Final Exam for Real Time Systems

2010 Apr 06, 08-14

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. State which problems you have solved in the following table.
5. Please handle in this coverage page together with your solutions.

Problem	Solved	Max. Points	Your Points
1		20	
2		20	
3		20	
4		10	
5		15	
6		15	
	SUMMA:	100	

Name : .....

Pers.no. : .....

Are you taking the course for 4.5p or 7.5p ? .....

Problem 1 *(20p)*

1. *Describe briefly three reasons why it is difficult to design predictable RT systems.*
2. *Describe briefly three major characteristics of RTOS.*
3. *Describe briefly the main drawnbacks of static scheduling.*
4. *What is the main difference between DMS and EDF?*
5. *What is the highest processor utilization can you achieve using RMS in a single processor multi-tasking system?*
6. *Describe briefly two methods to improve the average response times of soft real-time tasks.*
7. *Describe briefly how global scheduling works for multiprocessor systems. Is there any optimal scheduling algorithm for multiprocessor systems?*
8. *Is it possible to send messages with the same identity from different nodes on CAN bus? Why?*
9. *Explain briefly how the arbitration mechanism of CAN works.*
10. *Describe briefly two reasons why we need modelling and verification.*

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

1. *Program a task RESET, which checks whether a shared variable X is 50 every 1000 miliseconds and if it is, it resets X to 0.*
2. *Program another task ADD-ONE, which increases X by 1 every 20 miliseconds.*
3. *Program a WATCH-DOG. It checks the value of X every 100 miliseconds. If X is larger than 50, it will set the variable Warning to 1, and tell ADD-ONE to stop.*
4. *Model the tasks using timed automata, and*
5. *Write a query in UPPAAL to check whether X can be larger than 50. Is it possible for X to become 51?*

Problem 3 *(20p)*

1. *Describe briefly how partitioned scheduling works for multiprocessor systems using EDF and RMS. Describe briefly how to estimate the number of processors needed for each case.*
2. *Assume a set of 9 tasks with utilizations:* $0.5, 0.7, 0.5, 0.2, 0.4, 0.2, 0.5, 0.1, 0.6$. *How many processors needed to run these tasks when EDF and RMS are used for partitioned scheduling? Explain your answers. Assign the tasks to the processors for each of the two cases.*

Problem 4 *(10p) Assume a set of periodic tasks where the task deadlines are shorter than their periods.*

1. *Describe briefly the DMS priority assignment and run-time behaviour.*
2. *Describe how the DMS 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 5 *(15p)*

1. *Describe briefly the concept of periodic server.*
2. *Design a polling server for a non-periodic task with worst case execution time* C *and deadline* D.
3. *Describe how to estimate the period of the server.*

Problem 6 *(15p)*
(1) Explain the un-bounded priority inversion problem. (2) Explain briefly how the following priority ceiling protocol work:

- *Basic Inheritance Protocol (i.e. BIP)*
- *Immediate Priority Inheritance (i.e. HLP)*

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