EINDHOVEN UNIVERSITY OF TECHNOLOGY
Department of Mathematics and Computer Science

Examination Real-time Systems (2IN26)
on Monday, October 29^{th} *2012, 14.00h-17.00h.*

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. (1.0) A terminal, such as a sensor and an actuator, can be modeled as a component with three interface elements for control as illustrated in Figure 1. Briefly describe

   (a) the three elements of the interface;
   Answer: The control interface consists of *commands*, *observers*, and *events*; see slides Specification concepts.

   (b) the relevance of these elements for both sensors and actuators;
   Answer: The interface of a sensor typically contains a *command* (for initialization) and *observers* and/or *events* to inform the control system about state-changes. The interface of an actuator typically contains *commands*; see slides Specification concepts.

   (c) the relation with event-triggered and time-triggered tasks.
   Answer: a so-called *event* will (re-) activate an event-triggered task, whereas a so-called *observer* is particularly useful for a polling (i.e. time-triggered) task.

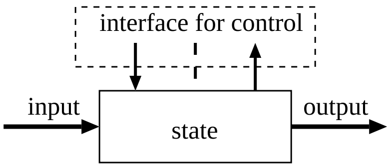

Figure 1: A component-model of a terminal.

2. (0.5) 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. Are the following conditions *necessary*, *sufficient*, or *irrelevant* for the application of the *HB*-bound? Motivate your answer.

   (a) Deadlines are at most equal to periods, i.e. $D_i \leq T_i$.
   Answer: The hyperbolic bound is meant for the rate monotonic (RM) algorithm, i.e. $D_i = T_i$. When the *HB* holds for $D_i = T_i$, the task set remains schedulable when deadlines are increased. Hence, when the assumptions for RM hold with the exception that $D_i \geq T_i$, the *HB*-bound is still a *sufficient* condition. The given condition $D_i \leq T_i$ is too weak for the *HB*-bound. In particular, it is *not necessary*, because $D_i \geq T_i$ is allowed, and it is *not sufficient*, because $D_i < T_i$ is not allowed.

   (b) 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. Formally: because $\sum_{1\leq i\leq n} U_i^\tau + 1 \leq \prod_{1\leq i\leq n}(U_i^\tau + 1)$ we immediately see that when $\sum_{1\leq i\leq n} U_i^\tau > 1$ the *HB*-bound will not hold. The condition is therefore *irrelevant*.

3. A so-called *deadlock* may occur when tasks share resources.

   (a) (0.5) Describe the notion of a deadlock and illustrate it with an example.
   Answer: See slides RTA.B4-Policies-3 and Fig. 7.11 (7.13) in the 2^{nd} (3^{rd}) edition of the book.

   (b) (1.0) Give two resource access protocols that prevent a deadlock, and illustrate for one of these protocols how it works using the same example.
   Answer: See slide 8 of RTA.B4-Policies-3 and book.

4. This question concerns fixed-priority scheduling with deferred preemption (FPDS).

   (a) (0.5) What may hamper the application of FPDS in practice?
   Answer: Placement of preemption-points in the code.

   (b) (0.5) Give an example of a taskset that is schedulable by fixed-priority pre-emptive scheduling (FPPS) but not by FPDS. Assume that the computation times of all subjobs are larger than zero.
   Answer: Whenever $C_1 = D_1 < T_1$, τ_1 cannot be deferred by any lower priority task.

5. Consider three tasks that are scheduled by means of FPPS, where τ_1 has highest and τ_3 has lowest priority, with arbitrary phasing and characteristics as given below.

	$WT = BT$	WD	BD	AJ	WC	BC
τ_1	5	3	1	1	2	2
τ_2	8	8	3	0	3	3
τ_3	25	24	8	0	5	4

   (a) (0.5) Determine the worst-case response time of τ_3 by means of the following recursive equation.

$$x = B_i + WC_i + \sum_{1 \le j < i} \left\lceil \frac{x + AJ_j}{WT_j} \right\rceil WC_j. \tag{1}$$

   Answer: Note that the *lowest* priority task is *never* blocked. Hence, $B_3 = 0$ irrespective of potential resource sharing. Using an iterative procedure, we find $WR_3 = 24$.

