

---

**Prüfungsteilnehmer****Prüfungstermin****Einzelprüfungsnummer**

---

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

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

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

**Herbst  
2013****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 zwei gemäß untenstehender  
Auswahlregel zu bearbeiten sind!**Anzahl der Druckseiten dieser Vorlage: **18**

---

Zu den zwei Themenschwerpunkten A (Datenbanksysteme) und B (Betriebssysteme) ist jeweils entweder die Teilaufgabe 1 oder 2 zu wählen.

Auf der Vorderseite des Kopfbogens sind im Feld „gewähltes Thema: Nr.“ die Nummern der beiden gewählten Teilaufgaben anzugeben (z. B. A2, B1)!

**Bitte wenden!**

## **Themenschwerpunkt A (Datenbanksysteme)**

### **Teilaufgabe 1:**

#### **1. Grundwissen**

##### **1.1 Bewerten Sie die folgenden Aussagen.**

Geben Sie für die folgenden Aussagen an, ob sie richtig oder falsch sind. Begründen Sie Ihre Antwort kurz.

- a) Ein perfekter Datenbankpuffer müsste die zukünftige Seitenreferenzfolge kennen.
- b) Ein Hash-Index eignet sich gut zur Beschleunigung von Bereichsanfragen.
- c) Fremdschlüsselwerte müssen eindeutig sein.
- d) Eine Relation kann mehrere Primärschlüsselattribute besitzen.
- e) Die Begriffe Datenbankverwaltungssystem und Datenbank sind äquivalent.

##### **1.2 Erläutern Sie kurz die folgenden Begriffe und Abkürzungen im Kontext von Datenbanksystemen.**

- a) Index
- b) Normalisierung
- c) Internes Schema
- d) Stored Procedure
- e) ORM

**Fortsetzung nächste Seite!**

## 2. Transaktionen

- a) Erläutern Sie kurz die vier wesentlichen Eigenschaften einer Transaktion.
- b) Können zwei parallel ablaufende Transaktionen, unter Garantie der Eigenschaften aus a), schreibend auf dasselbe Datenbankobjekt zugreifen? Begründen Sie Ihre Antwort.
- c) Kann bei zwei parallel ablaufenden Transaktionen, unter Garantie der Eigenschaften aus a), eine Transaktion schreibend auf ein Datenbankobjekt zugreifen, das von der anderen Transaktion gelesen wurde? Begründen Sie ihre Antwort.

## 3. Relationale Algebra

Relationenalgebraische Ausdrücke werden genutzt, um die Bearbeitung von SQL-Anfragen zu optimieren. Die Literatur kennt dabei eine Menge von Transformationsregeln. Untersuchen Sie die folgenden drei Regeln auf Richtigkeit bzw. Nichtrichtigkeit und begründen Sie dies kurz, wobei  $R(A_1, \dots, A_n)$ ,  $S(B_1, \dots, B_m)$  und  $T(C_1, \dots, C_k)$  Relationen sind.

- a) Der Verbund ist assoziativ, d. h. es gilt:  
 $\text{join}(\text{join}(R, S), T) = \text{join}(R, (\text{join}(S, T)))$
- b) Alle Mengenoperationen sind kommutativ.
- c) Welche Operation der relationalen Algebra wird durch SQL zunächst nur unvollständig umgesetzt (Verletzung der Relationeneigenschaften bei Anwendung)? Welche Eigenschaft fehlt dieser vereinfachten Umsetzung?
- d) Nennen Sie das SQL-Schlüsselwort, welches für die Operation aus c) ein Verhalten gemäß der relationalen Algebra erzwingt. Aus welchem Grund ist nicht Algebra konformes Verhalten in manchen Situationen besser?

