Final Exam for Real Time Systems

2008 Oct 20, 09-14

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		20	
3		20	
4		15	
5		15	
	SUMMA:	100	

Name : ....

Pers.no. : ....

Problem 1 *(30p)*

1. *Is it possible to send messages with the same identity from different nodes on CAN bus?*
2. *Explain briefly how the arbitration mechanism of CAN works.*
3. *Explain why RMS is stable but EDF is not. Are they optimal? If yes, explain in what sense.*
4. *Describe briefly one method to improve the average response times of soft real-time tasks in a system with both hard and soft real-time tasks.*
5. *What is the difference between Deadline-Monotonic Scheduling and Earlist Deadline First?*
6. *Describe briefly two static methods for fault-tolerance.*
7. *Describe briefly two dynamic methods for fault-tolerance.*
8. *Describe briefly two requirements on Operating Systems to be qualified as RTOS.*
9. *What is the best processor utilization can you achieve in RMS scheduling?*
10. *Are non-preemptive EDF and SJF optimal for scheduling real-time tasks with deadlines?*

Problem 2 *(20p) You don't have to use the precise syntax of Ada.*

1. *Program a periodic task as you have done in the Ada assignment, which checks whether a shared variable X is 25 every 100 miliseconds and if it is, it resets X to 0.*
2. *Program another task in Ada, which increases X by 1 every 20 miliseconds.*
3. *If the two tasks run in parallel, is it possible for X to become 26? Is it possible for X to become 1000?*
4. *Model the tasks using timed automata, and*
5. *Write a query in UPPAAL to check whether X can be larger than 25.*

Problem 3 *(20p) Assume a set of periodic tasks.*

1. *Describe briefly the RMS priority assignment and run-time behaviour.*
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.*
4. *Give a sufficient utilization bound for the schedulability of the task set if the tasks are implemented using 5-version programming technique for fault-tolerance (ignore the overheads and assume that each version of the same task has the same worst-case execution time).*
5. *Give a* necessary *utilization bound for the schedulability of the task set if a 5-processor system is used to compute the tasks (ignore the overheads).*

Problem 4 *(15p)*

1. *Describe briefly the concept of periodic server.*
2. *Design a polling server for a sporadic task with worst case execution time C, deadline D, and minimal inter-arrival time T, and*
3. *Describe briefly how to calculate the worst case response times for the task.*

Problem 5 *(15p)*
(1) Explain the un-bounded priority inversion problem. (2) Explain briefly how the following priority ceiling protocol work:

- *Basic Inheritance Protocol*
- *Immediate Priority Inheritance*

Is it possible to avoid deadlocks using these protocols? Explan your answers.