EINDHOVEN UNIVERSITY OF TECHNOLOGY

Department of Mathematics and Computer Science

Examination Real-time Architectures (2IN20)
on Friday, August 8^{th} 2008, 14.00h-17.00h.

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. Consider a set Γ of n independent, periodic real-time tasks $\tau_1, \ldots, \tau_n$, with arbitrary phasing and deadlines at most equal to periods. A typical recursive equation to determine the worst-case response time of a task τ_i under fixed-priority pre-emptive scheduling (FPPS) is given by

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

   (a) (0.5) Give assumptions concerning the priorities of tasks that need to hold to use this equation.
   Answer The priorities of tasks must be (a) unique and (b) decreasing for increasing indices of tasks, i.e. τ_1 has highest and τ_n has lowest priority.

   (b) (0.5) What can be concluded from the result of the evaluation of the relation

$$D_i \geq C_i + \sum_{j<i} \left\lceil \frac{D_i}{T_j} \right\rceil C_j. \qquad (2)$$

   Answer The left hand-side of the relation denotes the amount of execution time in an interval of length D_i, and the righthand-side denotes the maximum amount of execution requirements of tasks τ_1 till τ_i in that interval. Hence, if the evaluation yields true, then τ_i is schedulable. However, when the evaluation yields false, the task may, but need not, be schedulable.

   Let task τ_k have a fixed period T_k and an activation jitter of AJ_k.

   (c) (0.5) Is it allowed to use (1) for task τ_i when $i > k$? Motivate you answer.
   Answer No, because that would potentially yield an optimistic value, i.e. a value that is too small.

   (d) (0.5) What would be the impact of using $T_j - AJ_j$ rather than T_j in the denominator of (1)? Motivate you answer.
   Answer The result would become pessimistic, i.e. potentially yield a value that is too large.

2. Consider a subsystem $\mathcal{S}$ consisting of a single periodic task τ with a dedicated (associated) fixed-priority server σ, where τ and σ have characteristics as given in Table 1. Let the servers of all subsystems be scheduled by means of FPPS, and let σ have the highest priority among all servers of the subsystems. Determine the worst-case response time of τ for

(a) (0.5) a polling server,

Answer We first observe that the utilization U^σ of the server is equal to the utilization U^τ of the task, i.e. $U^\sigma = U^\tau$. As a result, the system will eventually reach a stable state.

A critical instant occurs when τ is released an infinitesimal ϵ *after* the release of σ. For such an instance, the polling server is depleted upon the activation of the task. The worst-case response time WR_1 of the *first* job of τ now becomes $3-\epsilon+3+0.8 = 6.8-\epsilon$. Hence, WR_1 is a supremum and has a value $WR_1 = \lim_{\epsilon\downarrow 0}(6.8-\epsilon) = 6.8$. Because $WR_1 > T$, the *second* job of task τ is *delayed* by the first job, and the second job may therefore have a larger response time than the first. Considering Figure 1, the worst-case response time WR_2 of the second job is a supremum and equal to 7.4, and WR_3 of the third job is also a supremum and equal to 6.2. From time $t = 16.2$, the

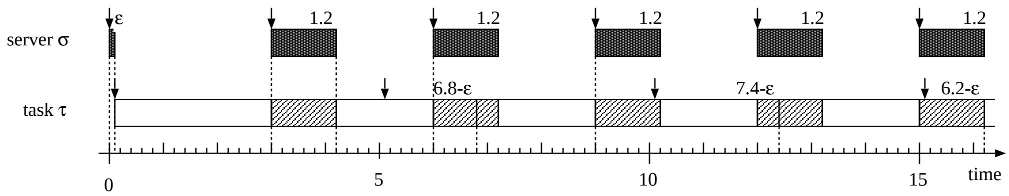

Figure 1: Timeline for $\mathcal{S}$ with a release of task τ an ϵ after the release of the polling server σ. The numbers at the top right corner of the boxes denote the response times of the respective releases.

behavioral pattern of the server and the task repeats itself with a length $T^{\mathcal{S}}$ equal to the least common multiple of the periods, i.e. $T^{\mathcal{S}} = \text{lcm}(3,5) = 15$.

Hence, the worst-case response time WR of τ is a supremum and equal to $WR = \max\{6.8, 7.4, 6.2\} = 7.4$.

(b) (0.5) a periodic server,

Answer A critical instant occurs when τ is released exactly when σ becomes depleted. Hence, the worst-case response time WR_1 of the first job of τ becomes $WR_1 = (3-1.2)+3+0.8 = 5.6$. Because $WR_1 > T$, we have to consider subsequent jobs as well; see Figure 2. Similar to the case with a polling server, the behavior of the

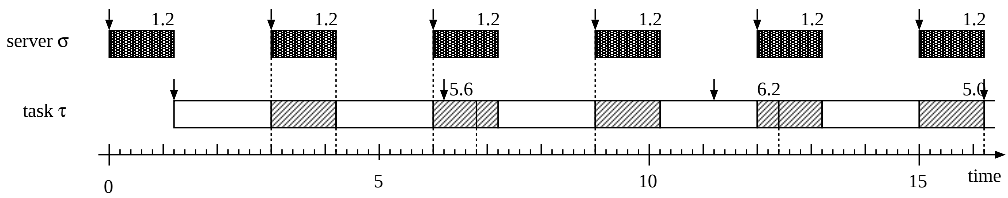

Figure 2: Timeline for $\mathcal{S}$ with a release of task τ an ϵ after the release of the polling server σ. The numbers at the top right corner of the boxes denote the response times of the respective releases.

system repeats itself from time $t = 16.2$ with a period $T^{\mathcal{S}}$ equal to 15. From Figure 2, we derive a worst-case response time $WR = \max\{5.6, 6.2, 5.0\} = 6.2$ for τ.