**Fortsetzung nächste Seite!**

**4. Normalformen**

- a) Die Normalisierung von Relationenschemata dient der Vermeidung von Redundanzen und dadurch bedingter Anomalien. Geben Sie ein Beispiel für eine nicht-normalisierte Relation an und erläutern Sie zwei mögliche Anomalien an diesem Beispiel.
- b) Normalisierung beruht auf dem Erkennen und Eliminieren von funktionalen Abhängigkeiten. Erläutern Sie in diesem Zusammenhang kurz die folgenden Begriffe
1. Funktionale Abhängigkeit
  2. Voll-funktionale Abhängigkeit
  3. Transitive funktionale Abhängigkeit
  4. Superschlüssel
  5. Determinante
- c) In welcher Normalform ist das folgende Beispiel? Zeigen Sie, dass alle Bedingungen für diese Normalform erfüllt sind. Welche Bedingung der nächsthöheren Normalform ist verletzt?

PersNr	Name	Land	Landesvorwahl	Vorwahl	Rufnummer
1	Maier	Frankreich	33	556	56881
2	Maier	Deutschland	49	821	4563
3	Pascal	Frankreich	33	556	56881
4	Konrad	Deutschland	49	841	923476
5	Schmidt	USA	1	0212	928357
6	Berner	Frankreich	33	556	183258

- d) Erklären Sie die Begriffe Verlustlosigkeit und Abhängigkeitsbewahrung bei der Zerlegung von Relationen.

**Fortsetzung nächste Seite!**

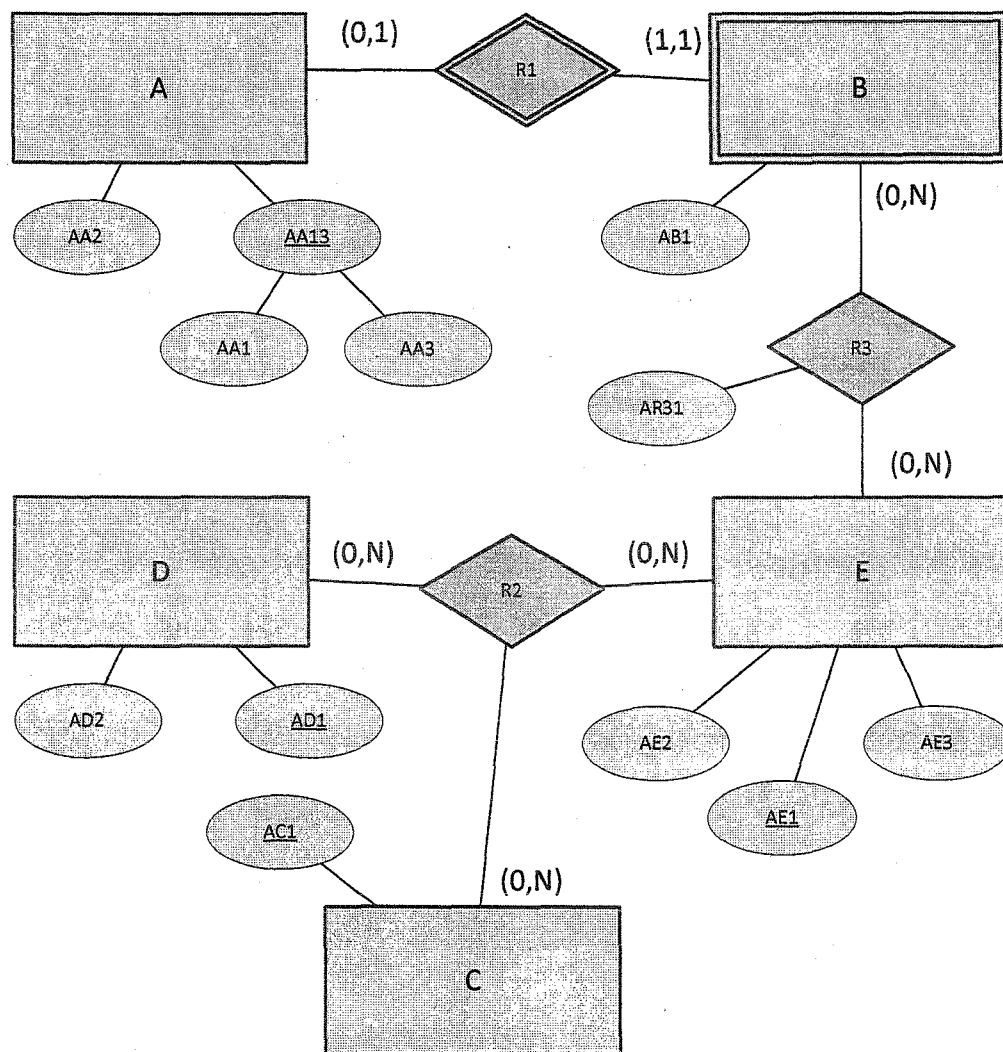
## 5. E/R-Modellierung und Relationenmodell

