
Prüfungsteilnehmer

Prüfungstermin

Einzelprüfungsnummer

Kennzahl: _____

Kennwort: _____

Arbeitsplatz-Nr.: _____

**Frühjahr
2008**

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 Auswahlregeln zu bearbeiten sind!

Anzahl der Druckseiten dieser Vorlage: **16**

Zu den zwei Themenschwerpunkten A (Datenbanksysteme) und B (Betriebssysteme) ist jeweils entweder die Aufgabe 1 oder 2 zu wählen. Auf der Vorderseite des Kopfbogens sind im Feld „Gewähltes Thema: Nr.“ die Nummern der beiden gewählten Aufgaben anzugeben (z. B. A1, B2)!

Bitte wenden!

Themenschwerpunkt A (Datenbanksysteme)

AUFGABE 1

I. Allgemeinwissen

1. Bewerten Sie die folgenden Aussagen!
Richtig oder falsch? Geben Sie für jede Aussage an, ob diese richtig oder falsch ist! Begründen Sie Ihre Aussage in jedem Fall!
 - a) Für den Datenaustausch zwischen Datenbank und Anwendungsprogrammen ist das Datenbankmanagementsystem verantwortlich.
 - b) Das Datenbankmanagementsystem erlaubt auch unkontrollierten Zugriff auf den Datenbestand.
 - c) Jeder Benutzer/ jedes Anwendungsprogramm eines Datenbanksystems muss für den Datenzugriff die eigentliche Organisation der Datenbank kennen.
 - d) Das Drei-Schichten-Modell nach ANSI/SPARC besteht aus der externen, der logischen und der physischen Schicht.
 - e) Der Zugriff auf sehr große Datenbestände ist per Datenbanksystem effizient möglich.
 - f) Integrität und Redundanzfreiheit wird mit Hilfe der mengenorientierten Datenmanipulation durch relationale Operatoren gewährleistet.
 - g) Das relationale Datenmodell diene als Grundlage zur Entwicklung des hierarchischen Modells und des Netzwerkmodells.
 - h) Die Ordnung der Tupel wird durch die Primärschlüsselwerte festgelegt.
 - i) Sichten dienen der Vermeidung von redundanter Speicherung.
 - j) Fremdschlüssel sind immer einmalig.
2. Beschreiben Sie kurz folgende Begriffe bzw. Abkürzungen im Kontext von Datenbanksystemen
 - a) Primärschlüssel
 - b) DML
 - c) Datenbankmanagementsystem
 - d) Satz

II. 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:

π = Projektion;	σ = Selektion;	\bowtie = Join;
\times = kartesisches Produkt;	\setminus = Mengendifferenz;	
\cap = Schnittmenge;	\cup = Vereinigungsmenge;	
ρ = Umbenennen;	\div = relationale Division	

1. Welche Operationen können sowohl auf Mengen als auch auf Relationen angewandt werden?
2. Definieren Sie die Operation „Division“! Auf welche Grundoperation kann die Division zurückgeführt werden und wie? Ein Beweis ist nicht erforderlich.
3. Eine Relation A mit den Attributen u, v, w, x habe 5 Tupel, eine Relation B mit den Attributen x, y, z habe 3 Tupel.
 - a) Wie viele Tupel und Attribute hat das kartesische Produkt aus A und B?
 - b) Wie viele Tupel und Attribute hat der Natural Join aus A und B?

Fortsetzung nächste Seite!

III. Datenbankentwurf

1. Wozu dient das Drei-Schichten-Modell nach ANSI/Sparc?
2. Geben Sie ein selbst gewähltes Beispielszenario, an dem Sie die Vorteile des Vorgehens nach ANSI/Sparc erläutern!

IV. Normalformenlehre

Die Normalformenlehre befasst sich mit der Fragestellung, wie „schlechte“ Relationenschemata erkannt und gegebenenfalls verbessert werden können.

1. Nennen Sie die drei möglichen grundlegenden Anomalien, die durch Normalisierung vermieden werden, beim Namen!
2. Welche Aussagen sind für funktionale Abhängigkeiten richtig (W, X, Y und Z sind beliebige Attributmengen)?
 - a) $X \rightarrow Y \Rightarrow Y \subseteq X$
 - b) $X \rightarrow Y$ und $Z \subseteq Y \Rightarrow X \rightarrow Z$
 - c) $X \rightarrow Y$ und $YW \rightarrow Z \Rightarrow XW \rightarrow Z$
 - d) $X \rightarrow Y$ und $Z \rightarrow Y \Rightarrow Z \rightarrow X$
3. Warum ist folgende Relation nicht in zweiter Normalform? Und warum werden Adressen dennoch häufig so gespeichert?
Adresse{Straße, Hausnummer, Postleitzahl, Wohnort}
4. Wie muss folgende Relation mit weiteren funktionalen Abhängigkeiten $B \rightarrow D$ und $D \rightarrow E$ zerlegt werden, damit sie in dritter Normalform vorliegt?
R{A, B, C, D, E}

V. Transaktionen und Transaktionsverarbeitung

1. Welche grundlegenden Eigenschaften müssen Datenbanktransaktionen heute garantieren? Benennen Sie jede Eigenschaft und beschreiben Sie jede mit max. einem Satz!
2. Können zwei parallel ablaufende Transaktionen (eine lesend, die andere schreibend), unter Garantie der oben definierten Eigenschaften, gleichzeitig auf dasselbe Datenbankobjekt zugreifen? Begründen Sie ihre Antwort in max. drei Sätzen!
3. Geben Sie an, welche Operationen auf Transaktionsebene Sie kennen, und beschreiben Sie jede mit max. zwei Sätzen!
4. Sie haben das so genannte „2 Phasen-Commit-Protokoll“ kennen gelernt. Skizzieren Sie das Szenario und erklären Sie die unterschiedlichen Abläufe, die entweder zu einem erfolgreichen oder nicht erfolgreichen Abschluss führen!

