Betriebssysteme

Wintersemester 2022/23 Übungsblatt 02

John Zeeh (4725202)

Daniel Burkhardt (4139647)

7.0/8

Aufgabe 1 a)

STOREIN ACC SP

- 1. IRd: Info aus I auf den R-Datenbus und damit die ALU
- 2. SPLd: Aus dem Stack pointer auf den L-Datenbus und in die ALU
- 3. ALUAd: Adresse in den SRAM
- 4. DDId: Adresse in den Datenbus DI
- 5. Steigende Taktflanke auf ACC, um dort die Info zu speichern

-0.25 Store bedeutet schreiben in den SRAM, also muss SPDd geöffnet werden

b)

MOVE IN2 ACC

- 1. IN2Dd: Info auf den Datenbus D
- 2. DDId: Info auf den Datenbus DI
- 3. Steigende Taktflanke auf ACC, um dort die Info zu speichern
- \Rightarrow 2 Treiber

c)

ADD ACC IN1

Man muss ja zwei Werte addieren, also es durch die ALU schicken. Folgender Execute-Plan:

- 1. IN1Ld: In1 auf den Datenbus L und damit die ALU
- 2. ACCDd: Wert aus ACC auf den Datenbus D
- 3. DRd: Wert aus Acc in die ALU
- 4. ALUDId: Wert aus der ALU auf den DI-Datenbus
- 5. Steigende Taktflanke auf ACC, um die Summe dort abzuspeichern.

d)

Wenn der Befehl in I steht, und man ihn durch die ALU leiten will, ohne den Index zu verändern (also einfach 0 addiert)

Bei nochmaligem Nachdenken würde man dann den IAd-Treiber anste<u>uern und ihn gar nicht erst durch</u>

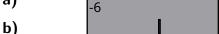
die ALU leiten.

-0.5 z.B. LOADI D i konkretter Befehl sollte genannt werden

Aber wenn ihr einen Immediate in ein Register schreiben wollt, wie kommt ihr dann vom Adressbus zu den Registern.

Aufgabe 2

a)



Aufgabe 3

brk(0x8e5000) = 0x8e5000

und alloziert neuen Speicher, passt schon

Der Systemaufruf gibt den aktuellen Ort der Programmunterbrechung an.

jeder Systemcall returnt etwas

 $openat(AT_FDCWD, "myfile.txt", O_WRONLY|O_CREAT|O_TRUNC, 0666) = 3$

Laut Programm wurde mit fopen und der Spezifikation 'w' eine leere Datei erstellt und geöffnet weil sie bearbeitet werden soll.

und was bedeutet w? passt schon

 $fstat(3, st_mode = S_IFREG|0644, st_size = 0, ...) = 0$

Zeigt Informationen zur erstellten Datei an, sie ist noch leer.

write(3, "012345678910111213" ..., 8192) = 8192

write(3, "60186118621863186418651866" ..., 3198) = 3198

fprintf lässt zu, dass die Datei beschrieben wird.

$$close(3) = 0$$

fclose f schließt die Datei.

 $openat(AT_FDCWD, "myfile.txt", O_RDONLY) = 3$

Mit openat und "r"wird die Datei geöffnet, um sie zu lesen.

 $fstat(3, st_mode = S_IFREG|0644, st_size = 11390, ...) = 0$

Zeigt Informationen zur erstellten Datei an, sie ist nicht mehr leer.

read(3, "012345678910111213" ..., 8192) = 8192

read(3, "60186118621863186418651866" ..., 8192) = 3198

Der Inhalt wird mit fscanf gelesen.

read(3, 8192) = 0

Gibt den gelesenen Programminhalt aus.

close(3) = 0

fclose f schließt die Datei.

 $exit_q roup(42) = ?$

-0.5 verlässt thread

+++exited with 42+++

das ist kein Symtemcall

Im Programm steht der exit code 42. exitgroup terminiert alle threads.