Final Exam for Real Time Systems

2012 Oct 26, 14:00–19:00 (five hours!)

Martin Stigge and Wang Yi

Important Instructions:

1. No course material or computer/calculator are allowed, only a pen and a dictionary.
2. Please mark which course you are registered for:

◯ **5hp** (1DT063)	◯ **10hp** (1DT004)
You need to solve problems 1–4 only.	You need to solve *all* problems.

. . . .

Problem 1 (30 points)

1. Describe briefly the concepts: feasibility, schedulability and optimality in real-time scheduling.
2. Describe briefly three main reasons why it is difficult to estimate the worst-case execution times of tasks.
3. Use examples to explain why Shortest Job First and Non-preemptive EDF are not optimal.
4. Describe briefly the concept of Resource Reservation and two methods for implementation.
5. Explain the semantics of **Delay** and **Delay until** in Ada.
6. Describe briefly how to implement periodic tasks in Ada.
7. Explain briefly how the arbitration mechanism of CAN works.
8. Explain the un-bounded priority inversion problem. Describe briefly how HLP and BIP work.
9. Describe briefly how to improve resource utilization in designing task parameters and how to handle tasks with the same periods in RMS scheduling.
10. What is the most important you have learnt from this course and why?

Problem 2 (30 points)

Consider following task set.

Task	Computing Time	Period
T1	20	100
T2	40	150
T3	20	150
T4	50	200

1. Assume RMS is used and T3 it is released by the end of each execution of T2. What is the release jitter for T3? Motivate your answers.
2. Assume RMS is used and T3 is released non-deterministically during the execution of each instance of T2. What is the release jitter for T3? Motivate your answers.
3. What is the priority order for RMS? Construct the run-time RMS schedule for the first 250 time units. What are the worst case response times of the tasks?
4. Assume EDF is used. Construct the run-time schedule for the first 250 time units. Is the task set schedulable using EDF? Motivate your answer.

Problem 3 (20 points)

Assume a CAN bus running at 0.5Mbits per second, connecting four stations (nodes) A, B, C and D.

1. On node A, there are two tasks. One is sending a message with identity 6 at most every 50ms and the other is sending a message with identity 8 at most every 60ms.
2. On node B, there are two tasks. One is sending a message with identity 9 at most every 100ms and the other is sending a message with identity 2 at most every 10ms.
3. On node C, there is a single task sending a message with identity 5 at most every 20ms.
4. On node D, there are five tasks sending messages with identity 1, 15, 25, 28, 49 and 44 at most every 15ms.

The transmitted messages are of fixed size (135 bits each). Assume that the CAN controllers have sufficient buffer capacity, no transmission errors, and no jitters. Calculate the worst case transmission delay for the messages with identity 8. You must (1) write down the necessary equations, (2) explain the intuitive meaning and (3) calculate a reasonable tight bound.

Problem 4 (20 points)

Assume a set of sporadic tasks with periods T_i, worst-case computing times C_i and deadlines D_i where $D_i \leq T_i$.

1. Describe briefly the DMS priority assignment and run-time behaviour.
2. Describe how the DMS sufficient schedulability test (i.e. using the utilization bound) works.
3. Assume the release jitter for each task is J_i and the overhead for loading or saving the task context is C_S. Assume further HLP is used for resource sharing. Describe how to calculate the worst case response times for each task (explain the meaning of each item in case you use the response time equation).
4. Modify your calculation for non-preemptive tasks.

Problem 5 (20 points)

1. Give two reasons motivating briefly why we should use multicore processors for embedded applications.
2. What is the main difficulty in developing real-time applications on multicore platforms? Explain why.
3. Describe briefly how partitioned scheduling works for multiprocessor systems using EDF and RMS. Assume the following tasks with utilizations: $0.1, 0.4, 0.5, 0.2, 0.2, 0.1, 0.3, 0.2, 0.2, 0.3, 0.1, 0.6, 0.7, 0.9$. Estimate the number of processors needed to run these tasks when EDF and RMS are used for partitioned scheduling. Explain your answers.
4. Describe an algorithm for partitioning a given task set onto a multiprocessor system. Partition the above task set using EDF.

Problem 6 (20 points)

Consider task set $\tau = \{\tau_1, \tau_2\}$ consisting of two sporadic tasks with following parameters for worst-case execution time (e_i), relative deadline (d_i) and period (p_i):

Task	e_i	d_i	p_i
τ_1	4	5	5
τ_2	5	12	19

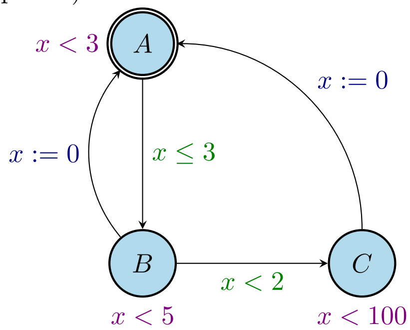
1. Which of the following is a *necessary and sufficient* EDF schedulability test for a sporadic task set $\tau = \{\tau_1, \ldots, \tau_n\}$? (Note that more than one option may be true.)
   - ◯ $\sum_i e_i/p_i \leq 1$
   - ◯ For all $t \geq 0$ it holds that $\sum_{\tau_i \in \tau} dbf_{\tau_i}(t) \leq t$.
   - ◯ For all $t \geq 0$ and all $\tau_i \in \tau$ it holds that $dbf_{\tau_i}(t) \leq t$.
   - ◯ For all $\tau_i \in \tau$ it holds that $R_i \leq d_i$ with R_i being the smallest value satisfying

$$R_i = e_i + \sum_{j<i} \left\lceil \frac{R_i}{p_j} \right\rceil \cdot e_j.$$

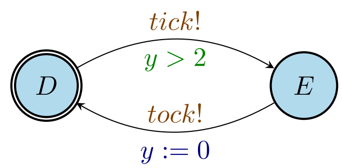
2. Draw the demand bound functions $dbf_{\tau_1}(t)$, $dbf_{\tau_2}(t)$ and $dbf(t)$ over the interval $[0, 40]$. Please use three separate diagrams instead of drawing them into just one.
3. Is τ schedulable with an EDF scheduler? Why/Why not?
4. For each of the four options in part 1 above, state whether it is a sufficient or a necessary EDF schedulability test.

Please turn the page!

Problem 7 (20 points)



(a) Timed automaton T_1



(b) Timed automaton T_2

Figure 1: Two timed automata in which x and y are clocks.

1. Take a look at the timed automaton T_1 in Figure 1(a). You can ignore T_2 for now.
   (a) Describe the set S of reachable states of T_1 (if it's running on its own, i.e., *not* together with T_2).
   (b) For each of the following properties, express it as a temporal formula and assess whether the property is satisfied.
      i. Is location C reachable?
      ii. Is it always the case that x is at least 1 at location B?
      iii. Can x ever have the value of exactly 5?
   (c) Is there an edge that can be removed without altering S?
2. Take a look at the timed automaton T_2 in Figure 1(b). Consider a combined system where T_1 and T_2 run together, i.e., concurrently. Further, add a synchronization constraint *tick*? to T_1 on the edge from A to B, and another synchronization constraint *tock*? to T_1 on the edge from B to A. With the combined system, answer the questions 1(b) and 1(c) from above again.