   (b) Determine the best-case response time of task τ_3

      i. (1.0) by drawing a time line with an optimal instant for τ_3.
      Answer: See Figure 2. Note that $BR_3 = 9$.

      ii. (0.5) by means of the following recursive equation.

$$x = BC_i + \sum_{1 \le j < i} \left(\left\lceil \frac{x - AJ_j}{BT_j} \right\rceil - 1 \right)^{+} BC_j \tag{2}$$

      Answer: Using $WR_3 = 24$ as initial value for the iterative procedure to determine the best-case response time BR_3, we find $BR_3 = 9$.

   (c) (1.0) Let $AJ_3 = 2$. Does this have an influence on the worst-case response time of task τ_3?
   Answer: First, observe that $WD_3 + AJ_3 = 26 > T_3 = 25$, hence using (1) is not appropriate in general. Next, because $WR_3 + AJ_3 = 26 > T_3 = 25$, using (1) does not necessarily yield WR_3, i.e. a next job of τ_3 may have a larger response time than the

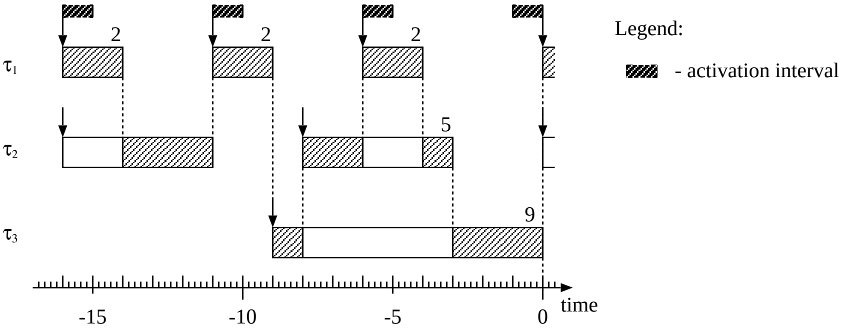

Figure 2: Timeline with an optimal instant for task τ_3.

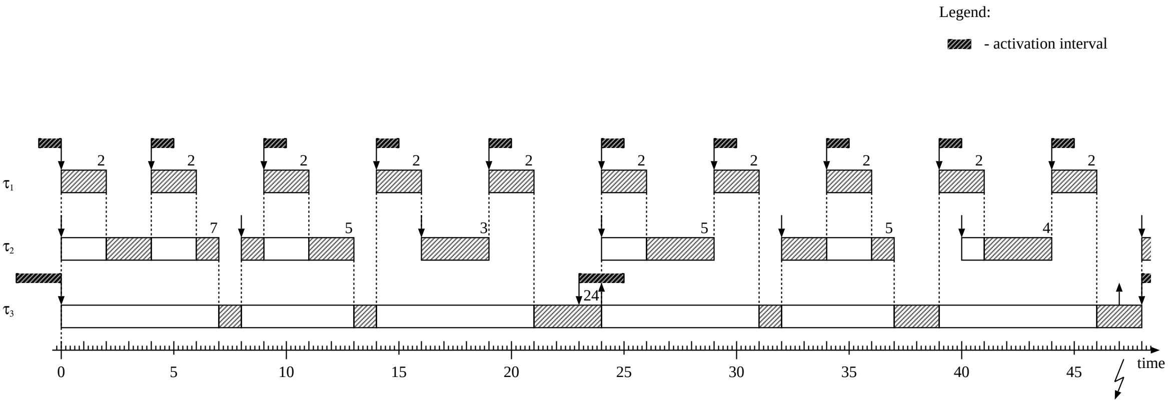

Figure 3: Timeline with a critical instant for task τ_3.

first job due to additional interference of its previous jobs. By drawing a time-line (see Figure 3), we find that the second job has a response time of 25, which is larger than WD_3. From the timeline, we also derive that the level-3 active period ends at the finalization of the second job, hence $WR_3 = 25 > WD_3 = 24$. Hence, the value found by means of (1) is not the worst-case response time of task τ_3 for $AJ_3 = 2$.

