $\bowtie$  = Join;  $\times$  = kartesisches Produkt;  $\setminus$  = Mengendifferenz;  $\cap$  = Schnittmenge;  $\cup$  = Vereinigungsmenge;  $\rho$  = Umbenennen;  $\div$  = relationale Division

- Beschreiben Sie ein vollständiges SQL-Statement (oder mehrere) und geben Sie an, welche Klausel mit welcher Operation aus der relationalen Algebra korrespondiert (alle Operationen müssen verwendet werden).
- Definieren Sie die mengenwertige Operation „Division“. Auf welche Grundoperation kann die Division zurückgeführt werden und wie? Ein Beweis ist nicht erforderlich.

## 2. Transaktionen und Transaktionsverarbeitung

- Nennen und definieren Sie die vier wesentlichen Merkmale einer Datenbanktransaktion.
- Erklären Sie steal, no-steal, force, no-force im Zusammenhang mit der Fehlerbehandlung.
- Fassen Sie die vier Kombinationen von force/no-force, steal/no-steal hinsichtlich der Anforderungen an die Redo- und Undo-Recovery auf folgende Weise zusammen:

	Force	No-Force
Steal	<ul style="list-style-type: none"> <li>• Kein Redo</li> <li>• Undo</li> </ul>	<ul style="list-style-type: none"> <li>•</li> <li>•</li> </ul>
No-Steal	<ul style="list-style-type: none"> <li>•</li> <li>•</li> </ul>	<ul style="list-style-type: none"> <li>•</li> <li>•</li> </ul>

Übertragen Sie die Tabelle und ergänzen Sie diese.

### 3. ER-Modellierung und Relationenmodell

**Szenario:** Sie sollen ein System zur Verwaltung von Firmen entwerfen.

- Eine Firma ist eindeutig bezeichnet durch ihre Handelsregisternummer. Daneben hat sie einen Namen und eine Adresse, die sich aus Straße, Hausnummer, Ort und Postleitzahl zusammensetzt.
- Firmen bestehen aus Abteilungen. Eine Firma kann dabei aus mehreren Abteilungen bestehen, eine Abteilung kann jedoch immer nur zu einer Firma gehören. Es kann Firmen geben, die nicht in Abteilungen eingeteilt sind, aber jede Abteilung gehört zu einer Firma. Abteilungen haben eine eindeutige AbteilungsID und einen Namen.
- Firmen beschäftigen Mitarbeiter. Ein Mitarbeiter kann nur für eine Firma arbeiten, eine Firma aber beliebig viele Mitarbeiter haben. Ein Mitarbeiter hat eine eindeutige Sozialversicherungsnummer (SVN), einen Namen (bestehend aus Name und Vorname), ein Geburtsdatum und ein Alter, das sich aus dem Geburtsdatum errechnet. Daneben hat ein Mitarbeiter eines oder mehrere Spezialgebiete, auf denen er Fachmann ist.
- Mitarbeiter können Angehörige haben. Zu diesen werden Geburtstag und Name gespeichert. Ein Mitarbeiter kann keine zwei Angehörigen mit gleichem Namen haben.
- Abteilungen bearbeiten Projekte. Ein Projekt ist gekennzeichnet durch seine ProjektID. Daneben besteht es aus einem Namen und einem Kunden. Ein Projekt kann von mehreren Abteilungen bearbeitet werden und eine Abteilung kann an mehreren Projekten arbeiten. Um nicht den Überblick zu verlieren, soll der Zeitraum gespeichert werden, in dem eine Abteilung ein Projekt bearbeitet. Dieser Zeitraum gliedert sich in die Felder „von“ und „bis“.

a) ER-Modellierung:

Erstellen Sie ein ER-Modell des oben beschriebenen Szenarios. Geben Sie Kardinalitäten von Beziehungen sowie Rollennamen an. Verwenden Sie auch abgeleitete, zusammengesetzte und mehrwertige Attribute, mehrstellige Beziehungstypen und schwache Entitätstypen, wo es sinnvoll ist.

b) Relationenmodell:

Überführen Sie das ER-Modell in ein Relationenschema.

**Notation:** Unterstreichen Sie Primärschlüssel; kennzeichnen Sie Fremdschlüssel durch Angabe der referenzierten Relation in eckigen Klammern.

**Beispiel:**

Autor(AID, Name)

Buch(ISBN, Schriftsteller[Autor])

## 4. SQL

Ein Internetauktionshaus nutzt eine relationale Datenbank zum Speichern seiner relevanten Daten. Angenommen ist folgender Auszug aus dem Tabellenschema zu den Benutzern, Auktionen und Geboten:

User(UId, Name,...)

