

ab 76 1.0

Fakultät IV – Elektrotechnik und Informatik

Fachgebiet Kommunikations- und
Betriebssysteme (KBS)

Dr. Gero Mühl

E-mail: gmuehl@ivs.tu-berlin.de

WWW: <http://www.kbs.cs.tu-berlin.de>

Zwischenklausur

Verteilte Systeme

Gruppe A

Sommersemester 2005

Berlin, den 25. Mai 2005

Hinweise:

Die Klausur ist als *Multiple Choice-Test* ausgelegt. Insgesamt besteht sie aus 23 Aufgaben mit jeweils 4 Antworten. Für die Lösung dient das angeheftete Lösungsblatt (letzte Seite). Ausschließlich die Angaben auf diesem Lösungsblatt werden bewertet! Beachten Sie unbedingt die weiteren Hinweise zur Beantwortung und zur Bewertung auf dem Lösungsblatt!

Ihnen stehen 90 Minuten zur Bearbeitung zur Verfügung. Geben Sie anschließend sowohl die Aufgabenblätter als auch das **ausgefüllte** Lösungsblatt ab.

Aufgabe 1: (Vektoruhren)

Gegeben sind die drei Vektorzeiten V_1 , V_2 , und V_3 in einem System mit 4 Prozessen zu einem Zeitpunkt t :

$V_1 = (5, 3, 4, 4)$ ist die Vektorzeit von P_1 zum Zeitpunkt t .

$V_2 = (5, 4, 4, 5)$ ist die Vektorzeit von P_2 zum Zeitpunkt t .

$V_3 = (6, 4, 5, 6)$ ist die Vektorzeit von P_3 zum Zeitpunkt t .

- a) Es gilt nicht $V_1 < V_2$
- b) Es gilt $V_1 \neq V_2$
- c) V_1 und V_3 können nicht beide zum gleichen Zeitpunkt die angegebenen Werte gehabt haben.
- d) Mit Hilfe von Vektoruhren lassen sich die Uhrenbedingung und auch die Umkehrung der Uhrenbedingung sicherstellen.

Aufgabe 2: (Verteilte Terminierungserkennung)

- a) Das einfache Zählverfahren bestimmt zuverlässig, ob ein System terminiert ist.
- b) Wird ein Spannbaum auf einer zusammenhängenden Topologie konstruiert, die *kein* Baum ist, so reicht es aus, die Sende- und Empfangszähler mittels nur einer Hin- und Rückwelle auf diesem Spannbaum zu bestimmen, um die Terminierung sicher festzustellen.
- c) Eine Phantom-Terminierung liegt vor, wenn ein Beobachter eine Terminierung feststellt, obwohl das System noch nicht terminiert ist.
- d) Hat jede Basisnachricht eine global eindeutige ID, so kann mittels einer einzigen Kontrollwelle bestehend aus Hin- und Rückwelle die Terminierung sicher festgestellt werden.

Aufgabe 3: (Echo-Algorithmus)

- a) Führt man den Echo-Algorithmus auf einem Baum mit nur einem Initiator aus, so werden bei n Knoten $n-1$ Explorer-Nachrichten verschickt.
- b) Bei einem bidirektionalen Ring werden vom Echo-Algorithmus bei einem Initiator $n+1$ Explorer-Nachrichten verschickt.
- c) Bei Ausführung des Echo-Algorithmus werden *nur* bei Bäumen über *jede* Kante zwei Nachrichten geschickt.
- d) Der Einsatz von Tabulisten beim Echo-Algorithmus spart am meisten Nachrichten, wenn es bei n Knoten $n - 1$ Kanten gibt.

Aufgabe 4: (Hypercubes)

- a) Die Dimension eines Hypercubes entspricht der Anzahl der Knoten.
- b) In Hypercubes ist die minimale Weglänge zwischen zwei beliebigen Knoten höchstens logarithmisch in der Anzahl der Knoten.
- c) In Hypercubes gibt es zwischen zwei beliebigen Knoten immer nur genau einen kürzesten Weg.
- d) Das Routing in Hypercubes ist nicht besonders effizient, weil es so viele verschiedene mögliche Routen gibt, die alle in Routing-Tabellen vorgehalten werden müssen.

Aufgabe 5: (Auswahl auf Ringen)

- a) Die Anzahl der Initiatoren hat beim Bully-Algorithmus keinen Einfluss auf die Anzahl der Nachrichten.
- b) Beim Algorithmus von Chang und Roberts kann ein Knoten ausgewählt werden, auch wenn er nicht die höchste ID unter allen Knoten besitzt.
- c) Die worst-case Nachrichten-Komplexität von Bully und Chang-Roberts ist $O(n^2)$.
- d) Fairness ist eine wichtige Anforderung an Auswahl-Algorithmen.

