EINDHOVEN UNIVERSITY OF TECHNOLOGY

Department of Mathematics and Computer Science

Examination Real-time Architectures (2IN60)
on Friday, April 16^{th} *2010, 9.00h-12.00h.*

First read the entire examination. There are 7 exercises in total. Grades are included between parentheses at all parts and sum up to 11 points. Good luck!

1. The hyperbolic bound $HB(n) : \prod_{1 \leq i \leq n}(U_i^{\tau} + 1) \leq 2$, where U_i^{τ} denotes the processor utilization of the task τ_i, is an example of a *sufficient* schedulability test.

   (a) (0.5) Briefly discuss the relation between the following conditions and the *HB*-bound.

   i. Deadlines are at most equal to periods, i.e. $D_i \leq T_i$.
   Answer: The condition $D_i = T_i$ must hold (as a precondition) to apply the *HB*-bound. The given condition is therefore too weak for the *HB*-bound.

   ii. The sum of the utilizations is at most equal to 1.
   Answer: This need not hold to *apply* the *HB*-bound. When the condition does not hold, the *HB*-bound will not hold either.

   (b) (1.0) Construct an example set of four tasks for which the left-hand side of $HB(4)$ is equal to 2 and where any *increase* of any computation time or *decrease* of any period will make the task set unschedulable under RMS. *Hint*: Construct the set of tasks by means of a timeline.
   Answer: Construction by means of a timeline is illustrated in RTA.Exercises-2. The result satisfies the requirements that $HB(4) = 2$ and that an *increase* of any computation time makes the task set unschedulable. The result does *not* satisfy the requirement that a *decrease* of any period will make the task set unschedulable, however. Even stronger, it is possible to decrease T_2, T_3, and T_4 to a value equal to T_1, without making the task set unschedulable. The conclusion is therefore that this can only be done when 3 tasks have a computation time equal to zero. For $C_i > 0$, this is therefore not possible.

2. Semaphores were conceived by Edsgar W. Dijkstra and can be used for "mutual exclusion".

   (a) (0.5) Explain the notion of "mutual exclusion" in your own words.
   Answer: See slides of 2IN60.prior-knowledge-II.

   (b) (1.0) Describe semaphores. *Hint*: Consider the *value* of a semaphore and its two *operations*, and describe the properties and semantics of the operations.
   Answer: See slides of 2IN60.prior-knowledge-II.

3. There are different types of tasks, e.g. periodic tasks with jitter, elastic tasks, and sporadic tasks.

   (a) (0.5) Describe the (distinguishing) characteristics of periodic tasks with jitter, elastic tasks, and sporadic tasks. *Hint*: Consider jitter and (best-case and worst-case) values for periods.
   Answer: The distinguishing characteristics are:

- A periodic task τ_i with jitter has a fixed period T_i and activation jitter AJ_i.
- An elastic task τ_i has a (finite) best-case period BT_i, a (finite) worst-case period WT_i, where $WT_i \leq BT_i$, and no jitter, i.e. $AJ_i = 0$.
- A sporadic task τ_i has an infinite best-case period BT_i, i.e. the inter-arrival times of its jobs may be an arbitrary amount of time apart, a (finite) worst-case period WT_i, and no jitter, i.e. $AJ_i = 0$.

(b) (1.0) Assume fixed-priority pre-emptive scheduling, and suppose you model a periodic task with jitter as an elastic task. Moreover, assume that the worst-case deadline is at most equal to the worst-case period minus the jitter. What would be the consequence for the calculated worst-case response time of (*i*) that task, (*ii*) tasks with a higher priority, and (*iii*) tasks with a lower priority? Motivate your answer.
Answer: Let the periodic task τ_i with jitter be specified with T_i and AJ_i. It is given that $WD_i \leq T_i - AJ_i$.
The maximum and minimum inter-arrival time of the jobs of τ_i are $T_{\max} = T_i + AJ_i$ and $T_{\min} = T_i - AJ_i$, respectively. Hence, the elastic task τ_i' has a best-case period $BT_i' = T_i + AJ_i$, a worst-case period $WT_i' = T_i - AJ_i$, and no jitter, i.e. $AJ_i' = 0$. The (worst-case) deadline is assumed to remain the same, i.e. $WD_i' = WD_i \leq T_i - AJ_i = WT_i' - AJ_i'$.
We now consider the consequence for the calculated worst-case response times. Let the set of indices of tasks with a higher and lower priority than τ_i be denoted by $hp(i)$ and $lp(i)$, respectively. For ease of presentation, let's assume that the tasks are independent. Because τ_i does not influence τ_j with $j \in hp(i)$, the way τ_i is modeled does not change the calculated WR_j; see also (1). Similarly, none of the differences between τ_i and τ_i' changes the outcome for the task itself. Finally, because the elastic task τ_i' can give rise to more interference than the periodic task with jitter τ_i to tasks τ_j with $j \in lp(i)$, the calculated worst-case response times of those lower priority tasks can become larger.

4. Consider two periodic tasks τ_1 and τ_2 and a deferrable server S_{DS} with characteristics as given in the following table.

	$T = D$	C
S_{DS}	5	1
τ_1	7	2
τ_2	9	3

Assume scheduling based on FPPS and a rate monotonic priority assignment.

(a) (1.0) Assuming arrivals of aperiodic requests at time $t = 4$ for an amount of 1.6 and at time $t = 7$ for an amount of 2, draw time-lines illustrating the execution of the tasks and the remaining capacity of the deferrable server in an interval of length 25.
Answer: See Figure 1.

