

## 4. Tutorium

# MIPS Assembler

Rechnerorganisation, Tutorium #13

Patrick Röper | 26. November 2019

FAKULTÄT FÜR INFORMATIK



## 1 MIPS

## 2 Aufgaben

Assemblerdirektiven oder Assembleranweisungen dienen zur Angabe von Zusatzinformationen

- .align n
- .data
- .text
- .globl
- .kdata
- .ktext
- .ascii
- .asciiz
- .byte
- .half
- .word
- .float
- .double
- .space

- Aufruf von Systemfunktionen über die Instruktion **syscall**
- **syscall** über eindeutigen **call code** identifiziert.
  - Muss ins register **\$v0** geladen werden.
- Operanden in Register **\$a0** bis **\$a3**.
- Fließkommawerte in Register **\$f12**.
- Ergebnisse werden in Register **\$v0** bzw. **\$f0** geschrieben.

## 1 MIPS

## 2 Aufgaben

## Aufgabe 1

Laden Sie sich den MIPS-Simulator MARS von der Projektseite <http://courses.missouristate.edu/kenvollmar/mars/> herunter und installieren Sie diesen auf ihrem System. Machen Sie sich anschließend mit der Bedienung des Programms vertraut und schreiben Sie ein Hello-World-Programm. Testen Sie bei der Ausführung des Programms auch die in MARS integrierte Debug- Funktionalität.

## Lösung 1

```
hello: .data                #Es folgen Daten im Datensegment
      .asciiz "Hello World! \n"  #auszugebender Text

      .text                  #Umschalten aufs Textsegment
      .globl main            #main als globales Symbol

main:  li $v0, 4              #Systemaufruf-Nr. fuer
                                #print_string -> $v0
      la $a0, hello          #Adresse des auszugebenen
                                #Texts -> $a0
      syscall                #Systemaufruf
      jr $ra                 #Programmende
```

## Aufgabe

Die Variablen i, j und k stehen in den Registern \$s3, \$s4 und \$s5.

```
if ( i == j )  
    k = i + j ;
```



## Aufgabe

Die Variablen i, j und k stehen in den Registern \$s3, \$s4 und \$s5.

```
if ( i == j )  
    k = i + j ;
```

## Lösung

```
        bne $s4 , $s3 , label  
        add $s5 , $s4 , $s3  
label:  ...
```

## Aufgabe

Die Variablen i, j und k stehen in den Registern \$s3, \$s4 und \$s5.

```
if ( i < j )  
    k = 1;  
else  
    k = 0;
```

## Aufgabe

Die Variablen i, j und k stehen in den Registern \$s3, \$s4 und \$s5.

```
if ( i < j )  
    k = 1;  
else  
    k = 0;
```

## Lösung

```
slt $s5 , $s3 , $s4
```

## Aufgabe 3

Führen Sie den folgenden MIPS-Kode aus und geben Sie die Änderungen in den 4 P. Register- und Speicherinhalten an.

```
addi $t3, $0, 0x20
```

```
lw $t1, 0x04($t3)
```

```
add $t4, $t3, $t0
```

```
sw $t4, 0x08($t3)
```

Register	Inhalt	Adresse	Inhalt
\$t0	0x10	\$0x20	0x22
\$t1	0x14	\$0x24	0x30
\$t2	0x16	\$0x28	0x40
\$t3	0x28	\$0x2C	0x50
\$t4	0x1234	\$0x30	0x60

## Lösung 3

Register	Inhalt	Adresse	Inhalt
\$t0	0x10	\$0x20	0x22
\$t1	<b>0x30</b>	\$0x24	0x30
\$t2	0x16	\$0x28	<b>0x30</b>
\$t3	<b>0x20</b>	\$0x2C	0x50
\$t4	<b>0x30</b>	\$0x30	0x60

# Was ihr jetzt kennen und können solltet...

- MARS
- Grundlagen des MIPS Assembler