

Grundlagen der Betriebssysteme

Tim Luchterhand, Paul Nykiel (Gruppe 017)

9. Juli 2018

1 Befehlsabarbeitung

(a) Der Prozessor wiederholt jeden Befehlszyklus folgende Aufgaben:

- a) Lade Befehlsregister aus PC (in Instruktionsregister)
- b) Interpretiere den Befehl
- c) Führe den Befehl aus
- d) PC inkrementieren

(b)

| Befehl | R ₀ | R ₁ | PC |
|--------|----------------|----------------|----|
| | e6 | 04 | 00 |
| 00 | a4 | 04 | 04 |
| 04 | a4 | 04 | a4 |
| a4 | 02 | 04 | a8 |
| a8 | 02 | 02 | ac |
| ac | 02 | 02 | b0 |
| b0 | 02 | 02 | b4 |
| b4 | 02 | 02 | a8 |
| a8 | 02 | 00 | ac |
| ac | 02 | 00 | b0 |
| b0 | 02 | 00 | 08 |
| 08 | 02 | 00 | 00 |
| 0c | 02 | 00 | |

2 Interrupts

- (a) Es kommt ein Syscall vor (durch Aufrufen des Stop Befehls bzw. des Ausgabe-Befehls), um dem Betriebssystem zu signalisieren, dass das Programm fertig ist bzw. eine Ausgabe zu tätigen.
- (b) Alle Register (auch PC und CCR) werden in Registern gesichert. Dann wird in den Kernel-Mode gewechselt und der PC für die ISR geladen.
- (c) Vor ausführen des Interrupts muss das passende Interrupt-Flag gelöscht werden (Befehl $5C_{16}$), damit der Interrupt nach ausführen nicht sofort erneut ausgeführt wird.

Nach ausführen des Interrupts (Befehl 64_{16}) muss der Befehl RTI aufgerufen werden, um den ursprünglichen Zustand wiederherzustellen.

- (d) Der Interrupt vor $B0_{16}$ könnte einerseits ein externer Interrupt sein (ausgelöst durch Timer oder Peripherie), andererseits ein User-Interrupt (ausgelöst durch das Aufrufen des Ausgabe-Befehls).
- (e) Außer Trap und externem Interrupt gibt es noch den internen Interrupt, der bei internen Fehlern/Exceptions (z.B. Division durch Null) auftritt.