Entwerfen Sie zum untenstehenden E/R-Diagramm ein Relationenschema in dritter Normalform (3. NF) mit möglichst wenigen Relationen!

Verwenden Sie dabei folgende Notation:

Primärschlüssel werden durch Unterstreichen gekennzeichnet und Fremdschlüssel durch die Nennung der Relation auf die sie verweisen in eckigen Klammern hinter dem Fremdschlüsselattribut. Attribute zusammengesetzter Fremdschlüssel werden durch runde Klammern als zusammengehörig markiert.

Beispiel: Relation1 (Primärschlüssel, Attribut1, Attribut2, Fremdschlüssel[Relation2])



Fortsetzung nächste Seite!

## 6. Relationen und SQL

Gegeben ist folgendes Relationenschema zur Verwaltung von Studenten und Kursen:

STUDENT (Matrikelnummer, Name, Geburtsdatum)

KURS (Kursnummer, Kursname, Kursleiter)

Die Primärschlüssel der Relationen sind wie üblich durch Unterstreichung gekennzeichnet.

Verwenden Sie für die zu formulierenden SQL-Anfragen nur standardkonformes SQL.

- a) Erweitern Sie das obige Schema so, dass eine m:n-Beziehung zwischen STUDENT und KURS dargestellt werden kann: Ein Kurs kann aus mehreren Studenten bestehen, und ein Student kann an mehreren Kursen teilnehmen. Es ist darauf zu achten, dass alle Primärschlüssel nach wie vor genau ein Tupel identifizieren. Die Ausgangsrelationen STUDENT und KURS dürfen nicht verändert werden.
- b) Formulieren Sie ein SQL-Statement, welches die Namen der Kursleiter alphabetisch sortiert ausgibt (keine Mehrfachnennung).
- c) Formulieren Sie ausgehend von Ihrem in Teilaufgabe a) erweiterten Schema eine Anfrage in SQL, die die Namen und Leiter aller Kurse liefert, in denen Studenten mit Namen „Maier“ teilnehmen.
- d) Formulieren Sie eine Anfrage in SQL, die die Namen und Geburtsdaten aller Kommilitonen liefert, die der Student mit der Matrikelnummer „12345678“ in mindestens einem seiner Kurse außer sich selbst noch trifft? (keine Mehrfachnennungen)
- e) Formulieren Sie eine SQL-Abfrage, die die Namen der 10 ältesten Studenten ausgibt, absteigend sortiert nach Alter. Gehen Sie davon aus, dass es keine zwei Studenten genau gleichen Alters gibt.

**Teilaufgabe 2:****1. ER-Modellierung**

Entwerfen Sie ein ER-Modell für die folgende Miniwelt, die sich mit Konzerten, Veranstaltern, Bands usw. befasst. Geben Sie Kardinalitäten in (*min*, *max*)-Notation an. Sie brauchen Schlüsselattribute nicht zu kennzeichnen.

