



Technische Informatik II im SS 2007

10. Übungsblatt

Abgabetermin: 05. Juli 2007, 13:00 Uhr

Prof. Dr. J. Henkel

Am Zirkel 2, Geb. 20.20
D-76131 Karlsruhe

Dr.-Ing. T. Asfour

Telefon: +49-721-608-7379
Fax: +49-721-608-8270
Email: asfour@ira.uka.de
<http://ti.ira.uka.de>

Aufgabe 1

(4 Punkte) Schreiben Sie ein MIPS-Assemblerprogramm, welches

eine Integer-Zahl einliest und das entsprechende ASCII-Zeichen in Hochkomma ausgibt, z.B.: Eingabe 42, Ausgabe "★". Falls die eingegebene Zahl nicht im gültigen Bereich für ASCII-Zeichen (32 bis 126) liegt, soll eine Fehlermeldung ausgegeben werden.

Aufgabe 2

(6 Punkte) Schreiben Sie ein MIPS-Assemblerprogramm, welches eine maximal vierzig Zeichen lange Zeichenkette und eine 8-Bit Integerzahl (den Schlüssel) einliest. Anschließend werden jeweils ein Byte der Zeichenkette mit dem Schlüssel „exklusiv-oder“ verknüpft und das Ergebnis jeder dieser Verknüpfungen als Integer-Zahlen ausgegeben.

Eine bestimmte Zeichenkette, die mit der Integer-Zahl 78 verschlüsselt wurde, ergibt folgende Zahlenfolge:

3 39 37 60 33 62 60 33 52 43 61 61 33 60 43 32 98 110 29 62 43 39 45 38 43 60 98 110 30 43
60 39 62 38 43 60 39 43

Wie lautet die Zeichenkette?

Aufgabe 3

(7 Punkte)

1. Gegeben ist das folgende MIPS-Programmstück:

```
anfang:    addi $t0, $zero, 0
           addi $t1, $zero, 1
schleife:  slt  $t2, $a0, $t1      # if ($a0<$t1) then $t2=1 else $t2=0
           bne $t2, $zero, ende
           add  $t0, $t0, $t1
           addi $t1, $t1, 2
           j   schleife
ende:      add $v0, $t0, $zero
```

Das Register \$a0 ist mit einer positiven Integerzahl n initialisiert. Das Register \$v0 wird für die Ausgabe des Ergebnisses verwendet.

- i.) Welche Funktion erfüllt das angegeben Programmstück?
 - ii.) Welche Werte stehen im Register `$v0` nach Abarbeitung des Programmstücks, wenn das Register `$a0` mit 9 bzw. mit 10 initialisiert wird.
2. Geben Sie die echten MIPS-Befehle bzw. Befehlsfolgen zu den folgenden MIPS-Pseudobefehlen an. Verwenden Sie das Register `$at` zur Speicherung temporärer Variablen.
- `move $t5, $t3` # `$t5 = $t3`
 - `clear $t5` # `$t5 = 0`
 - `bgt $t5, $t3, marke` # `if ($t5 > $t3) goto marke`
 - `bge $t5, $t3, marke` # `if ($t5 >= $t3) goto marke`
3. Beschreiben Sie die Funktion der folgenden MIPS-Befehle:
- i.) `lw $s1, 100($s2)`
 - ii.) `sw $s1, 100($s2)`
 - iii.) `jal mystery`

Aufgabe 4

(10 Punkte)

Schreiben Sie ein MIPS-Assemblerprogramm, welches einen Fließkommawert mit doppelter Genauigkeit (*double-precision*) einliest und dessen Sinus-Funktion berechnet und ausgibt. Verwenden Sie dazu die Taylorreihenentwicklung:

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} \pm \dots = \sum_{n=0}^{\infty} (-1)^n \cdot \frac{x^{2n+1}}{(2n+1)!} \quad (\text{x ist im Bogenmaß})$$

Zur Implementierung gehen Sie wie folgt vor:

1. Schreiben Sie ein Unterprogramm **fakultaet** zur Berechnung der Fakultätsfunktion. Achten Sie darauf, dass als Ergebnis ein Fließkommawert geliefert wird, da die Division innerhalb der Summanden der Potenzreihe mit Fließkommawerten durchgeführt wird.
 2. Schreiben Sie ein Unterprogramm **exponent** zur Berechnung der Potenz einer Fließkommazahl mit ganzzahligem Exponenten. Das Ergebnis muss wie in 1) eine Fließkommazahl sein.
 3. Implementieren Sie die Potenzreihe unter Verwendung von **fakultaet** und **exponent** als Unterprogramm **sinus**. Die Summation soll abgebrochen werden, wenn sich das Ergebnis der Partialsummen durch die Addition eines weiteren Summanden nicht mehr verändert.
- Hinweis:** Verwenden Sie eine Variable m , die beginnend mit $m = 1$ alle ungeraden Zahlen aufzählt. Die Quotienten der Summanden ergeben sich dann als $\frac{x^m}{m!}$. Für $m = 1, 5, 9, \dots$ werden diese Quotienten addiert und sonst subtrahiert. Betrachtet man m als Binärzahl, so kann mittels der zweit-niedrigstwertigen Bits unterschieden werden, ob ein Summand addiert oder subtrahiert wird.
4. Schreiben Sie ein Hauptprogramm, das den Fließkommawert einliest, die Sinus-Funktion berechnet und das Ergebnis ausgibt.