6. The following questions concern guest-lectures.

   (a) (0.5) Dr. Isovic presented the Real-Time Talk (RTT) method for the development of real-time systems, improving on the common practice.

      i. What is the common practice to get from an application to a set of tasks?
      Answer: An application is split into a set of tasks.

      ii. How does RTT improve on that approach?
      Answer: An application consists of a number of *operation modes* during its lifetime, each mode may consist of a number of *transactions*, and each transaction consists of a set of *tasks*.

   Answer: See RT.Design for further details and an example.

   (b) (1.0) The example with the truck-bed shown by Dr. Isovic had two tasks with a dependency relation. The tasks had the same period. Give at least two ways to implement the dependency relation between these tasks using fixed-priority preemptive scheduling (FPPS).
   Answer: In the example of the truck-bed, two dependencies exists:

- a *precedence* relation between ALARM and CONTROL;
  The precedence relation can be resolved in FPPS by
  i. Giving the first task a higher priority than the second, and making sure that the second task is not released before the first task.
  ii. Giving (*i*) the tasks the same priority and (*ii*) the second task an appropriate offset compared to the first, making sure that the second task is released only after the first task completes.
  iii. Using a semaphore S, initialized by zero, a $V(S)$ executed by the first and a $P(S)$ executed by the second task.
- *resource sharing* between CONTROL and ERROR.
  These two tasks have *different* periods. Resource sharing can be resolved by a resource access protocol, such as SRP, or by executing the tasks non-preemptively.

(c) (1.0) During his presentation on Scalable Video Algorithms, Prof. Hentschel mentioned that the number of operations (such as additions, multiplications, and memory accesses) alone is not a convenient measure for the resource needs of an algorithm. Explain why and briefly describe an alternative.
Answer: From his slides: "*Resources on function level are difficult to manage and to control*". As an alternative a (-n estimated) resource-budget for the processor described in terms of, e.g. a *capacity* and a *period* can be used.

7. This question concerns the paper "[Shin et al 03] I. Shin and I. Lee, *Periodic resource model for compositional real-time guarantees*, In: Proc. 24^{th} IEEE Real-Time Systems Symposium (RTSS), pp. 2-13, December 2003".

(a) (1.0) In [Shin et al 03], a *resource supply bound function* $\texttt{sbf}_\Gamma(t)$ of a time interval of length t is defined that calculates the minimum resource supply of Γ during t units. Given a periodic resource $\Gamma(\Pi, \Theta)$, draw $\texttt{sbf}_\Gamma$ as a function of t for $0 \leq t \leq 4\Pi$, where $\Theta = \Pi/3$. Note that Π represents the period and Θ the capacity.
Answer: See paper. Note that the so-called "blackout duration" (i.e. the longest time without resource supply) is equal to $2(\Pi - \Theta) = \frac{4}{3}\Pi$. Further note that a similar question was asked in the exams of April 16^{th}, 2010 and April 16^{th}, 2012.

(b) (0.5) Which of the following two servers can be used to implement the periodic resource model? Motivate your answer.
  i. polling server;
  Answer: No, because a polling server looses all its capacity when there is no work pending. Hence, the "blackout duration" for a polling server is equal to $2\Pi - \Theta$ (rather than $2(\Pi - \Theta)$).
  ii. deferrable server.
  Answer: Yes, because a deferrable server can provide its capacity in every period irrespective of pending work.