- Es gibt *Bands*, die jeweils einen Namen tragen. Zu jeder Band soll ein Gründungs- und ein Auflösungsdatum vermerkt werden.
- *Musiker* haben einen Namen. Musiker spielen *Instrumente* in Bands. Dabei können sie in verschiedenen Zeitintervallen (von-bis) verschiedene Instrumente in wechselnden Bands spielen, auch mehrere zur gleichen Zeit.
- *Instrumente* haben einen Namen.
- Bands spielen *Konzerte*. Jedes Konzert wird von genau einer Band gespielt, zusätzlich kann es noch bis zu zwei Vorgruppen geben. Konzerte finden zu einem bestimmten Datum an einem bestimmten Ort statt. Weitere Musiker, die nicht zur Band gehören, können bei Konzerten als Gast auftreten.
- *Veranstalter* haben einen Namen und veranstalten Konzerte. Jedes Konzert hat genau einen Veranstalter.
- *Besucher* gehen zu Konzerten. Ein Konzert hat mindestens 100 und höchstens 100.000 Besucher. Zu einem Konzertbesuch soll notiert werden, wie hoch der Eintrittspreis war. Dieser kann für jeden Besucher verschieden sein.

Ihr Modell sollte folgendes ausdrücken können. Sie brauchen diese Fakten nicht zu modellieren, sie dienen nur zur Kontrolle, ob Ihr Modell die o. g. Anforderungen erfüllt.

- Der Musiker Eddie van Halen spielte vom 1.1.1977 bis zum 6.6.1994 Gitarre und Keyboard in der Band "Van Halen"; vom 3.4.1986 bis zum 12.8.1988 spielte er Bass in der "Sammy Hagar Band".
- Am 3.6.2007 spielte die Band "Linkin Park" ein Konzert in Hamburg, das 7.000 Besucher hatte. Vorgruppe war "P.O.D.", als Gast trat Fred Durst auf. Das Konzert wurde veranstaltet von Undercover Entertainment.
- Der Besucher Karl Napf war bei diesem Konzert und hat dafür 53 Euro Eintritt gezahlt.
- Die Besucherin Liese Laut war auch beim o. g. Konzert, hat aber nur 48 Euro Eintritt gezahlt.

**Fortsetzung nächste Seite!**

## 2. Relationale Algebra

Entsprechend dem Modell aus der vorigen Aufgabe sei eine Miniwelt über Konzerte, Veranstalter usw. entsprechend den folgenden Relationen modelliert:

- Besucher(persnr, name)
- Konzerte(konzertnr, bandnr, datum, ort, veranstalternr)
- Veranstalter(veranstalternr, name)
- KonzertBesuche(persnr, concertnr, preis)
- Vorgruppen(bandnr, concertnr)
- Bands(bandnr, name, gründungsdatum, auflösungsdatum)
- Musiker(musikernr, name)
- Instrumente(instrumentnr, name)
- Spielt(musikernr, bandnr, instrumentnr, von, bis)
- Gast(musikernr, concertnr)

Geben Sie je einen Ausdruck in relationaler Algebra für folgende Anfragen an:

**2.1** In welchen Bands (Nummern sind gesucht) hat Musiker Nr. 12 gespielt?

**2.2** Wer spielte schon (Musikernummern sind gesucht) in der Band "The Hooters"?

**2.3** Welche Musiker (Nummern sind gesucht) waren schon einmal Gast bei einem Konzert der "Dave Matthews Band"?

**Fortsetzung nächste Seite!**



### 3. SQL

Für diese Aufgabe betrachten wir eine relationale Datenbank in Anlehnung an das Schema aus Aufgabe 2.

**3.1** Geben Sie je ein CREATE-TABLE-Statement zum Erzeugen der Relationen *Bands* und *Spielt* an. Achten Sie auf den Primärschlüssel und auf möglichst restriktive Fremdschlüssel und Integritätsbedingungen:

Ein Musiker soll ein bestimmtes Instrument in einer gegebenen Band nur einmal spielen können (nicht mehrere Zeitintervalle).

Wenn ein Instrument verschwindet, soll das entsprechende Attribut in *Spielt* auf NULL gesetzt werden. Ein Musiker soll nicht gelöscht werden können, solange er noch in einer Band spielt. Wenn Bands gelöscht werden, sollen die entsprechenden Tupel in *Spielt* auch verschwinden.