Auction(AId, Name, Begin\_Auction, End\_Auction, Seller[User], ...)

Bid(UId, AId, Bid\_Time, Amount)

Das Auktionshaus will ein Bewertungssystem einführen, um die Risiken für Verkäufer eindämmen zu können.

Folgende Tabelle soll zu den existierenden Tabellen hinzugefügt werden:

Review(RId, AId[Auction], BuyerID[User], Rating, Rating\_Time)

RId, AId und BuyerID sind beliebige ganzzahlige Datentypen, Rating ist ein ganzzahliger Datentyp, der die Werte 1,2,3,4,5 annehmen kann, und Rating\_Time ist ein timestamp. RId ist Primärschlüssel und die beiden Fremdschlüssel dürfen nie leer sein.

- a) Geben Sie das vollständige CREATE-Statement zum Anlegen von Review an, das möglichst viele Constraints erfüllt.
- b) Erstellen Sie eine Sicht mit dem Namen Auction\_Review mit den Attributen (AId, UId, Durchschnittsrating):  
Zu jeder Auktion (gekennzeichnet mit AId) soll das Durchschnittsrating jedes Nutzers (UId) angegeben werden, der auf diese Auktion geboten hat. Die Durchschnittsratings sollen auf allen Ratings des Nutzers basieren, die zeitlich vor Ende der betrachteten Auktion vergeben wurden.
- c) Erstellen Sie eine Sicht mit dem Namen Auction\_Max\_Amount mit den Attributen (AId, MaxAmount):  
Zu jeder abgeschlossenen Auktion (aktueller Zeitpunkt mit sysdate() ) soll der größte gebotene Betrag angegeben werden.
- d) Ausgehend von der Sicht Auction\_Max\_Amount sollen die 5 TOP-Verkäufer (Umsatz) aus dem Jahr 2005 bestimmt werden (Auktionsende in 2005). Geben Sie die SQL-Query an.

## AUFGABE 2

### 1. Integritätsbedingungen

Eine Aufgabe von Datenbankmanagementsystemen ist die Erhaltung der Datenintegrität.

- a) Welche Arten von Integritätsbedingungen gibt es?
- b) Geben Sie für jede Art der Integritätsbedingungen je ein Beispiel an.

### 2. Modellierung

Der Veranstalter der Ausstellung "Becit" möchte diese in einer relationalen Datenbank verwalten:

Zu jedem Ausstellungsraum gibt es eine eindeutige Raumnummer (RNr). Ein Raum hat eine bestimmte Fläche (in  $m^2$ ), eine Höhe (in  $m$ ) und zusätzliche Sonderausstattungen wie Beamer, Fernseher, Stühle, Bänke, usw. Die Ausstellungsstände werden in Räumen aufgestellt. Es hat sich eingebürgert, den Ständen innerhalb von Räumen Buchstaben zuzuordnen (SId). So ist beispielsweise Stand "3B" der 2. Stand in Raum 3. Zu Ständen ist bekannt, wieviel Fläche (in  $m^2$ ) sie benötigen, die Anzahl benötigter Tische und Stühle und möglicherweise benötigte Sonderausstattungen des entsprechenden Raumes.

Die Stände werden dann den Anbietern zugeordnet. Für die Anbieter werden eindeutige Ids vergeben. Zusätzlich sind von diesen Name, Adresse und Telefonnummer bekannt.

Das Personal kümmert sich um den reibungslosen Ablauf der Ausstellung. Jedes Mitglied des Personals besitzt eine Personalnummer (PNr). Ebenfalls bekannt ist der Lohn des Personals.

Das Personal unterteilt sich in Auf-/Abbau-Personal, Sicherheitspersonal und Reinigungspersonal. Das Auf-/Abbau-Personal bekommt bestimmte Stände zugewiesen, die es auf- und abbaut. Das Sicherheitspersonal kümmert sich um die Sicherheit der Ausstellung. Die einzelnen Mitglieder des Sicherheitspersonals patrouillieren hierfür in den ihnen zugewiesenen Ausstellungsräumen. Sie können über ein Funkgerät verfügen und ggf. auch bewaffnet sein. Das Reinigungspersonal kümmert sich um die Endreinigung der Ausstellungsräume. Da hier in verschiedenen Räumen unterschiedlich viel Arbeit anfallen kann, können vorab keine Räume zugeordnet werden.

Zu beachten ist insbesondere, dass sämtliche Personen des Personals auch mehrere Aufgaben übernehmen können, d.h. sie können beim Aufbau mithelfen, zusätzlich während der Ausstellung für die Sicherheit sorgen und am Ende ggf. die Räume mit reinigen.

