Probeklausur Rechnerarchitektur

Prof. Chen/Prof. Teubner

13.07.2020

Hinweise

1. Bevor Sie mit der Bearbeitung der Aufgaben beginnen, tragen Sie **auf allen Blättern** Ihren Namen und Ihre Matrikelnummer ein.
2. Bei der Bearbeitung der **Probeklausur** sind **Hilfsmittel zugelassen**, bei der eigentlichen Klausur werden **keine Hilfsmittel** zugelassen.
3. Benutzen Sie einen **dokumentenechten**, nicht roten Stift.
4. Der Klausurtext enthält ausreichend Platz zur Lösung der Aufgaben. Benutzen Sie NICHT die Rückseiten der Blätter für Ihre Lösungen, höchstens für Nebenrechnungen. Wenn Sie zusätzliches Papier benötigen, wenden Sie sich an die Klausuraufsicht.
5. Sollte Ihre Lösung nicht unmittelbar bei der Aufgabenstellung stehen, machen Sie einen entsprechenden Hinweis.

Wir wünschen Ihnen viel Erfolg!

erreichte Gesamtpunktzahl | |

Note | |

Aufgabe 1 Speedup (4 Punkte)

Gemäß Amdahl hängt der Gesamtspeedup S_{tot} eines Programms vom Anteil an parallelisierbarem Code P und dem Speedup S für diesen Anteil bei paralleler Ausführung ab. Es wird dabei von der idealen Parallelisierbarkeit des entsprechenden Code-Anteils ausgegangen.

1. Wie lautet die formale Beschreibung von Amdahls Gesetz?

(2 Punkte)

2. Die Gesetze von Gustafson und Amdahl stehen scheinbar im Widerspruch. Erklären Sie, durch welche Unterschiede zwischen den beiden Gesetzen sich der Widerspruch aufheben lässt.

(2 Punkte)

Aufgabe 2 Rechner-Klassifikation (8 Punkte)

Ordnen Sie die vier Klassen von Multiprozessorsystemen nach der Klassifikation von Flynn in die folgende Tabelle und beschreiben Sie eine wichtige Eigenschaft jeder der vier Klassen kurz.

(Je korrekter Klasse in Tabelle 0,5 Punkte; je Eigenschaft 1,5 Punkte; max. 8 Punkte

		Befehlsströme	
		1	>1
Datenströme	1		
	>1		

Klasse: _____

Eigenschaft:

Klasse: _____

Eigenschaft:

Klasse: _____

Eigenschaft:

Klasse: _____

Eigenschaft:

Aufgabe 3 Synchronisierung/Multithreading (10 Punkte)

a) Nennen Sie zwei verschiedene Hardware-Instruktionen mit denen sich einfache Locks realisieren lassen.

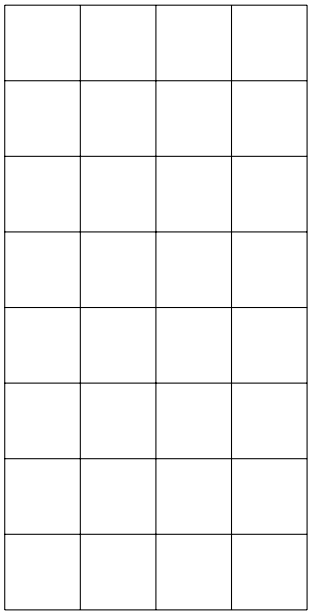
(2 Punkte)

b) Nennen Sie zwei angestrebte Eigenschaften von Locks, die über die Grundfunktion (Sicherheit des Locks) hinausgehen.

(2 Punkte).

c) Nennen Sie **zwei** aus der Vorlesung bekannte Arten von Multithreading und stellen **eine** grafisch dar. Für die grafische Darstellung können Sie das untenstehende Muster verwenden (z.B. mittels Nummerierung unterschiedlicher Threads durch Ziffern oder Buchstaben). Geben Sie an, welche der Möglichkeiten grafisch dargestellt wurde.

(je 1 Punkt pro Nennung, 2 Punkte für Grafik)



Aufgabe 4 Multithreading - Programmanalyse (12 Punkte)

a) Erklären Sie (kurz) was das unten stehende Programm (offensichtlich) machen soll. Kann es in diesem Programm zu Problemen kommen? Wenn ja, welche?

```
int counter;
int result;
void *producer() {
  for (int i = 0; i < 42; i++)
    counter++;
  return NULL;
}

void *consumer() {
  for (int i = 0; i < 42; i++)
    result += counter;
  return NULL;
}

int main() {
  counter = 0;
  result = 0;

  pthread_t t1,t2;
  pthread_create (&t1, NULL, &producer, NULL);
  pthread_create (&t2, NULL, &consumer, NULL);
  pthread_join (t1,NULL);
  pthread_join (t2,NULL);

  printf("Das Ergebnis lautet: %d\n",result;)
  return 0;
}
```

(8 Punkte)

b) Wie können die Probleme gelößt werden?

