

## **Das Interruptsystem**

Eine Unterbrechung (Interrupt) erzwingt den Ablauf eines bestimmten Programms (des Unterbrechungsbehandlers) anstelle des gerade laufenden Programms. Ist die Behandlung der Unterbrechung beendet, kann das unterbrochene Programm fortgesetzt werden.

*Hierzu ist es erforderlich, den Kontext des Programms (bzw. den aktuellen Maschinenzustand) zu retten und wieder einzustellen (Save/Restore).*

Unterbrechungsauslösung:

- durch externen Bedingungen (Interrupteingänge)
- durch interne Bedingungen (Ausnahmen, Maschinenfehler, Zeitgeber usw.)

Mindest-Voraussetzungen in der Hardware:

- Bereitstellen der Eintrittsadresse,
- Retten des aktuellen Befehlszählers,
- Verhindern weiterer Unterbrechungen.

Es genügt eine einzige Eintrittsadresse, wenn die Ursache der jeweiligen Unterbrechung programmseitig abgefragt werden kann.

Unterbrechungsauslösung erfordert typischerweise hardwareseitige Vorbereitung (Interruptcontroller).

Weitergehende Vorkehrungen zur Kontextumschaltung sind selten (Registerfenster, umschaltbare Registersätze, fest zugeordnete Speicherbereiche). Nostalgische Beispiele: RCA Spectra 70, Zilog Z 80.

Der wesentliche Kennwert: die *Latenzzeit* - von der Auslösung bis zum ersten Befehl des Behandlers.

*Was - u. a. - die Latenzzeit verlängert:*

- lange dauernde Maschinenbefehle,
- komplexe Systemumgebungen.

Lange dauernde Maschinenbefehle entweder in sich unterbrechbar machen (IBM /370, Motorola 68k ) oder in eine Schleife aus (wenigstens) zwei Befehlen auflösen (Intel).

*Tricks:*

- fest zugeordnete Rettungsbereiche,
- Interrupthandler stets resident halten (vom Paging ausschließen),
- System von Grund auf entsprechend auslegen.