VI. ER-Modellierung und Relationenmodell

1. ER-Modellierung

Erstellen Sie das ER-Diagramm einer fiktiven Zimmerverwaltung eines Krankenhauses!

Wo möglich bzw. sinnvoll, sollen 3-fache Beziehungen und Generalisierung/Spezialisierung verwendet werden. Attribute von Entitäten und Beziehungen sind anzugeben; Schlüsselattribute werden durch Unterstreichen gekennzeichnet. Die Kardinalitäten von Beziehungen und - falls nötig - Rollennamen sollen ins Diagramm aufgenommen werden. Verwenden Sie zur Angabe der Kardinalität die (Min, Max)-Notation! Führen Sie Surrogatschlüssel nur ein, falls es nötig ist!

„Im Krankenhaus“

Zimmer werden durch eine eindeutige Zimmernummer identifiziert. Weiterhin sind noch die Anzahl der Betten und die Größe in qm des Zimmers angegeben.

Das Krankenhaus beschäftigt **Personal**, wobei es zwei besondere Personengruppen gibt: **Ärzte** und **Schwestern**. Jede Person hat eine Personalnummer, Vorname und Nachname. Schwestern haben zusätzlich eine vereinbarte Wochenarbeitszeit und Ärzte ihr Approbationsjahr und die -behörde gespeichert.

Das Personal ist in verschiedenen **Stationen** (also Krankenhausabteilungen) beschäftigt. Jede Station wird durch genau eine Schwester geleitet. Außerdem gibt es zu jeder Station genau einen Arzt, der zu ihr in Chefarzt-Beziehung steht. Es gibt im Krankenhaus Ärzte, die nicht Chefarzt einer Station sind.

Jeder **Dienstplan** hat ein Beginn- und Enddatum. Durch den Dienstplan wird festgelegt, welche Schwester welches Zimmer betreuen muss. Es gibt keine leeren Dienstpläne und auch keine untätigen Schwestern; manche der Zimmer werden allerdings nicht von Schwestern betreut, da sie nicht der Patientenunterbringung dienen.

2. Relationenmodell

Ausgehend von Ihrem ER-Diagramm-Entwurf aus Aufgabenteil 1 ist ein Relationenschema in dritter Normalform (3. NF) zu entwerfen. Primärschlüssel werden dabei durch Unterstreichen, Fremdschlüssel durch Nennung der referenzierten Relation in eckigen Klammern hinter dem Attributnamen kenntlich gemacht. Attribute, die nicht Primärschlüssel sind, aber dennoch „NOT NULL“ bzw. „UNIQUE“ sein müssen, sind zu kennzeichnen.

Folgende willkürliche Beispiele dienen als Hinweis auf die Notation:

```
Haus (Straße, OrtId[Ort])
Ort (OrtId, PLZ, Name)
Mitarbeiter (PersNr, Vorname, Nachname, SozialversNr)
SozialversNr: UNIQUE
```

VII. SQL

Beachten Sie:

Primärschlüssel werden durch Unterstreichen, Fremdschlüssel durch Nennung der referenzierten Relation in eckigen Klammern hinter dem Attributnamen kenntlich gemacht.

Beispiel:

```
Haus (Straße, OrtId[Ort])
Ort (OrtId, PLZ, Name)
```

Fortsetzung nächste Seite!

Interpretation:

Das Attribut `OrtId` der Relation `Haus` verweist als Fremdschlüssel auf das Attribut `OrtId` der Relation `Ort`.

„Kundenverwaltung“

Ein Unternehmen verwendet folgendes Datenbankschema für seine Verwaltung:

Kunde (KNr, Name, Vorname)

Bestellung (BNr, Datum)

Artikel (ANr, Bezeichnung, KatNr[Kategorie])

Best_K_A (BNr[Bestellung], ANr[Artikel], KNr[Kunde], Anzahl)

Kategorie (KatNr, Bezeichnung)

In `Best_K_A` sind Teilbestellungen gespeichert.

Formulieren Sie folgende Datenbankoperation in SQL:

Einfache Anfragen:

- Bestellungen sollen als Ganzes geliefert werden. Für Verhandlungen mit einem Lieferdienst soll ermittelt werden, wie viele Artikel eine Bestellung durchschnittlich enthält.
- Es soll eine Kundenliste erstellt werden und dabei ausgegeben werden, wann der Kunde zuletzt eine Bestellung aufgegeben hat. Darzustellen sind KNr, Name, Vorname und Datum der letzten Bestellung. Das Ergebnis soll nach Datum absteigend sortiert sein.

Datenänderungen:

- Ein neuer Kunde ist einzufügen. Er heißt „Hans Mueller“ und er soll die Kundennummer bekommen, die 1 größer ist als die bisher größte!
- Einige Artikel wurden noch keiner Kategorie zugeordnet (`Artikel.KatNr` ist NULL). Diese Artikel sind der Kategorie zuzuordnen, die die Bezeichnung 'Sonstiges' trägt.
- Um zukünftige Produktverwechslungen bei Bestellungen auszuschließen, sollen alle Artikel mit der Bezeichnung `Waage` aus der Kategorie `Kuechengerat` in `Kuechenwaage` umbenannt werden.
- Im Datenbestand ist ein Fehler aufgetreten. Natürlich ergeben Teilbestellungen keinen Sinn, wenn `Anzahl=0` ist. Daher sind alle Einträge aus `Best_K_A` zu löschen, deren `Anzahl=0` ist.

