

3.

Was ist präemptives Scheduling?

Scheduling: CPU-Zuteilungsplan

präemptiv: vorbeugend, zuvorkommend

präemptives Scheduling: Scheduler entrißt einem Prozess die CPU und gibt sie einem anderen Prozess (vorausgeplant)

Realisiert durch regelmäßiges Entreißen der CPU von einem Prozess und zuteilen per Algorithmus

Wie quasiparallele Abarbeitung Prozesse?

- durch regelmäßiges Entreißen in kleinen, vom Benutzer unbemerkten Zeitabständen (z.B. 20 ms)

Beispiel nebenläufige Prozesse, gemeinsame Daten

Bankkonto: zwei Leute heben gleichzeitig ab:

- problematisch, da nur ein Automat Geld gibt, da zwei Prozesse gleichzeitig aufgerufen werden und Geldauszahlung Ende für beide ist
- Lösung: gleichzeitige Prozesse vermeiden

Programmable Interval Timer

PIT

- unterbricht regelmäßig Prozess

Wie funktioniert es?

- generiert Output-Signal, wenn Zeit um  
→ Unterbrechung

Konfigurationssequenz

- 4 Parts:
- 0x40 Counter-Register Channel 0
  - 0x41 Counter-Register Channel 1
  - 0x42 Counter-Register Channel 2
  - 0x43 Initialisierung

Frequenz: 1153182 Hz  $\approx 1,2$  MHz

Interrupt:

- Unterbrechung (asynchron)
- eigentliche Unterbrechungen
- von Geräten ausgelöst (nicht Programm)
- Brücke zwischen Hardware und Software
- dadurch können Geräte mit Steuerprogrammen kommunizieren

## Interrupts unterscheiden

- Interrupt - Controller - Baustein

- 256 verschiedene Interrupt-Vektoren

0: divide error

1: debug exception

2...19: Hauptsächliche Programmfehler

20...31: Intel Reserved

32...255: Geräte - Interrupts

(im Realmode mit 4 multiplizieren)

- Interrupttabelle

## Interrupt behandeln

- 1. Gerät setzt Unterbrechung über Signalleitung (PIT) an

- 2. CPU wechselt Systemmodus (Systemadressraum)

- 3. CPU speichert Wert von Programm Counter (PC) und Statusregister auf Interrupt-Stack

- 4. CPU indiziert Vektortabelle mit Unterbrechungsummer

-5. springt an Adresse Handler

Können Interrupts während Interrupts auftreten?

JA

- bei Gleichzeitigkeit der mit höherer Priorität zuerst (neuer, bei gleich)

A: 10 ms      B: 100ms:

A hohe, B niedrige Priorität:

a, a, a, a, a, a, a, a, a, b

A, B gleich:

a, a, a, a, a, a, a, a, a, b

B hohe, A niedrige Priorität

b, a

- wahrscheinlich hat A mit kürzerer Zeit immer höhere Priorität