
Prüfungsteilnehmer**Prüfungstermin****Einzelprüfungsnummer**

Kennzahl: _____**Frühjahr****Kennwort:** _____**2001****66113****Arbeitsplatz-Nr.:** _____

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen
- Prüfungsaufgaben -

Fach: **Informatik (vertieft studiert)****Einzelprüfung:** **Rechnerarchitektur, Datenb., Betriebssys.****Anzahl der gestellten Themen (Aufgaben):** 2**Anzahl der Druckseiten dieser Vorlage:** 6

Bitte wenden!

Thema Nr. 1**Sämtliche Teilaufgaben sind zu bearbeiten!*****Teilgebiet 1: Rechnerarchitektur und Rechnernetze*****Aufgabe 1.1: Kommunikation in Rechnernetzen**

Ein lokales, nach außen abgeschottetes Netzwerk soll mit einem Wahlserver versehen werden. Dazu wird auf einem bestimmten Rechner ein Dienst installiert, der von allen anderen Rechnern mittels Fernaufruf (remote procedure call, RPC) genutzt wird. Der Dienst soll zwei Prozeduren zur Verfügung stellen:

- **vote:** Die Prozedur dient zur Stimmenabgabe und enthalte als Parameter den Namen des Kandidaten (als Zeichenkette) und eine Wahlnummer (als ganze Zahl), um mehrfache Stimmenabgabe verhindern zu können.
- **result:** Die Prozedur enthalte als Parameter den Namen eines Kandidaten und als Ergebnisparameter die Zahl der für ihn abgegebenen Stimmen.

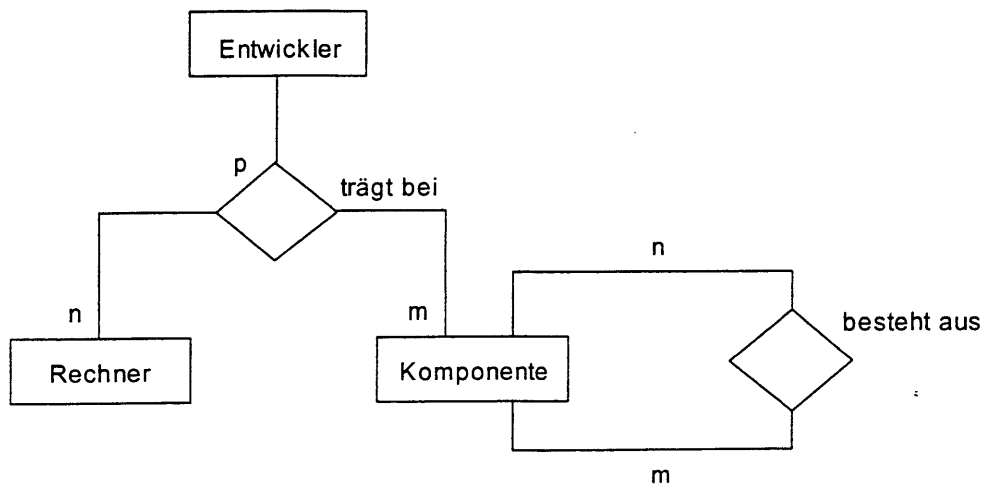
Der Server führe zwei Listen, nämlich ein Wählerverzeichnis zur Verhinderung mehrfacher Stimmabgabe und eine Stimmenliste, in der zu jedem Kandidaten die für ihn abgegebenen Stimmen mitgezählt werden.

- 1.1.1 Man entwerfe für die Prozedur vote ein Flussdiagramm für den Fall, dass die Fernaufrufe vom Server serialisiert werden.
- 1.1.2 Welche Ergänzungen sind erforderlich, wenn Fernaufrufe nebenläufig abgewickelt werden?
- 1.1.3 Welche der Aufrufsemantiken 'maybe', 'at least once' oder 'at most once' ist für die Lösung der Aufgabenstellung am besten geeignet? Begründung!
- 1.1.4 Welche zusätzlichen (Schutz-)Maßnahmen wären erforderlich, wenn das Netzwerk Unbefugten zugänglich wäre, und wie könnten sie realisiert werden?