Komplexe Anfragen:

(Es dürfen kein `TOP(n)` oder `LIMIT` verwendet werden! Sichten sind ausdrücklich erlaubt!):

- Es sollen die fünf Kunden mit der größten Anzahl an bestellten Artikeln (Summe) ermittelt werden. Auszugeben sind Vorname, Name und Gesamtzahl der verkauften Artikel sortiert nach Name aufsteigend.
- Es sollen die fünf am meisten (nicht am häufigsten) bestellten Artikel (Summe der Anzahl) ermittelt werden. Auszugeben sind Artikelbezeichnung und Gesamtzahl sortiert nach Artikelbezeichnung aufsteigend.

Achtung: Unterschiedliche Artikel (unterschiedliche Artikelnummer) können die gleiche Bezeichnung tragen!

AUFGABE 2

I. Das Entity-Relationship Modell

Das Fremdenverkehrsamt will sich einen besseren Überblick über Zirkusse verschaffen. In einer Datenbank sollen dazu die Zirkusse, die angebotenen Vorstellungen, die einzelnen Darbietungen in einer Vorstellung sowie die zugehörigen Dompteure und Tiere verwaltet werden.

Ein Zirkus wird eindeutig durch seinen Namen gekennzeichnet und hat einen Besitzer. Vorstellungen haben eine `VorstellungsID` und ein Datum. Darbietungen haben neben der eindeutigen `ProgrammNr` eine Uhrzeit. Ein Dompteur hat eine eindeutige `AngestelltenNr` sowie einen Künstlernamen. Tiere sind eindeutig durch eine `TierNr` bestimmt und haben außerdem eine Bezeichnung der Tierart.

Vorstellungen werden von genau einem Zirkus angeboten. Ein Zirkus bietet mehrere Vorstellungen an und stellt mehrere Dompteure an. Ein Dompteur ist genau bei einem bestimmten Zirkus angestellt. Eine Darbietung findet in einer bestimmten Vorstellung statt. In einer Vorstellung finden mehrere Darbietungen statt. Des Weiteren trainiert ein Dompteur mehrere Tiere, ein Tier kann allerdings auch von mehreren Dompteuren trainiert werden. In einer Darbietung tritt genau ein Dompteur mit mindestens einem Tier auf.

- Erstellen Sie ein Entity-Relationship-Diagramm für obige Datenbank!
- Setzen Sie das in a) erstellte Entity-Relationship-Diagramm in ein Relationenschema um! Dabei sind Relationships mit einer möglichst geringen Anzahl von Relationen zu realisieren, wobei unnötige Redundanzen vermieden werden sollen. Ein Relationenschema ist in folgender Form anzugeben: `Relation(Attribut1, Attribut2, . . .)`. Schlüsselattribute sind dabei zu unterstreichen. Achten Sie bei der Wahl des Schlüssels auf Eindeutigkeit und Minimalität!

II. Normalisierungstheorie

Gegeben sei das Relationenschema $R(\underline{A}, B, C, D, E, F, G)$. Die Attribute seien atomar, d. h. R ist in 1. Normalform. Neben $\{A\}$ gibt es keine weiteren Schlüsselkandidaten.

Zusätzlich zu den durch die Schlüsseleigenschaft geltenden funktionalen Abhängigkeiten gelten die folgenden funktionalen Abhängigkeiten: $D \rightarrow E$, $D \rightarrow F$, $F \rightarrow G$.

- Ist R in 2. Normalform? Begründen Sie Ihre Antwort!
- Ist R in 3. Normalform? Begründen Sie Ihre Antwort!
- Zerlegen Sie R in mehrere Relationen, die alle in 3. Normalform sind (wenden Sie ein Verfahren Ihrer Wahl an)! Geben Sie für jede der neuen Relationen einen Schlüssel an!

Gegeben sei das Relationenschema $S(\underline{A}, \underline{B}, C, D, E, F)$. Die Attribute seien atomar, d. h. S ist in 1. Normalform. Neben $\{A, B\}$ gibt es keine weiteren Schlüsselkandidaten.

Zusätzlich zu den durch die Schlüsseleigenschaft geltenden funktionalen Abhängigkeiten gelten die folgenden funktionalen Abhängigkeiten: $A \rightarrow C$, $A \rightarrow D$; $B \rightarrow E$.

- Ist S in 2. Normalform? Begründen Sie Ihre Antwort!
- Ist S in 3. Normalform? Begründen Sie Ihre Antwort!
- Zerlegen Sie S in mehrere Relationen, die alle in 3. Normalform sind (wenden Sie ein Verfahren Ihrer Wahl an)! Geben Sie für jede der neuen Relationen einen Schlüssel an!

III. SQL

Gegeben sei das folgende relationale Datenbankschema (Schlüsselattribute sind jeweils unterstrichen):

Personal (PNR, Name, Geburtsdatum, Gehalt, ANR)

Abteilung (ANR, Bezeichnung, LeiterPNR, FNR)

Filiale (FNR, Ort)

In der Datenbank sind die Daten von Mitarbeitern, Abteilungen und Filialen eines Handelsunternehmens gespeichert. Jeder Mitarbeiter hat eine Personalnummer (PNR), einen Namen, ein Geburtsdatum, ein Gehalt und ist in einer bestimmten Abteilung beschäftigt. Jede Abteilung hat eine Abteilungsnummer (ANR), eine Bezeichnung, die Personalnummer des Abteilungsleiters (LeiterPNR) und gehört zu einer bestimmten Filiale. Jede Filiale hat eine Filialnummer (FNR) und ist in einem Ort beheimatet.

