

## Übung 5

### Aufgabe 1

Gegeben seien 3 periodische Tasks  $t_1, t_2, t_3$  mit Periodendauern  $\Delta t_i$  und maximalen Ausführungszeiten  $\Delta e_i$ :

<b>Task</b>	<b><math>\Delta t_i</math></b>	<b><math>\Delta e_i</math></b>
1	3	1
2	6	2
3	18	5

- Entwickeln Sie einen brauchbaren Plan nach monotonen Raten für die Taskmenge
- Entwickeln Sie einen zyklischen Plan für die Taskmenge. (Gehen Sie davon aus, dass jede der Tasks, falls sinnvoll und erforderlich, in kleinere Teiltasks aufgeteilt werden kann.)

### Aufgabe 2

Gegeben seien 3 periodische Tasks  $t_1, t_2, t_3$  mit Periodendauern  $\Delta t_i$ , maximalen Ausführungszeiten  $\Delta e_i$  und von der Anwendung vorgegeben Prioritäten (nach "Wichtigkeit")  $p_i$ :

<b>Task</b>	<b><math>\Delta t_i</math></b>	<b><math>\Delta e_i</math></b>	<b><math>p_i</math></b>
1	100	30	3
2	6	1	2
3	25	5	1

- Beweisen Sie, dass es für die Taskmenge keinen brauchbaren Plan nach festen Prioritäten gibt.
- Ändern Sie die Prioritäten so, dass die Taskmenge einen brauchbaren Plan hat und beweisen Sie dies (ohne den Plan aufzustellen)
- Entwickeln Sie einen brauchbaren Plan für die Taskmenge mit den Prioritäten nach (b)

### Aufgabe 3

Gegeben seien 3 periodische Tasks  $t_1, t_2, t_3$  mit Periodendauern  $\Delta t_i$  und maximalen Ausführungszeiten  $\Delta e_i$  und Prioritäten  $p_i$  nach monotonen Raten:

<b>Task</b>	<b><math>\Delta t_i</math></b>	<b><math>\Delta e_i</math></b>	<b><math>p_i</math></b>
1	19	3	1
2	9	2	2
3	4	2	3

Führen Sie für diese Taskmenge (a) den LL-Test, (b) den HB-Test und (c) eine Antwortzeitanalyse durch.

#### Aufgabe 4

Gegeben seien 4 periodische Tasks  $t_1, t_2, t_3, t_4$  mit Startzeiten  $s_i$ , maximalen Ausführungszeiten  $\Delta e_i$  und Prioritäten  $p_i$ . Die Tasks benutzen zwei kritische Bereiche die durch zwei Semaphore A und B geschützt sind. Die ununterbrochene Ausführung der Tasks ist beschrieben durch entsprechende Ausführungsfolgen. Jeder Großbuchstabe in einer Anweisungsfolge entspricht einer Zeiteinheit (A=rechnet im kritischen Bereich A, B= rechnet im kritischen Bereich B, E=rechnet außerhalb des kritischen Bereichs, vgl. Skript S. 6-28ff).

Das jeweils erste Auftreten einer Aktion im kritischen Bereich wird eingeleitet durch eine *wait*-Operation (wenn der Semaphor nicht frei ist, wird die Task blockiert). Diese ist in der Ausführungsfolge explizit dargestellt durch einen Kleinbuchstaben (*a* bzw. *b*). Nach der letzten Aktion in einem kritischen Bereich findet eine *signal*-Operation statt. Diese ist in der Ausführungsfolge explizit dargestellt durch einen "gestrichenen" Kleinbuchstaben (*a'* bzw. *b'*). Eventuell auf den Semaphor wartende Tasks werden dann wieder rechenbereit. Sowohl *wait*- als auch *signal*-Operationen haben aber keine signifikante Zeitdauer.

<b>Task</b>	<b><math>p_i</math></b>	<b><math>s_i</math></b>	<b><math>\Delta e_i</math></b>	<b>Ausführungsfolge</b>
1	4	7	4	EaAAa'bBb'
2	3	2	5	EbBaAa'Bb'E
3	2	5	4	EEEE
4	1	0	5	EaAAAA'E

- Beschreiben Sie die Ausführung der drei Tasks auf einem Prozessor beim Planen nach monotonen Raten mit Verdrängen (*preemptive Multitasking*).
- Beschreiben Sie, wie sich das Verhalten unter Anwendung von *Prioritätsvererbung* ändert.