(c) (1.0) a deferrable server.

Answer A critical instant occurs when τ is simultaneously released with σ; see

Figure 3. Similar to the case of the polling server and the periodic server, the worst-case response time is *not* assumed for the first job, despite the fact that the worst-case response time of the first job of τ is less than its period. From Figure 3, we derive a worst-case response time $WR = 4.4$.

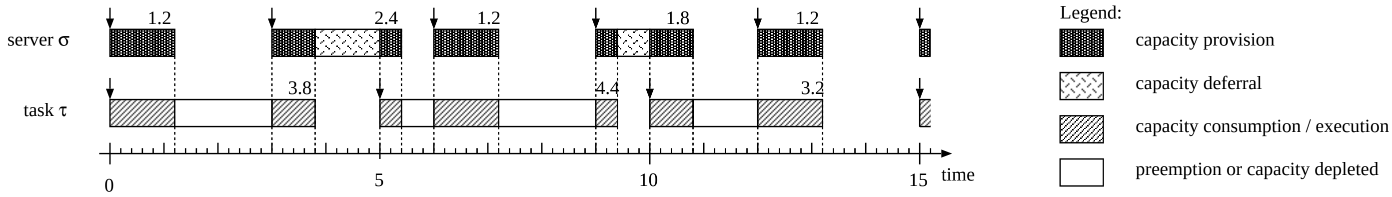

Figure 3: Timeline for $\mathcal{S}$ with a simultaneous release of task τ and deferrable server σ, including a graph with the remaining capacity of σ.

Hints Determine a critical instant for τ and derive the worst-case response time of τ by means of a timeline. Note that the deadline D of τ is not constrained.

	T	C
σ	3	1.2
τ	5	2

Table 1: Characteristics of τ and σ of subsystem $\mathcal{S}$.

3. Consider Table 2, describing two sets of hard real-time, periodic tasks, Γ_1 and Γ_2. The tasks are characterized by means of their periods T_i and (worst-case) computation times C_i.

   (a) (1.0) Determine whether or not task sets Γ_1 and Γ_2 are schedulable under RMS and arbitrary phasing.
   Answer
   - Task set Γ_1 is not schedulable (task τ_3 misses its deadline). This becomes immediately clear when a timeline is drawn.
   - Task set Γ_2 is schedulable, which also becomes clear by drawing a timeline.

   (b) (1.0) Determine for task set Γ_1 and Γ_2 with which factor the processor speed should be increased or decreased to make the task set precisely schedulable. Motivate your answer.
   Answer

Name	$T_i (= D_i)$	C_i
τ_1	5	2
τ_2	7	2
τ_3	11	2.5
τ_4	20	1

Name	$T_i (= D_i)$	C_i
τ_1	7	4
τ_2	14	3
τ_3	28	5

Table 2: Two sets of tasks, Γ_1 (left-hand side) and Γ_2.

- When all tasks of Γ_1 are simultaneously released at time 0, task τ_3 needs to execute 2.5 units of time in an interval of length 10 (i.e. till the next activation of task τ_1). As a result, a total computation of 10.5 units of time has to be executed in an interval of length 10. The speed of the processor must therefore be *increased* with a factor $\frac{21}{20}$, effectively reducing the computation times with a factor $\frac{20}{21}$.
- Note that the periods of the tasks of Γ_2 are harmonic, i.e. for every i, j with $i > j$ there exists a $k \in Nat$ such that $T_i = kT_j$. For this particular case, the necessary test $U \leq 1$ is also an exact test. Because $U_2 = \frac{27}{28}$, the speed of the processor can therefore be *decreased* with a factor $\frac{27}{28}$, effectively increasing the computation times with a factor $\frac{28}{27}$.

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	6	7	0	10	3
τ_2	0	0	0	8	0
τ_3	4	14	8	0	0
τ_4	0	11	0	9	7

   (a) (1.5) Priority Inheritance Protocol.
   Answer Similar to Exercise 7.5 of the 2^{nd} edition of the book of Buttazzo. Compared to that exercise, the columns have been exchanged ($A \to E \to B \to A$ and $C \leftrightarrow D$), and all non-zero values have been increased by 1.

   (b) (1.5) Priority Ceiling Protocol.
   Answer Similar to Exercise 7.6 of the 2^{nd} edition of the book of Buttazzo.

5. One of the lectures concerned *A QoS approach for multimedia consumer terminals with media processing in software.* The aim of the QoS approach was *cost-effective high-quality video processing in software for multimedia consumer terminals*, motivated by the requirements for *openness* and *flexibility* of these systems, and having as boundary condition that *the existing system qualities should be preserved.*

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

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

   Answer See slides of the lecture.