Final Exam for Real Time Systems

2013 Oct 25, 8:00 – 13:00 (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) You need to solve problems 1–4 only.	◯ **10hp** (1DT004) You need to solve *all* problems.

. . . .

Problem 1 (30 points)

1. (2p) Describe briefly the main difference between embedded and general-purpose computer systems.
2. (2p) Describe briefly three main features of an RT programing language like Ada.
3. (2p) Describe briefly why it is difficult to design predictable real-time systems.
4. (2p) Describe briefly why we need task models.
5. (2p) What is the most reasonable transmission rate for a CAN bus of 100 meters? If it is 200 meters, what is your answer?
6. (4p) Explain briefly how the arbitration mechanism of CAN works.
7. (4p) What is the essential difference between EDF and DMS? Are they optimal? If yes, why?
8. (4p) Describe briefly how to implement periodic tasks in Ada.
9. (4p) Describe briefly two periodic servers.
10. (4p) Assume periodic tasks: A and B where B is released by A during its execution in each period. To reduce the release jitter of B, what will you do? Explain why we should avoid jitters.

Problem 2 (10 points)

Assume a set of periodic tasks (T_i, C_i) where T_i stands for period and C_i for computing time.

1. Describe briefly the RMS priority assignment and the run-time behaviour of a RMS scheduler.
2. Describe how the RMS sufficient schedulability test (i.e. using the utilization bound) works.
3. Describe how to calculate the worst case response times for each task. You may ignore jitters, and overheads for context switch etc. Modify your calculation for non-preemptive tasks.

Problem 3 (10 points)

Explain the un-bounded priority inversion problem. Explain briefly how the following resource sharing protocols work:

1. Basic Inheritance Protocol (i.e. BIP)
2. Immediate Priority Inheritance (i.e. HLP)

Is it possible to avoid deadlocks using these protocols? Explan your answer. If yes, what would you do to avoid deadlocks?

Please turn the page!

Problem 4 (10 points)

Assume a system with one processor and three periodic tasks:

Task	T_i	C_i	D_i
A	260	60	260
B	200	50	60
C	150	50	125

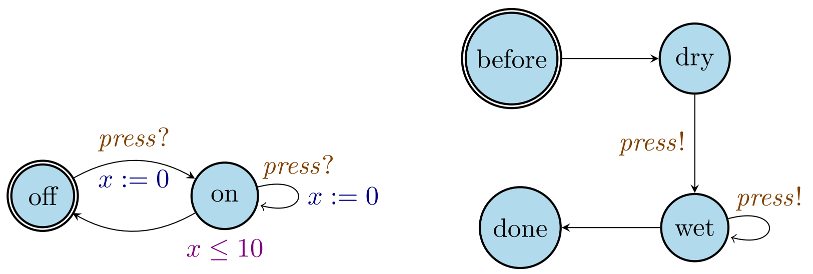
where T stands for period, C for computing time, och D for deadline.

1. Assume that DMS is used to schedule the tasks:
   (a) What is the priority order?
   (b) Construct the run time schedule for the first 260 time units.
   (c) Is the task set schedulable? Motivate your answer.
2. Assume that EDF is used to schedule the tasks:
   (a) Construct the run time schedule for the first 260 time units.
   (b) Is the task set schedulable? Motivate your answer.

Problem 5 (20 points)

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

Problem 6 (20 points)



(a) Timed automaton A_{Shower} (b) Timed automaton A_{User}

Figure 1: Two timed automata synchronizing on a channel *press*.

Take a look at the timed automata A_{User} and A_{Shower} in Figure **??**. They model a user that is about to take a shower.

1. The condition "$x \leq 10$" is a *location invariant* on location "on" of automaton A_{Shower}. Explain what this means.
2. Formulate an UPPAAL query that asks for a deadlock in which A_{User} is *not* in location "done".
3. Is there such a deadlock? Motivate your answer.
4. Describe a change to the model so that we can ask for the property "It is possible that the user is out of the shower but the water is still running for more than 10 more time units". Formulate the query as well. Is it satisfied?
5. Use Part **??** to illustrate the difference between the concepts *Testing* and *Verification.*