EINDHOVEN UNIVERSITY OF TECHNOLOGY

Department of Mathematics and Computer Science

Examination Real-time Architectures (2IN20)
on Wednesday, August 22^{nd} *2007, 14.00h-17.00h.*

This document contains (draft) answers.

First read the entire examination. There are 5 exercises in total. Grades are included between parentheses at all parts and sum up to 11 points. Good luck!

1. A recursive equation to determine the best-case response time of a periodic task τ_i is given by

$$x = C_i + \sum_{j<i} \left(\left\lceil \frac{x}{T_j} \right\rceil - 1 \right) C_j.$$

(a) (0.5) For which class of scheduling algorithms is this equation applicable?
Answer Fixed-priority pre-emptive scheduling (FPPS).

(b) (0.5) Give at least four assumptions that need to hold to use this equation.
Answer The equation is applicable for FPPS when deadlines are at most equal to periods, and task τ_j has a higher priority than task τ_i if and only if $j < i$. See Section 4.1. of the book for general assumptions for periodic task scheduling.

(c) (1.0) Is the value ι_i given by

$$\iota_i = \frac{C_i}{1 - U_{i-1}},$$

where $U_{i-1} = \sum_{j<i} \frac{C_j}{T_j}$, an appropriate initial value for the iterative procedure to determine the best-case response time of τ_i? Motivate your answer.
Answer Yes it is.
The best-case response time BR_i is the largest positive value satisfying the recursive equation, and the iterative procedure therefore starts with an upper bound. Hence, we have to prove that ι_i is an upper bound for BR_i, i.e. $BR_i \leq \iota_i$. Intuitively, $1 - U_{i-1}$ is the percentage of the processor that is available to τ_i *on average*. The value ι_i therefore represents an average response time of τ_i, which is larger than or equal to the best-case response time. More formally, we derive

$$\begin{aligned} BR_i &= C_i + \sum_{j<i} \left(\left\lceil \frac{BR_i}{T_j} \right\rceil - 1 \right) C_j \\ &\leq C_i + \sum_{j<i} \frac{BR_i}{T_j} C_j \\ &= C_i + BR_i \cdot \sum_{j<i} \frac{C_j}{T_j} = C_i + BR_i \cdot U_{i-1}. \end{aligned}$$

Hence, for $U_{i-1} < 1$, we get $BR_i \leq \frac{C_i}{1-U_{i-1}}$.
Note: This question is similar to 1(c) of the examination of Wednesday, August 30^{th}, 2006.

2. The book of Buttazzo illustrates three anomalies expressed by a theorem of Graham for an optimally scheduled task set on a multiprocessor with some priority assignment, a fixed number of processors, fixed computation times, and precedence constraints, i.e. the schedule length can increase when (a) the number of processors is increased, (b) the computation times are reduced, and (c) the precedence constraints are weakened.

(a) (1.0) Can the schedule length also increase when the speed of a processor is increased? If no, explain why. If yes, give an example using the precedence graph shown in Figure 1 scheduled on a parallel machine with three processors.

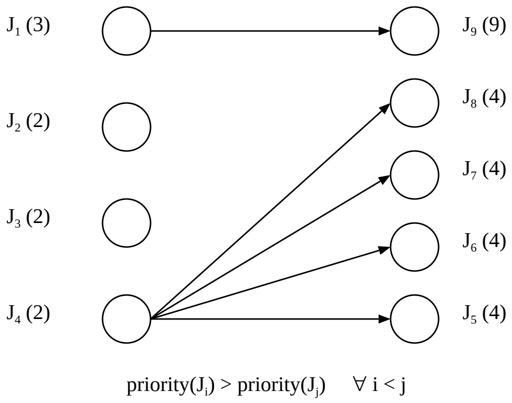


Figure 1: Precedence graph of the task set J; numbers in parentheses indicate computation times.

Answer The answer is yes. The optimal schedule of task set J with a length of 12 is shown in Figure 2.17 in the book of Buttazzo. The schedule can increase when jobs J_5 till J_8 start *before* job J_9. This can be accomplished by increasing the speed of either processor P_2 or P_3 with, for example, a factor 4/3. The resulting schedule has a length of 16; see Figure 2 for an increase of the speed of processor P_2. Note that the computation times of J_2, J_4, J_6, and J_8 on P_2 become 1.5, 1.5, 3, and 3, respectively.

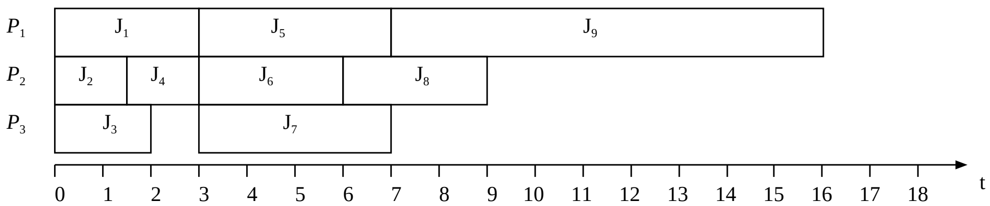


Figure 2: Schedule of task set J when the speed of processor P_2 is increased with a factor 4/3.

