EINDHOVEN UNIVERSITY OF TECHNOLOGY
Department of Mathematics and Computer Science

Examination Real-time Architectures (2IN20)
on Thursday, June 23rd, 2005, 9.00h-12.00h.

This document includes (draft) answers.

First read the entire examination. There are 5 exercises in total. Grades are included between parentheses at all parts and sum up to 11 points. You may use slides and papers for reference purposes. Good luck!

1. There exist three main classes of schedulability tests: *necessary*, *exact*, and *sufficient* tests.

   (a) (0.5) Give definitions for each of these classes of tests and explain the differences between them.
   Answer Let $schedulable(\Gamma, \Sigma)$ denote a boolean predicate that holds if and only if Γ is schedulable using a particular scheduling algorithm Σ. Similarly, let $nt(\Gamma, \Sigma)$, $et(\Gamma, \Sigma)$, and $st(\Gamma, \Sigma)$ denote boolean predicates for a necessary, exact, and sufficient test for the task set Γ using Σ. The three kinds are now characterised by:

   $$\begin{aligned} schedulable(\Gamma, \Sigma) &\Rightarrow nt(\Gamma, \Sigma) \\ schedulable(\Gamma, \Sigma) &\Leftrightarrow et(\Gamma, \Sigma) \\ schedulable(\Gamma, \Sigma) &\Leftarrow st(\Gamma, \Sigma) \end{aligned}$$

   Hence, *et* is both necessary and sufficient, *nt* is *optimistic*, and *st* is *pessimistic*.

   (b) (0.5) Give an example for each of these classes of tests for fixed-priority preemptive scheduling (FPPS).
   Answer Let U denote the utilization of the task set Γ consisting of n hard real-time, periodic tasks, which are denoted by $\tau_1, \tau_2, \ldots \tau_n$. Moreover, let task τ_i be characterised by a (release) period T_i, worst-case computation time C_i, and a deadline D_i. Finally, let the tasks be ordered according to their priority (i.e. lowest number has highest priority).

   - A necessary test for FPPS is given by $U \leq 1$.
   - A suffient test for RMS (!) is given by the so-called Liu and Layland bound $U \leq n(2^{1/n} - 1)$.
   - For an exact test for FPPS, we can check whether or not the worst-case response times R of all tasks are at most equal to their deadlines D, i.e. $R_i \leq D_i$ for $1 \leq i \leq n$. The worst-case response time R_i of task τ_i is given by the smallest positive value satisfying the following recursive equation

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

   (c) (1.0) Describe which conditions should be met in order to make the test applicable.
   Answer The necessary test has a single pre-condition, i.e. the tasks must be periodic.
   The necessary test holds for *any* scheduling algorithm (not just for FPPS).
   The sufficient test only holds for RMS, i.e.

- the tasks are periodic, preemptive, independent, deadlines are (hard and) equal to periods, tasks can immediately start upon release and do not suspend themselves, and
- priorities are assigned inversely proportional to periods (e.g. the task with the shortest period gets the highest priority).

For the exact test, similar conditions hold, with the following exceptions:

- deadlines of tasks can be less than or equal to periods;
- priorities may be assigned arbitrary.

See also a book of Buttazzo, e.g. the terms sufficient and necessary may be found in Section 4.4 in the 1^{st} edition and in Section 4.5 in the 2^{nd} edition.

2. (1.0) Periodic tasks are typically characterized by means of their period T, worst-case computation time C and deadline D. Give an appropriate domain for these three notions (e.g. natural numbers, integer numbers, or real numbers), and motivate your answer.
   Answer The teacher asked this question a number of times to the audience during the lectures, and the students were "warned" that it may be asked during the exam. There are two magic terms: "clock" and "dense intervals" (from the domain of verification of real-time systems). When either of these terms appeared in the solution, the full points were earned.
   A model can be defined in which all tasks parameters are reals, and preemptions are allowed at any time. Alternatively, we may consider a model in which all task parameters are integers, i.e. $T, C, D \in \mathbb{Z}^+$, and preemptions are restricted to integer time points. Typically, the literature is not very strict when domains are concerned and about when preemptions can occur; see a book of Buttazzo for an example.
   If we look at the underlying physical problem, we may conclude that integer task parameters are the proper abstraction, and that restricting preemptions to integer values is preferable. Simply stated: a clock is discrete.

   In the domain of verification of real-time systems, it is argued that a so-called dense domain is required c.q. desired, i.e. between every two poits in time there exists an intermediate point. For a dense domain, the (non-negative) rational numbers can be used.

   Finally, a reason for introducing a model based on for exampe reals is that it may ease proofs of theorems and lemmas, and that in many cases the results can equally well be applied for a discrete model.

3. Consider a hybrid set of tasks Γ, consisting of a set Γ_H of n hard real-time periodic tasks and a set Γ_S of soft real-time tasks. The hard-real time tasks are denoted by $\tau_1, \tau_2, \ldots \tau_n$. We assume fixed-priority preemptive scheduling and arbitrary phasings of hard real-time tasks. A hard real-time task τ_i is characterized by a period T_i, a worst-case computation time C_i, and a deadline D_i. For notational convenience, we assume that task τ_j has a higher priority than task τ_i if and only if $j < i$.
   Consider a system with a deferable server DS for handling the soft real-time tasks next to the hard real-time tasks. The server has a capacity C_{DS} and a period T_{DS}. Assume that the server receives the highest priority.

