Final Exam for Real Time Systems, 1DT004 (10 credits)

2017 Oct 26, 8:00 – 13:00 (five hours)

Teachers: Pontus Ekberg and Wang Yi

Important Instructions:

1. No course material or computer/calculator are allowed, only writing tools (pen, eraser, ruler etc.) and a dictionary are allowed.
2. Hand in answers to *all* questions (or as many as you can).
3. Please answer all questions precisely and concisely. To help you with this, we indicate how long the answer of each question is (at most) expected to be. *Excessively* verbose answers will not be counted, whereas it is fine to provide shorter answers as long as they contain all relevant information. The word limits are only rough guidelines for you, not strict requirements. (200 words is about 1/2 page in A4 format, depending on your handwriting.)
4. Draw figures when requested. Feel free to do so for other questions as well, if you think they help.
5. Check that this is the correct exam for you:

> This is the written exam for the **10 credits** version of the Real Time Systems course (1DT004).
> If you are registered to the 5 credits version (1DT063), this is the *wrong exam for you.*
> In that case you should immediately talk to the exam administrators to sort it out.

Good luck!

. . . .

Problem 1 (14 points)

1. Describe briefly three reasons for why it is difficult to estimate the worst-case execution time (WCET) of a real-time task? *(100 words)*
2. Describe what the utilization of a task set is. Show how to calculate the utilization $\mathrm{U}(\mathcal{T})$ of a task set $\mathcal{T} = \{\tau_1, \tau_2, \ldots, \tau_n\}$ of n periodic tasks. *(100 words)*
3. Assume that you have a task set $\mathcal{T}$ of implicit-deadline periodic tasks to execute on one preemptive processor. For each of the following statements, indicate whether or not it is true and briefly motivate why.
   (a) $\mathcal{T}$ is EDF-schedulable if $\mathrm{U}(\mathcal{T}) \leqslant 1$.
   (b) $\mathcal{T}$ is EDF-schedulable if $\mathrm{U}(\mathcal{T}) \leqslant 0.69$.
   (c) $\mathcal{T}$ is FP-schedulable with RM priority ordering if $\mathrm{U}(\mathcal{T}) \leqslant 1$.
   (d) $\mathcal{T}$ is FP-schedulable with RM priority ordering if $\mathrm{U}(\mathcal{T}) \leqslant 0.69$.
   (100 words)
4. Describe briefly three features we typically expect a real-time operating system (RTOS) to have (in addition to what is found in a general purpose operating system). *(100 words)*
5. Describe briefly how the CAN bus arbitration mechanism works. *(100 words)*
6. Is it possible to send messages with the same identity from two different nodes on a CAN bus? Why/why not? *(100 words)*
7. Describe briefly two main drawbacks of static scheduling (cyclic executives). *(100 words)*

Problem 2 (6 points)

1. Describe the differences between the `delay` and `delay until` statements in Ada, and a use case for each. *(100 words)*
2. Write the code for an Ada procedure or task that does the following:
   - It reads the current temperature every 100 ms. You can assume that there is a function `Current_Temperature()` available that returns the current temperature in degrees Celsius as a Float.
   - Every second it prints the average temperature of the last ten readings.
   - If it ever reads a temperature higher than 45 °C, it immediately prints the text `WARNING`. It then continues as before (it doesn't stop).

   If you don't remember the Ada syntax exactly, do your best and indicate which parts may have incorrect syntax. Make a note of what those parts as *supposed* to do. *(100 words + code)*

Problem 3 (15 points)

Assume we have a system with one preemptive processor and a task set $\mathcal{T}$ containing the following synchronous periodic tasks:

Task	C_i	D_i	T_i
τ_1	1	4	4
τ_2	3	12	15
τ_3	2	3	5

For a task τ_i, we let C_i denote its WCET, D_i its relative deadline, and T_i its period, respectively. Assume that the tasks are scheduled by a *fixed-priority* (FP) scheduler using *deadline monotonic* (DM) priority ordering.

1. Describe briefly how an FP scheduler works, and how DM priority ordering works. *(100 words)*
2. What is the DM priority ordering of the three tasks above? Is DM a good choice for a priority ordering in this example? Why/why not? *(100 words)*
3. Draw the schedule for the first 15 time units. Make sure that you clearly indicate release times, deadlines and scheduled execution. *(50 words + drawing)*
4. Write down the recurrence relation for calculating the worst-case response time R_i of a task τ_i that is scheduled by an FP scheduler on one preemptive processor. Explain how it is used to calculate the worst-case response time. *(50 words + equation)*
5. Use the formula you gave in part 4 to calculate the worst-case response time of task τ_2. Write down all the steps of the calculation. Based on your calculation, is τ_2 schedulable? Why/why not? *(50 words + calculation)*

Problem 4 (10 points)

Assume we have a system with one preemptive processor and a task set $\mathcal{T}$ containing the following synchronous periodic tasks:

Task	C_i	D_i	T_i
τ_1	1	2	5
τ_2	2	3	4
τ_3	5	16	20

For a task τ_i, we let C_i denote its WCET, D_i its relative deadline, and T_i its period, respectively. Assume that the tasks are scheduled by an *earliest deadline first* (EDF) scheduler.

