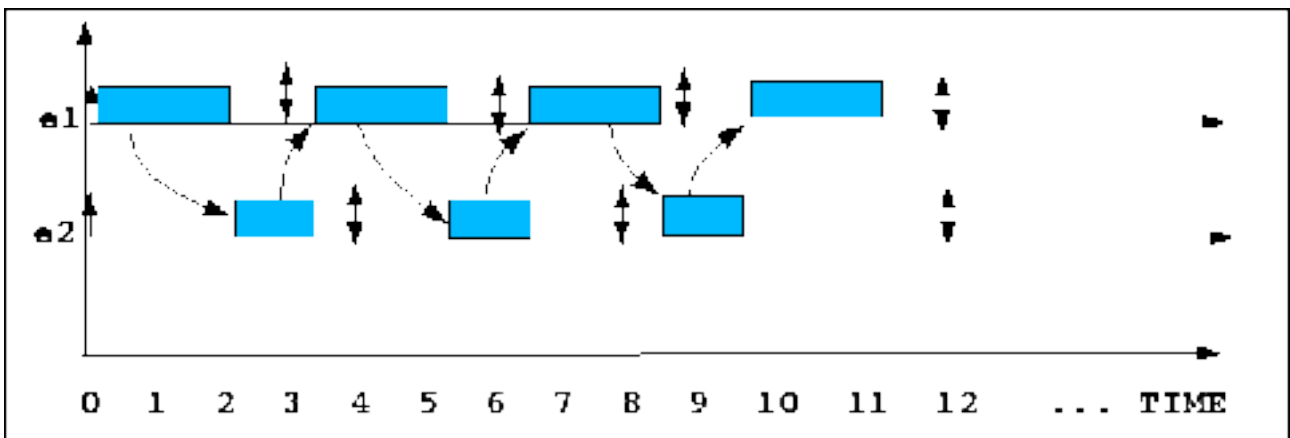
	Fachbereich 2 Studiengang: <i>Informatik</i>
	Realzeitsysteme – Übungsaufgabe Nr.2
	Letzte Änderung: 24.Oktober 2004 Seite // 3


Das Verhalten eines Realzeitsystems planen

Hintergrund

Im Rahmen der Vorlesung geht es jetzt zunehmend darum, Ereignisse, die als Input für Realzeitsysteme auftreten können, durch diese Systeme zu bearbeiten. Dazu ist es wichtig, vorab zu klären, ob diese Ereignisse bei den gegebenen Ressourcen so geplant werden können, dass Sie ihre zeitlichen Vorgaben ('Deadlines') einhalten können. Dazu muss ein Plan erstellt werden.



Obige Grafik zeigt ein Beispiel: Zwei *Ereignisse* e_1 und e_2 treten regelmässig (*periodisch*) auf. Jedes Ereignis hat eine eigene Zeitlinie. Alle Ereignisse beginnen bei der Zeitmarke 0. Ereignis e_1 hat die *Periode* $T=3$, Ereignis e_2 hat die Periode $T=4$; dies bedeutet, dass Ereignis e_1 alle 3 Zeiteinheiten (ab 0) auftritt und e_2 alle 4 Zeiteinheiten (ab 0). Das *Auftreten* (*occurrence*) eines Ereignisses auf der Zeitachse wird durch einen nach oben zeigenden Pfeil angezeigt. In diesem Beispiel wird zusätzlich angenommen, dass bei beiden Ereignissen die *relative Deadline* D mit der Periode zusammenfällt, d.h. $D = T$. Dies bedeutet, die zu einem Ereignis gehörende *Task* muss spätestens bis zur Deadline D abgearbeitet sein, also, wegen $D = T$, bis spätestens zum Auftreten des nächsten Ereignisses. Deadlines werden angezeigt durch einen nach unten zeigenden Pfeil; fallen Deadline und Periode zusammen, dann benutzt man einen Doppelpfeil. Ereignisse treten 'punktuell' auf, zu einem bestimmten Zeitpunkt t auf der Zeitachse. Wenn ein Ereignis e_i als Input eines Systems S auftritt, dann ist normalerweise festgelegt, welche *Aufgabe* (*Task*) t aufgrund des Auftretens des Ereignisses e_i ausgeführt werden soll. Nehmen wir vorläufig an, jedem Ereignis wird ein eigener Task zugeordnet, also es soll gelten: $\text{task}(e_i) = t_i$. Ferner gilt, dass jeder Task t_i eine typische *Ausführungszeit* (*execution time*) C_i besitzt.

	Fachbereich 2 Studiengang: <i>Informatik</i>
	Realzeitsysteme – Übungsaufgabe Nr.2
	Letzte Änderung: 24.Oktober 2004 Seite // 3

Setzt man nun voraus, dass man nur *ein einziges* System S hat und dieses System zur gleichen Zeit nur *einen einzigen Task* ausführen kann, dann gibt es ein Problem, sobald mehr als ein Ereignis gleichzeitig oder 'nah beieinander' auftreten. Die zugehörigen Tasks können nicht gleichzeitig ausgeführt werden.

