

Übungsblatt 7

Rechnerarchitektur im SoSe 2020

Zu den Modulen K

Besprechung: Besprechung der Übungsaufgaben in den Übungsgruppen vom 15. – 19. Juni 2020

Aufgabe Ü15: Assemblerprogrammierung unter SPIM

(– Pkt.)

Im Folgenden soll ein MIPS-Assembler Programm vervollständigt werden, welches als Nutzereingabe eine 8-stellige Binärzahl von der Konsole entgegennimmt, sie in ihre dezimale Repräsentation umwandelt und diese auf der Konsole ausgibt. Die Eingabe wird vom Programm als String entgegengenommen. Danach soll in einer Schleife über die Zeichen diese Strings iteriert und geprüft werden, ob es sich beim aktuellen Zeichen um eine „0“ oder eine „1“ handelt und die entsprechende Wertigkeit aufsummiert werden. Zudem muss der Fall behandelt werden, dass das Ende des Strings, welches durch ein Byte mit dem Zahlenwert 0 markiert ist, erreicht wurde. Beachten Sie, dass der Sting nach 8 von der Konsole gelesenen Zeichen automatisch übernommen und Null-terminiert wird. Dementsprechend ist kein Zeilenumbruch enthalten.

Laden sie sich die `binarytodecimal.s` von der Homepage herunter und ergänzen Sie den dort angegebenen Coderahmen um insgesamt **6 Zeilen Code**, so dass das Programm wie beschrieben funktioniert. Tragen Sie Ihre Lösung unter den mit „# Ihre Loesung:“ markierten Stellen direkt in den Coderahmen der heruntergeladenen Datei ein.

Aufgabe Ü16: Interaktion mit dem Stack

(– Pkt.)

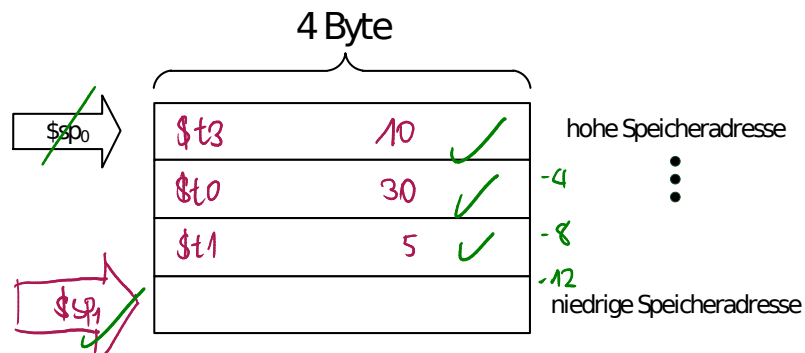
Sei folgendes MIPS-Codefragment gegeben:

```
1      li $t0, 30
2      li $t1, 5
3      li $t2, 15
4      li $t3, 10
5      addi $sp, $sp, -12
6      sw $t2, 12($sp) 15
7      sw $t3, 12($sp) 10
8      sw $t0, 8($sp) 30
9      sw $t1, 4($sp) 5
```

Handwritten notes:
\$sp = stack pointer
(move \$sp 3 down) // um 12 herabgesetzt.
Verschieben
↑

Skizzieren Sie in folgenden Vorlage den Inhalt des Stacks nachdem alle Programmzeilen ausgeführt wurden. Kennzeichnen Sie auch die neue Position des Stackpointers!

$12(\$sp) = (12 + \$sp)$ → Address Operator



Stackpointer : zeigt im Stack die Add der nächste freie Speicherzelle.

Program Counter : Next Instruction to execute



Stack : Super Easy to free up memory !
Just move the \$sp up !