Aufgabe 6: (Logische Ringe)

- a) Es gibt eine Möglichkeit, einen logischen Ring über eine physikalische Baumstruktur zu legen, so dass ein Ring-Durchlauf nur n Nachrichten benötigt.
- b) Für die Konstruktion eines logischen Rings kann sowohl eine Tiefensuche als auch eine Breitensuche verwendet werden.
- c) Mit Hilfe eines logischen Ringes können bekannte Ringalgorithmen in beliebigen physikalischen Topologien angewendet werden.
- d) Ein logischer Ring kann prinzipiell auf jedem beliebigen (zusammenhängenden, bidirektionalen) Graphen konstruiert werden.

Aufgabe 7: (Auswahl auf Bäumen)

- a) Auswahl auf Bäumen kann effizienter als in Ringen realisiert werden.
- b) Am Ende der Kontraktionsphase begegnen sich zwei Kontraktionsnachrichten auf genau *einer* Kante.
- c) Die Anzahl der Nachrichten in der Kontraktionsphase ist unabhängig von der Anzahl der Initiatoren.
- d) Die Anzahl der Nachrichten in der Explosionsphase ist unabhängig von der Anzahl der Initiatoren.

Aufgabe 8: (Grundlegende Eigenschaften Verteilter Systeme)

Gemeint sind hier die grundlegenden Eigenschaften realer verteilter System, ohne weitere Annahmen.

- a) In einem verteilten System haben die Knoten Zugriff auf einen gemeinsamen primären Speicher.
- b) Die Laufzeit von Nachrichten ist unbestimmt und variiert unvorhersagbar.
- c) Nachrichten können verloren gehen, sich gegenseitig überholen und korrumpiert werden.
- d) Ausgefallene Prozesse oder Links können von langsamen unterschieden werden.

Aufgabe 9: (Modelle für Verteilte Systeme)

- a) Im synchronen Modell haben Nachrichten keine Laufzeit.
- b) Im Atommodell können sich keine Nachrichten überholen.
- c) Welches Modell betrachtet wird, hat keine Auswirkungen auf, ob bzw. wie effizient Algorithmen implementiert werden können.
- d) Das synchrone Modell und das Atommodell sind Spezialisierungen des asynchronen Modells.

Aufgabe 10: (Verteilte Terminierungserkennung)

- a) Das Zeitzonenverfahren erkennt Nachrichten aus der Vergangenheit. Ist eine Nachricht aus der Vergangenheit empfangen worden, so ist das System noch nicht terminiert.
- b) Beim Vektorverfahren werden die Vektoren der Prozesse eingesammelt, bis sich der Nullvektor ergibt. Dann ist das System terminiert.
- c) Beim Vektorverfahren ist es sinnvoller, einen Prozess zu besuchen, dessen Komponente im Kontrollvektor 0 ist, als einen Prozess, dessen Komponente einen von 0 verschiedenen Wert hat.
- d) Bei der Kreditmethode werden die Kreditanteile von Nachrichten und Prozessen als hochgenaue Fließkommazahlen gespeichert.

Aufgabe 11: (Wechselseitiger Ausschluss)

- a) Beim Algorithmus von Lamport kann ein Knoten in den kritischen Abschnitt eintreten, wenn (i) seine Anforderung unter den Anforderungen in seiner Warteschlange diejenige mit dem kleinsten Zeitstempel ist und (ii) er von allen anderen Knoten eine Nachricht mit einem größeren Zeitstempel empfangen hat.
- b) Der Algorithmus von Lamport setzt zuverlässige Kanäle, jedoch keine FIFO-Kanäle voraus.
- c) Der Algorithmus von Ricart und Agrawala braucht $2(n - 1)$ Nachrichten pro Anforderung.
- d) Der Algorithmus von Maekawa braucht nur $O(\log n)$ Nachrichten pro Anforderung.

Aufgabe 12: (Verteilte Speicherbereinigung)

- a) Bei der Referenzzählungsmethode muss verhindert werden, dass eine Inkrement-Nachricht eine Dekrement-Nachricht überholt.
- b) Die Referenzzählungsmethode kann auch Objekte bereinigen, die Zyklen im Objektgraph bilden.
- c) Die gewichtete Referenzzählungsmethode ist von der Kreditmethode für die verteilte Terminierung abgeleitet worden.
- d) Bei der Verwendung der „Markieren-und-Ausfegen“-Verfahren ist es ausreichend, das System vor Beginn der „Ausfegen“-Phase zu stoppen.

