# Final Exam for Real Time Systems

**2007 Oct 19, 14-19**

Wang Yi

**Instructions**:

1. You may use a mini-calculator (not a computer!) and a dictionary.
2. Do NOT write on the back side.
3. Put page number on each page.
4. You may write in English or Swedish.
5. State which problems you have solved in the following table.
6. Please handle in this coverage page together with your solutions.

| Problem | Solved | Max. Points | Your Points |
|---|---|---|---|
| 1 | | 30 | |
| 2 | | 12 | |
| 3 | | 12 | |
| 4 | | 12 | |
| 5 | | 12 | |
| 6 | | 12 | |
| 7 | | 12 | |
| | SUMMA: | 100 | |

Name : ..........................................................................

Pers.no. : ..........................................................................

**Problem 1** *(30p)*

1. *What is the most important you have learnt from this course, and why?*
2. *What is the essential difference between hard and soft real-time systems?*
3. *What is the essential difference between testing and verification.*
4. *Explain briefly how the arbitration mechanism of CAN works.*
5. *Explain briefly how to use RMS to schedule a task set containing tasks with the same periods? How about schedulability analysis?*
6. *Explain briefly the concepts: "Multi-tasking" and "Multiprocessor".*
7. *What is the meaning of* **delay 20; x:=10** *in Ada?*
8. *Give three most important features of Ada for real-time programming.*
9. *Are non-preemptive EDF and SJF optimal for scheduling hard real-time tasks?*
10. *What is the difference between DMS and EDF? Are they optimal? If yes, explain in what sense.*

**Problem 2** *(12p) Consider a system consisting of a number of tasks and a watch-dog monitoring the response time of a particular periodic task whose period and deadline are 10 and 8 respectively. The task should inform the watch-dog whenever (1) it arrives (or released) and (2) it finishes its computation within the period. If the deadline is violated, the watch-dog should set a variable WARNING to 1. When WARNING is 1 (initially it should be assigned with 0), the task stops.*

- *Draw two timed automata to model the behaviour of the watch-dog and the task.*
- *Write a property formula in UPPAAL to check that WARNING may become 1.*
- *Transform the two automata into two Ada-like tasks with similar behaviours.*

**Problem 3** *(12p) Assume a system containing three periodic tasks:*

$$\{(200, 1000), (400, 1500), (200, T3)\}$$

*with worst case execution times 200, 400 and 200 and task periods: 1000, 1500 and T3. Assume that the system is scheduled using RMS, the instances of the third task always arrive at the same time as the second task, and it may be released at any time during the execution of the second task. What are the period $T3$, release jitter and worst case response time of the third task?*

**Problem 4** *(12p) Assume a CAN bus running at 0.5 Mbits per second, connecting 4 stations (nodes) A, B, C and D.*

1. *On node A, there are two tasks. One is sending a message with identity 3 at most every 2ms. The other is sending a message with identity 7 at most every 6ms.*
2. *On node B, there is a single task sending a message with identity 8 at most every 10ms.*
3. *On node C, there is a single task sending a message with identity 5 at most every 5ms.*
4. *On node D, there is a single task sending a message with identity 2 at most every 1ms.*

*The transmitted messages are of fixed size (120 bits each). Assume that the CAN controllers have sufficient buffer capacity, no transmission errors, and no jitters. What is the worst case transmission delay (i.e. time from queuing to completed message transmission) for the messages with identity 8? Motivate your answer. You should at least write down the equation to calculate the worst case transmission delay for the message, and explain the meaning of each item in the equation.*

**Problem 5** *(12p) Consider a system consisting of a number of periodic tasks and a sporadic task with worst case execution time 1. Assume that every 60 time units, a sequence of 5 instances of the sporatic task will be released. How do you schedule the mixed set of tasks such that the worst-case response times for all tasks can be calculated? Describe how to calculate the worst case response time for the sporadic task.*

**Problem 6** *(12p)*

1. *Describe the un-bounded priority inversion problem.*
2. *Describe the resource access protocol: BIP (Basic Priority Inheritance Protocol). Use an example if needed.*
3. *Can BIP prevent deadlocks? Motivate your answer.*
4. *The two standard operations P and V on semaphores are implemented according to the following pseudo-code:*

| P(S) | V(S) |
|---|---|
| Disable-interrupt;<br>if S.counter > 0<br>then S.counter -- 1;<br>else {<br>insert(current-task, S.queue);<br>schedule()};<br>Enable-interrupt | Disable-interrupt;<br>If non-empty(S.queue)<br>then {<br>next-to-run := get-first(S.queue);<br>insert(next-to-run, Ready-queue);<br>schedule()}<br>else S.counter ++ 1;<br>Enable-interrupt |

*Modify the above code to implement BIP. You should also describe what information should be kept in the TCB (task control block) and SCB (semaphore control block) for your implementation.*

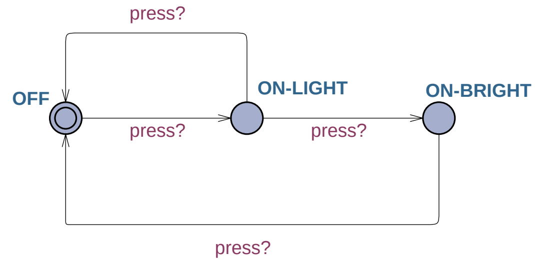


Figure 1: A light controller.

**Problem 7** *(12p) The automaton in Fig. 1 shows a 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. If you press the button twice within 2 seconds, it will switch to ON-BRIGHT; otherwise, it will be on and then off.*

1. Add clock constraints in the automaton to describe the timing behaviour of the controller properly.
2. Add clock constraints such that the controller will turn off automatically when it is on for 1000 seconds.
3. Model a "slow" person with timed automaton, who is trying to turn on the light, but too slow to turn it to ON-BRIGHT.