Final Exam for Real Time Systems

2014 Oct 24, 8:00 – 13:00 (five hours!)

Philipp Rümmer 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 1–5 only.	You need to solve *all* problems.

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. (200 words is about 1/2 page in A4 format).

....

Problem 1 (10 points)

1. Describe briefly three reasons why it is difficult to estimate the worst-case execution times of real-time tasks. *(50 words)*
2. Describe briefly the main advantage and disadvantage of static scheduling. *(50 words)*
3. Describe briefly three major characteristics of RTOS and real-time programming languages. *(50 words)*
4. What is the main difference between DMS and EDF? Are they optimal, and if yes, in what sense are they optimal? *(50 words)*
5. What is the highest processor utilization you can achieve using RMS in a single processor multi-tasking system? *(50 words)*

Problem 2 (10 points)

Assume periodic tasks A and B, where B is released by A when it finishes its execution in each period; i.e., the result of A will be used by B as input in each period.

1. Is it possible to remove the release jitter of B? If yes, explain how. *(100 words)*
2. Explain why we should avoid jitter. *(100 words)*

Problem 3 (15 points)

1. Describe the unbounded priority inversion problem. *(200 words)*
2. Describe how the resource access protocol BIP (Basic Priority Inheritance Protocol) works. Use an example if needed. *(200 words)*
3. Can BIP prevent deadlocks? Motivate your answer with an example. If not, how to avoid deadlocks? *(200 words)*

Please turn the page!

4. The two standard operations P and V on semaphores are implemented according to the following pseudo-code:

P(S)	V(S)

Disable-interrupt;
if S.counter $>$ 0
then S.counter -- 1;
else {
insert(current-task, S.queue);
schedule()};
Enable-interrupt
	Disable-interrupt;
	If non-empty(S.queue)
	then {
	next-to-run := get-first(S.queue);
	insert(next-to-run, Ready-queue);
	schedule()}
	else S.counter ++ 1;
	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. *(200 words + code)*

Problem 4 (15 points)

Assume a system with one processor and three periodic tasks:

Task	T_i	C_i	D_i
A	52	12	52
B	40	10	12
C	30	10	25

where T stands for period, C for WCET, och D for deadline.

1. Assume that $D_i = T_i$ (i.e., ignore the deadlines given in the table) and RMS is used to schedule the tasks. *(200 words + diagrams)*
   (a) What is the priority order?
   (b) Construct the run time schedule for the first 52 time units.
   (c) Is the task set schedulable? Motivate your answer.
2. Assume that DMS is used to schedule the tasks. *(200 words + diagrams)*
   (a) What is the priority order?
   (b) Construct the run time schedule for the first 52 time units.
   (c) Is the task set schedulable? Motivate your answer.
3. Assume that EDF is used to schedule the tasks. *(200 words + diagrams)*
   (a) Construct the run time schedule for the first 52 time units.
   (b) Is the task set schedulable? Motivate your answer.

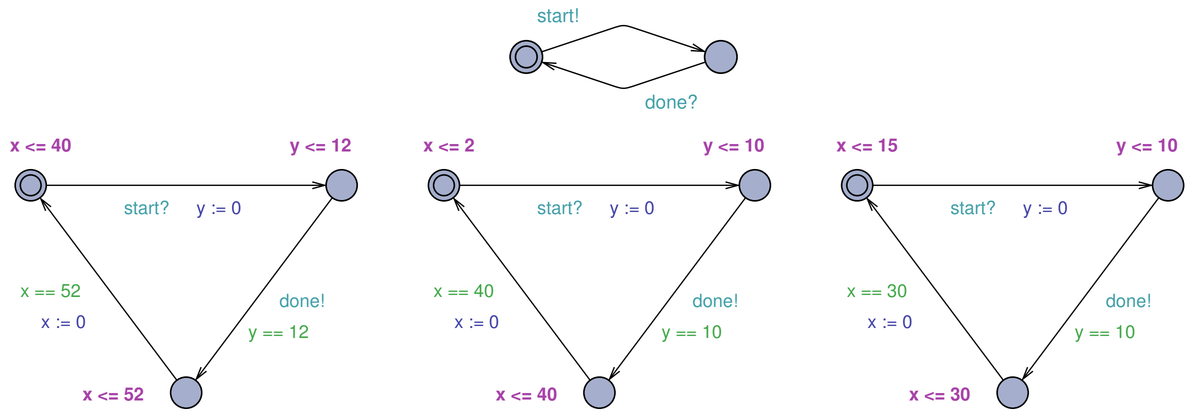
Problem 5 (10 points)

Assume a CAN bus running at 0.5 Mbits per second, connecting 3 stations (nodes) A, B and C.

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 are two tasks sending a message with identity 5 at most every 5ms, and a message with identity 2 at most every 1ms.

The messages are of fixed size (120 bits each). Estimate the worst case transmission delay (i.e., the time period from queueing to completed message transmission) for messages with identity 7. Motivate your answer. *(200 words)*

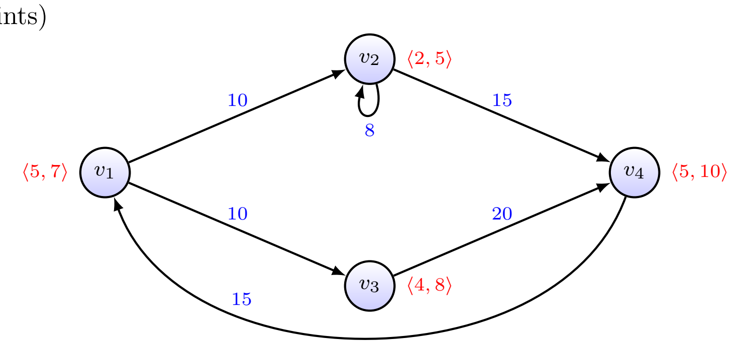
Problem 6 (20 points)



Consider the network of timed automata shown above. All clocks in the automata are local and start at 0, the automata are running concurrently and synchronise on the channels start and done.

1. Are any *deadlocks* reachable in the network? Motivate your answer. *(50 words)*
2. Are any *timelocks* reachable in the network? Motivate your answer. *(50 words)*
3. Assume that there is an additional clock z that starts at 0 and is never reset, and consider the UPPAAL query E$\diamond$ $z = 80$. Express the query in natural language. Is this query satisfied by the system? Motivate using a diagram or table. *(100 words + diagram/table)*
4. Now compare the network to the system in Problem 4, and consider the query E$\diamond$ $z = 1560$. Explain what kind of scheduling is modelled by the timed automata network and the query, and why the constant "1560" was chosen. *(200 words)*

Problem 7 (20 points)



Consider the above task workload in the Digraph (DRT) model. Each node v_i is annotated with WCET and deadline, each edge is annotated with the minimum inter-release delay.

1. Write an Ada program that generates this workload. It is not necessary that your program performs any meaningful computation, it should just use the statements `loop`, `if .. then .. else` and `delay until` to simulate the different jobs and inter-release delays. *(50 words + code)*
2. Use the algorithm introduced in the lectures to draw the demand bound function $dbf(t)$ for this model, for intervals $0 \leq t \leq 30$. *(50 words + diagram)*
3. Is the workload schedulable with an EDF scheduler? Motivate. *(50 words)*

   Hint: you do *not* have to consider the demand bound function to answer this question.
4. Are demand bound functions $dbf(t)$ sub-additive, super-additive, or none of them? Motivate! *(100 words)*

   (A function f is sub-additive if the inequality $f(x + y) \leq f(x) + f(y)$ holds, and super-additive if $f(x + y) \geq f(x) + f(y)$ holds.)