EINDHOVEN UNIVERSITY OF TECHNOLOGY
Department of Mathematics and Computer Science

Examination Real-time Architectures (2IN60)
on Wednesday, November 3^{rd} 2010, 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. Automotive domain:

   (a) (0.5) It has been estimated that (*i*) 80% of automotive innovation is based on electronics and is realized by software (*ii*) 35% of the total cost are due to software. Notably, the complexity of HW in terms of number of ECUs tends to *decrease*. Explain this latter trend and the consequences for software development.
   Answer: See slides of 2IN60.prior-knowledge-I.pdf.

   (b) (0.5) Five main automotive domains have been distinguished i.e. powertrain, chassis, body, telematics, and passive safety. Classify the safety requirements of these domains as *very high*, *high*, or *low*, and motivate your answer.
   Answer: See slides of 2IN60.prior-knowledge-I.pdf.

2. Interrupts generated by I/O peripheral devices represent a big problem for the predictability of a real-time system. Three approaches are described in the book of Buttazzo to handle interrupts, two based on disabling interrupts and by using polling (by either application tasks or dedicated kernel routines) and a third where all external interrupts are left enabled.

   (a) (0.5) Describe the third approach. *Hint*: Distinguish between *device driver* and *device manager* and address the role of the operating system.
   Answer: See book.

   (b) (0.5) Describe at least 2 advantages of this third approach.
   Answer: Advantages (see book): (*i*) busy wait during I/O operations is eliminated; (*ii*) interrupt interference is minimized (i.e. to the *activation* of a proper task only); (*iii*) priority of the task managing the device is completely independent from other priorities and can be set according to the application requirements.

3. Semaphores were conceived by Edsgar W. Dijkstra and can be used for resource sharing.

   (a) (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.

   (b) (0.5) Which specific assumption for semaphores is needed to make them applicable in the context of Real-Time Systems?
   Answer: See slides RTS.B4-Policies-3-RAP.

   (c) (0.5) Describe how semaphores can be used to implement resource sharing, assuming mutual exclusive access.
   Answer: See slides RTS.B4-Policies-3-RAP.

4. Assume a set Γ of n tasks $\tau_1, \tau_2, \ldots, \tau_n$, where tasks are indexed in order of decreasing priority, i.e. τ_1 has highest and τ_n has lowest priority. When the tasks are scheduled by means of fixed-priority pre-emptive scheduling, recursive equation (1) can used to determine the worst-case response time of a task τ_i, assuming certain additional conditions.

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

   (a) (0.5) Give at least 3 conditions that need to hold to use this equation.
   Answer: Conditions include (see also RTS.B5-Analysis-2-FPPS): (*i*) tasks have unique priorities; (*ii*) no self-suspension of tasks; (*iii*) (worst-case) relative deadline WD_i is at most equal to the (worst-case) period WT_i minus activation jitter AJ_i, i.e. $WD_i \le WT_i - AJ_i$; (*iv*) overhead of scheduling and context switching is ignored.
   Note: arbitrary phasing is not a condition (or constraint). For a specific phasing, the equation can stil be used, but may yield a pessimistic result.

   (b) (0.5) What do the left-hand side and the right-hand side of the equation represent?
   Answer: LHS - amount of time *available* in $[0, x)$. RHS - *max.* amount of time *requested* by τ_i and τ_j with $1 \le j < i$ in $[0, x)$.

   (c) (0.5) Give a *sufficient* worst-case condition for schedulability of Γ using (1).
   Answer: When $WD_i \le WT_i - AJ_i$ holds for *all* $1 \le i \le n$, we can use

$$\underset{1 \le i \le n}{\forall} \; WD_i \ge B_i + WC_i + \sum_{1 \le j < i} \left\lceil \frac{WD_i + AJ_j}{WT_j} \right\rceil WC_j. \tag{2}$$

5. Consider three tasks that are scheduled by means of fixed-priority pre-emptive scheduling, where τ_1 has highest and τ_3 has lowest priority, with arbitrary phasing and characteristics as given in the following table.

	T	WD	BD	AJ	WC	BC
τ_1	7	10	2	0	2	2
τ_2	9	4	2	1	3	3
τ_3	31	27	10	0	8	6

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

      i. (0.5) drawing a time line with an optimal instant for τ_3.
      Answer: See Figure 1. Note that $BR_3 = 11$.

      ii. (1.0) means of the following recursive equation.

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

      Answer: Using (1) we first determine the worst-case response time $WR_3 = 25$ as initial value for the iterative procedure to determine the best-case response time BR_3. Using (3), we subsequently find $BR_3 = 11$.

   (b) (0.5) Is task τ_3 schedulable?
   Answer: Yes, because $BD_3 = 10 \le BR_3 = 11$ and $WR_3 = 25 \le WD_3 = 27$.

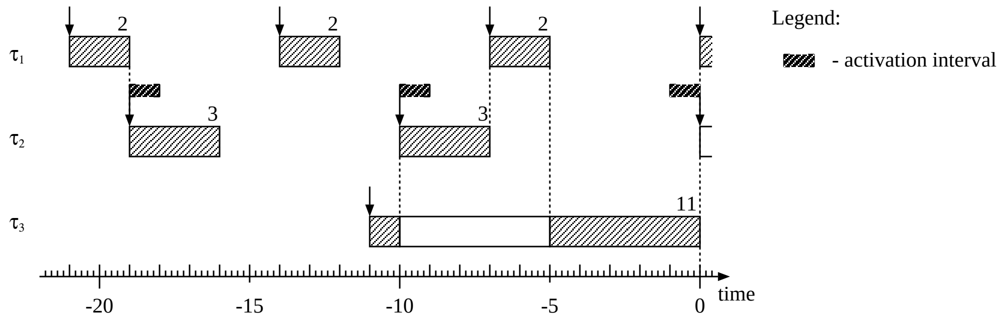

Figure 1: Timeline with an optimal instant for task τ_3.

