Betriebssysteme

Übungsblatt 2

19.5/20

sehr gut Arbeit n_n

Micha Erkel Felix Ruh

schön auf den Punkt gebracht
ohne rießige Textberge,
weiter so ^_^

8/8

Aufgabe 1

a) **STOREIN ACC SP i**
ACCLd (legt ACC auf die linke Seite L der ALU)
IRd (um i auf die rechte Seite R der ALU zu legen)
ALUAd (legt die berechnete Adresse auf den Adressbus A)
SPDd (legt den Inhalt von SP auf den Datenbus D)
ASMd (gibt die Daten vom Adressbus weiter in den Speicher)

b) **MOVE IN2 ACC**
Man muss mindestens 2 Treiber verwenden:
IN2Dd (legt den Inhalt von IN2 auf den Datenbus D)
DDid (verbindet den Datenbus D mit dem internen Datenbus DI)

c) **ADD ACC IN1**
IN1Dd (legt IN1 auf den Datenbus D)
DRd (legt IN1 auf die rechte Seite R der ALU)
ACCLd (legt ACC auf die linke Seite L der ALU)
ALUDid (legt das Ergebnis der ALU auf den internen Datenbus DI)

d) **LOADI D i**
Der Befehl lädt die Zahl i in das Zielregister D, diese steht jedoch nur im Instruktionsregister I. Die einzige Möglichkeit, i auf DI bzw. in das Register D zu legen, verläuft über die ALU. Damit i sich dabei nicht verändert muss auf der linken Seite L der ALU 0Ld anliegen.

Aufgabe 2

6/6

a) **push():**
STOREIN SP ACC 0
SUBI SP 1

pop():
LOADIN SP ACC 1
ADDI SP 1

b) Grundsätzlich ist die Situation folgende. Das Betriebssystem löst ein Interrupt aus und versucht diesen mit einem zusätzlichen Programm zu beheben. Dieses kann nur entweder intern, in dem gerade ausgeführten Programm oder extern, in einem gesonderten Register, enthalten sein.

es braucht keinen JUMP, es geht auch das PC-Register direkt zu manipulieren

Falls es intern wäre, müsste es mit einem Jump-Befehl erreicht werden. Dies ist allerdings nicht möglich, da der Jump-Befehl eine feste Schrittzahl für den Sprung braucht, das Programm sich aber an jedem Punkt befinden könnte. Es müsste die Schrittzahl für jeden Fall individuell bestimmt werden. Das wiederum erfordert aber ein Programm zur Abstandsbestimmung -> unmöglich.

Falls es extern wäre, müsste entweder in das entsprechende Programm gesprungen - wofür es keinen Befehl gibt - oder in das laufende Programm hinein geladen werden. Dies würde mit einer extra while-Schleife bewerkstelligt, was allerdings höchst umständlich und nur wenig hilfreich wäre -> also auch unmöglich.

Die einfachste Lösung ist also ein Befehl, welcher in das zusätzliche Programm hinein bzw. wieder heraus springt => INT i / RTI.

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

5.5/6

Aufgabe 3

Der Befehl **strace ./test** bewirkt, dass das Programm test.c ausgeführt wird und jeder 'system call', sowie der Rückgabewert im Terminal ausgegeben wird.

Befehl fopen löst die systemcalls openat(...) = 3 und fstat(3, ...) aus.
Befehl fprintf löst den systemcall write(3, ...) aus.
Befehl fclose löst den systemcall close(3) aus.

Befehl fopen löst die systemcalls openat(...) = 3 und fstat(3, ...) aus.
Befehl fscanf löst den systemcall read(3, ...) aus.
Befehl fclose löst den systemcall close(3) aus.

eigentlich sollten Beschreibungstexte was welcher Befehl macht dazugeschrieben werden. Das ist allerdings meiner Meinung nach Zeitverschwendung, daher ziehe ich dafür keine Punkte ab, da das zurodnen ja voraussetzt, dass ihr irgendwie verstanden habt, was die Systemcalls ungefähr machen

-0.5 exitgroup fehlt

Begründung: Die Informationen in den man-pages, sowie die Namen lassen eine Zusammengehörigkeit erahnen.