1. Describe briefly how an EDF scheduler works. *(50 words)*
2. Draw the schedule for the first 25 time units. Make sure that you clearly indicate release times, deadlines and scheduled execution. *(50 words + drawing)*
3. Describe what the hyper-period of a task set is, and what its significance is. Calculate the hyper-period of the task set $\mathcal{T}$. *(100 words)*
4. Is $\mathcal{T}$ EDF-schedulable? Motivate your answer clearly. *(100 words)*

Problem 5 (15 points)

1. Explain the unbounded priority inversion problem that can occur if tasks share locked resources without a suitable resource access protocol. *(200 words)*
2. Explain briefly how the following resource access protocols work.
   (a) The Non-Preemptive Protocol (NPP)
   (b) The Priority Inheritance Protocol (PIP) (sometimes called the *Basic* Priority Inheritance Protocol (BIP))
   (c) The Highest Locker Protocol (HLP) (sometimes called Immediate Priority Ceiling (IPC) or Immediate Priority Inheritance)
   (100 words each)
3. Is it possible to avoid deadlocks using these protocols? Explain your answers. *(200 words)*

Problem 6 (15 points)

1. Explain why worst-case execution time (WCET) analysis is even more difficult on a multicore processor than on a single-core processor. *(200 words)*
2. Describe briefly the following three paradigms in multiprocessor scheduling. Indicate pros and cons for each.
   (a) Global scheduling
   (b) Partitioned scheduling
   (c) Semi-partitioned scheduling
   (100 words each)
3. Assume that you have the following task set $\mathcal{T}$ of sporadic tasks that you want to schedule on a multiprocessor platform using partitioned EDF.

Task	C_i	D_i	T_i
τ_1	3	10	10
τ_2	4	20	20
τ_3	25	100	100
τ_4	3	30	30
τ_5	13	50	50
τ_6	13	100	100
τ_7	120	200	200
τ_8	300	400	400
τ_9	7	20	20
τ_{10}	78	100	100

For a task τ_i, we let C_i denote its WCET, D_i its relative deadline, and T_i its period, respectively. Try to find a partitioning of $\mathcal{T}$ in which all tasks are schedulable and the number of used processors is minimized. Explain your workflow and reasoning clearly. *(400 words)*

Problem 7 (10 points)

In the Digraph Real-Time (DRT) task model, the behavior of a task is expressed by a labeled directed graph, as for the task τ in Figure 1. Each vertex j_i is labeled with a WCET and relative deadline pair $\langle e_i, d_i \rangle$, and each edge (j_i, j_k) is labeled with a minimum inter-release separation delay $p(j_i, j_k)$.

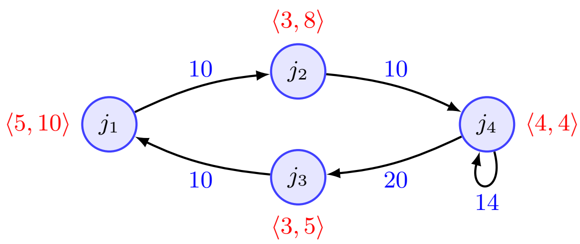

Figure 1: The DRT task τ.

1. Briefly describe the semantics of a DRT task. What job sequences can it generate? *(200 words)*
2. The DRT task model is more expressive than the sporadic task model. Why do we not always use the most expressive task model available? *(100 words)*
3. Draw the demand bound function $\mathrm{dbf}(\tau, t)$ of the task τ in Figure 1, for all values of $t \leqslant 25$. Do this by first computing all *demand pairs* that can fit into an interval of length 25, using the method described in the lectures. Show all steps of your calculations. *(200 words + calculations + drawing)*

Problem 8 (15 points)

The (unfinished) timed automaton in Figure 2 shows a simple light controller. It is supposed to work as follows: In the OFF-location, you may turn it on by pressing its button once or twice. The first button press will switch it from OFF to ON-LIGHT. If the second button press is within 2 seconds of the first button press, it will switch from ON-LIGHT to ON-BRIGHT. If the second button press is more than 2 seconds after the first, it will instead switch to OFF again. Pressing the button in location ON-BRIGHT will always switch it to OFF.

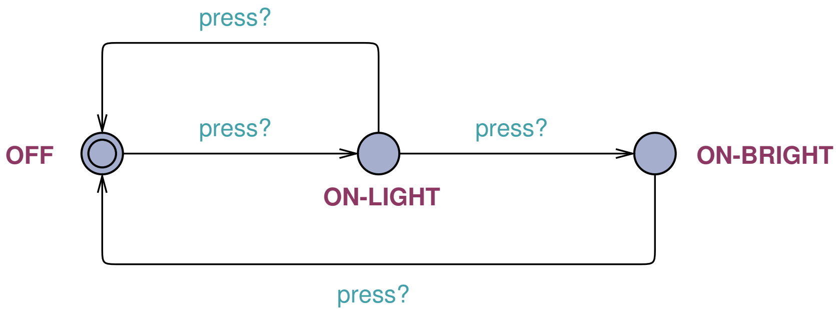

Figure 2: A light controller.

1. Add clock constraints in the automaton to describe the above timing behavior of the controller properly. Draw the resulting automaton. *(50 words + drawing)*
2. Further extend the controller so that it will automatically turn the light off if it has been on continuously (in either ON-LIGHT or ON-BRIGHT) for 1000 seconds. Draw the resulting automaton. *(50 words + drawing)*
3. Based on your automaton for part 2 above, write a query that asserts that the automatic shut-off functions correctly. Write your query using logical symbols (as you would in UPPAAL) and also give an (unambiguous) English translation. *(50 words + query)*