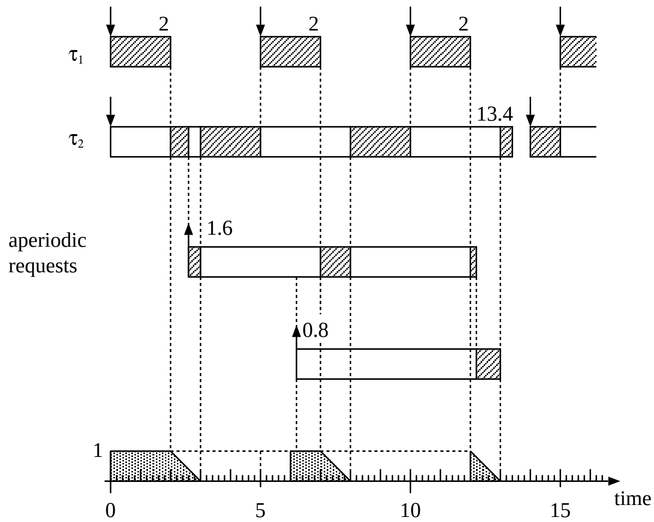

Figure 2: Timeline with a periodic server assuming gain-time consumption.

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

	$T = D$	C
S_{PS}	6	1
τ_1	5	2
τ_2	14	5

Assume scheduling based on FPPS, a rate monotonic priority assignment, and a simultaneous release of the task with a replenishment of the server at time $t = 0$. Moreover, assume arrivals of a-periodic requests at time $t = 2.6$ for an amount of 1.6 and at time $t = 6.2$ for an amount of 0.8. Draw time-lines illustrating the execution of the tasks and the remaining capacity of a periodic server in an interval of length 16 assuming

(a) (1.0) *gain-time consumption*.
Answer: See Figure 2.

(b) (0.5) *idling*.
Answer: See Figure 3.

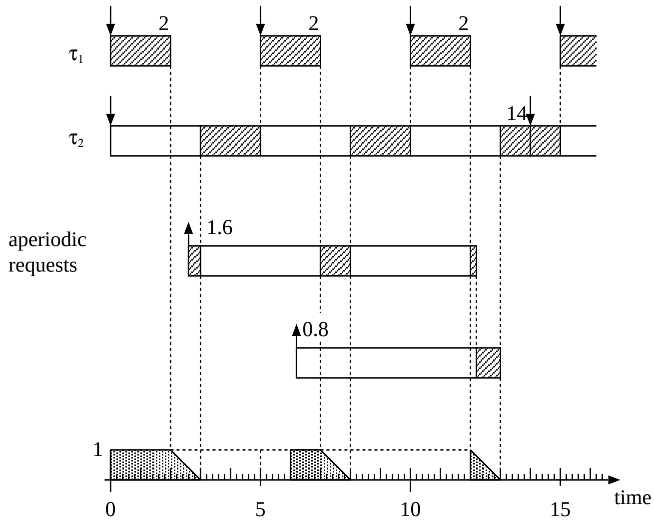

Figure 3: Timeline with a periodic server assuming idling.

Note: explicitly explain (or visualize) which a-periodic request is served when by S_{PS}.

7. Consider a task set consisting of four hard real-time tasks τ_1, τ_2, τ_3, and τ_4, which share three resources R_1, R_2, and R_3. The tasks are scheduled using fixed priority scheduling, where task τ_i has a higher priority than τ_j iff $i < j$, i.e. τ_1 has highest and τ_4 has lowest priority. Figure 4 illustrates a situation of chained blocking when no resource sharing protocol is applied. Draw the timeline (including the dynamic priorities of the tasks) that results when applying the following resource sharing protocols for the example.

   (a) (1.0) Priority Inheritance Protocol (PIP).

   (b) (1.0) Stack Resource Protocol (SRP).

   Answers: See RTA.Exercises-8.

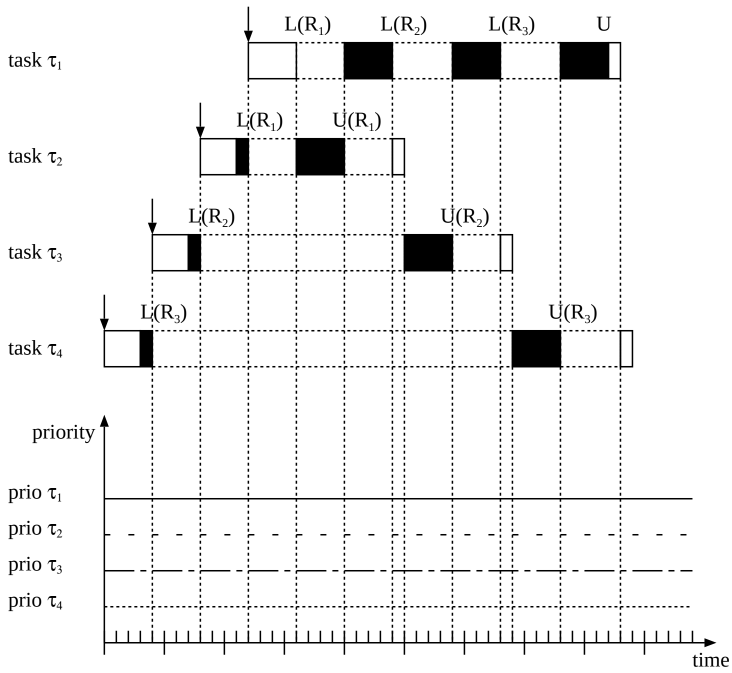

Figure 4: Example of chained blocking.