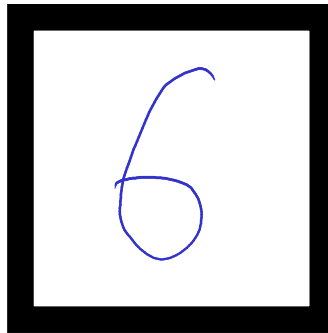


Vorlesung Rechnerorganisation Wintersemester 2020/21

- Übungsblatt 4 -

Tutoriumsnummer



Name, Vorname: Slavov, Velislav

Matrikelnummer: 2385786

Studiengang: Informatik BSc

Name des Tutors: Jonas Heinle



/25 Punkte

A1

- W
- F
- F
- W
- ~~W~~



A2

1. Dies bedeutet, dass die Adresse eines Datums durch 4 teilbar ist (32 bit = 4 Byte). Somit kann die CPU direkt zur Adresse navigieren und sich die nächste 32 bit vom Speicher holen.




2. Sie gibt an, ob das niedrigwertigste Byte an der kleinste oder an der größte Adresse zu finden ist.



A3

- RISC
- CISC
- RISC
- CISC
- CISC
- CISC



A4 Das Programm Findet das größte Element des Arrays $[36, 20, 27, 15, 1, 62, 41]$. 

Wobei:

- \$t0 speichert den array Index des aktuellen Elements
- \$s0 speichert das aktuell größte Element
- \$s1 speichert die Länge des Arrays
- bge \$t0, \$s1, m3 beachtet, ob der aktuelle Index kleiner ist als die Länge des Arrays
- mul \$t1, \$t0, 4 berechnet wie viele Bytes nach dem Anfang des Arrays sich das nächste Element befindet ($\text{index} * 4$)
- ble \$t2, \$s0, m2 Sorgt dafür, dass nur Elemente die größer sind aus dem aktuellen max in \$s0 gespeichert werden
- m2 incrementiert den Index und springt zurück zum Loop
- m3ibt das max Element aus und ruft "exit" auf
g

A5 1i) Speichert die Summe aller ungeraden Zahlen die $\leq n$ sind im Register \$v0.

Wobei:

- \$t0 speichert die aktuelle Summe

- \$t1 speichert die aktuelle Zahl

- `slt $t2, $a0, $t1` sorgt dafür, dass die aktuelle Zahl $\leq n$ ist

- `add $t0, $t0, $t1` aktualisiert die Summe

- `addi $t1, $t1, 2` berechnet die nächste ungerade Zahl



1ii) Für $n=9$ kommt 25 raus

Für $n=10$ kommt wieder 25 raus



2i) Lädt den Wert an der Adresse $100 + \langle \text{Wert aus Register } \$s2 \rangle$ im Register \$s1 ✓

2ii) Speichert den Wert aus Register \$s1 in der Speicheradresse $100 + \langle \text{Wert aus Register } \$s2 \rangle$ ✓

2iii)



A 6

1. Der "syscall" Befehl

2.

1 = print_int

6 = read_float

2 = print_float

7 = read_double

3 = print_double

8 = read_string

4 = print_string

9 = sbrk

5 = read_int

10 = exit

3. Register \$v0