Teilgebiet 2: Datenbanken**Aufgabe 2.1: Schemaentwurf im Relationenmodell**

Die Entwicklung eines großen Programmsystems soll durch eine Datenbank unterstützt werden. Sie soll festhalten, welche Komponenten von den einzelnen Entwicklern beigesteuert werden und auf welchen Rechnern die Komponenten gespeichert sind. Außerdem soll sie Auskunft über die Zerlegung von Komponenten in Teilkomponenten geben. Zur Lösung dieser Aufgabe wurde folgendes Entity-Relationship-Diagramm entwickelt:

Fortsetzung nächste Seite!



- 2.1.1 Entwerfen Sie ein Schema nach dem Relationenmodell, das diesem Diagramm entspricht. Bestimmen Sie die Primär- und Fremdschlüssel.

Entwickler (E) habe die Attribute ENR, Name (ENAME) und Ort (EORT).

Komponente (K) habe die Attribute KNR, Bezeichnung (KBEZ), Programmiersprache (KSPRA) und verwendeter Compiler (KCOMP).

Rechner (R) bestehe aus IP-Adresse (RADR) und Komponentennummer (KNR).

Beitrag (B) wird durch ENR, KNR und Datum (DATUM) beschrieben.

Komponentenliste (KL) hat die Attribute Ober-Komponente (OKN) und Teilkomponente (TKN).

- 2.1.2 Drücken Sie die folgenden beiden Anfragen in umgangssprachlicher Formulierung aus:

1. $\pi_{TKN}(KL \mid \bowtie_{OKN=TKN} \pi_{TKN}(KL \mid \bowtie_{OKN=TKN} \pi_{TKN} \sigma_{OKN='Abrechnung'} KL))$
2. $\pi_{RADR}(R \mid \bowtie_{KNR=KNR} (B \mid \bowtie_{\sigma_{EORT='Bangalore'} E}))$

Aufgabe 2.2: Bibliotheksdatenbank

Zur Verwaltung des Ausleihwesens benutzt eine Universitäts-Bibliothek folgende Datenbank:

LESER:	L (<u>LSNR</u> , NAME, VORNAME, WOHNORT)
BUCH:	B (<u>ISBN</u> , TITEL, VERLAG, EXEMPLARE)
EXEMPLAR:	E (<u>ISBN</u> , <u>EXPNR</u> , STANDORT)
AUSLEIHE:	A (<u>LSNR</u> , <u>ISBN</u> , <u>EXPNR</u> , DATUM)

Formulieren Sie folgende Anfragen in SQL:

- 2.2.1 Welche Bücher sind an mehreren Standorten vorhanden?
- 2.2.2 Von welchen Buchtiteln sind sämtliche Exemplare ausgeliehen?

Fortsetzung nächste Seite!

Aufgabe 2.3: Algebraische Optimierung

Zur Organisation des Prüfungswesens werde folgende Datenbank benutzt:

Professor:	P	(PNAME, FBNR)
Student:	S	(MATRNR, NAME, VORNAME, FBNR)
Fachbereich:	F	(FBNR, FNAME, DEKAN)
Prüfung:	PR	(PNAME, MATRNR, NOTE)

- 2.3.1 Folgende Anfrage soll nach den Heuristiken „Selektion möglichst früh“ und „Projektion möglichst spät“ optimiert werden.

$\pi_{\text{MATRNR, NAME, VORNAME}} \sigma_{\text{FNAME='Informatik'}} S \mid > < \mid F$

(Liste der Studenten, die im Fachbereich Informatik eingeschrieben sind.)

- 2.3.2 Erläutern Sie an dem Beispiel, inwiefern obige Heuristiken (evtl. abhängig vom Mengengerüst) zu einer Optimierung der Anfrage führen.

Teilgebiet 3: Betriebssysteme

Aufgabe 3.1: Prozess/Thread

- 3.1.1 Beschreiben Sie die Gemeinsamkeiten und Unterschiede zwischen einem Prozess und einem Thread.
- 3.1.2 Welche Zustände kann ein Thread während seiner Laufzeit annehmen?

Aufgabe 3.2: Koordinierung

Skizzieren Sie ein Programm-Modul (Klasse), das einen Ringpuffer implementiert, mit den Prozeduren (Methoden): **einfügenElement** und **entnehmenElement**. Die Prozeduren sollen von mehreren Threads parallel aufrufbar sein, d.h. Sie müssen insbesondere auf die Koordinierung achten. Als Element können Sie einen beliebigen Datentyp wählen.