Im obigen Beispiel treten die beiden Ereignisse e1 und e2 bei Zeitpunkt 0 gleichzeitig auf. Zu diesem Zeitpunkt gilt:

$t_1(0, 2.5, 3, 3)$

$t_2(0, 1.7, 4, 4)$


Dabei wird das Schema benutzt:

$t(r_0, C, D, T)$ mit $0 \leq C \leq D \leq T$

(mit der Bedeutung r_0 := erstmalige Auftretenszeit des Ereignisses, das Task t hervorruft; C := Ausführungszeit; D := Deadline; T := Periode).

Zum Zeitpunkt $r=0$ muss also entschieden werden, welche der beiden Tasks soll nun zuerst ausgeführt werden? Wir nehmen für diese Übung an, dass zur gleichen Zeit nur ein Task ausgeführt werden kann; der andere muss warten, bis ein schon laufender Task beendet ist. Um die Frage zu entscheiden, benötigt man eine *Entscheidungsregel*. Eine verbreitete Entscheidungsregel für *periodische* Tasks ist:

Entscheidungsregel für <i>periodische</i> Tasks (RM := Rate Monotonic)
(i) Höchste Priorität P hat der Task mit der kürzesten Periode T, also $T_i < T_j \Rightarrow P_i > P_j$ (ii) $D = T$ (iii) Die Berechnung, welcher Tasks als nächstes ausgeführt werden soll, wird immer dann vorgenommen, wenn (a) gerade ein laufender Task beendet wurde und (b) noch Tasks zur Ausführung anstehen.

	Fachbereich 2 Studiengang: <i>Informatik</i>
	Realzeitsysteme – Übungsaufgabe Nr.2
	Letzte Änderung: 24.Oktober 2004 Seite 8 // 3

Teilaufgaben

1. (Max.Pkt: 2) Überlegen Sie sich eine realistische Anwendungssituation, in der mindestens drei verschiedene periodische Ereignisse $\{e_1, e_2, e_3, \dots\}$ auftreten, auf die ein System mit unterschiedlichen Tasks $\{t_1, t_2, t_3, \dots\}$ antworten muss. Nehmen Sie auch an, dass ihr System nur einen Task gleichzeitig ausführen kann und ein laufender Task nicht unterbrochen werden darf. Als Entscheidungsregel gilt die in der Tabelle angegebene RM-Regel. Beschreiben sie die Anwendungssituation.
2. (Max.Pkt: 1) Erstellen Sie ein Zeitdiagramm wie im vorausgehenden Beispiel: pro Task eine Zeitlinie. Tragen Sie die Periode und die Deadlines für jedes Ereignis ein. Achten Sie darauf, dass ihr Zeitfenster lange genug ist. Ordnen sie die Tasks entsprechend der RM-Entscheidungsregel an.
3. (Max.Pkt: 3) Erstellen sie ein formales Konzept für ein Computerprogramm, aus dem hervorgeht, dass ihr Programm die in den Punkten (1-2) gestellten Forderungen erfüllen kann.
4. (Max.Pkt: 4) Implementieren Sie ihr Konzept mittels der Sprache C oder C++. Eingabe für den Algorithmus sind Ereignisse und zugehörige Tasks in der Form $t_1(0,2.5,3,3)$, $t_2(0,1.7,4,4)$, ... Das Format kann entweder eine Liste im ASCII-Format sein oder eine Datei im DIF-Format (Data Interchange Format); Letzteres hat den Vorteil, dass man die Daten in ein Spreadsheet eingeben kann und diese dann automatisch konvertiert werden koennen.
5. Die Ausgabe soll eine Liste von Zeitpunkten sein und die zu jedem Zeitpunkt auszuführenden Tasks, geordnet nach Priorität.

Abgabe

Die Übungsaufgabe gilt als erfüllt, wenn folgende Bedingungen erfüllt sind:

1. In der Übung liegen alle Dokumente ausgedruckt vor (vergewissern sie sich rechtzeitig, wie und wo Sie ausdrucken können)
2. Sie haben ihr Programm auf einem elektronischen Datenträger und können die Dateien von diesen Datenträgern auf ihren Zielrechner laden (Diskette oder CD-ROM)(Vergewissern Sie sich rechtzeitig, wie Sie Daten entsprechend präparieren)
3. Sie können ihren Beitrag auf dem Zielrechner und per Beamer vorführen (Bedenken sie, dass unterschiedliche Versionen von Compilern und Präsentationsprogrammen gibt. Vergewissern sie sich, dass sich ihr programm und ihr Beitrag auf dem Zielsystem so verhält, wie auf ihrem Entwicklungssystem).
4. Alle an der Erstellung der Übung Beteiligten sind anwesend und erläutern den Beitrag (Wer nicht anwesend ist bekommt keine Punkte).