- a) Entwerfen Sie für das beschriebene Szenario ein ER-Diagramm. Bestimmen Sie hierzu:
  - die Entity-Typen, die Relationship-Typen und jeweils deren Attribute;
  - ein passendes ER-Diagramm;
  - die Primärschlüssel der Entity-Typen, welche Sie anschließend in das ER-Diagramm eintragen;
  - die Kardinalitäten der Relationship-Typen, welche Sie ebenfalls in das ER-Diagramm eintragen.

- b) Überführen Sie das ER-Modell aus Aufgabe a) in ein verfeinertes relationales Modell. Geben Sie hierfür die verallgemeinerten Relationenschemata an.
- c) Bei den Zuordnungen der Sonderausstattungen kann es zur Verletzung einer Integritätsbedingung kommen.
- Begründen Sie, woran dies liegt.
  - Wie lässt sich das Problem innerhalb einer Datenbank verhindern?

### 3. Normalformen

Gegeben sei das folgende Relationenschema

*Kaeufe*: {[ *ArtikelId*, *ArtikelBezeichnung*, *EinkaufsPreis*, *VerkaufsPreis*, *Datum*,  
*TagesAktion*, *KundeName*, *KundeVorname*, *KundeGebdatum*, *Filiale*, *Verkaeuer* ]}

und eine Menge  $F$  von funktionalen Abhängigkeiten. **Hinweis:** Die Abkürzungen innerhalb der Menge  $F$  beziehen sich auf die Großbuchstaben in den Attributnamen der Relation *Kaeufe*.

$$F := \{ \begin{array}{l} AI \rightarrow AB \\ AI, AB \rightarrow EP \\ KN, KV \rightarrow KG \\ AI, D, KN, KV \rightarrow V \\ AI, AB, KN, KV, D, V \rightarrow F, VP \\ D \rightarrow TA \\ V \rightarrow F \end{array} \}$$

- Ermitteln Sie sämtliche Schlüsselkandidaten von *Kaeufe*.
- Zeigen Sie, dass das Relationenschema *Kaeufe* nicht in dritter Normalform ist.
- Bestimmen Sie eine kanonische Überdeckung  $F_c$  zu  $F$ .
- Bringen Sie *Kaeufe* mit dem Synthesealgorithmus in 3NF.

### 4. SQL und Relationale Algebra

Gegeben sei folgendes Schema der relationalen Datenbank eines Aktienhändlers:

*Aktie*{*ANr*, *Name*, *Firma*[*Firma*], *Ausgabedatum*, *Ausgabepreis*}

*Aktienkurs*{*ANr*[*Aktie*], *Datum*, *Zeit*, *Wert*}

*Firma*{*Name*, *Adresse*, *Telefon*}

*Kunde*{*KndNr*, *Name*, *Geburtsdatum*, *Telefon*}

*Konto*{*KntNr*, *KndNr*[*Kunde*], *Eroeffnungsdatum*, *Saldierungsdatum*}

Fortsetzung nächste Seite!

*Buchung*{*KntNr*[*Konto*], *BNr*, *ANr*[*Aktie*], *Aktienzahl*, *EKdatum*, *EKzeit*, *EKpreis*,  
*VKdatum*, *VKzeit*, *VKpreis* }

**Hinweis:** Für Fremdschlüssel wird nach dem Format <Attr> [<RefTable>] die referenzierte Tabelle angegeben. Beispielsweise ist *KndNr*[*Kunde*] in der Relation *Konto* ein Fremdschlüssel der Relation *Kunde*.

Sämtliche verfügbaren Aktien sind in der Relation *Aktie* enthalten. Die zugehörigen Kurse werden in der Relation *Aktienkurs* festgehalten. Zu jedem (im Rahmen dieser Aufgabe notwendigen) Zeitpunkt existiert ein korrespondierendes Tupel. Die Aktiengesellschaften (also Firmen, welche Aktien ausgeben) sind in der Relation *Firma* gespeichert. Die Kunden des Aktienhändlers werden in der Relation *Kunde* verwaltet. Jeder Kunde kann beliebig viele Konti eröffnen. Diese werden in der Relation *Konto* aufgeführt und dem Kunden direkt zugeordnet. Aktienkäufe und Aktienverkäufe erfolgen paketweise. Kauft ein Kunde ein Paket einer Aktie, so wird ein neues Tupel in der Relation *Buchung* erstellt. Die Buchungsnummern werden für jedes Konto jeweils in aufsteigender Reihenfolge beginnend bei 1 vergeben. Innerhalb eines neuen Tupels werden Umfang des Pakets (*Aktienzahl*), Datum und Zeitpunkt des Kaufs und insbesondere der Preis des Aktienpakets festgehalten. Die entsprechenden Attribute für einen Verkauf werden zu *NULL* initialisiert. Wird später ein Verkauf des Aktienpakets durchgeführt, so werden die Werte für *VKdatum*, *VKzeit* und *VKpreis* entsprechend gesetzt. Aktuell sei der Zeitpunkt 13:00:00 Uhr und das Datum 2008/07/20. Für Datumsangaben gelte insbesondere, dass eine Verschiebung des Datums um  $t$  Tage durch folgende Addition durchgeführt werden kann: 2008/07/20 +  $t$ . Beispielsweise ergibt 2008/07/20 + 20 = 2008/08/09.

