Final Exam, Real Time Systems

2003 Oct 22, 14-19

Wang Yi

Instructions:

1. Do not write on the back side.
2. Do not write more than necessary.
3. Put page number on each page.
4. You may use calculators, text books and lecture notes.
5. You may write in English or Swedish.
6. State which problems you have solved in the following table.
7. Please handle in this coverage page together with your solutions.

Problem	Solved	Max. Points	Your Points
1		20	
2		10	
3		20	
4		10	
5		10	
6		10	
7		20	
	SUMMA:	100	

Name : ....

Pers.no. : ....

Problem 1 (20p) Answer the following questions briefly.

1. What is the essential difference between testing and (formal) verification?
2. Why is FPS (fixed priority scheduling) widely used in practice, but not EDF?
3. What are the main advantages of verification?
4. What are the differences between precise and utilization tests for schedulability?
5. What is the main difference between hard and soft real time tasks?
6. Why is so difficult to achieve *predictability* in implementing a real time system?
7. What is the difference between EDF and DMS?
8. What are QoS, SoC, WCET and UML standing for?
9. What are the differences between modeling and programming (models and programs)?
10. What are the main differences between general-purpose and real-time operating systems

Problem 2 (10p) A watch-dog is monitoring a control software which must reset a variable X to 0 at least once every 100 time units. If it doesn't, the watch-dog should send out a warning. Model the watch-dog using a timed automaton.

Problem 3 (20p)

Describe the un-bounded priority inversion problem **using an example of your own**. Why is the blocking (due to priority inversion) un-bounded?

Assume three tasks: T_1, T_2, T_3 with priority order $P(T_1) > P(T_2) > P(T_3)$. We use a tuple $(A, C_1, P(S), C_2, V(S))$ to represent a task meaning that the tast arrives at time A, computes for C_1 time units, locks semaphore S, computes C_2 time units and then releases S. Let $T_1 = (5, 1, P(S_1), 5, V(S_1))$, $T_2 = (7, 1, P(S_2), 50, V(S_2))$ and $T_3 = (0, 2, P(S_1), 10, V(S_1))$.

Illustrate how NPP, BIP and HLP work at run-time using **the three tasks**. What are the essential differences between NPP and HLP? What are the essential differences between BIP and HLP?

Problem 4 (10p) Using RMS, what is the best processor utilization? How do you achieve it? Give one example.

Problem 5 (10p) Assume a system with one processor and three periodic tasks:

Task	T_i	C_i	D_i
A	300	100	250
B	400	100	120
C	530	120	520

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

1. Assume that RMS is used to schedule the tasks (ignore the deadlines given in the table)
   (a) What is the priority order?
   (b) Draw the run-time schedule for the first 520 time units.
   (c) Is the task set schedulable? Motivate your answer.
2. Assume that DMS is used to schedule the tasks:
   (a) What is the priority order?
   (b) Draw the run-time schedule for the first 520 time units.
   (c) Is the task set schedulable? Motivate your answer.
3. Assume that EDF is used to schedule the tasks:
   (a) Draw the run-time schedule for the first 520 time units.
   (b) Is the task set schedulable? Motivate your answer.

Problem 6 (10p) Assume a set of non-periodic and preemptive tasks each of which is characterized as a pair (C_i, D_i): computing time and Deadline.

1. Explain the notion of *schedulability* and how do you test the schedulability for such task sets without constructing a schedule. (2p)
2. Give an example to show that non-preemptive *Earliest Deadline First* is not optimal. (2p)
3. Explain (prove informally) why Preemptive *Earliest Deadline First* (EDF) is optimal.(4p)
4. Can EDF be used to improve the average response time for soft real time tasks? Motivate your answer.(2p)

Problem 7 (20p) There is a set of tasks sending messages periodically over a bus (or a network) through a controller which decides who should send and who should wait. They send messages of the same size. The controller should guarantee that whenever a task gets the right to send, it should finish sending the same message. Now assume that the period for each task is T_i and the transmission time for a message is C_i (each task sends only one message each period). Design a control algorithm for such a controller, and show how to calculate the maximal delay for a message to reach the receiver sent by each task.