Formulieren Sie die folgenden Anfragen in SQL:

- a) Geben Sie für jede Abteilung die Nummer und die Bezeichnung der Abteilung zusammen mit der Personalnummer und dem Namen des Abteilungsleiters aus! Das Ergebnis soll aufsteigend nach der Abteilungsnummer sortiert werden.
- b) Geben Sie diejenigen Orte aus, in denen mehr als zwei Filialen beheimatet sind!
- c) Finden Sie die Personalnummern derjenigen Mitarbeiter, für die ein anderer Mitarbeiter mit dem gleichen Geburtsdatum existiert!

Themenschwerpunkt B (Betriebssysteme)

AUFGABE 1

I. Speicherverwaltung

In einem System mit Seitenadressierung (paged address space), Adresslänge = 16 Bit, Seitengröße = 4 KByte, Hauptspeichergröße = 64 KByte wird ein Programm durch einen Prozess ausgeführt (zur Erinnerung: $4096_{10} = 1000_{16}$).

Die erste Seite des virtuellen Adressraums wird nicht genutzt. Das Textsegment des Prozesses umfasst 2 Seiten, das Datensegment umfasst ebenfalls 2 Seiten, direkt im Anschluss daran. Das Stack-Segment umfasst eine Seite ganz am Ende des virtuellen Adressraums.

Die Seiten des Textsegments liegen im Hauptspeicher in den Seitenrahmen (= Kacheln) auf Adresse 0x4000 und 0x7000, die erste Seite des Datensegments liegt in Seitenrahmen 0x6000, die zweite Seite ist ausgelagert und liegt auf der Platte im Block 0x37000. Die Seite des Stacks liegt in Seitenrahmen 0x9000. Das (sehr kleine) Betriebssystem belegt die ersten 4 Seitenrahmen.

Die Seitenrahmen von 0xa000 bis 0xf000 sind durch einen anderen Prozess belegt.

1. Skizzieren Sie den Aufbau des virtuellen Adressraums des Prozesses sowie den Aufbau und die Belegung des Hauptspeichers und die Abbildungen dazwischen!
2. Die Seiten des Textsegments seien read-only in den Adressraum abgebildet, alle anderen Seiten zum Lesen und Schreiben. Die Ausführung von Maschinenbefehlen aus dem Daten- oder Stack-Segment ist nicht zulässig.
Skizzieren Sie die Datenstrukturen, die die MMU für die Umsetzung des beschriebenen virtuellen Adressraums benötigt und wie dabei die logische Adresse 0x3240 umgesetzt wird!
3. In dem Programm stehen folgende Anweisungen:

```
void fl() {  
    static int *p = (int *)0x4018;  
    int i;  
    ...  
    *p = 41;  
    i = 5;  
    ...  
}
```

Beschreiben Sie, was bei der Ausführung des Prozesses passiert, wenn er die beiden obigen Zuweisungen ausführt:

- a) Welche Aktivitäten laufen der Reihe nach in der Anwendung und im Betriebssystem ab? (Nummerieren Sie diese Schritte!)
- b) Welche Wechsel des Prozesszustands finden hierbei bei welchen Schritten statt?
- c) Tragen Sie in Ihre Skizzen zu den Teilaufgaben 3a) und 3b) die Änderungen ein, die im Laufe der Ausführung der Anweisungen erfolgen, und markieren Sie die entsprechenden Stellen mit c)! Tragen Sie dabei auch die Orte ein (ungefähr), an denen die Werte 41 und 5 im Hauptspeicher hingeschrieben werden!

Fortsetzung nächste Seite!

II. Verklemmungen

1. Welche Bedingungen müssen gegeben sein, damit eine Verklemmung auftreten kann?
2. Welche drei grundsätzlichen Verfahren gibt es, um mit der Verklemmungsproblematik umzugehen?
 - a) Beschreiben Sie jedes Verfahren! Wie ist jeweils die grundsätzliche Vorgehensweise?
 - b) Geben Sie ein Beispiel für einen Algorithmus an, der bei einem dieser Verfahren zum Einsatz kommt und beschreiben Sie den Algorithmus!

Für die folgenden Teilaufgaben III., IV. und V. wird das Datensicherungs- und Archivsystem eines Unternehmens betrachtet. Hierzu besitzt das unternehmenseigene Rechenzentrum einen Bandroboter mit fünf Bandlaufwerken und einem Regallager für die Bänder. Das System wird zur täglichen Datensicherung sowie zur gezielten Archivierung von Datenbeständen genutzt.

III. Dateisystem

1. Wie würden Sie den Dateibestand eines großen Softwareentwicklungsprojekts auf der Festplatte organisieren, damit die Programmquellen (mehrere Anwendungsprogramme sowie Funktionsbibliotheken), die Dokumentation sowie mehrere, aktuell an Kunden ausgelieferte Binärversionen der Software sowohl als Ganzes, als auch in den beschriebenen Teilen einfach archiviert werden können? Erläutern Sie Ihre Entscheidung am besten anhand einer Skizze!
2. Welche Informationen über Dateien bzw. über Kataloge würden Sie archivieren? Wozu werden welche Informationen im Archiv bzw. bei oder nach einem Restaurieren des Archivs benötigt?
3. Skizzieren Sie in einer Programmiersprachen-ähnlichen Form den Aufbau eines einfachen Archivierungsprogramms, das eine Archivierung entsprechend 1. und 2. realisiert! Gehen Sie dabei vereinfachend davon aus, dass der gesamte Bandroboter über die Datei "/dev/roboter" angesprochen werden kann und die Auswahl des Bandes und des Laufwerks (in dieser Teilaufgabe) automatisch erfolgt!