- a) Formulieren Sie die Anfragen in den folgenden Teilaufgaben (i - ii) in der Anfragesprache SQL. Zur Vereinfachung dürfen Sie sich beliebige Sichten (Views) definieren.
- i) Gesucht sind alle Aktienanteile, welche Kunde "Müller" aktuell hält. Geben Sie für diese Kontonummer, Buchungsnummer, Aktiennummer und die zugehörigen Firmennamen aus.
  - ii) Nehmen Sie an, das Ergebnis aus Teilaufgabe i) sei als View *AktuelleAktienMueller* verfügbar.  
Geben Sie nun gruppiert nach den Firmennamen die Gewinne/Verluste aus, welche Müller beim Verkauf seiner aktuellen Aktienanteile machen würde.
- b) Formulieren Sie folgende Anfrage in der relationalen Algebra:  
Für welche Firmen hat sich der Aktienkurs im 1. Jahr nach der Ausgabe mindestens verdoppelt?

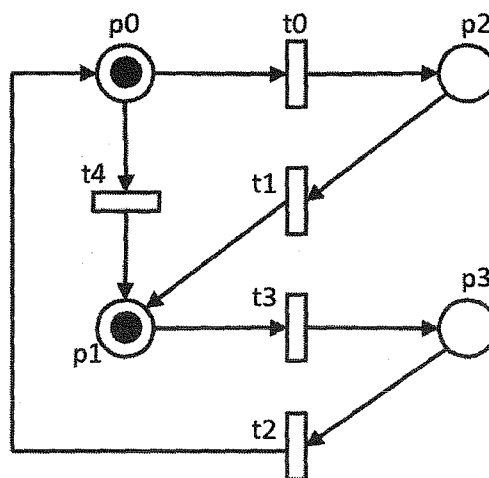
Themenschwerpunkt B

(Betriebssysteme)

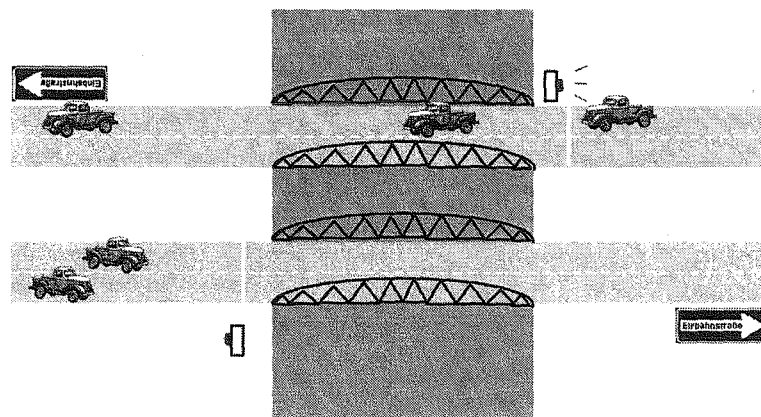
## AUFGABE 1

## 1. Petrinetze

- a) Gegeben sei folgendes boolsches Petri-Netz (boolsch bedeutet: Alle Stellen haben Kapazität 1):



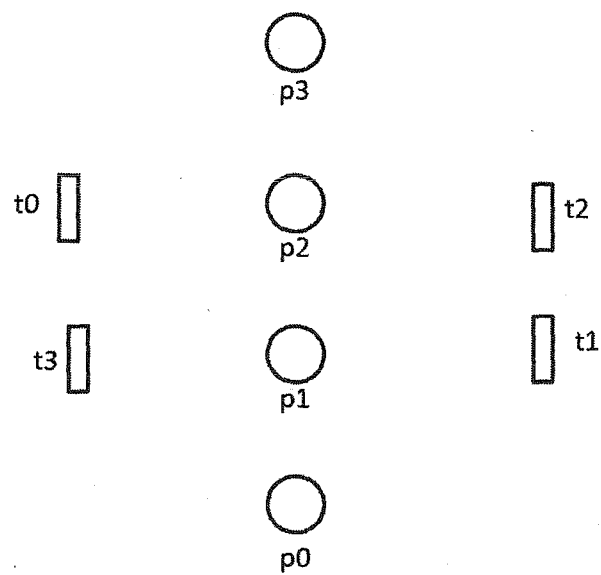
- Geben Sie den Erreichbarkeitsgraphen für dieses Petrinetz an!
  - Argumentieren Sie anhand des Erreichbarkeitsgraphen, ob das Netz verklemmt ist!
- b) Zwei Ufer sind durch zwei Brücken mit nur je einer Fahrtrichtung verbunden. Eine Brücke kann jedoch immer nur ein Fahrzeug tragen.



