
Grundlagen der Betriebssysteme

Praktische Einführung mit Virtualisierung

Stefan Bosse

Universität Koblenz - FB Informatik

Zeitmanagement

Timer

In einem Betriebssystem müssen häufig zeitliche Abläufe synchronisiert werden.

- Zeitauflösung für Ereignisse könnte Mikro- oder Millisekunden sein
 - Eine solch hohe Auflösung ist aber meistens nicht erforderlich
 - Beispiel: Ein Mikroprozessr arbeitet mit einem Kerntakt von 72MHZ. Dann beträgt die minimale Ausführungszeit einer Maschineninstruktion ca. 20 ns, typischerweise benötigt ein Funktionsaufruf einer kleinen Funktion ca. 50 Instruktionen, was 1 μ s entspricht.
- Daher wird in den meisten Betriebssystemen eine virtuelle Zeiteinheit genutzt, die Ticks.
 - Ein Tick entspricht Zeitintervallen typischerweise im Bereich 10-100 ms
 - Prozessscheduling wird bei Zeitscheibenverfahren in Ticks behandelt, d.h. eine Zeitscheibe häufig ein Tick

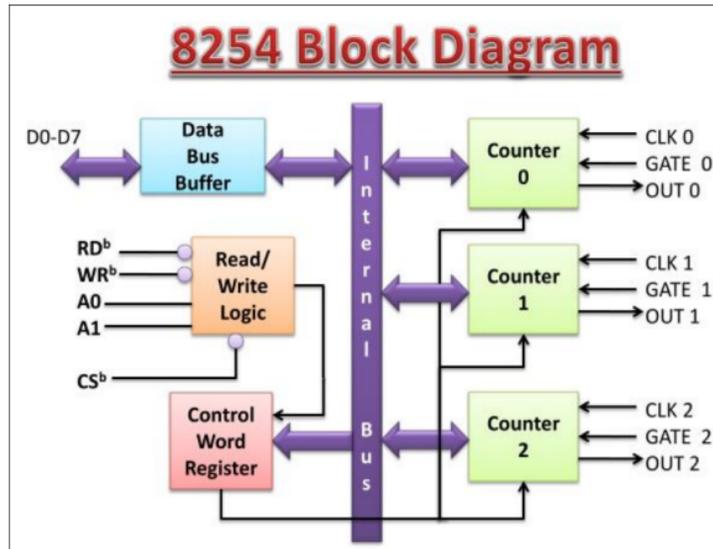
Programmable Interval Timer



Für die "Erzeugung" von Ticks werden Intervalltimer verwendet, also binäre Zähler. Diese stellen die Systemuhr des betriebssystem zur Verfügung. Nicht zu verwechseln mit der Real Time Clock (RTC), die eine nicht flüchtige Echtzeit in Sekunden usw. ausgibt.

Der Programmable Interval Timer (PIT) war ursprünglich ein spezieller Baustein (Chip) im IBM-PC, der für die zeitliche Steuerung von Prozessen sorgte. Die am häufigsten verwendeten Timer-Chips waren die Typen 8253 und 8254 von Intel mit drei 16-Bit breiten Zählern. [Wikipedia]

Programmable Interval Timer



[A. Sahu, Dept. of Comp. Sc. & Eng. IIT Guwahati]

Abb. 1. Die Architektur eines PITs (8254)

