

# Grundlagen der Betriebssysteme

## Blatt 03

### Gruppe 055

Marco Deuscher  
Ibrahim Hasan

Mai 2019

## 1 Befehlsbearbeitung

(a)

1. Initialer Zustand: R0=0xe6 R1=0x04 PC=0x00
2. Ausführen von 0x00: R0=0xa4 R1=0x04 PC=0x04
3. Ausführen von 0x04: R0=0xa4 R1=0x04 PC=0xa4
4. Ausführen von 0xa4: R0=0x02 R1=0x04 PC=0xa8
5. Ausführen von 0xa8: R0=0x02 R1=0x02 PC=0xac
6. Ausführen von 0xac: R0=0x02 R1=0x02 PC=0xb0 ⇒ Ausgabe von R1
7. Ausführen von 0xb0: R0=0x02 R1=0x02 PC=0xb4
8. Ausführen von 0xb4: R0=0x02 R1=0x02 PC=0xa8
9. Ausführen von 0xa8: R0=0x02 R1=0x00 PC=0xac
10. Ausführen von 0xac: R0=0x02 R1=0x00 PC=0xb0 ⇒ Ausgabe von R1
11. Ausführen von 0xb0: R0=0x02 R1=0x00 PC=0x08
12. Ausführen von 0x08: R0=0x02 R1=0x00 PC=0x0c ⇒ Ausgabe von R0
13. Ausführen von 0x0c: R0=0x02 R1=0x00 PC=0x10

(b) Zunächst liest der Prozessor aus dem Programm-Counter die Adresse der nächsten Instruktion. Dann liest er diese ein und interpretiert diese. Im Fall einer MOV-Anweisung erwartet er als Parameter einen Wert und ein Register. An dieser Stelle wird der Befehl ausgeführt. Nach der Ausführung wird der Programm-Counter inkrementiert (außer es handelt sich um eine Branch-Anweisung). Dannach beginnt der Zyklus von vorne mit der nächsten Instruktion.

## 2 Interrupts

(a) Es gibt drei verschiedene Arten von Interrupts. Externe Interrupts bspw. durch Peripheral-Devices, Hardwaretimer etc. Bei internen Interrupts handelt es sich um interne Ausnahmen, bspw. Division mit 0 oder ein Zugriffsfehler bzw. Aufruf ohne vorhandene Erlaubnis.

Die dritte Interruptart sind die user/software Interrupts auch Traps genannt. Diese werden immer aufgerufen, wenn ein Übergang vom Usermode in den Kernelmode von Nöten ist. Also beispielsweise um etwas auszugeben oder auf ein externes Gerät zuzugreifen.

In dem in Aufgabe 1 behandelten Programm kommen mit Sicherheit Traps vor, da etwas ausgegeben werden soll und dafür wird der Print-Systemcall benötigt.

(b) Wenn ein Interrupt eintritt sichert die CPU ihre Register, i.d.R. werden diese auf den Stack gepusht. Der Programm-Counter wird so aktualisiert, dass als nächstes die ISR ausgeführt wird. Es wird hiermit der Wechsel vom Usermode in den Kernelmode vollzogen. Im Kernelmodus wird dann der Interrupt entsprechend behandelt. Nach der Behandlung des Interrupt werden die alten Registerwerte incl. des Programmcounters wieder eingetragen und die Programmausführung geht an dem Punkt weiter, an welchem sie durch den Interrupt unterbrochen wurde.

(c) Es muss die Flag gecleared werden, welche den entsprechenden Interrupt signalisiert um sicherzustellen, dass der Interrupt nicht doppelt aufgerufen wird. Nach der ISR sollte noch RTI (Return from Interrupt) aufgerufen werden um den vorherigen Zustand wiederherzustellen.

(d) Bei dem in c) auftretenden Interrupt handelt es sich um einen external Interrupt oder einen user Interrupt. Ein external Interrupt kann jeder Zeit auftreten bspw. durch Peripherie-Geräte. Hier ist allerdings ein user Interrupt wahrscheinlicher, da zuvor ein Print-Syscall gemacht wurde und dieser nun behandelt wird.

(e) Die dritte Art von Interrupt welche in der Vorlesung behandelt wurde ist der interne Interrupt. Dieser tritt in Fehlersituationen auf. Beispiele hierfür wären die Division durch 0 oder der Versuch einen Kernelbefehl im Usermode auszuführen.