(4 Punkte)

Aufgabe 5 Dynamisches Scheduling (10 Punkte)

a) Welche grundlegenden Unterschiede gibt es zwischen statischem und dynamischem Scheduling? Welche Probleme gibt es bei beiden Ansätzen? Unter welchen Voraussetzungen für die Hardware bzw. die Programmierung ist der Einsatz der einen oder der anderen Methode vorzuziehen?

(5 Punkte)

b) Erklären Sie kurz den Unterschied zwischen den erwähnten Verfahren des *Scoreboardings* und dem Verfahren von Tomasulo. Welche Unzulänglichkeiten, die beim *Scoreboarding* auftreten, werden von dem Verfahren von Tomasulo beseitigt? Warum?

(5 Punkte)

Aufgabe 6 Sprungvorhersage (8 Punkte)

Die 2-Bit-Sprungvorhersage stellt einen Sonderfall der n-Bit-Sprungvorhersage dar. In dieser Aufgabe wird der Zustandsautomat der 2-Bit Sprungvorhersage betrachtet. Geben Sie ein Zustandsübergangsdiagramm des Automaten an (als Moore-Automat)

Aufgabe 7 Speicherhierarchie: Austauschverfahren (10 Punkte)

a) Mit welchen Strategien können Sie auswählen, welcher Eintrag innerhalb einer Speicherhierarchie im Falle eines *Misses* überschrieben wird?

(1 Punkt je Strategie (außer für LRU), max. 2 Punkte)

b) Wie funktioniert die Ersetzungsstrategie *least-recently-used* (LRU)?

(1 Punkt)

c) Nehmen Sie an, dass wir uns das relative Alter der Einträge mit Hilfe einer Dreiecksmatrix $f[i, j]$ merken wollen, wobei i die Zeile und j die Spalte angibt. Zur Vermeidung von lästigen Fallunterscheidungen ist es nachfolgend zulässig, statt der Speicherung nur der Dreicksmatrix die Speicherung der vollständigen Matrix $f[i, j]$ (allerdings ohne die Diagonale) anzunehmen. Wie sind die Einträge in dieser Matrix definiert?

(2 Punkte)

d) Welche Operation müssen Sie auf der Matrix ausführen, wenn der Eintrag i der jüngste wird?

(1 Punkt)

e) Welche Operation müssen Sie auf der Matrix ausführen, wenn Sie den ältesten Eintrag suchen?

(1 Punkt)

f) Worauf reduziert sich die Realisierung von LRU mittels einer Dreiecksmatix für den Fall eines 2-fach Satz-assoziativen Caches?

(1 Punkt)

g) Bitte nennen Sie eine zweite Technik zur Realisierung von LRU!

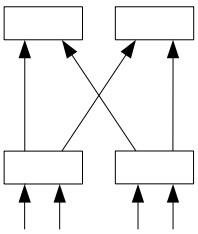
(2 Punkte)

Aufgabe 8 Netzwerke/Routing (10 Punkte)

a) Vervollständigen Sie folgende Zeichnung zu einem Butterfly-Netzwerk mit 16 Ein-/Ausgängen.

Hinweis: Butterfly-Netzwerke verwenden *bit-reversal*-Routing

(5 Punkte)



b) Was zeichnet ein Benes-Netzwerk gegenüber einem Butterfly-Netzwerk aus, wenn es parallele Kommunikation zwischen beliebigen Start-Ziel-Permutationen gibt?

(2 Punkte)

c) Beantworten Sie für ein Netzwerk mit N Knoten folgende Fragen:

(je 1 Punkt, insg. 3 Punkte)

- Welche der in der Vorlesung vorgestellten Topologien hat die größte Halbierungsbandbreite?

- Welchen Durchmesser hat eine binäre Baum-Topologie?

- In welcher der in der Vorlesung vorgestellten Topologien der Dimension k haben *alle* Knoten den Knotengrad k?

Aufgabe 9 Parallelisierung durch OpenMP (10 Punkte)

Fügen Sie passende OpenMP-Pragmas hinzu, damit die nachfolgende Schleife 100 Zufallszahlen summiert. Auch Teile einer Lösungen werden bewertet.

```
int main() {

  int sum = 0;

  for (unsigned int i = 0; i < 100; i++) { // Summiere 100 Zufallszahlen

      sum = rand() % 10; // 0-9

  }

  printf("Summe aus 100 Zufallszahlen: %d\n", sum);

  return 0;
}
```

Aufgabe 10 OpenCL (10 Punkte)

a) Wie sieht das Ausführungs- bzw. Berechnungsmodell von OpenCL aus?

(3 Punkte)

b) Wie sieht das Speichermodell von OpenCL aus?

(4 Punkte)

c) Betrachten Sie bitte folgende Code-Fragmente:

(3 Punkte)

1	`memoryObjects[0] = clCreateBuffer(context,` `CL_MEM_READ_ONLY | CL_MEM_ALLOC_HOST_PTR, buffSize, NULL, &err);`
2	`cl_int* inputA = (cl_int*)clEnqueueMapBuffer(commandQueue, memoryObjects[0],` `CL_TRUE, CL_MAP_WRITE, 0, bufferSize, 0, NULL, NULL, &errorNumber);`
3	`int i = get_global_id(0);`

Bitte erklären Sie, was das jeweilige Code-Fragment bedeutet bzw. macht.

1.

2.

3.