   (a) (1.5) Derive an equation to determine the worst-case response times of the hard real-time tasks.
   Answer This question is taken from sheet 13 of lecture "RTA.B4-Policies-2". The

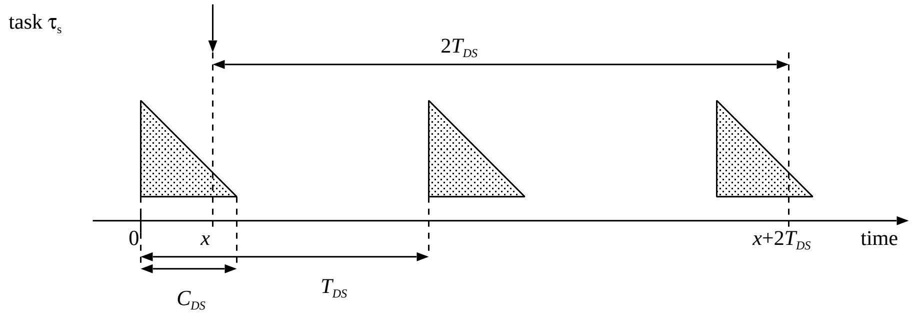

Figure 1: Longest response time of τ_s.

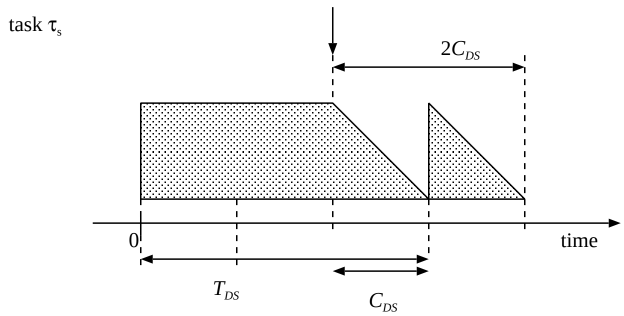

Figure 2: Shortest response time of τ_s.

magic word is "(release) jitter".
The longest continuous interval of time that a deferable server can preempt a hard real-time task is equal to $2C_{DS}$, i.e. when the capacity of the server is consumed as late as possible in one period and as early as possible in the next period. A deferable server can therefore be modelled as a task with a *release jitter* of $T_{DS} - C_{DS}$. We can therefore apply the theory of release jitter to determine the worst-case response time R_i of a hard real-time task τ_i, i.e. R_i is the smallest positive x satisfying the following equation:

$$x = C_i + \lceil \frac{x + T_{DS} - C_{DS}}{T_{DS}} \rceil C_{DS} + \sum_{j<i} \lceil \frac{x}{T_j} \rceil C_j$$

(b) Consider a soft real-time task τ_s with a computation time $C_s = 2C_{DS}$. Assume that the task can immediately start upon its release.

i. (0.5) Determine the longest response time of τ_s.
Answer Consider Figue 1, with a release of τ_s at time x, where $0 < x < C_{DS}$. Note that an amount x of the capacity has already been consumed by another soft task. We immediately see that the longest response time equals $2T_{DS}$.

ii. (0.5) Determine the shortest reponse time of τ_s.
Answer Consider Figure 2, with a release of τ_s at time $T_{DS} - C_{DS}$, where the server was idle from its replenishment at time 0. We immediately see that the shortest reponse time equals $C_s = 2C_{DS}$.

4. Consider three periodic tasks τ_1, τ_2, and τ_3 (having decreasing priority), which share three resources, A, B, and C. Compute the maximum blocking time B_i for each task for the

following two protocols, knowing that the longest duration $D_i(R)$ for a task τ_i on resource R is given by the following table (there are no nested critical sections):

	A	B	C
τ_1	2	0	2
τ_2	2	3	0
τ_3	3	2	5

(a) (1.5) Priority Inheritance Protocol.
Answer See 2^{nd} edition of book of Buttazzo Exercise 7.2.

(b) (1.5) Priority Ceiling Protocol.
Answer See 2^{nd} edition of book of Buttazzo Exercise 7.3.

Note: this exercise has been literally taken from Buttazzo's book.

5. Consider the lecture "Video streaming over network". By using at most 1 page for this entire exercise, explain:

(a) (1.0) Which real-time problems were addressed in the example systems.
Answer Two aspects have been addressed:

- *Application domain characteristics* (user perception issues):
  - both *relative jitter* (frame-rate fluctuations) and *absolute jitter* (drift) are undesirable;
  - deadline misses [frame-skips] are undesirable. Note that to decode a frame, it must be available *entirely*.
- *resource (wireless network) characteristics*:
  - lossy transmissions and (bursty) fluctuating bandwidth;
  - existing protocols either yield *relative jitter* and *deadline misses* of frames (TCP) or packet losses (RTP).

(b) (1.0) How these problems were solved.
Answer

- *application level*: other encoding schemes (e.g. layered MPEG);
- *network level*: combining existing protocols (TCP-RTM):
  - dropping late frames (taken from RTP);
  - re-transmission of lost packages (taken from TCP).

(c) (0.5) Why these approaches were taken.
Answer

- *application level*: exploits relative importance of layers (base layer is the most important);
- *network level*: new protocol combines best of both worlds.

Note that these questions were described as goals on sheet 5 of the lecture "RTA.A1-Overview". Further note that for each of the questions both the *application* and the *system* (or network) level plays a role.