IV. Prozesse und Threads

Da von modernen Plattensystemen schneller gelesen wird, als auf ein Bandgerät geschrieben werden kann, sollen die Daten auf mehrere Bänder parallel geschrieben werden. Hierzu soll das Archivierungsprogramm folgendermassen organisiert werden:

- Ein Prozess P_0 leistet die oben in Teilaufgabe 3 beschriebene Funktionalität und schreibt die Daten in einen großen Speicherbereich (10 MB), der in Blöcken zu je 1 MB organisiert ist und als Ringpuffer betrieben wird.
 - Für jedes Bandgerät i ($i = 1 \dots 5$) gibt es einen Prozess P_i , der jeweils den nächsten verfügbaren Block aus dem Ringpuffer entnimmt und auf das Bandgerät schreibt.
1. Wie kann der Ringpuffer als gemeinsamer Speicher in einem Betriebssystem mit Seitenadressierung realisiert werden? In welchen Datenstrukturen müssen hierzu welche Einträge erfolgen?
 2. Skizzieren Sie die logischen Adressräume der Prozesse P_0 und P_1 und den Zusammenhang zu dem physikalischen Hauptspeicher!

3. Die beschriebene Realisierung mit Hilfe von Prozessen ist teuer. Wodurch entstehen diese Kosten (Speicher, Laufzeit)?
4. Eine Realisierung mit Hilfe von Threads wäre günstiger. Welche Arten von Threads kennen Sie, wodurch unterscheiden sie sich, was ist der Unterschied zu Prozessen?
5. Welche Thread-Art ist für die beschriebene Aufgabe geeignet und sollte statt einer Realisierung mit Prozessen genutzt werden? Welche Kosten spart man dadurch ein, welche Abläufe werden effizienter?
6. Warum ist die andere Thread-Art für die Aufgabe nicht geeignet?

V. Koordinierung

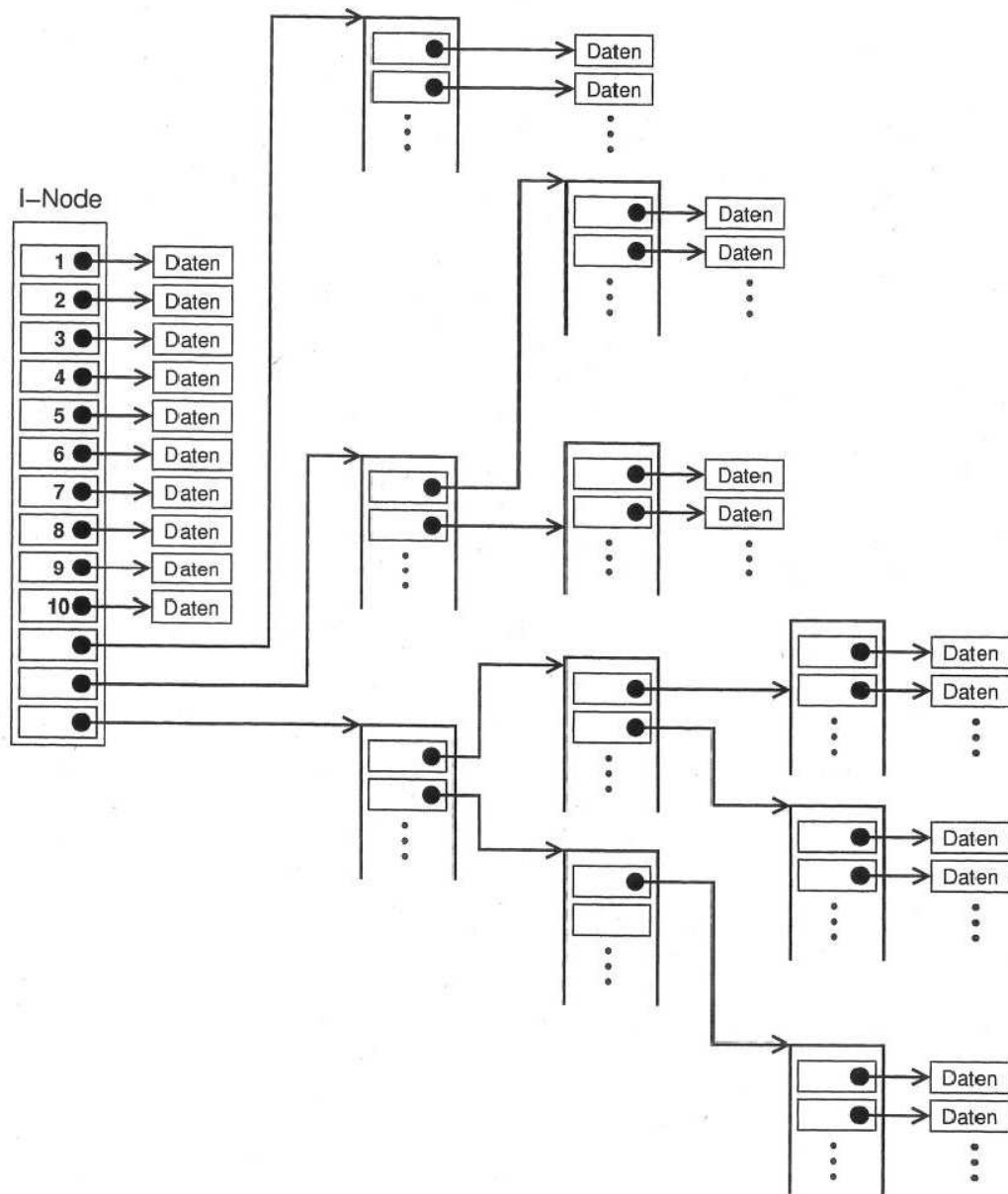
Unter der Annahme, dass mehrere Archivierungen bzw. Restaurierungen gleichzeitig anfallen können und dabei jeweils parallel zwei bis fünf Laufwerke genutzt werden, gibt es eine ganze Reihe von Koordinierungsproblemen:

- Zugriff auf einzelne Bänder
 - Zugriff auf Bandlaufwerke
 - Zugriff auf Ringpuffer
1. Welche Art von Koordinierungsproblem liegt in den obigen drei Situationen jeweils vor?
 2. Die Archivierung von vier unterschiedlichen Softwareprojekten wird fast gleichzeitig gestartet. Entsprechend der Aufgabe „IV. Prozesse und Threads“ soll jeweils auf drei Laufwerken parallel geschrieben werden. Welche Problematik hinsichtlich Verklemmungen existiert hier?
 3. Nennen Sie zwei unterschiedliche Maßnahmen zur Verklemmungsvorbeugung im obigen (in Teilaufgabe 2 beschriebenen) Szenario! Welche der für Verklemmungen notwendigen Bedingungen werden dadurch jeweils entkräftet?
 4. Skizzieren Sie in einer Programmiersprachen-ähnlichen Form zwei Funktionen put und get, die den Zugriff auf den Ringpuffer realisieren!

AUFGABE 2

I. Dateisysteme

Unix-Filesysteme implementieren das Konzept der I-Nodes. Dabei findet eine strikte Trennung des eigentlichen Dateiinhalts von den Verwaltungsinformationen (z. B. Dateigröße, Dateityp, Eigentümer, Gruppe, Zugriffsrechte) statt. Die Anzahl der I-Nodes sowie die Größe der Blöcke werden bei der Formatierung des Dateisystems festgelegt.



Gehen Sie in dieser Aufgabe von folgenden Annahmen aus:

- Jeder I-Node enthält 10 Felder für direkte Adressen auf Datenblöcke der Datei, sowie jeweils einen Verweis auf den ersten, zweiten und dritten Indirektionsblock - insgesamt also 13 Adressfelder (siehe Grafik).
- Der erste Indirektionsblock (Indirektionsblock erster Stufe) enthält Adressen von Datenblöcken. Der zweite Indirektionsblock (Indirektionsblock zweiter Stufe) enthält Adressen von weiteren Indirektionsblöcken erster Stufe, die ihrerseits Adressen von Datenblöcken enthalten. Analog für den dritten Indirektionsblock mit wieder einer zusätzlichen Stufe.
- Alle Adressen sind 4 Bytes lang.

Fortsetzung nächste Seite!

- Die Blockgröße beträgt einheitlich 4 KBytes (gilt sowohl für Daten- als auch für Indirektionsblöcke)

Beantworten Sie die folgenden Fragen!

- a) Wie groß ist der Inhalt einer Datei **mindestens** (genaue Angabe in Bytes), wenn ihr I-Node auf genau zwei Datenblöcke direkt verweist?
- b) Wie groß ist der Inhalt einer Datei **höchstens** (genaue Angabe in Bytes), wenn ihr I-Node auf genau zwei Datenblöcke direkt verweist?
- c) Es soll auf die Datei `/baum/zweig/ast` zugegriffen werden. Wie viele I-Nodes und Datenblöcke müssen dazu mindestens gelesen werden?
- d) Warum ist es sinnvoll, den Dateityp im I-Node festzuhalten?
- e) Mit Hilfe des Befehls `ln` lassen sich in Unix Verknüpfungen (Links) auf eine Datei erstellen. Dabei wird zwischen Hardlinks (`ln quelleDatei zielDatei`) und symbolischen Links (`ln -s quelleDatei zielDatei`) unterschieden. Wodurch unterscheiden sich diese beiden Varianten?

Auf einem Rechner wird das Dateisystem Ext2 (ohne Journaling) mit einer Blockgröße von 4 KBytes verwendet. Folgende Situation ist gegeben:

- Der I-Node mit der Nummer 332 adressiert die Datenblöcke mit den Adressen `0xA0A8AFA0`, `0xA0A8AFA4` und `0xA0A8AFA8` direkt. Die verbleibenden sieben Felder für direkte Adressen, sowie die drei Felder für die Adressierung von Indirektionsblöcken enthalten keine Adressen (NULL). Der Referenzzähler (Linkzähler) steht auf 1.
 - Der Datenblock `0x8124AAA0` beschreibt das Verzeichnis `studies`, welches unter anderem die Datei `notes.txt` enthält. Der entsprechende Verzeichniseintrag verweist auf den I-Node 332.
 - Der Datenblock `0x011100B0` beschreibt das Verzeichnis `info`, welches unter anderem die Datei `notes_copy.txt` enthält. Der entsprechende Verzeichniseintrag verweist ebenfalls auf den I-Node 332.
- f) Worin besteht in diesem Beispiel die Inkonsistenz?
 - g) Können die Dateien `notes.txt` und `notes_copy.txt` in den jeweiligen Verzeichnissen vom Nutzer gelesen werden, auch wenn die Inkonsistenz nicht behoben wird? Geben Sie an, ob keine, eine oder beide Dateien gelesen werden können und begründen Sie Ihre Antwort!
 - h) Was passiert mit dem I-Node 332, wenn der Nutzer nur die Verknüpfung `notes_copy.txt` aus dem Verzeichnis `info` entfernt (`rm notes_copy.txt`)? (Mit Begründung.)