- 3.2.1 Skizzieren Sie in einer Ihnen geläufigen Programmiersprache (bzw. in einer Programmiersprachen ähnlichen Notation) die verwendeten Datenstrukturen.
- 3.2.2 Skizzieren Sie in ähnlicher Notation („prozedural“) die beiden Prozeduren (Methoden) zum Einfügen und Entfernen eines Elements. Der Schwerpunkt soll auch hierbei auf den notwendigen Koordinierungsmaßnahmen liegen.
- 3.2.3 Beschreiben Sie die Funktionsweise des von Ihnen verwendeten Koordinierungsmechanismus.

Thema Nr. 2

Sämtliche Teilaufgaben sind zu bearbeiten!

Aufgabe 1: Addierwerke und Multiplexer

Ein Halbaddierer bildet für Bit-Paare die Summe mod2 und den Übertrag.

- a) Man synthetisiere aus zwei Halbaddierern das Schaltnetz für die Binäraddition (Volladdierer).
- b) Man synthetisiere Halb- und Volladdierer ausschließlich aus NAND-Gattern.

Man zeige, wie sich mit Hilfe des Volladdierers

- c) ein serielles 4-Bit Addierwerk,
- d) ein paralleles 4-Bit Addierwerk aufbauen lässt.
- e) Man zeichne das Schaltnetz für einen Multiplexer mit 8 Eingängen ein.

Aufgabe 2: Pipelining

Die Abarbeitung einer Maschineninstruktion erfordere neben dem Zugriff auf die Arbeitsregister, auf den Operanden- und den Instruktionscache die Befehlsdecodierung und -ausführung.

- a) Man zeichne das entsprechende Phasendiagramm.
- b) Man skizziere eine Befehlspipeline für dieses Phasendiagramm unter Berücksichtigung von „Internal Forwarding“.
- c) Man nenne und beschreibe vier Arten von Pipeline-Konflikten, die den Fluss der Befehlsabarbeitung in einer Pipeline hemmen können.
- d) In einer fünf-stufigen Befehlspipeline werden nach jedem bedingten Springbefehl drei „Stall“-Zyklen eingeschoben.
Was sollen diese Zyklen bewirken? Man zeichne die Phasendiagramme für die Abarbeitung von vier Maschinenbefehlen, von denen der erste ein bedingter Sprung ist, mit und ohne Stallzyklen. Wie sehr vermindern die Stallzyklen die Leistung der Pipeline, wenn 20 % der Instruktionen bedingte Sprünge sind?
- e) Man bestimme die Beschleunigung (Speed-Up) einer k-stufigen Pipeline für die Abarbeitung von n Instruktionen, wenn keine Hemmnisse auftreten.
Wie groß ist die maximal erreichbare Beschleunigung?

Fortsetzung nächste Seite!

Aufgabe 3: Übertragungstheorie

1. Erstellen Sie ein Modell einer Übertragungsstrecke.
2. Welche Rolle erfüllt hierbei die
 - a) Quellcodierung?
 - b) Kanalcodierung?
 - c) Leitungscodierung?

Aufgabe 4: Kanalcodierung

1. Geben Sie die Definition von *Hamming-Distanz* an.
2. Wie groß ist die Hamming-Distanz bei einem Code mit einfacher Paritätsprüfung?
3. Welche Hamming-Distanz hat ein zyklischer Hamming-Code und welche Eigenschaften lassen sich daraus ableiten?
4. Ein zyklischer Hamming-Code wird durch das Generatorpolynom $G(u) = u^3 + u + 1$ festgelegt.
 - a) Welche Eigenschaften besitzt das Generatorpolynom $G(u)$?
 - b) Wie ist die Anzahl von Nachrichten- und Kontrollstellen für den durch $G(u)$ erzeugten Hamming-Code?
 - c) Geben Sie die durch $G(u)$ erzeugte Prüfmatrix an.

Aufgabe 5: Quellcodierung

1. Definieren Sie die folgenden Begriffe:
 - a) Informationsgehalt
 - b) Entscheidungsgehalt
 - c) Redundanz
2. Codieren Sie die Zeichen der folgenden Quelle nach dem Verfahren von Huffman und berechnen Sie die Redundanz.

x_i	x_1	x_2	x_3	x_4	x_5
$p(x_i)$	0,3	0,3	0,2	0,15	0,05

3. Ist es immer möglich eine Quelle redundanzfrei zu codieren?