Gegeben sei folgende unvollständige Skizze für ein Petrinetz, das diese Bedingung garantiert. Die Stellen haben jeweils die Kapazität  $\infty$ .

Fortsetzung nächste Seite!





- i) Vervollständigen Sie die Skizze indem Sie geeignete Anfangsmarkierungen, Übergänge und Benennungen für Stellen und Transitionen eintragen!
- ii) Begründen Sie, warum Ihr Ansatz das Gewünschte leistet!

## 2. Wechselseitiger Ausschluss, Deadlocks

- a) Betrachten Sie folgende Ansätze zur Garantie des wechselseitigen Ausschlusses (Assembler-artiger Code):

### Ansatz A:

```
enterCriticalRegion:
    LOAD REGISTER, LOCK
    MOVE LOCK, #1
    CMP REGISTER, #0
    JNE enterCriticalRegion
    RET
leaveCriticalRegion:
    MOVE LOCK, #0
    RET
```

### Ansatz B:

```
enterCriticalRegion:
    TSL REGISTER, LOCK
    CMP REGISTER, #0
    JNE enterCriticalRegion
    RET
leaveCriticalRegion:
    MOVE LOCK, #0
    RET
```

- i) Welcher der beiden Ansätze garantiert den wechselseitigen Ausschluss? Begründen Sie Ihre Antwort!
- ii) Was ist ein Nachteil beider Ansätze?
- b) In Java kann der wechselseitige Ausschluss unter Verwendung des Schlüsselwortes `synchronized` realisiert werden. Diese Konstrukte realisieren das Monitor-Konzept zum wechselseitigen Ausschluss. Dies funktioniert intern mit *intrinsic locks*. Jedem Objekt ist ein *intrinsic lock* zugeordnet. Threads, die eine `synchronized` Methode eines Objekts betreten, erwerben das *intrinsic lock* des Objekts. Nun kann kein anderer Prozess eine `synchronized` Methode des Objekts mehr betreten.

Erklären Sie warum es in folgendem Code zu einem Deadlock kommen kann! Überprüfen Sie die vier Deadlock-Bedingungen (Coffman-Bedingungen) an diesem Beispiel!

```
public class Duchess {
    private final String name;
    public Duchess(String name) {
        this.name = name;
    }
    public String getName() {
        return this.name;
    }
    public synchronized void kiss(Duchess kisser) {
        System.out.println(this.name + ": " + kisser.getName() + " has kissed me!");
        kisser.kissBack(this);
    }
    public synchronized void kissBack(Duchess kisser) {
        System.out.println(this.name + ": " + kisser.getName() + " has kissed me back!");
    }
}

public class Deadlock {
    public static void main(String[] args) {
        final Duchess josephine = new Duchess("Josephine");
        final Duchess emily = new Duchess("Emily");
        new Thread(new Runnable() {
            public void run() { emily.kiss(josephine); }
        }).start();
        new Thread(new Runnable() {
            public void run() { josephine.kiss(emily); }
        }).start();
    }
}
```

Fortsetzung nächste Seite!

### 3. Prozess-Scheduling

Thema dieser Aufgabe sind verschiedene Prozessorverwaltungs-Strategien (Scheduling-Strategien). Als Modell sei dazu ein einfaches Ein-Bediener-System gegeben. Das heißt wir betrachten Aufträge mit Bedienzeiten (Rechenzeiten) sowie Ankunftszeiten (der Zeitpunkt, ab dem der Auftrag im System vorhanden ist und berechnet werden kann) und einen Prozessor, auf dem jeweils immer nur einer der Aufträge bearbeitet werden kann. Wir nehmen vereinfachend an, dass ein Prozesswechsel keine Zeit kostet.

Die Ausführung eines solchen Systems kann man sehr gut anhand von Scheduling-Diagrammen visualisieren, bei dem auf der x-Achse die Zeiteinheiten und auf der y-Achse die verschiedenen Aufträge angetragen sind. Dabei ist jeder Auftrag durch eine Linie von seinem Ankunftszeitpunkt bis zum Zeitpunkt seiner abgeschlossenen Berechnung eingetragen, wobei die Linie gestrichelt ist solange er wartet und durchgezogen, solange ihm der Prozessor zugeteilt ist.

