

RISC-V CPU Emulator

Christian Krause

christian.krause@stud.uni-heidelberg.de

Was passiert, bevor ein Programm ausgeführt wird?

1. C-Programm (hello_world.c)
2. Assembly Datei (hello_world.s)
3. **Ausführbare Datei** (“.exe” auf Windows) (hello_world)

```
addi    s0,sp,32
li     a5,3
sw    a5,-20(s0)
li     a5,37
sw    a5,-24(s0)
lw     a5,-20(s0)
mv    a4,a5
lw     a5,-24(s0)
addw   a5,a4,a5
sw    a5,-28(s0)
lw     a5,-28(s0)
addiw  a5,a5,2
sw    a5,-28(s0)
lw     a5,-28(s0)
mv    a1,a5
```

Was passiert, während ein Programm ausgeführt wird?

- (Beispiel: **sl**)
 1. Program wird in den Arbeitsspeicher geladen
 2. CPU lädt eine **Instruction** (z.B. addieren, speichern etc.) aus dem RAM
 3. CPU führt die Instruction aus *
 4. Wird wiederholt bis das Programm abgeschlossen ist
 - * Eigentlich passiert da noch ein bisschen mehr

Was passiert, während ein Programm ausgeführt wird?

- (Beispiel: **sl**)
 1. Program wird in den Arbeitsspeicher geladen
 2. CPU lädt eine **Instruction** (z.B. addieren, speichern etc.) aus dem RAM
 3. CPU führt die Instruction aus *
 4. Wird wiederholt bis das Programm abgeschlossen ist

Was hat **RISC-V** damit zu tun?

Werbung

Wir werden

- Verstehen wie ELF/Assembly Dateien aufgebaut sind
 - Eine virtuelle CPU-Architektur implementieren
 - Programme auf unserer CPU ausführen
 - optional: (Linux ausführen, Doom cross-compile, (halbes) Betriebssystem implementieren)
- (+ danach könnt ihr Assembly)