

Übungsblatt 3

Abgabe: 21.11.05 - 23.11.05

Aufgabe 1

Punkte: 4

Gegeben sind 2 Programme P_1 und P_2 , die für die Systeme S_1 bzw. S_2 übersetzt wurden. P_1 und P_2 enthalten Befehle, die sich in Bezug auf ihre Ausführungszeit in 4 verschiedene Klassen einteilen lassen (vgl nachfolgende Tabelle). Ausserdem führt P_1 500 Speicherzugriffe aus, P_2 800.

	Anzahl P_1	Anzahl P_2
Befehle mit CPI=2	20000	10000
Befehle mit CPI=4	5000	2000
Befehle mit CPI=10	2000	5000
Befehle mit CPI=50	30	300
Speicherzugriffe	500	800

Anmerkung: CPI ist die Abkürzung für „Cycles per Instruction“. Damit wird die Anzahl der benötigten Taktzyklen des Prozessors für einen bestimmten Befehl bezeichnet. Das bedeutet für die Tabelle: Programm P_1 verwendet 20000 Befehle, die zur Ausführung 2 Taktzyklen benötigen, 5000 Befehle, die 4 Taktzyklen benötigen, usw.

a) Bestimmen Sie die Ausführungszeit von Programm P_2 . Nehmen Sie dazu an, daß ein Speicherzugriff genau 100 Takte dauert und das System mit einer Taktrate von 1 GHz getaktet wird.

Die Programme P_1 und P_2 dienen zur Berechnung von Skalarprodukten. Von Programm P_1 ist bekannt, daß insgesamt 1500 Skalarprodukte berechnet werden.

b) Wieviele Skalarprodukte müsste System S_2 bei gleicher Taktrate berechnen, damit beide Systeme dieselbe Performance aufweisen? Nehmen Sie an, daß Programm P_1 zur Ausführung auf System S_1 genau 131,5 μs benötigt.

c) Um welchen Faktor müsste die Taktfrequenz des langsameren Systems erhöht werden, damit beide Programme gleich schnell ausgeführt werden?

Im Folgenden soll ein Speicherzugriff auf System S_1 nun 10 ns benötigen, auf System S_2 7 ns. System S_1 werde mit einem Takt von 2 GHz betrieben.

d) Wie hoch muss die Taktfrequenz von System S_2 sein, damit beide Systeme gleich schnell laufen?

Für die folgenden Aufgaben benötigen Sie die in der Vorlesung bereits vorgestellte Toy-Machine. Eine Java-Version finden Sie auf der Webseite zu dieser Vorlesung im Bereich „Downloads zur Bearbeitung der Übungsblätter“.

Hilfreich ist es, die zahlreichen Beispielprogramme anzuschauen. Diese findet man unter „File→Open Example Programs“. Dort sehen Sie auch, wie man Toy-Machine-Programme sinnvoll kommentiert. Die volle Punktzahl für die Programmieraufgaben gibt es nur für in diesem Stil dokumentierte Abgaben!

Aufgabe 2

Punkte: 2

Betrachten Sie folgendes Toy-Machine-Programm:

```
10: 7101
11: 7207
12: 7301
13: 1333
14: 2221
15: D213
16: 0000
```

Was steht bei der Terminierung in Register 3?

Aufgabe 3

Punkte: 4

Schreiben Sie ein Programm `cube.toy`, das eine Zahl a vom Stdin einliest und a^3 auf dem Stdout ausgibt. Testen Sie Ihr Programm mit mindestens 5 verschiedenen Zahlen und dokumentieren Sie die Ergebnisse.

Aufgabe 4

Punkte: 4

Schreiben Sie ein Programm `iseven.toy`, das eine Zahl vom Stdin einliest und 0001 ausgibt, falls die Zahl gerade ist, bzw 0000, falls die Zahl ungerade ist. Testen Sie auch hier Ihr Programm mit mindestens 5 verschiedenen Zahlen und dokumentieren Sie die Ergebnisse.

Aufgabe 5

Punkte: 6

Mitte der 80er Jahre des letzten Jahrhunderts erfand A.K. Dewdney als Zerstreung für Programmierer das Computerspiel „Core Wars“. Dabei treten zwei in einer Assembler-Sprache geschriebene Programme gegeneinander an. Ziel des Spieles ist es, das jeweils andere Programm zu zerstören. Dies geschieht im Allgemeinen dadurch, dass man versucht, dem anderen Programm eine ungültige Anweisung in den Speicher zu schreiben, die es dann nicht mehr ausführen kann. Ein ganz einfaches „Core-Wars“-Programm, der sogenannte „Dwarf“, wirft in bestimmten Abständen „Bomben“ (= ungültige Anweisungen) in den Speicherraum und versucht dadurch die Befehle des Gegners zu überschreiben.

Ziel dieser Aufgabe ist es, einen solchen „Dwarf“ für die Toy-Machine zu entwerfen.

Dazu nehmen Sie an, dass 0FFF die ungültige Anweisung ist, die ein anderes Programm zum Absturz bringt. Schreiben Sie also ein Programm, das in bestimmten Abständen „0FFF“-Bomben in den Speicher schreibt. Achten Sie bei der Wahl der Abstände darauf, dass das Programm sich nicht selbst überschreibt, sondern endlos weiterläuft.

Die Abgabe der Lösungen zu den theoretischen Aufgaben erfolgt schriftlich in den jeweiligen Übungsgruppen. Die Abgabe der Lösungen zu den Programmieraufgaben klären sie bitte mit Ihrem Tutor (Email, Diskette, CD, ...).

Bitte schreiben Sie unbedingt Ihre Namen und Ihre Matrikelnummern auf die Ausarbeitung! Die Bearbeitung erfolgt ausnahmslos in Zweiergruppen.