EINDHOVEN UNIVERSITY OF TECHNOLOGY

Department of Mathematics and Computer Science

Examination Real-time Architectures (2YN26)
on Monday, January 23^{rd} *2012, 10.30h-12.00h.*

First read the entire examination. There are 6 exercises in total. Grades are included between parentheses at all parts and sum up to 10 points. Good luck!

1. Let a sensor indicate whether or not a particular situation has occurred, such as the *end-sensor* in the control example of a truck bed. Assume the system has to react within D^{system} on an occurrence of the situation (e.g. detection of the end-position).

   (a) How often should a time-triggered task τ that reads (and acts upon) the sensor at least be activated?

   i. (0.5) when $T^\tau = D^\tau$;

   ii. (0.5) when $T^\tau > D^\tau$?

   (b) (0.5) Which constraint holds for the deadline of an event-triggered task that reads (and acts upon) the sensor?

   (c) (0.5) What is the disadvantage of a time-triggered approach? *Hint*: consider small values of D^{system}.
   Answer: High overhead.

   Answer: See slides RT Design and RTA.D0-Water-Vessel.

2. In the following two fixed-priority scheduling (FPS) situations, the response times of all jobs in a so-called level-i active period have to be considered to determine the worst-case response time of task τ_i. Explain in your own words why looking at the first job alone is not sufficient.

   (a) (0.5) Fixed-priority preemptive scheduling (FPPS) and *arbitrary deadlines*.

   (b) (0.5) Fixed-priority scheduling with deferred preemption (FPDS) and *deadlines at most equal to periods*.

   Answer: See slides of the lectures.

3. The following questions concern the practical factors.

   (a) *Context switches*

   i. (0.5) Assuming independent tasks, how many jobs can a job preempt?
   Answer: at most once.

   ii. Assume a set $\mathcal{T}$ of two tasks and a single shared resource R (next to the processor). Every job of every task uses (i.e. locks and unlocks) R exactly once. How many times can a job of τ_1 at most preempt a job of τ_2, for the following synchronization protocol?

   A. (0.5) Priority Inheritance Protocol (PIP).
   Answer: at most twice, once when τ_2 has locked (but not yet unlocked) R and once when τ_2 subsequently unlocks R.

B. (0.5) Stack Resource Policy (SRP).
Answer: at most once; see slides.

(b) (0.5) *External interrupts*
The basic recursive equation for worst-case response time analysis is given by

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

Extend the equation with a term for interrupt handling.
Hint: assume a set of sporadic tasks $\mathcal{T}_s$, a minimum inter-arrival time T_k of the interrupt corresponding with sporadic task τ_k, and a cost IH_k for handling the interrupt for τ_k.
Answer: see slides of the lecture.

4. This question concerns the CAN-bus.

   (a) (0.5) Why is "*bit-stuffing*" applied and how does it work?

   (b) (1.0) Briefly explain the arbitration mechanism on a CAN-bus.

   Answers: See slides.

5. The following questions concern Hierarchical Scheduling Frameworks (HSFs) and *independent* applications.

   (a) (0.5) Does the analysis presented in "[Davis et al 05] R.I. Davis and A. Burns, *Hierarchical Fixed Priority Pre-Emptive Scheduling*, In: Proc. 26th IEEE Real-Time Systems Symposium (RTSS), pp. 389-398, December 2005." support independent analysis of applications? Motivate your answer.
   Answer: No, because the interference of servers allocated to higher priority applications is taken into account in the worst-case response time analysis of tasks as well.

   (b) (0.5) Is the analysis described in "[Shin et al 03] I. Shin and I. Lee, *Periodic resource model for compositional real-time guarantees*, In: Proc. 24th IEEE Real-Time Systems Symposium (RTSS), pp. 2-13, December 2003" applicable when a periodic resource with a period Π and capacity Θ is supplied by a deferrable server with the same period and capacity?
   Answer: Yes, because the supply bound function $\texttt{sbf}(t)$ for a periodic resource with period Π and capacity Θ never provides more resources in an interval of length t than a deferrable server with a period Π and a capacity Θ.

6. The following questions concern Hierarchical Scheduling Frameworks (HSFs) and *dependent* applications.

   (a) (1.0) Explain the difference between SIRAP and H-SRP.
   Answer: see papers.

   (b) Temporal isolation between applications is *not* guaranteed by either SIRAP or H-SRP.

      i. (0.5) Explain why for H-SRP.
      Answer: the maximum *execution time of a critical section* guarding global resource access is not enforced.

ii. (1.0) Explain why for SIRAP.
Answer: the maximum *resource-holding time* is not enforced. Note that the maximum resource-holding time *includes* preemptions of tasks with a higher priority than the resource ceiling. Because H-SRP uses the highest priority of tasks within an application as *local* resource ceiling, preemptions of other tasks during global resource access are prevented.