II. Zustands-Prozessmodelle

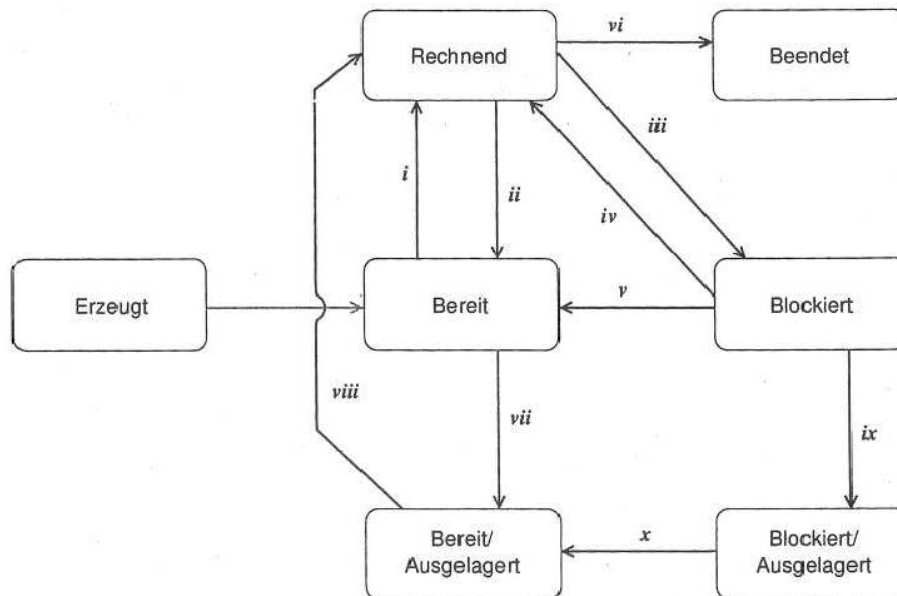
Diese Aufgabe beschäftigt sich mit dem gängigen 7-Zustands-Modell zur Beschreibung möglicher Ausführungszustände eines Prozesses sowie deren Übergänge. Die fünf Grundzustände sind Erzeugt, Bereit, Rechnend, Blockiert und Beendet, wobei in den beiden Zuständen Bereit und Blockiert jeweils unterschieden wird, ob der jeweilige Prozess sich aktuell im Hauptspeicher befindet (resident) oder ausgelagert wurde (swapped/suspended).

- a) Geben Sie für jede der folgenden Transitionen an, ob es sich um einen **gültigen Übergang** handelt! Falls ja, geben Sie ein Beispiel an, welches den jeweiligen Zustandswechsel auslösen könnte! Falls nein, begründen Sie, warum dieser Zustandsübergang nicht sinnvoll ist!

Fortsetzung nächste Seite!

Beispiel:

Erzeugt → Bereit: Gültig; Beispiel: Das Prozess-Image liegt vollständig im Speicher, und der Prozess ist nun rechenbereit. Er wartet darauf, vom Scheduler zur Ausführung ausgewählt zu werden.



- | | |
|---------------------------|-----------------------------------------------|
| i) Bereit → Rechnend | vi) Rechnend → Beendet |
| ii) Rechnend → Bereit | vii) Bereit → Bereit-Ausgelagert |
| iii) Rechnend → Blockiert | viii) Bereit-Ausgelagert → Bereit |
| iv) Blockiert → Rechnend | ix) Blockiert → Blockiert-Ausgelagert |
| v) Blockiert → Bereit | x) Blockiert-Ausgelagert → Bereit-Ausgelagert |

- b) Welche(r) der Zustände Bereit, Rechnend und Blockiert könnte(n) eingespart werden, wenn das Betriebssystem in reinem Batch-Betrieb (d. h. kein Multiprogramming) arbeitet? (Mit Begründung.)

III. Round-Robin-Scheduling und Prozesswechsel

Beim Scheduling nach der Round-Robin-Strategie wird eine FIFO-Warteschlange von Prozessen verwaltet, wobei jeweils dem in der Schlange ersten Prozess für eine feste Zeitdauer Q (die Länge der sog. Zeitscheibe) die CPU zugeteilt wird. Ein Prozess, dessen Zeitscheibe abgelaufen ist, wird am Ende der Warteschlange wieder eingereiht. Die mittlere Bedienzeit T eines Prozesses bezeichne die Zeitspanne, für die ein Prozess im Mittel die CPU benötigt, bevor er das Warteschlangensystem aufgrund einer angestoßenen E/A-Operation bzw. bei Prozessende freiwillig verlässt. Es handelt sich bei T also um die reine Rechenzeit des Prozesses. Die mittlere Verweilzeit V hingegen bezeichne die gesamte Zeitspanne, die ein Prozess im Mittel im System verweilt (ggf. wechselnd zwischen den Zuständen Bereit und Rechnend). Die mittlere Wartezeit W sei die durchschnittliche Summe der Zeiträume, in denen ein Prozess zwar im System verweilt, aber nicht rechnet. Ein Prozesswechsel benötige eine Zeitdauer S , die als Overhead verloren geht, aber unvermeidbar ist.

Zusammenfassend ergeben sich die folgenden Zusammenhänge:

- Größe der Zeitscheibe: Q
- Mittlere Bedienzeit (reine Rechenzeit) eines Prozesses: T
- Mittlere Verweilzeit eines Prozesses: V , $V \geq T$
- Mittlere Wartezeit eines Prozesses: W , $W = V - T$
- Zeit für Prozesswechsel: S

Fortsetzung nächste Seite!

- a) Beschreiben Sie kurz, was beim Prozesswechsel passiert und welche Aufgabe das Betriebssystem dabei erfüllt! Erklären Sie in diesem Zusammenhang in kurzen Stichworten den Begriff des Prozesskontextes!
- b) Die Zeitdauer S wird also wesentlich von der Komplexität des Prozesswechsels beeinflusst. Wovon hängt diese Komplexität maßgeblich ab?
- c) Gegeben sind die Prozesse P_1 bis P_4 mit den in der folgenden Tabelle angegebenen Ankunfts- und Bedienzeiten. Es gelte: $Q = 2$ und vereinfachend $S = 0$ (also vernachlässigbar klein). Eine Zeitscheibe muss nicht voll ausgenutzt werden. Terminiert ein Prozess vor Ablauf von Q , so wird diese eine Zeitscheibe entsprechend verkürzt und der nächste Prozess kann sofort aktiviert werden.

