

Übung 10

1. Erklären Sie den Unterschied zwischen Interrupts und Polling.

Bei einem Interrupt melden sich die externen Geräte bei der CPU, um ihr mitzuteilen, wenn ein Event sofort behandelt werden muss (z.B. User bewegt die Maus). Dabei wird das aktuell laufende Programm unterbrochen und der Interrupt wird durch ein Unterprogramm (Interrupt-Service-Routine) abgearbeitet.

Im Gegensatz dazu fragt die CPU beim Polling ständig die Geräte nacheinander ab, ob diese Events haben, die zu behandeln sind. Die Abfrage macht also beim Polling den Programmablauf aus. Polling wird von manchen Geräten zum Beispiel beim Anschließen an USB-Ports genutzt.

2. Welche Varianten beim Interruptpolling gibt es?

Beim Interruptpolling gibt es Varianten ohne Prioritäten und mit Prioritäten.

Beim Polling ohne Prioritäten geht die CPU die Liste von Geräten durch und behandelt den Interrupt vom jeweiligen Gerät, wenn vorhanden. Anschließend geht die CPU von diesem Punkt aus die Liste weiter durch.

Beim Polling mit Prioritäten geht die CPU ebenfalls eine Liste von Geräten durch. Allerdings sind diesmal die Geräte nach ihrer Priorität sortiert. Wenn die CPU den Interrupt eines Gerätes behandelt hat, geht sie auch nicht weiter von diesem Punkt durch die Liste, sondern startet wieder am Anfang bei dem Gerät mit der höchsten Priorität.

Polling ohne Prioritäten

- CPU geht Liste von Geräten durch
- wenn etwas behandelt wird und CPU fertig ist geht sie von diesem Punkt aus weiter die Geräte durch

Polling mit Prioritäten

- CPU geht durch Liste von einem bestimmten Punkt aus
- wenn CPU etwas behandelt hat, geht sie nicht von der Position aus weiter durch die Liste der Geräte, sondern startet wieder am Anfang, da dort die Geräte mit der höchsten Priorität sind
- Geräte werden in der Reihenfolge ihrer Priorität abgefragt
- Bsp. Daisy Chaining

3. Erklären Sie kurz „IVT“ und „ISR“:

- a. Was sind „IVT“ und „ISR“?
- b. Was steht jeweils drin?
- c. Wer legt sie an?

IVT steht für "Interrupt Vector Table". Sie enthält die Startadressen der ISR's. Eine Interruptquelle sendet der CPU eine Interrupt-Nummer, die auf einen Eintrag in der IVT zeigt. Die ISR mit der Adresse, die in dem Eintrag steht, wird dann gestartet und behandelt den Interrupt. Das Betriebssystem beschreibt und liest die IVT. Die Startadresse der IVT ist vom Prozessor vorgeschrieben. (Z.B. 0000:0000 auf 80x86 Prozessoren)

ISR steht für "Interrupt Service Routine". Wie eben bereits angedeutet, ist eine ISR verantwortlich für die Abarbeitung eines Interrupts. Dabei gibt es verschiedene ISR's für die verschiedenen Interrupts. Eine ISR ist dabei einfach ein sehr kurzes Unterprogramm. Besonders wichtig ist hier im Vergleich zu anderen Unterprogrammen, dass das Prozessor Status Word (PSW) und die Register, die verändert werden, vor Behandlung des Interrupts gesichert werden müssen. Das Betriebssystem entscheidet, wo die ISR's abgelegt werden.

IVT

- Interrupt Vektor Table
 - enthält die Startadressen der ISR's
 - Interruptquelle sendet eine Interrupt Nummer, die auf einen Eintrag in der IVT zeigt
 - die ISR, mit der Adresse, die in dem Eintrag steht wird dann gestartet und behandelt den Interrupt
 - Interrupt Liste kann fest codiert aber auch veränderbar sein
 - werden vom Hersteller angelegt des Prozessors angelegt -> in Handbuch steht die Liste der Interrupts und was verändert werden darf (wenn überhaupt)
 - Angelegt vom Betriebssystem
- <https://github.com/torvalds/linux/blob/master/arch/ia64/kernel/ivt.S>

ISR

- Interrupt Service Routine
 - verantwortlich für Abarbeitung eines Interrupts -> gibt verschiedene ISR's für verschiedene Interrupts
 - müssen vor behandlungs des Interrupts das PSW und Register, die verändert werden, sichern
 - angelegt vom Betriebssystem
- <https://unix.stackexchange.com/questions/5788/how-is-an-interrupt-handled-in-linux>

4. Kann eine Interruptbehandlung von einem neuen Interrupt unterbrochen werden?

Es kommt drauf an. Theoretisch können Interrupts von anderen Interrupts unterbrochen werden. Allerdings gibt es bei jedem interrupt eine Phase, in der sie nicht unterbrochen werden dürfen. Nämlich wenn die ISR gestartet wird und gerade das Program Status Word (PSW) und die Register, die die ISR während der Abarbeitung verändert, vorher sichert. Zusätzlich kann bewusst entschieden werden, dass ein Interrupt nicht von einem anderen Interrupt unterbrochen werden darf. Dies kommt vor wenn Interrupts ebenfalls Prioritäten zugewiesen bekommen. In diesem fall kann ein Interrupt mit höherer Priorität nicht von einem Interrupt mit niedrigerer Priorität unterbrochen werden.

5. Zeichnen Sie eine Daisychain mit 5 I/O-Geräten.

6. Warum sind Daisy-Chains sinnvoll?

- Prioritäten werden berücksichtigt

Daisy-Chains haben den Vorteil, dass sie die Prioritäten der verbundenen Geräte berücksichtigen. Zudem bieten sie ein sehr einfaches Modell für die Umsetzung von Polling in der Hardware, da Softwarelösungen sehr Zeitaufwendig sind.

7. Beschreiben Sie den Ablauf einer Interrupt-Behandlung nach dem Daisy-Chain-Verfahren.

Beim Daisy-Chain-Verfahren sind die Geräte in Prioritätsreihenfolge mit dem Prozessor verbunden. Es gibt eine Interruptleitung, mit der alle Geräte verbunden sind. Wenn eine der Quellen einen Interrupt hat, sendet diese ein Signal über die Interruptleitung. Der Prozessor reagiert darauf, indem er kontrolliert, ob er in diesem Moment unterbrochen werden darf. Wenn dies der Fall ist, weiß der Prozessor aber noch nicht, wer den Interrupt gemeldet hat, da alles über eine Leitung kommt. Der Prozessor geht nun die Geräte nach Priorität von vorne nach hinten durch und fragt die Geräte, ob sie den Interrupt gemeldet haben. Der Prozessor gibt dabei dem ersten Gerät die Erlaubnis, die Interrupt-Nummer auf den Bus zu legen, wenn dieses Gerät einen Interrupt hat. Wenn es keinen hat, gibt dieses erste Gerät die Erlaubnis weiter an das nächste Gerät in der Reihe und immer so weiter.

8. Sind Daisy-Chains fair?

Das kommt auf die Definition von Fairness an. Da Daisy-Chains nach Priorität sortiert sind, werden manche Geräte bevorzugt behandelt. Daher sehen wir Daisy-Chains nicht als "fair" an.