**3.2** Geben Sie SQL-Statements für folgende Anfragen an:

**3.2.1** In welchen Bands (Nummern sind gesucht) hat Musiker Nr. 28 gespielt?

**3.3** In welchen Bands (Nummer gesucht) hat der Musiker namens "Phil Collins" gespielt?

**3.4** Erzeugen Sie eine Sicht, die eine Übersicht über die Veranstalter und die Anzahlen ihrer Konzerte enthält.

Die Sicht soll also Tupel der Form (*Veranstalternr*, *Veranstaltername*, *Anzahl Konzerte*) enthalten.

**3.5** Geben Sie eine Anfrage an, die für jede Band den Namen der Band und die Gesamtzahl der jemals dort spielenden Musiker ausgibt. (Vorsicht, Musiker mit mehreren Instrumenten nicht mehrfach zählen.)

**3.6** Was leistet folgende SQL-Anfrage:

```
select v.name
from konzerte k, konzertbesuche b, veranstalter v
where k.konzertnr = b.konzertnr
and v.veranstalternr = k.veranstalternr
group by v.veranstalternr
having sum(b.preis) >= all (
    select sum(bb.preis)
    from konzerte kk, konzertbesuche bb
    where kk.konzertnr = bb.konzertnr
    group by kk.veranstalternr
)
```

**3.7** Ist die Unteranfrage in 3.6 korreliert oder nicht? (Begründung.)

**Fortsetzung nächste Seite!**

#### 4. Normalisierung

Gegeben seien die funktionalen Abhängigkeiten

$$\begin{aligned}
 F = \{ & ABC \rightarrow E, & (i) \\
 & B \rightarrow AC, & (ii) \\
 & E \rightarrow BCD, & (iii) \\
 & F \rightarrow ABE, & (iv) \\
 & D \rightarrow EF \} & (v)
 \end{aligned}$$

der Relation  $R(A, B, C, D, E, F)$ .

Berechnen Sie eine kanonische Überdeckung  $F_C$  von  $F$ . Geben Sie alle Zwischenschritte und die angewendeten Transformationen an.

#### 5. Normalisierung

Gegeben sei folgendes Schema, das die Bücher eines Verlages repräsentiert:

Bücher (ISBN, AutorNr, Autorname, Autoradresse, LektorNr, Lektorname, Titel, Seiten, Preis, Auflage, Jahr)

mit den funktionalen Abhängigkeiten

$$\begin{aligned}
 F = \{ & \text{AutorNr} \rightarrow \text{Autorname, Autoradresse}, & (i) \\
 & \text{LektorNr} \rightarrow \text{Lektorname}, & (ii) \\
 & \text{ISBN} \rightarrow \text{Titel, AutorNr, LektorNr}, & (iii) \\
 & \text{ISBN, Auflage} \rightarrow \text{Seiten, Preis, Jahr} \} & (iv)
 \end{aligned}$$

*Hinweis:* Sie können folgende Abkürzungen verwenden:

I	ISBN	T	Titel
A#	AutorNr	S	Seiten
An	Autorname	P	Preis
L#	LektorNr	J	Jahr
Ln	Lektorname	Au	Auflage
		Ad	Autoradresse

5.1 In welcher Normalform ist dieses Schema? (Begründung.)

5.2 Nennen Sie je ein Beispiel für eine Einfüge-, Update- und Löschanomalie in diesem Schema.

## Themenschwerpunkt B (Betriebssysteme)

### Teilaufgabe 1:

#### 1. Schedulingstrategien

- a) Definieren Sie den Begriff *Auslastung*

Definieren Sie den Begriff *Antwortzeit*

Inwieweit widersprechen sich die beiden Schedulingziele *maximale Auslastung* und *minimale Antwortzeit*?

- b) In einem System wird das Scheduling gemäß *Round-Robin* mit einer Zeitscheibengröße von  $n$  durchgeführt. Ein Prozesswechsel dauert  $k$  Zeiteinheiten.

Was passiert wenn  $n$  unendlich gross wird?