Although an increase of the speed of processor P_1 with a factor 5/2 also allows jobs J_5 till J_8 to start before job J_9, the resulting schedule reduces, i.e. it now has a length of $8\frac{4}{5}$; see Figure 3. Note that the computation times of J_1, J_4, J_5, J_8, and J_9 on P_1 become $1\frac{1}{5}$, $\frac{4}{5}$, $1\frac{3}{5}$, $1\frac{3}{5}$, and $3\frac{3}{5}$, respectively.

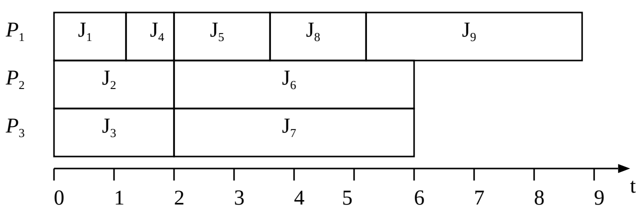


Figure 3: Schedule of task set J when the speed of processor P_1 is increased with a factor 5/2.

(b) (1.5) Show by means of an example that a feasible schedule of a set of tasks with fixed computation times scheduled by FPPS on a *single* processor can become infeasible when the computation time is reduced. *Hint:* Assume a fixed phasing and resource sharing.

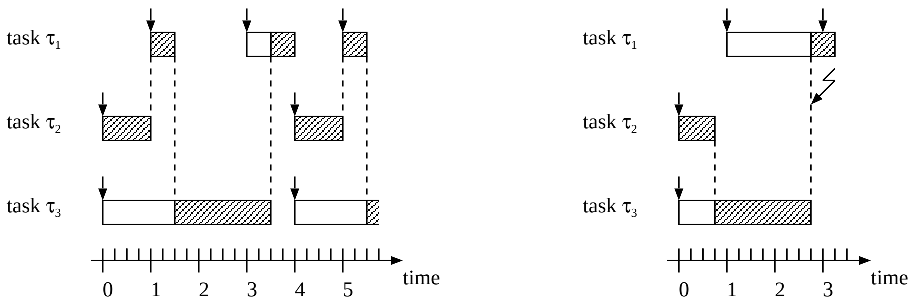
Answer Consider task set $\mathcal{T}$ consisting of three periodic tasks, with characteristics as given in Table 1. Note that tasks τ_1 and τ_3 both need resource R for the entire duration of their computation time. Further note that $C_3 = T_1$, i.e. when τ_3 blocks τ_1 for more than $T_1 - C_1 = 1.5$, τ_1 will miss its deadline. The construction of the example is based on this latter observation.

Figure 4(a) illustrates a feasible schedule for $\mathcal{T}$. The schedule in the interval $[0, 4)$

	$D = T$	C	φ	R
τ_1	2	0.5	1	0.5
τ_2	4	1	0	0
τ_3	4	2	0	2

Table 1: Characteristics of tasks τ_1, τ_2 and τ_3 of $\mathcal{T}$.

is repeated in the intervals $[4n, 4(n+1))$, with $n \in \mathbb{N}^+$. When the computation time of task τ_2 is reduced to 0.75, task τ_3 will block τ_1 for an amount of 1.75, and task τ_1 therefore misses its deadline at time 3; see Figure 4(b).



(a) A feasible schedule. (b) Task τ_1 misses deadline.

Figure 4: Schedules for $\mathcal{T}$, where $C_2 = 1$ in (a) and $C_2 = 0.75$ in (b).

3. (1.0) Consider two periodic tasks τ_1 and τ_2 that share a budget implemented by a periodic server σ, with characteristics as given in Table 2. Under which conditions are the two tasks schedulable?

	$D = T$	C
τ_1	5	2
τ_2	8	3
σ	3	2

Table 2: Characteristics of tasks τ_1 and τ_2 and server σ.

Answer Never, because the utilization of the tasks ($U_1^{\mathcal{T}} + U_2^{\mathcal{T}} = \frac{C_1^{\mathcal{T}}}{T_1^{\mathcal{T}}} + \frac{C_2^{\mathcal{T}}}{T_2^{\mathcal{T}}} = 0.4 + 0.375 = 0.775$) is *larger* than the utilization of the server ($U^\sigma = \frac{C^\sigma}{T^\sigma} = \frac{2}{3}$).

4. Consider four periodic tasks τ_1, τ_2, τ_3 and τ_4 (having decreasing priority), which share five resources, A, B, C, D, and E. Compute the maximum blocking time B_i for each task for the following two protocols, knowing that the longest duration $D_i(R)$ for a task τ_i on resource R is given in the following table (there are no nested critical sections).

	A	B	C	D	E
τ_1	2	5	9	0	6
τ_2	0	0	7	0	0
τ_3	0	3	0	7	13
τ_4	6	0	8	0	10

(a) (1.5) Priority Inheritance Protocol.
Answer See book of Buttazzo Exercise 7.5.

(b) (1.5) Priority Ceiling Protocol.
Answer See book of Buttazzo Exercise 7.6.

5. One of the lectures, given by Alina Weffers-Albu, P.D. Eng., concerned *Behavioural Analysis of Real-Time Systems with Interdependent Tasks.*

   (a) (1.0) Explain which real-time problems were addressed.

   (b) (1.0) Explain how these problems have been solved.

   (c) (0.5) Explain why these approaches were taken.

   Answers See slides of that lecture.
   Note that these questions were described as goals on sheet 6 of the lecture 'RTA.A1-Overview', and have also been asked for the same lecture in the examination of June 27^{th}, 2007.