Prozess	Ankunftszeitpunkt	Bedienzeit
P_1	0	4
P_2	1	5
P_3	3	2
P_4	6	4

- i) Ermitteln Sie für jeden Prozess die **individuelle Verweil- und Wartezeit**! Erstellen Sie eine Tabelle nach folgendem Muster!

Tipp: Machen Sie sich dazu die Ausführungsreihenfolge der Prozesse durch ein Diagramm (z. B. Gantt-Chart) klar, und zeichnen Sie auch die Warteschlange (separates Blatt)! Für die Bewertung spielen die Zeichnungen allerdings keine Rolle.

Prozess	Verweilzeit	Wartezeit
P_1		
P_2		
P_3		
P_4		

- ii) Berechnen Sie die **mittlere Verweil- und Wartezeit** für dieses System!
- d) Unter der Annahme, dass in einem System stets rechenbereite Prozesse zur Verfügung stehen, lässt sich die CPU-Auslastung ρ wie folgt definieren: $\rho = \frac{T}{T+n \cdot S}$, wobei n die Anzahl der pro Prozess durchschnittlich anfallenden Prozesswechsel ist. Dabei soll gelten: Ein Prozesswechsel gehört zu dem Prozess, der zuletzt (also unmittelbar **vor** dem Prozesswechsel) gerechnet hat. Terminiert ein Prozess, so gehört der anschließende Wechsel zu einem rechenbereiten Prozess also auch noch zu dem gerade terminierten Prozess.
- i) Wie wirkt sich ein hoher Wert für S auf die CPU-Auslastung aus?
- ii) Geben Sie für n eine Formel in Abhängigkeit von T und Q an!
- iii) Welche CPU-Auslastung ergibt sich, wenn gilt: $Q > T$? Gegen welche Ihnen bekannte Scheduling-Strategie konvergiert Round Robin in diesem Fall?
- iv) Welche CPU-Auslastung ergibt sich, wenn gilt: $Q = S$?

IV. Seitenersetzung

In dieser Aufgabe sollen verschiedene Seitenersetzungsstrategien (Paging-Strategien) am konkreten Beispiel verglichen werden. Dabei sei die Menge der Seiten gegeben durch $N = \{0, 1, 2, 3, 4\}$. Die Menge der Seitenrahmen, die für die Speicherung der Seiten im Arbeitsspeicher zur Verfügung stehen, sei gegeben durch $F = \{f_0, f_1, f_2\}$. Auf die fünf Seiten der Menge N werde in folgender Reihenfolge zugegriffen:

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

Fortsetzung nächste Seite!

Ein Seitenfehler liegt immer dann vor, wenn sich eine referenzierte Seite nicht im Arbeitsspeicher befindet. Dieser ist zu Beginn leer.

- a) Ermitteln Sie die Anzahl der Seitenfehler für die Paging-Strategie LRU (Least Recently Used), indem Sie den Zustand des Speichers nach jedem Zugriff in einer Tabelle nach folgendem Muster dokumentieren! Informationen, die aufgrund der Paging-Strategie zusätzlich benötigt werden, tragen Sie ebenfalls ein!

Referenzierte Seiten	f_0	f_1	f_2	Summe Seitenfehler
0				
4				
3				
4				
3				
1				
3				
4				
0				
3				
2				
1				
0				

- b) Ermitteln Sie nun die Anzahl der Seitenfehler für die Paging-Strategie LFU (Least Frequently Used), indem Sie den Zustand des Speichers nach jedem Zugriff in einer Tabelle nach folgendem Muster dokumentieren! Informationen, die aufgrund der Paging-Strategie zusätzlich benötigt werden, tragen Sie ebenfalls wieder ein!

Referenzierte Seiten	f_0	f_1	f_2	Summe Seitenfehler
0				
4				
3				
4				
3				
1				
3				
4				
0				
3				
2				
1				
0				

- c) In der Praxis ergibt sich ein signifikantes Problem beim Einsatz der zweiten Strategie (Least Frequently Used). Welches? Wie könnte man LFU modifizieren, um dieses Problem zu minimieren? Tipp: Überlegen Sie sich, was passiert, wenn auf eine Seite zunächst über einen langen Zeitraum häufig zugegriffen wird, später dann die Seite nicht mehr benötigt wird (da z. B. der zugehörige Prozess terminiert wurde)!

Fortsetzung nächste Seite!

V. Verklemmungen (Deadlocks)

Gegeben sind die drei Prozesse P_1 , P_2 und P_3 sowie der vier Betriebsmittel R_1 bis R_4 . Zum betrachteten Zeitpunkt t gilt:

- P_1 nutzt R_3 und fordert R_1 an.
 - P_2 nutzt R_4 und fordert R_3 an.
 - P_3 nutzt R_1 und fordert R_2 und R_4 an.
- a) Zeichnen Sie den **Resource-Allocation-Graph (Betriebsmittelgraph)** für den Zeitpunkt t unter Berücksichtigung aller vorhandenen Betriebsmittel!
- b) Welche Betriebsmittelanforderung(en) ist (sind) sofort erfüllbar?
- c) Entscheiden und begründen Sie anhand des Resource-Allocation-Graphen, ob nach Erfüllung der erfüllbaren Anforderung(en) ein Deadlock vorliegt!