Was ist das kleinstmögliche  $n$ ? Wie verhält sich das System in diesem Falle und wann ist dies sinnvoll?

- c) Gegeben sind folgende Prozesse mit ihren Bereitzeitpunkten und ihrem Rechenzeitbedarf. Führen Sie für eine Einprozessormaschine das nicht-preemptive Scheduling gemäß *Shortest Processing Time First* (SPF) und das preemptive Scheduling gemäß *Round-Robin* (RR) mit Zeitscheibenlänge 1 durch. Der Overhead für den Prozesswechsel ist hier vernachlässigbar.

Prozess	Ankunftszeit	Benötigte Rechenzeit
A	0	3
B	1	6
C	3	3
D	5	5
E	9	1
F	15	2

Ein Task kann zum Zeitpunkt seiner Ankunft bereits ausgeführt werden (Beispiel siehe Prozess A: Ankunftszeit = 0, Prozessor-Zuteilung = 0).

**Fortsetzung nächste Seite!**

Übertragen Sie die folgenden drei Tabellen auf Ihr Arbeitsblatt:

Tragen Sie in der Zeile Ausführung jenen Prozess ein, der für die entsprechende Zeitscheibe dem Prozessor zugeteilt wird. Die restlichen Zeilen der Tabelle stellen die Ready-Queue dar. Tragen Sie hier die Prozesse ein, die in der Warteschlange auf Zuteilung warten (beginnend in der obersten Zeile mit dem Prozess, der als nächster den Prozessor zugeteilt bekommt, usw.).

Shortest Processing Time First:

	0				5				10				15			
<b>Ausf.</b>																
<b>Ready-Queue</b>																

Round Robin:

	0				5				10				15			
<b>Ausf.</b>																
<b>Ready-Queue</b>																

Geben Sie für jeden Prozess die Verweildauer im System an:

<b>Prozess</b>	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>	<b>F</b>
<b>SPF</b>						
<b>RR</b>						

Fortsetzung nächste Seite!

## 2. Synchronisation

- a) Nennen Sie die vier hinreichenden und notwendigen Bedingungen für Verklemmungen.
- b) Betrachten Sie die folgenden zwei Programme A und B, die konkurrierend auf den Puffer buf und auf die Logdatei lf zugreifen. Auf die Logdatei lf greifen auch noch andere Programme zu. Die beiden Semaphore SemBuf und SemLog sind jeweils mit 1 initialisiert!

Program A:	(1)	Program B:
while (true) {	(2)	while (true) {
...	(3)	P(SemBuf);
P(SemLog);	(4)	infoB=buf.removeNext();
write(lf, "A: Got "+infoA);	(5)	P(SemLog);
P(SemBuf);	(6)	write(lf, buf.length());
buf.add(infoA);	(7)	V(SemBuf);
write(lf, buf.length());	(8)	write(lf, "B: Got "+infoB);
V(SemLog);	(9)	V(SemLog);
V(SemBuf);	(10)	...
}		}

Können diese Programme verklemmungsfrei laufen? Begründen Sie die Verklemmungsfreiheit bzw. beschreiben und begründen Sie eine Änderung der Programme, so dass keine Verklemmungen auftreten können.

**Fortsetzung nächste Seite!**

### 3. Segmentierung

- a) Erläutern Sie warum in heutigen Rechnern virtuelle Adressierung anstelle von physikalischer Adressierung verwendet wird. Gehen Sie dabei auf die vier Probleme physikalischer Adressierung ein.
- b) Gegeben sei ein Rechner mit 5 Bit physikalischem Adressraum und 5 Bit segmentbasiertem, virtuellen Adressraum und 5 Bit breiten Worten. Der Inhalt des physikalischen Speichers ist unten angegeben.
- Es gibt 4 Segmente, d. h. die ersten (=höherwertigsten) 2 Bit einer Adresse bilden die Segmentnummer
  - Die Basisadresse der Segmenttabelle ist 00000.
  - Ein Eintrag in der Segmenttabelle benötigt jeweils zwei aufeinanderfolgende Speicherworte:
    - das erste Wort eines Eintrags gibt die Segmentbasisadresse an
    - vom zweiten Wort enthalten die ersten (=höherwertigsten) 3 Bit die Länge des Segments in Speicherworten, das 4. Bit sei hier nicht von Bedeutung und das 5. Bit sei das *present*- bzw. *mapped*-Bit.

