

Register

- Behälter für Daten
- sehr schneller Zugriff

MIPS

- $\$v_0, \v_1 für Ergebnisse
- $\$a_0 - \a_3 für Argumente
- $\$t_0 - \t_7 für temp. Variablen
- $\$s_0 - \s_7 für langlebige Variablen

Systemaufrufe

z.B. `System.out.print("abc");` in JAVA

SPIM: 1 Befehl für alle 10 Systemaufrufe
SYSCALL

Welche Funktion gewünscht?

→ Nummer der Fkt zuvor in $\$v_0$ laden

z.B. 4: Print String
5: Read Int

ggf. Argumente in $\$a_0, \a_1

ggf. Rückgabewerte über $\$v_0$

Aufbau eines SPIM-Programms

Data-Direktive:

`.data`

`marke: .word ...`

`.ascii ...`

Data-Segment: `0x10010000`

`0x10010004`

Textdirektive:

.text

main:

li \$t0, 5

addi \$t1, \$t0, 7

;

li \$v0, 10

syscall

Textsegment

0x00400024

li, \$t0, 5

0x00400028

addi ...

Speicherlauf

Stacksegment



Datasegment

Textsegment

.data

m1:

.word 4, 5, 8, 11, 20

.ascii "abcde"

(HEX !!)

0x10010000:

0x00000004

0x10010004:

0x00000005

0x10010010:

0x00000014

0x10010014:

0x64636261

17 16 15 14

0x10010018:

660065165

~~.ascii "e"~~

.. " "
asciiz "e"

asciiz "f"

0x10010018

660065|65
 ^e ^e

0x1001001B

0x1001001C

0x 00