(b) (0.5) Are the tasks and the deferrable server schedulable under arbitrary phasing? Motivate your answer by means of *calculations*. *Hint*: Determine the worst-case response times of the tasks by means of the following recursive equation.

$$x = WC_i + \sum_{j<i} \left\lceil \frac{x + AJ_j}{WT_j} \right\rceil WC_j \qquad (1)$$

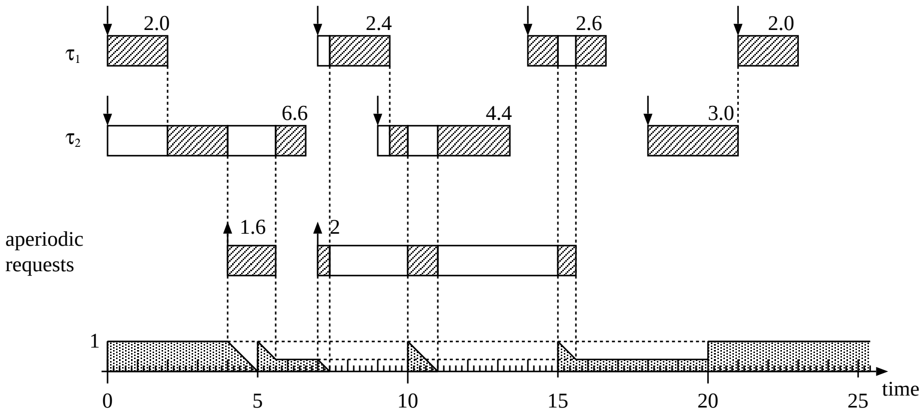

Figure 1: Example of high-priority Deferrable Server.

Answer: Note that $AJ_{\text{DS}} = T_{\text{DS}} - C_{\text{DS}} = 4$. Based on (1), we find $WR_1 = 4 < D_1 = 7$ and $WR_2 = 10 > D_2 = 9$. Hence, task τ_2 misses its deadline and task τ_2 is therefore not schedulable.

5. Consider three tasks with arbitrary phasing and characteristics as given in the following table.

	T	WD	BD	AJ	WC	BC
τ_1	5	6	2	0	2	2
τ_2	10	9	2	1	3	3
τ_3	37	35	15	0	9	7

(a) (0.5) Give a necessary and sufficient condition for the schedulability of the three tasks.
Answer: A necessary and sufficient, i.e. exact, condition is given by

$$\underset{1 \le i \le 3}{\forall} (BD_i \le BR_i \wedge WR_i \le WD_i)\,, \tag{2}$$

where BD_i and WD_i are the best-case and worst-case deadline, respectively, and BR_i and WR_i are the best-case and worst-case response time, respectively

(b) Assume fixed-priority pre-emptive scheduling, where τ_1 has highest and τ_3 has lowest priority.

i. (0.5) Draw a time line with an optimal instant for task τ_3.
Answer: See Figure 2. Note that $BR_3 = 16$.

ii. (1.0) Determine the best-case response time of task τ_3 using the following recursive equation.

$$x = BC_i + \sum_{j<i} \left(\left\lceil \frac{x - AJ_j}{BT_j} \right\rceil - 1 \right)^{+} BC_j \tag{3}$$

Answer: Using (1) we first determine the worst-case response time $WR_3 = 35 = WD_3$ as initial value for the iterative procedure to determine the best-case response time BR_3. Using (3), we subsequently find $BR_3 = 16 > BD = 15$.

6. Without a resource access protocol, a so-called *deadlock* may occur when tasks share resources.

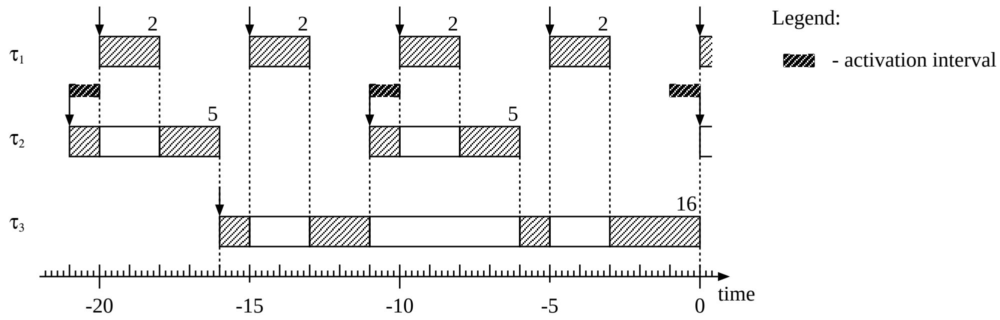

Figure 2: Time line with an optimal instant for task τ_3.

(a) (0.5) Describe the notion of a deadlock and illustrate it with an example.
Answer: See slides RTA.B4-Policies-3 and book Fig. 11 on p. 210.

(b) (0.5) Describe how deadlocks can be avoided *without* a resource access protocol.
Answer: See slide 10 of RTA.B4-Policies-3.

(c) (1.0) Give two resource access protocols that prevent a deadlock, and show by means of a timeline for one of these protocols how its application resolves a deadlock.
Answer: See slide 8 of RTA.B4-Policies-3 and book.

7. (1.0) Give an example illustrating transitive adjustment of priorities for the Priority Inheritance Protocol (PIP).
Answer: See slides RTA.B4-Policies-3 and book Fig. 7.8.