

# Betriebssysteme Übungsblatt 2

Marius Birmele mb504 3543404

4.11.2022

19/20 sehr gute  
Arbeit! n\_n

7/8

## Aufgabe 1:

a)

STOREIN ACC SP i: Der Speicher an Adresse ACC + i wird mit SP beschrieben.

SPDd um die Daten des SP auf den Datenbus zu legen.

ACCLd und IRd um den Wert des ACC mit i vom IR in der ALU zusammenzuzählen.

ALUAd um die Adresse auf den Adressbus zu legen

ASMD um die Adresse vom Adressbus in den Speicher zu übertragen.

b) MOVE IN2 ACC: Der Inhalt des IN2 wird in den ACC kopiert.

Die minimale Anzahl an Treibern ist 2. Erstens muss der IN2Dd aktiv sein, um die Daten vom IN2 auf den Datenbus zu legen, zweitens muss der DDId aktiv sein, um die Daten vom Datenbus auf den Indexregister-Datenbus zu schalten.

c) ADD ACC IN1: Dem Inhalt des ACC wird IN1 hinzuaddiert. Der ACCLd muss aktiv sein, dass der Inhalt des ACC auf den linken Bus kommt.

Der IN1Dd muss aktiv sein, damit der Inhalt des IN1 auf den Datenbus und dann mit aktivem DRd auf den rechten Bus gelegt wird. Nach dem Addieren in der ALU muss der ALUDd aktiv sein, damit die Daten auf den DI-Bus und zum ACC kommen.

d) -

-1

## Aufgabe 2:

a)

push():

STOREIN SP ACC 0

SUBI SP 1

pop():

LOADIN SP ACC 1

ADDI SP 1

6/6

b)

Das Betriebssystem hat Berechtigungen auf anderen Speicher (oder z.B. I/O Schnittstellen)

zuzugreifen als der Benutzer, deswegen wird der INT i Befehl benötigt.

Um zum Programmcode zurückzukehren benötigt es dann den RTI Befehl

beste Antwort bisher

aber die RETI macht das nicht, aber andere Architekturen machen sowas

## Aufgabe 3:

Da die Aufgabe seltsam gestellt ist, gibt es hierfür trotz allem volle Punkte.

brk(0x177b000)

= 0x177b000

Ändert die Datensegmentgröße für den Prozess.

6/6

openat(AT\_FDCWD, "myfile.txt", O\_WRONLY|O\_CREAT|O\_TRUNC, 0666) = 3

Öffnet die Datei in dem Dateipfad bzw. erstellt die Datei, hat also mit "fopen" zu tun und "w" aus dem Bibliotheksbefehl wird zu "O\_WRONLY".

fstat(3, {st\_mode=S\_IFREG|0644, st\_size=0, ...}) = 0

perfekt ^\_^

Gibt Informationen über die spezifizierte Datei wieder.

```
write(3, "0 1 2 3 4 5 6 7 8 9 10 11 12 13 "..., 8192) = 8192
```

```
write(3, "60 1861 1862 1863 1864 1865 1866"... , 3198) = 3198
```

Schreibt in die geöffnete Datei, also der "fprintf"-Befehl in der For-Schleife, da auch die aufsteigenden Zahlen aus der Schleife vorkommen.

```
close(3) = 0
```

Schließt die vorher geöffnete Datei, hat also mit "fclose" zu tun.

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

Öffnet die Datei in dem Dateipfad, hat also wieder mit "fopen" zu tun, "r" wird dann zu "O\_RDONLY" also "read only".

```
fstat(3, {st_mode=S_IFREG|0644, st_size=11390, ...}) = 0
```

```
read(3, "0 1 2 3 4 5 6 7 8 9 10 11 12 13 "..., 8192) = 8192
```

```
read(3, "60 1861 1862 1863 1864 1865 1866"... , 8192) = 3198
```

```
read(3, "", 8192) = 0
```

Liest die Datei in den Puffer ein, muss also mit "fscanf" zu tun haben.

```
close(3) = 0
```

Schließt die vorher geöffnete Datei, hat also wieder mit "fclose" zu tun.

```
exit_group(42) = ?
```

Schliesst alle Threads des Prozesses und gibt den Rückgabewert zurück.

```
+++ exited with 42 +++
```

Der Rückgabewert des Programms.