In Abb. 1 ist ein solches Diagramm zur Verdeutlichung für eine fiktive Scheduling-Strategie dargestellt.

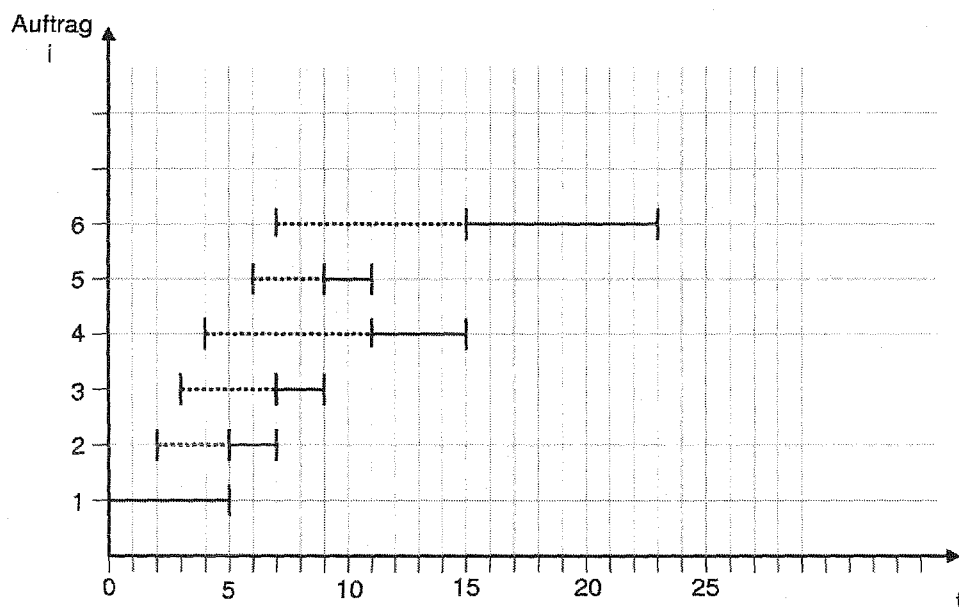


Abbildung 1: Scheduling Diagramm (fiktive Strategie)

Im Modell-Bediensystem existieren sechs Aufträge  $A_1, \dots, A_6$  mit den Bedienzeiten  $b = (4, 3, 2, 3, 2, 6)$  (bedeutet  $A_1$  benötigt 4 Zeiteinheiten,  $A_2$  3 usw.). Der Vektor der Ankunftszeiten sei durch  $a = (2, 3, 5, 6, 7, 8)$  (bedeutet  $A_1$  ist ab Zeitpunkt 2,  $A_2$  ab 3 usw. in der Warteschlange) gegeben. Vor dem Zeitpunkt  $t = 0$  sei das Bedienungssystem leer.

- a) aa) Zeichnen Sie entsprechende Diagramme für das angegebene Beispiel-Bediensystem für folgende Scheduling-Strategien:
- FCFS (First Come First Served): FCFS bedeutet, dass die Aufträge in der Reihenfolge ihrer Ankunft ausgeführt werden. Auftragsunterbrechung ist nicht möglich.
  - SRPT (Shortest Remaining Processing Time): SRPT bedeutet, dass jeweils der Auftrag mit kürzester Restbedienzeit ausgeführt wird. Auftragswechsel sind nur bei

Fortsetzung nächste Seite!

Ankunftszeitpunkten neuer Aufträge möglich.

Wir legen für unser Bediensystem fest: Haben zwei Prozesse gleiche Restbedienzeiten und rechnet einer davon gerade, wird zunächst dem gerade rechnenden Prozess der Vorrang gegeben. Haben ansonsten zwei Prozesse gleiche Restbedienzeiten wird dem Prozess Vorrang gegeben, der zuerst im System angekommen war.

bb) Warum ist SRPT schwierig umzusetzen? (1 Satz)

cc) Warum ist es sinnvoll, im Falle von realen (nicht-idealisierten) Systemen bei gleichen Restbedienzeiten dem gerade rechnenden Prozess den Vorrang zu geben? (1 Satz)

b) aa) Berechnen Sie die *mittlere Verweilzeit*  $\bar{V}$  und die *mittlere Wartezeit*  $\bar{W}$  für die beiden Strategien der vorangehenden Teilaufgabe. Nutzen Sie zur Berechnung folgende Formeln:

- $c_i$  bezeichnet die **Abgangszeit** (der Zeitpunkt seiner fertigen Berechnung) des Auftrags  $A_i$ ,  $a_i$  seine Ankunftszeit und  $b_i$  seine Bedienzeit;
- $v_i = c_i - a_i$  bezeichnet die **Verweilzeit** des Auftrags  $A_i$ ;
- $w_i = v_i - b_i = c_i - a_i - b_i$  bezeichnet die **Wartezeit** des Auftrags  $A_i$ ;
- Gibt es  $n \in N$  Aufträge, so gilt für die **mittlere Verweilzeit**:

$$\bar{V} = \frac{1}{n} \sum_{i=1}^n v_i$$

- Für die **mittlere Wartezeit** gilt:

$$\bar{W} = \frac{1}{n} \sum_{i=1}^n w_i$$

bb) Welche Ziele sollen beim Prozess-Scheduling in Batch-Systemen optimiert werden?

cc) Welches Ziel ist in interaktiven Systemen besonders wichtig?

c) aa) Wie funktionieren Zeitscheiben-basierte Scheduling Verfahren prinzipiell?

bb) Warum führt man hier zusätzlich oft Prioritäten ein?

cc) Nennen und erläutern Sie kurz einen Vorteil und einen Nachteil kleiner Zeit-Quanten!

#### 4. Memory Management

- a) Sei eine Menge von Seiten (pages)  $N = \{0, 1, 2, 3, 4, 5, 6\}$  und eine Menge von Seitenrahmen (frames)  $F = \{f_1, f_2, f_3\}$  gegeben. Nun wird in folgender Reihenfolge auf die Seiten zugegriffen:

$w = 4 \ 2 \ 4 \ 0 \ 5 \ 5 \ 6 \ 3 \ 1 \ 3 \ 2 \ 6 \ 4$

Vollziehen Sie die Seitenersetzungs-Strategien LRU (Least Recently Used) und FIFO (First In First Out) schrittweise nach, indem Sie die entsprechenden folgenden Tabellen auf Ihr Blatt übertragen und vervollständigen!

Strategie LRU:

Anforderung	$f_1$	$f_2$	$f_3$	Page-Fault j/n
4			4	j
2		2	4	j
4		2	4	n
0	0	2	4	j
5				
5				
6				
3				
1				
3				
2				
6				
4				

Strategie FIFO:

Anforderung	$f_1$	$f_2$	$f_3$	Page-Fault j/n
4			4	j
2		2	4	j
4		2	4	n
0	0	2	4	j
5				
5				
6				
3				
1				
3				
2				
6				
4				

Fortsetzung nächste Seite!

- b) Gegeben sei ein virtueller Adressraum von 24 Bit und eine Seiten- und Kachelgröße von je 4096 Byte. Der Speicher sei Byte-weise adressierbar. Weiterhin sei eine Seiten-Kacheltabelle gegeben, die u.a. folgende Einträge hat (Format: erste Spalte: Dezimal; zweite Spalte: Binär):

129	001101
130	101111
131	110100
132	000101
133	001100
134	101110
135	010110
136	110111
...	...

Ermitteln Sie zu der virtuellen Adresse

$v = (000010000011000011100011)$

die physikalische Adresse!

## AUFGABE 2

### Teilaufgabe 1

Die folgenden vier Jobs stehen zu den angegebenen Zeiten (in Minuten) zur Ausführung an:

Job	Ankunftszeit	Bearbeitungsdauer
A	0	15
B	1	8
C	6	10
D	10	1

- a) Stellen Sie die zeitlichen Abläufe in Gantt-Diagrammen (Balkendiagramme über der Zeitachse) dar und berechnen Sie dazu den Turnaround eines jeden Jobs, sowie den Average Turnaround bei den folgenden Scheduling-Strategien:
1. First-Come-First-Served (FCFS)
  2. Shortest-Job-First (SJF)
  3. Preemptive-Shortest-Job-First (Preemptive SJF=Shortest Remaining Time First)
  4. Round-Robin (RR) mit Zeitquantum  $q = 8$
- b) Nun soll eine „2-Level-Feedback-Strategie“ verfolgt werden derart, dass je eine Warteschlange für die Kategorien „Kurzläufer“ und „Langläufer“ existiert, wobei das Scheduling bei den Kurzläufern mit Round-Robin und dem Zeitquantum  $q = 1$  und bei den Langläufern nach FCFS abläuft.
- Jeder neu ankommende Job wird zunächst als Kurzläufer eingestuft. Erst wenn ein neuer Job eine Bearbeitungszeit von 2 Zeitquanten überschreitet, muss er in die Kategorie der Langläufer wechseln. Die Warteschlange der Kurzläufer hat Priorität über die Warteschlange der Langläufer, d.h. falls gerade ein Langläufer bearbeitet wird und es kommt ein neuer Job an, so wird der aktive Langläufer-Job solange angehalten, bis keine Kurzläufer mehr im System sind. Geben Sie jeweils die Wartezeit eines jeden Jobs an.