- Die Ausgänge der drei Zähler können/sind mit dem Interruptcontroller verbunden

Programmable Interval Timer

- Der PIT wird, unabhängig von der Rechengeschwindigkeit, stets im Takt von (idealerweise) 1,193181818... MHz betrieben; das ist $\frac{1}{3}$ der Farbträger-Frequenz des NTSC-Fernsehsystems!
- In den ersten PCs von IBM wurde diese merkwürdig anmutende Frequenz gewählt, da für die Einbindung einer CGA-Grafikkarte, die auch Fernsehgeräte als Monitor verwenden konnte, ohnehin ein Quarz mit einem Vielfachen des Farbträgers vorhanden sein musste.
- Aus diesem Quarz konnte die PIT-Frequenz dann leicht mittels Frequenzteilung gewonnen werden; diese Vorgehensweise war günstiger als der Einsatz weiterer Quarze, da mit nur einem Quarz in der Schaltung keine störenden Schwebungen das Bildsignal beeinträchtigen können, die dann mit weiterem Schaltungsaufwand herausgefiltert werden müssten.
- Auch der CPU-Takt der ersten PCs von 4,772727 MHz ergab sich aus einer solchen Erwägung, er betrug $\frac{4}{3}$ der Farbträgerfrequenz.

Programmable Interval Timer

Der PIT hat verschiedene Betriebsarten:

1. Standard Interrupt Modus: Interrupt bei Erreichen eines einstellbaren Zählwerts (Über- oder Unterlauf)
2. Periodischer Modus: Erzeugung eines Rechtecksignals
3. Periodischer Modus: Erzeugung von Pulssignalen
4. Software Modus: Softwaresteuerung des Zählers

Programmable Interval Timer

- Im Standard-Modus zählt der Timer vom Startwert (maximal 65535) auf 0 hinunter und fängt beim folgenden Überlauf wieder von vorne an.
- Bei jedem solchen Überlauf sendet er ein Signal, den Timer-Interrupt (IRQ0), an den Prozessor.
- Wenn man die Taktfrequenz durch die Takte bis zum erneuten Überlauf dividiert, erhält man die Frequenz, mit der der IRQ0 erzeugt wird.
- Unter MS-DOS und älteren Windows-Versionen bis einschließlich Windows 98 und Windows ME wurde der Maximalwert 65535 in den Timer geladen, man erhält somit $1193181,8181... \text{ Hz} / 65536 = \text{ca. } 18,2065 \text{ Hz}$.
- Es kommt auf diese Art ein Intervall von ca. 55 ms zwischen den Aufrufen des IRQ 0 zustande.
- Modernere Betriebssysteme (Linux, Windows NT und Abkömmlinge) verwenden kürzere Intervalle zwischen 1 und 10 ms.

Programmable Interval Timer

Mehr Info gibt es hier:

https://wiki.osdev.org/Programmable_Interval_Timer

Ticks,Frequenzen, Zeiten und Timer

Folgende Parameter sind die Zutaten des Zeitmanagements im Betriebssystem:

$$\text{TIMER_FREQ} = 1193182$$

$$\text{TICKS_PER_SECOND} = 20$$

$$\text{TICKS_MILLITIME} = \frac{1}{\text{TICKS_PER_SECOND}}$$

$$\text{TIMER_COUNT} = \frac{\text{TIMER_FREQ}}{\text{TICKS_PER_SECOND}}$$

D.h. der Timer muss immer mit `TIMER_COUNT` initialisiert werden um die Vorgabe `TICKS_PER_SECOND` zu erreichen.

- Im Basekernel werden die Ticks als Clicks bezeichnet.

Ticks,Frequenzen, Zeiten und Timer



Die kleinste von Programmen (und dem Kernel selber) messbare Zeiteinheit ist `1/TICKS_PER_SECOND`, also z.B. 20 ms

- Benötigt man eine genauere Zeitmessung so müssen entweder andere Timer oder der aktuelle Timercount des PIT benutzt werden.
- Bekannt ist mindestens die Anzahl der Ticks (Clicks), dann kann man den aktuellen Zählerwert des PIT auslesen und kann die Zeit zwischen zwei Ticks interpolieren.
- Bei ca. 1 MHz Timerfrequenz ist die kleinste Auflösung 1 μ s.



Aufgabe: Implementiere im Basekernel eine neue Funktion `clock_micros` die die Systemzeit in Mikrosekunden zurück gibt.