

Was ist präemptives Scheduling und wie kann es realisiert werden?

Im Gegensatz zum kooperativen Scheduling wird die CPU nicht von den Prozessen "freiwillig" untereinander weitergereicht, sondern die CPU wird den Prozessen entzogen bzw. übergeben. Dies wird über Interrupts realisiert und es muss folglich eine Synchronisation stattfinden.

(Siehe Scheduling Folie 16)

Wie kann man eine quasiparallele Abarbeitung von Prozessen durch ein präemptives Scheduling erreichen?

Bei einer Quasiparallelität gibt es mehr Prozesse als Prozessoren. Sie kann über eine Zeitscheibenvergabe realisiert werden. [Je kürzer die Zeitabschnitte, die ein Prozess zur Ausführung erhält, desto näher kommt es der wirklichen Parallelität.]

Was wäre ein passendes Beispiel für einen nebenläufigen Zugriff auf gemeinsame Daten?

In der realen Welt wäre ein Beispiel eine Veranstaltung mit mehreren Eingängen, bei denen die Anzahl der Besucher beschränkt ist. Hier müssen alle Sicherheitsleute auf dem gleichen Stand der Anzahl der aktuell auf der Veranstaltung befindlichen Menschen sein.

Was ist ein Programmable Interval Timer?

Ein PIT ist ein Baustein, meist in Form des Typs "8253" oder "8254", der für eine zeitliche Kontrolle von Prozessen genutzt wird. Es werden hierfür einer oder zwei PITs verwendet.

Wie funktioniert der Programmable Interval Timer (PIT)?

Der PIT ist mit anderen Bestandteilen des Computers verkabelt (u. a. mit dem PIC). Der bzw. die PIT(s) können durch je vier Ports angesprochen werden können. Mithilfe eines Steuerworts, das an dem Kontrollport (0x43) übergeben wird, werden Einstellungen wie der Zählmodus übergeben. Dann können vom PIT Interrupts ausgelöst werden.

Wie kann man diesen konfigurieren (Beschreibt eine Konfigurationssequenz!)?

Zunächst muss entschieden werden, in welchem Zeitintervall die Interrupts ausgegeben werden. Darauf aufbauend wird dann ermittelt, wie viele Ticks notwendig sind, um dieses Intervall zu erfüllen. Das Kontrollwort wird danach in den Kontrollport (0x43) übergeben, die benötigte Anzahl Ticks wird dem Datenport übergeben (0x40), wozu ggf. (bei 16 Bit) mehrere Anläufe benötigt werden.

Was ist ein Interrupt?

Interrupts sind asynchrone Unterbrechungen (Traps: synchron), die von Geräten bzw. Hardwarebestandteilen ausgelöst werden und bilden eine Brücke zur Software. Sie können jederzeit auftreten und sind nicht zeitlich, aber funktional vorhersagbar.

(Unterbrechungen Folie 2, 6, 7)

Wie kann die CPU Interrupts unterscheiden?

Wie werden Interrupts behandelt?

1. Kontextwechsel zur Behandlungsroutine (HW | Maschinenebene)
2. Unterbrechungsbehandlungsroutine (SW | Betriebssystemebene)
 - Interrupt am PIC quittieren
 - Gerätezustand überprüfen
 - Wartende Prozesse aufwecken
3. Kontextwechsel zum unterbrochenen Programm (HW | Maschinenebene)

Können neue Interrupts während einer Interruptsbehandlung auftreten?

Das Zulassen von Interrupts kann verboten oder erlaubt werden. Je nachdem sind Interrupts während einer Interruptsbehandlung möglich oder unmöglich.

Wenn ja, werden diese dann umgehend behandelt?

Im Allgemeinen nicht. Interrupts werden hierbei geschachtelt und nach einer Prioritätssortierung abgearbeitet. Eine Interruptbehandlung kann unterbrochen werden, wenn dies durch einen Interrupt mit höherer Priorität geschieht.

Wenn nein, werden die blockierten Interrupts durch das Gerät später nochmal ausgelöst?

Wird das Auslösen von Interrupts wieder reaktiviert, so können wieder Interrupts auftreten. Interrupts die während ausgeschalteter Interruptbehandlung auftreten, sind verloren.

Zwei Prozesse A und B sollen in einer Endlosschleife immer wieder ihren Namen ausgeben. Diese Ausgabe dauert 10 Takte. Wenn Prozess A eine Zeitscheibe von 10 Takten und Prozess B eine Zeitscheibe von 100 Takten zugewiesen wird, welche Ausgaben sind zu erwarten?

A BBBBBBBBBBB A BBBBBBBBBBB A BBBBBBBBBBB ...