## Teilaufgabe 2

- a) Vor einem Supermarkt sind die verfügbaren 100 Einkaufswagen alle in einer Reihe **A** aufgestellt. Jeder Wagen ist durch eine kleine Kette mit dem nächsten Wagen gekoppelt. Der erste Wagen der Reihe wird mit einer Kette an einen Stahlträger gekoppelt. Jeder Kunde des Supermarkts benötigt für seinen Einkauf genau einen Einkaufswagen und er hält hierfür eine 1€-Münze parat, mit der die Kette zum derzeit letzten Einkaufswagen der Reihe gelöst werden kann. Sind alle 100 Einkaufswagen belegt, so wartet ein neuer Kunde darauf, dass ein anderer Kunde seinen Einkaufswagen wieder zurückbringt.
- b) Implementieren Sie das Vorgehen der Kunden und die Synchronisation bei der Nutzung der Einkaufswagen in (Java-ähnlichem) Pseudo-Code unter ausschließlicher Verwendung binärer Semaphore.
- c) Nach einem Ausbau des Supermarkts werden für die Kunden zusätzlich zur vorhandenen Reihe **A** für die 100 Einkaufswagen an einer anderen Stelle **B** des Parkplatzes weitere 50 Einkaufswagen desselben Typs als Reserve angeboten. Damit die Kunden nicht unnötig zu **A** oder **B** gehen müssen, hat die Leitung des Supermarkts große, von überall sichtbare, elektronische Anzeigetafeln anbringen lassen, auf denen jeweils die aktuelle Zahl verfügbarer Einkaufswagen bei **A** und **B** angeschrieben ist. Beschreiben Sie diese kundenfreundlichere Lösung mit „Counting Semaphores“ und kommentieren Sie mögliche Sonderfälle von „Race Conditions“.



## Teilaufgabe 3

- a) Wie kann man seitens eines Betriebssystems für die aktuell vorhandenen Prozesse und Betriebsmittel erkennen, ob zu einem bestimmten Zeitpunkt gerade ein Deadlock vorliegt? Erläutern Sie knapp das Vorgehen.
- b) Für den Fall, dass für eine aktuell vorhandene Gruppe von Prozessen und Betriebsmitteln zu einem bestimmten Zeitpunkt ein Deadlock erkannt wurde, wie sollte das Betriebssystem in solch einer Situation reagieren? Erläutern Sie knapp mögliche Strategien sowie deren jeweilige Vor- und Nachteile.
- c) Der exklusive „Verein zur Pflege philosophischer Diners“ hat über einen Aufnahmeantrag eines neuen Bewerbers zu entscheiden. Dieser führt in seinem Antrag aus, dass mit seiner Aufnahme die Gefahr von Deadlocks bei den traditionellen Diners beseitigt würde, da er Linkshänder sei, während alle bisherigen fünf Vereinsmitglieder Rechtshänder seien. Bei diesen Diners am runden Tisch sind für die fünf Philosophen je ein Teller und ein Essstäbchen vorhanden sowie für alle in der Tischmitte eine Schüssel mit Reis. Um essen zu können, benötigt ein Philosoph die beiden Essstäbchen, die links und rechts seines Tellers liegen. In der Tat hatte es in der Vereinsgeschichte schon Deadlocks gegeben, bei der die Philosophen, die bisher ausnahmslos Rechtshänder waren, grundsätzlich zuerst das Essstäbchen rechts des jeweiligen Tellers ergriffen und dann geduldig darauf warteten, dass das Essstäbchen links ebenfalls verfügbar wird.

Zeigen Sie mit Hilfe eines Widerspruchsbeweises über die notwendigen Bedingungen bei Deadlocks, dass der neue Bewerber als Linkshänder, der immer zuerst nach dem Essstäbchen zur Linken seines Tellers greifen wird, tatsächlich die Deadlock-Gefahr für den Verein beseitigen würde.

## Teilaufgabe 4

In einem Demand-Paging System mit drei Kacheln sei die folgende Seitenreferenz-Folge (von links nach rechts gelesen) zu bearbeiten:

**0, 1, 2, 3, 0, 2, 1, 3**

- a) Notieren Sie zeilenweise nach jeder Seitenreferenz die komplette Speicherbelegung des Systems und markieren Sie die Zeilen, die nach einem Seitenfehler (page fault) entstanden. Das erste Laden einer Seite zählt dabei ebenfalls als ein Seitenfehler.
- b) Ermitteln Sie die Anzahlen der Seitenfehler für die beiden Ersetzungsstrategien

Optimale Ersetzung („Belady’s Strategy“)

Second Chance („Clock Policy“)