



Rechnerarchitekturen 1*

Übungen zum MIPS-Assembler

Prof. Dr. Alexander Auch

*Teilweise entnommen aus “Mikrocomputercomputertechnik 1” von Prof.Dr-Ing. Ralf Stiehler und Rechnerarchitekturen von Dr. Leonhard Stiegler

Vorbemerkungen

- Jede Gruppe sollte aus 2-3 Personen bestehen
- Sie können selbst die jeweiligen Breakout-Räume wählen
- Sie bekommen Bearbeitungszeit
- Bei Problemen sprechen Sie bitte mich an

Aufgabe 1

- Schreiben Sie ein Programm, das einen int-Wert von der Konsole liest, und diesen Wert anschließend ausgibt

```
.data
msg1:
    .asciiz "Bitte Zahl eingeben: "
msg2:
    .asciiz "Sie haben folgenden Wert eingegeben: "

.text
.globl main
main:
    # Ihr Code
    li $v0, 10 # exit
    syscall
```

Aufgabe 2

- Schreiben Sie ein Programm, das von 1 bis 10 zählt, die einzelnen Werte ausgibt, deren Summe berechnet und ebenfalls ausgibt.

```
.data
cr:
    .asciiz "\n"
msgSum:
    .asciiz "Summe: "

.text
.globl main
main:
    #...
    li $v0, 10 # exit
    syscall
```

Aufgabe 3 – Funktionsaufruf & Rekursion

- Schauen Sie sich die Datei fact.asm in MARS an.
- Beantworten Sie folgende Fragen:
 - Warum wird hier ein Stack verwendet?
 - Was würde es für die Programmiersprache bedeuten, wenn man keinen Stack hätte?
 - Wozu dient \$fp?
 - In Zeile 48 wird \$a0 auf dem Stack gesichert. Ist dies nötig? Bitte begründen Sie Ihre Antwort
 - Warum wird \$ra auf dem Stack gesichert? Gäbe es eine elegantere Lösung?
- Im Hinblick auf den Prozedureinstiegs- und ausstiegscode: inwiefern macht die Konvention mit \$tX und \$sX Sinn?