EINDHOVEN UNIVERSITY OF TECHNOLOGY

Department of Mathematics and Computer Science

Examination Real-time Architectures (2IN25)
on Thursday, March 12^{th} *2009, 14.00h-17.00h.*

First read the entire examination. There are 6 exercises in total. Grades are included between parentheses at all parts and sum up to 9 points. Good luck!

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

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

   (a) (0.5) For which class of scheduling algorithms is this equation applicable?

   (b) (0.5) Give at least four assumptions that need to hold to use this equation.

   Answer: See RTA examination of August 30^{th} 2006.

2. Consider two periodic tasks τ_1 and τ_2, with characteristics as given in the following table, which are scheduled by means of fixed-priority scheduling with deferred pre-emption (FPDS), where τ_1 has a higher priority than τ_2.

	$T = D$	C
τ_1	3	1
τ_2	4	$1\frac{1}{2}$

   (a) (1.0) Determine the worst-case length WL_2 of the level-2 active period.
   Answer: Draw a timeline with a simultaneous release of both tasks at time $t = 0$. Because the pending load becomes zero at time $t = 2\frac{1}{2}$, the worst-case length $WL_2 = 2\frac{1}{2}$.

   (b) (1.0) Determine the worst-case response times WR_1^{D} of τ_1 and WR_2^{D} of τ_2 using

$$WR_{ik}^{\mathrm{D}} = \begin{cases} WR_i^{\mathrm{P}}(B_i^{\mathrm{D}} + kC_i - F_i) + F_i - (k-1)T_i & \text{for } i < n \\ WO_i^{\mathrm{P}}(kC_n - F_n) + F_n - (k-1)T_n & \text{for } i = n \end{cases}, \tag{1}$$

   and the following recursive equations for $WR_i^{\mathrm{P}}(C)$ and $WO_i^{\mathrm{P}}(C)$, respectively.

$$x = C + \sum_{j<i} \left\lceil \frac{x}{T_j} \right\rceil C_j, \tag{2}$$

$$x = C + \sum_{j<i} \left(\left\lfloor \frac{x}{T_j} \right\rfloor + 1 \right) C_j. \tag{3}$$

   Answer: For task τ_1, it is only necessary to consider the first job: $WR_1^{\mathrm{D}} = 2\frac{1}{2}$. Because the worst-case length WL_2 of the level-2 active period contains $\left\lceil \frac{WL_2}{T_2} \right\rceil = 1$ job, we need to consider one job for task τ_2. Hence $WR_2^{\mathrm{D}} = WR_{2,1}^{\mathrm{D}} = 2\frac{1}{2}$.

3. (1.0) Consider a system consisting of a periodic server, which is used to service aperiodic requests, and a set of hard real-time tasks. Describe the consequence of selecting either an *idling* or a *gain-time providing* periodic server for determining the best-case and worst-case response times of hard-real-time tasks.
Answers: The choice only potentially influences the best-case response time of hard real-time tasks with a priority lower than the periodic server; see RTA.Exercises-5 slide 8.

4. (2.0) Let tasks τ_1 and τ_2 both use resources r_1 and r_2. Task τ_1 first locks r_1 and subsequently r_2, and τ_2 first locks r_2 and subsequently r_1, which may give rise to a deadlock without a resource access protocol; see Figure 1. Discuss what happens when a resource access protocol is used and illustrate the behaviour by means of appropriate drawings for PIP, HLP, PCP, and SRP.

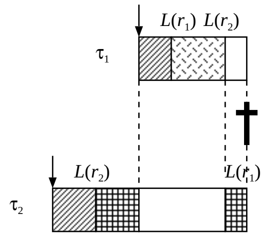

Figure 1: A deadlock situation.

Answer: See RTA examination of June 13^{th} 2008 or RTA.Exercises-8+solutions.

5. The following questions concern presentations of assignments.

   (a) *Hierarchical scheduling of dependent applications*: Two approaches were presented for global resource access, one approach termed H-SRP and the other termed SIRAP. The approaches differed with respect to their behavior when the amount of remaining capacity of a reservation was less than the amount of time needed for accessing the global resource.

   i. (0.5) Explain the difference between both approaches.
   Answer: H-SRP assumes budget *overrun* (optionally with *pay-back*) and SIRAP assumes (*self-*) *blocking*.

   ii. (0.5) What are the consequence of SIRAP for a supporting mechanism?
   Answer: Upon entering a critical section, a check is needed whether or not there is still enough remaining capacity to use the global resource, i.e. to execute the critical section.

   (b) (1.0) *Fixed Priority Scheduling in Multiprocessor Systems*: Is the following *sporadic* task system schedulable on two identical processors? Motivate your answer. Assume fixed-priority pre-emptive scheduling and that task τ_1 has highest and τ_3 has lowest priority.

	T	D	C
τ_1	2	1	1
τ_2	3	1	1
τ_3	6	6	5

Answer: No; see slides.

6. The following questions concern guest-lectures.

   (a) (0.5) Mike Holenderski gave a guest lecture entitled *Swift mode changes.* He considered two scheduling approaches for mode changes, fixed-priority pre-emptive scheduling (FPPS) and fixed-priority scheduling with deferred pre-emption (FPDS). Which approach is better and why?
   Answer: Assuming that processing a mode corresponds with a single sub-job, the worst-case lead-time of a mode-change is larger for FPPS than for FPDS, because for FPDS we need to wait for at most 1 sub-job to complete.

   (b) (0.5) Alina Weffers-Albu gave a guest-lecture entitled *Behavioural Analysis of Real-Time Systems with Interdependent Tasks.* She explained that the behaviour of a chain of components scheduled by means of fixed-priority pre-emptive scheduling (FPPS) assumes a repetitive pattern after an finite amount of time. How can that amount of time for the so-called initial phase be minimized?
   Answer: By giving the first task in the chain the lowest priority and/or by minimizing the size of the buffers prior to the lowest priority component in the chain.