Prof. Dr. K. Wüst
FH Gießen–Friedberg, FB MNI

WS 2002/2003
Studiengang Informatik

Praktikum Maschinennahe Programmierung
Aufgabenblatt 11

Themen: Zeiger, Zeitmarkenzähler

Aufgabe 11.1 Testen Sie die untenstehende Funktion und finden Sie heraus, warum sie nicht funktioniert! Betrachten Sie dazu den erzeugten Assemblercode!

```
void vertausche(int x, int Y) {
int temp;
    temp=x;
    x=y;
    y=temp;
```

Aufgabe 11.2 Verändern Sie die Funktion `vertausche(...)` aus der obigen Aufgabe so, dass sie funktioniert. Schreiben Sie die Funktion dabei zunächst in reinem C/C++ und testen Sie an Beispielzahlen im Rahmen eines C-Programmes.

Aufgabe 11.3 Schreiben Sie nun den Rumpf der Funktion `vertausche(...)` in Inline-Assembler und testen Sie wieder!

Aufgabe 11.4 Ermitteln Sie für die folgende Befehlssequenz die in Prozessortakten gemessene Ausführungszeit.

```
    mov esi,eax
    mov ebx,edx
    sub edi,esi
    xor edi,edi
    and edi,ecx
    or ebx,ecx
    add eex,edx
```

Fügen Sie diese Befehle nach und nach ein, so dass Sie zuerst nur die Ausführungszeit für den ersten Befehl ermitteln, dann für die ersten beiden, für die ersten drei usw.

Hinweise: Für die Ermittlung der Ausführungszeit kann der Zeitmarkenzähler (time stamp counter) des Pentium-Prozessors benutzt werden, ein 64-Bit-Zähler, der bei jedem Prozessortakt um eins erhöht wird. Mit dem Befehl `RDTSC` wird der Zeitmarkenzähler ausgelesen und in die Register EDX (höherwertige 32 Bit) und EAX (niederwertige 32 Bit) übertragen. Die Ausführungszeit für eine Befehlssequenz kann dann als Differenz zweier Zählerstände ermittelt werden, wenn der zuerst gelesene Zählerstand irgendwo gesichert wird. `RDTSC` braucht allerdings selbst schon 31 Prozessortakte! Das alles kann auch in

Inline-Assembler durchgeführt werden, die Ausgabe des Ergebnisses ist dann sehr bequem.

Aufgabe 11.5 Ermitteln Sie nun mit der gleichen Methode jeweils die Ausführungszeit für die Befehle:

```
imul eax,ebx
div ebx
finit
fsin
printf("Hallo Welt!")
```

Aufgabe 11.6 Ein Feld von 80 vorzeichenlosen 8-Bit-Zahlen soll in einer Schleife bearbeitet werden. Dabei sollen alle Werte um zwei erhöht werden.

```
unsigned char Zahlen[80];

main() {
int i;
for (i=0; i<80; i++) Zahlen[i]=2*i;

// erhöhe alle Werte um 2 ...

}
```

Lösen Sie die gestellte Aufgabe

1. in normalem C,
2. in Inline-Assembler,
3. in Inline-Assembler mit MMX-Befehlen.

Bestimmen Sie für alle drei Lösungen die Ausführungszeit! Stellen Sie nun den Compiler so ein, dass die Ausführungszeit optimiert wird, („Geschwindigkeit erhöhen“) und übersetzen Sie die erste der drei Lösungen (reines C) erneut, messen Sie wieder die Ausführungszeit.

Hinweise: Hersteller-Angaben von intel zum Pentium4-Prozessor sind in den Dateien `pentium4_ba.pdf` und `pentium4_isr.pdf` auf dem Server verfügbar und enthalten weitere Einzelheiten zu MMX.