

Übung zur Kernvorlesung: "Eingebettete Systeme" Übungsblatt Nr. 4

Ausgabe am 21.05.03; Bearbeitung bis 28.05.03, 12:00; Besprechung am 30.05.03

Aufgabe 1 Prioritätsvererbung

(2+5+6=13P)

a) Was versteht man unter *Prioritätsumkehr*? Wieso ist dies für Echtzeitsysteme problematisch?

b) Gegeben sind die in der untenstehenden Tabelle aufgeführten Prozesse, wobei $Priorität(P_i) < Priorität(P_{i+1})$ für alle i . Es werden Semaphore zur Synchronisation verwendet. In der Tabelle sind jeweils die Zeitpunkte relativ zum Startzeitpunkt des Prozesses angegeben, zu denen ein Prozess Semaphore freigibt oder setzt. Der Verbrauch von Rechenzeit, ohne Semaphore-Operationen zu verwenden, ist in der Tabelle mit "-" gekennzeichnet. Die Startzeitpunkte s_i der Prozesse sind relativ zu einem Zeitpunkt t , wobei gilt: $s_1=t+0$, $s_2=t+1$, $s_3=t+2$, $s_4=t+3$ und $s_5=t+4$. Es bekomme immer der rechenbereite Prozess mit der höchsten Priorität den Prozessor zugeteilt. Die Zuteilung finde nur zu Beginn einer Zeitscheibe statt. Erstellen Sie ein Zeitdiagramm für den Fall, dass keine Prioritätsvererbung angewendet wird. Wie lange wird der Prozess P_5 blockiert? Aus dem Zeitdiagramm soll ersichtlich werden, wann welcher Prozess rechnet. Kennzeichnen Sie dabei eine nicht erfolgte Zuteilung eines Semaphors s mit „P(s“, die erfolgte Zuteilung mit „P(s“. Erfolgt eine vorher blockierte Zuordnung P(s und ist der nächste Rechenschritt R, dann werde dies mit „)R“ gekennzeichnet. Vergleiche hierzu das Diagramm im Skript Seite 2-111.

Zeitpunkt	P_1	P_2	P_3	P_4	P_5
s_i+0	P(a)	P(b)	P(b)	-	P(a)
s_i+1	-	-	P(a)	-	P(c)
s_i+2	P(b)	V(b)	V(b)	-	V(a)
s_i+3	V(b)	P(c)	V(a)	-	-
s_i+4	V(a)	V(c)	-	-	V(c)

c) Erstellen Sie nun entsprechende Zeitdiagramme, falls im unter 1b) beschriebenen Beispiel die Technik der Prioritätsvererbung eingesetzt wird. Aus dem Zeitdiagramm soll ersichtlich werden, wann welcher Prozess rechnet und welche Priorität der aktuell rechnende Prozess zu Beginn der jeweiligen Zeitscheibe hat. Wie lange wird der Prozess P_5 jetzt blockiert?

Aufgabe 2 Priority Ceiling

(2+8+3=13)

- a) Was versteht man unter dem Begriff *Deadlock*? Mit welcher im Skript genannten Technik, können Deadlocks verhindert werden?
- b) Nochmals werde die Situation von Aufgabe 1b) mit den gleichen Prozessen, Startzeiten und Semaphoreoperationen untersucht. Erstellen Sie diesmal das Zeitdiagramm (vgl. Skript Seite 2-111) für den Fall, dass das Priority-Ceiling-Verfahren angewendet wird. Beachten Sie folgende beiden Ergänzungen zum im Skript angegebenen Algorithmus:
- Die jeweiligen Prozessorzuteilungen erfolgen nicht sofort, sondern erst zu Beginn der nächsten Zeitscheibe. Dabei erhält wieder der rechenbereite Prozess mit der höchsten Priorität den Prozessor.
 - bei Punkt 2 wird ein Unterfall c) ergänzt, so dass:
c) Sei P_k der Prozess, der das Semaphore s' besitzt mit $\text{ceil}(s) \leq \text{ceil}(s')$ für alle s aus lockedsem .

- $P_i = P_k$ - $P(s)$ wird beendet
 - $\text{lockedsem} = \text{lockedsem} \cup \{s\}$
 - P_i erhält Prozessor

Aus dem Zeitdiagramm soll ersichtlich werden, wann welcher Prozess rechnet, welche Priorität der aktuell rechnende Prozess hat zu Beginn einer Zeitscheibe, welchen Wert *actceil* am Ende der jeweiligen Zeitscheibe hat und welche Semaphore in der Menge *lockedsem* am Ende der jeweiligen Zeitscheibe enthalten sind. Wie lange wird der Prozess P_5 jetzt blockiert?

- c) Im Skript wird beim Priority Ceiling Verfahren angegeben, dass ein hoch priorisierter Prozess in seiner Ausführungszeit nur einmal von einem niedriger priorisierten blockiert werden könne. Belegen Sie diese Aussage! Weshalb wurde in Aufgabenteil b) der Unterfall c) in Punkt 2 des Priority Ceiling Verfahrens ergänzt? Wo spielt das bei dem Beispiel aus Aufgabenteil b) eine Rolle?