Aufgabe 13: (Lamport's Logische Uhren)

- a) Aus $C(a) < C(b)$ folgt $a \rightarrow b$.
- b) Aus $C(a) = C(b)$ folgt $a \parallel b$.
- c) Die logischen Uhren von Lamport definieren zusammen mit eindeutigen Prozessidentitäten eine totale Ordnung auf den Ereignissen.
- d) Aus $a \rightarrow b$ folgt $C(a) < C(b)$.

Aufgabe 14: (Vektoruhren und Physikalische Uhren)

- a) Mit Hilfe von Vektoruhren lässt sich ein kausaler Broadcast realisieren.
- b) Mit Hilfe von Vektoruhren lässt sich eine Ablieferung von Nachrichten in einer die kausale Ordnung respektierenden Reihenfolge sicherstellen, auch wenn jede Nachricht nur an einen Prozess ausgeliefert wird.
- c) Mit physikalischen Uhren lässt sich die Umkehrung der Uhrenbedingung sicherstellen, wenn die Uhren genügend genau synchronisiert sind.
- d) Mit physikalischen Uhren lässt sich die Uhrenbedingung sicherstellen, wenn die Uhren genügend genau synchronisiert sind.

Aufgabe 15: (Schnappschüsse)

- a) Ein Schnappschuss besteht aus den lokalen Zuständen aller Prozesse und aus den Nachrichten, die folgende zwei Bedingungen erfüllen: (i) Die Nachricht wurde gesendet, bevor der Sender seinen Schnappschuss gesichert hat. (ii) Die Nachricht wurde noch nicht empfangen, als der Empfänger seinen Schnappschuss gesichert hat.
- b) Beim Schnappschussalgorithmus von Lai und Yang sichert ein Prozess seinen Schnappschuss, wenn er die Aufforderung dazu oder eine rote Nachricht empfängt.
- c) Beim Schnappschussalgorithmus von Lai und Yang lässt sich anhand der Differenz der Sende- und Empfangszähler feststellen, wie viele Nachrichten in dem Schnappschuss enthalten sein müssen.
- d) Der Schnappschussalgorithmus von Chandy und Lamport verwendet das Kanal-Flushing-Prinzip und ist deswegen nicht auf FIFO-Kanäle angewiesen.

Aufgabe 16: (Grundlegende Begriffe der Fehlertoleranz)

- a) Ein System dessen interner Zustand mit Fehlern behaftet ist, kann immer noch nach außen hin korrekt arbeiten.
- b) Maskierende Fehlertoleranz soll sicherstellen, dass ein System nach einem Ausfall wieder korrekt arbeitet.
- c) „Triple Modular Redundancy“ ist ein bekanntes Verfahren der nichtmaskierenden Fehlertoleranz.
- d) Ein Ersatzreifen lässt sich aktiver Replikation zuordnen.

Aufgabe 17: (Agreement)

- a) Agreement lässt sich in einem synchronen System bei Verwendung von „oral messages“ sicherstellen, wenn mehr als die Hälfte der Teilnehmer korrekt arbeitet.
- b) Agreement lässt sich in einem synchronen System bei Verwendung von „signed messages“ unabhängig von der Anzahl der fehlerhaften Teilnehmer sicherstellen.
- c) Auch in einem asynchronen System ist Agreement möglich.
- d) Nur die Generäle können Verräter sein; nicht aber der Anführer.

Aufgabe 18: (Nicht-Maskierende Fehlertoleranz)

- a) Bei „Fail Safe“ wird die Lebendigkeit des Systems sichergestellt.
- b) Bei „Graceful Degradation“ wird im Fehlerfall versucht, das System mit eingeschränkter Funktionalität weiter zu betreiben.
- c) Eine Rekonfiguration im Fehlerfall hat das Ziel, dass das System nach Abschluss der Rekonfiguration wieder korrekt arbeitet.
- d) Nichtmaskierende Fehlertoleranz ist nicht notwendigerweise auf Redundanz angewiesen.

Aufgabe 19: (Gegenseitiger Ausschluss)

- a) Beim Lift-Algorithmus wandert das Token in Richtung der Kanten eines Baumes und dreht dabei die Richtung der verfolgten Kante um.
- b) Der zentrale Algorithmus benötigt von allen vorgestellten Verfahren die wenigsten Nachrichten pro Anforderung.
- c) Beim einfachen Token-Ring-Verfahren kreist das Token auch dann, wenn gar kein Knoten zugreifen möchte.
- d) Beim Algorithmus von Maekawa sind „Request Granting“-Mengen der Größenordnung $O(\log n)$ minimal.