Adresse	Inhalt	Adresse	Inhalt	Adresse	Inhalt	Adresse	Inhalt
00000	10101	01000	01110	10000	11111	11000	00110
00001	10011	01001	00011	10001	11010	11001	01010
00010	11011	01010	10010	10010	01011	11010	00010
00011	00100	01011	00000	10011	01101	11011	10001
00100	01000	01100	11000	10100	00101	11100	10100
00101	11101	01101	10000	10101	11100	11101	01001
00110	01111	01110	11110	10110	00111	11110	10110
00111	11001	01111	01100	10111	10101	11111	00001

1. Wieviele Segmente sind *present*?
2. Welche Daten stehen an den Adressen 01000, 00111 und 10001? Geben Sie jeweils auch die Adressumrechnungen mit an.
3. Geben Sie die virtuelle Adresse für die reale Adresse 10010 an.

Fortsetzung nächste Seite!

#### 4. Virtueller Speicher und Seitenverdrängung

- a) Gegeben sei ein Rechner mit 32 Bit virtuellem Adressraum und 1 GByte byteweise adressiertem, realem Speicher. Eine Seite sei 8 kBytes groß.

Hinweis:

• 1kByte = 1024 Byte =  $2^{10}$  Byte

• 1GByte = 1024 MByte =  $2^{30}$  Byte

Wie viel Bit einer Adresse werden für den Offset benötigt?

Wie viele virtuelle Seiten gibt es?

Wie viele Seitenrahmen gibt es?

- b) Erklären Sie, nach wie vielen Schritten der Second Chance Algorithmus bei einer Speichergöße von N Seiten spätestens eine zu verdrängende Seite gefunden hat.

**Teilaufgabe 2:****Aufgabe 1**

- a) Nennen Sie mindestens fünf wichtige Kategorien von Informationen, die in einem Prozesskontrollblock zu speichern sind.
- b) Welche wesentlichen Eigenschaften unterscheiden Prozesse und Threads?
- c) Schildern Sie knapp, welche Schritte bei der Initialisierung eines Prozesses üblicherweise durchlaufen werden.
- d) Wofür werden die beiden Modi „User-Modus“ und „Supervisor-Modus“ benötigt? Nennen Sie hierzu jeweils mindestens ein Beispiel für einen Wechsel von User-Modus zu Supervisor-Modus und von Supervisor-Modus zu User-Modus.

**Aufgabe 2**

Gegeben sei ein Prozess-System mit vier Prozessen:  $P_1, \dots, P_4$  und die folgenden drei Ressourcen:  $R_1$  (vier Einheiten) und  $R_2$  (zwei Einheiten) und  $R_3$  (drei Einheiten). Zu einem gegebenen Zeitpunkt besitzt

$P_1$  drei Einheiten von  $R_1$

$P_2$  eine Einheit von  $R_1$

$P_3$  eine Einheit von  $R_2$

$P_4$  zwei Einheiten von  $R_3$

- a) Zusätzlich fordert nun  
 $P_1$  je eine Einheit von  $R_1$  und  $R_3$  an,  
 $P_2$  fordert zwei Einheiten von  $R_2$ ,  
 $P_3$  und  $P_4$  je eine Einheit von  $R_3$ .

Zeichnen Sie den zugehörigen Betriebsmittelgraphen (Resource Allocation Graph) nach Holt.