Aufgabe 20: (Auswahlalgorithmus von Chang und Roberts)

- a) Die average-case Nachrichtenkomplexität ist $n H_k$.
- b) Der best-case tritt ein, wenn die k Initiatoren in absteigender Reihenfolge auf dem Ring angeordnet sind und in aufsteigender Reihenfolge die Auswahl initiieren.
- c) Im best-case werden genau n Nachrichten für die Auswahl benötigt.
- d) Die randomisierte Variante für bidirektionale Ringe hat eine etwa 30% geringere Nachrichtenkomplexität.

Aufgabe 21: (Echo, Fluten und Broadcast)

- a) Jeder Broadcast-Algorithmus hat als untere Schranke eine Nachrichtenkomplexität von $\Omega(n \log n)$.
- b) Ein Broadcast wird immer mittels Fluten realisiert.
- c) Ein durch den Echo-Algorithmus gebildeter Spannbaum ist von der Laufzeit der Explorer-Nachrichten abhängig.
- d) Fluten mit Bestätigung benötigt doppelt so viele Nachrichten wie Fluten ohne Bestätigung.

Aufgabe 22: (Las Vegas- und Monte Carlo-Algorithmen)

- a) Las Vegas-Algorithmen liefern niemals ein falsches Ergebnis.
- b) Die worst-case Laufzeit von Las Vegas-Algorithmen ist beschränkt.
- c) Monte Carlo-Algorithmen liefern manchmal ein falsches Ergebnis.
- d) Monte Carlo-Algorithmen terminieren nie.

Aufgabe 23: (Ordnungsrelationen auf Ereignismengen)

- a) Sind nicht transitiv.
- b) Sind reflexiv.
- c) Sind antisymmetrisch.
- d) Eine partielle Ordnung ordnet jedes Paar von Ereignissen eindeutig.

Name: _____ Vorname: _____ Matrikelnr.: _____

Übersetzungstabelle

8	4	2	1	
a) 0 / 1	b) 0 / 1	c) 0 / 1	d) 0 / 1	Lösungsziffer
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	A
1	0	1	1	B
1	1	0	0	C
1	1	0	1	D
1	1	1	0	E
1	1	1	1	F

Name: _____ Vorname: _____ Matrikelnr.: _____

Lösungsblatt Gruppe A

Füllen Sie Ihre persönlichen Daten am oberen Rand des Blattes ein! Schreiben Sie **gut leserlich!**

Zu den Antworten **a** bis **d** jeder Aufgabe finden Sie in der folgenden Tabelle jeweils ein Kästchen. Tragen Sie eine **1** ein, wenn Sie der Meinung sind, dass die entsprechende Aussage richtig ist. Eine **0** ist einzutragen, wenn Sie die Aussage für falsch halten. Für jede Teilaufgabe **muss** entweder **0** oder **1** eingetragen werden. Aus den binären Lösungsziffern der einzelnen Teilaufgaben ergibt sich für jede Aufgabe eine hexadezimale Lösungsziffer. Die Übersetzung können Sie bei Bedarf der beiliegenden Übersetzungstabelle entnehmen. Haben Sie aus Versehen eine falsche Ziffer eingetragen, korrigieren Sie die Ziffer bitte, indem Sie die falsche Ziffer waagrecht durchstreichen und die andere Ziffer eintragen. Achten Sie unbedingt darauf, dass diese Korrektur gut erkennbar ist! Bitte benutzen Sie einen **dokumentenechten Stift** (keinen Bleistift)!

Lösung (bitte jeweils 0 oder 1 eintragen)					
	8	4	2	1	Lösungsziffer
Aufgabe	a) 0 / 1	b) 0 / 1	c) 0 / 1	d) 0 / 1	0 - F
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					
19					
20					
21					
22					
23					

Zur Bewertung: Jede korrekte binäre Ziffer ergibt einen Punkt, jede nicht korrekte binäre Ziffer einen Minuspunkt. Ergäbe sich für eine Aufgabe eine negative Punktzahl, so wird diese mit 0 Punkten bewertet. Ein Gesamtergebnis von mindestens 50 Prozent der erreichbaren Punkte genügt in jedem Fall, um die Klausur zu bestehen. Bitte beachten Sie, dass von uns nur die angegebene *hexadezimale* Lösungsziffer ausgewertet wird. Fehlt die hexadezimale Lösungsziffer, so wird die gesamte Aufgabe mit 0 Punkten bewertet.