- b) Überprüfen Sie mit dem Banker's Algorithmus, ob bei Zuteilung der Ressourcen, wie in Teil a) von den Prozessen angefordert, ein sicherer Zustand vorliegen würde. Die Summe der belegten und der in Teil a) angeforderten Ressourcen stellen jeweils bereits die Maximalforderungen der Prozesse dar.
- c) Konstruieren Sie für ein System von drei Prozessen und drei Ressourcentypen einen Systemzustand von Ressourcenbelegungen und -anforderungen, der einen geschlossenen Zyklus von Anforderungen und Belegungen enthält, der aber trotzdem kein Deadlock ist. Zeichnen Sie den zugehörigen Betriebsmittelgraphen.

**Fortsetzung nächste Seite!**



**Aufgabe 3**

Ein neu eröffneter Supermarkt hat zwei Selbstbedienungskassen eingerichtet, bei denen sich die Kunden per Chipkarte identifizieren, dann die Artikel unter den Kassenscanner legen und den Rechnungsbetrag direkt von ihrem Konto abbuchen lassen. Bei diesen beiden Kassen handelt es sich um eine „normale Automatik-Kasse“ A und eine „Express-Kasse“ E.

a) Anfangs ist die Nutzung der beiden Kassen folgendermaßen geregelt:

1. Jeder Kunde  $K(i)$ ,  $i > 0$ , kann sich an der Express-Kasse E anstellen, wenn die Anzahl der Artikel  $n(i)$  in seinem Warenkorb höchstens gleich drei ist.
2. Andernfalls muss er sich an der normalen Kasse A anstellen.

Implementieren Sie das Prozessverhalten des Kunden  $K(i)$  und der beiden Kassen A und E in Pseudo-Code. Die Synchronisation der Kunden  $K(i)$ , der Kasse A und der Kasse E soll ausschließlich über allgemeine Semaphore, also Counting Semaphore mit `wait()` und `signal()`, geschehen.

b) Nach einem mehrmonatigen Betrieb der Selbstbedienungskassen werden anstatt des bisherigen Verfahrens die folgenden beiden Regeln für die Nutzung der Kassen A und E in Kraft gesetzt:

1. Warten bereits drei Kunden an der Express-Kasse E, so muss sich jeder weitere ankommende Kunde  $K(i)$  bei der normalen Kasse A anstellen, unabhängig von der Anzahl seiner Artikel im Warenkorb.
2. Kunden  $K(i)$  dürfen sich nun auch mit einer beliebigen Anzahl von Artikeln bei der Kasse E anstellen. Allerdings muss ein  $K(i)$  mit mehr als drei Artikeln direkt vor der Kasse E prüfen, ob mindestens ein weiterer Kunde hinter ihm in der Warteschlange steht. Ist dies der Fall, so muss  $K(i)$  zur Kasse A wechseln. Sonst wird er bei E bedient. Hat  $K(i)$  höchstens drei Artikel, so wird er ohne Prüfen der Länge der Warteschlange bei E bedient.

Implementieren Sie hier ebenfalls das Verhalten der Kunden  $K(i)$  in Pseudo-Code. Die Prozesse für die Kassen A und E sollen wie bei a) realisiert werden. Verwenden Sie wiederum counting Semaphore und zusätzlich zur Kontrolle der Warteschlangenlänge bei der Kasse E eine globale Zählvariable `anzahl_bei_E`.

**Aufgabe 4**

In einem Arbeitsspeicher existieren die folgenden freien Blöcke in der angegebenen Reihenfolge mit den Größen

50KB, 48KB, 120KB, 66KB, 62KB

Geben Sie schrittweise die verbleibenden freien Speicherblöcke für die folgenden Anforderungen an (ebenfalls in dieser Reihenfolge) nach Blöcken der Größe:

48KB, 58KB, 64KB, 44KB

Verwenden Sie dabei die Vergabeverfahren:

- a) First-Fit
- b) Best-Fit
- c) Worst-Fit
- d) Rotating-First-Fit. Bei dieser Variante von First-Fit werden die freien Speicherblöcke in einer zyklisch verketteten Liste verwaltet. Nach jeder erfolgreichen Vergabe eines angeforderten Blocks startet die Suche für die nächste Anforderung jeweils beim Nachfolger des zuletzt betrachteten Speicherblocks.