Parallel and Distributed Real-Time Systems
EDA421/DIT171

Final Exam, June 04, 2013 at 8:30 – 12:30 in the V building

Examiner:
Professor Jan Jonsson

Questions:
Risat Pathan, phone: 076 214 8509

Content:
The written exam consists of 4 pages (including cover), containing 7 problems worth a total of 60 points.

Grading policy:
24–35 ⇒ grade 3
36–47 ⇒ grade 4
48–60 ⇒ grade 5

Restrictions:
Books, notes or calculators are NOT allowed (only writing material and dictionaries)

Solution:
No solution provided.

Results:
Posted on the course home page on Tuesday, June 18, 2013 at 09:00.

Inspection:
Room 4128, Rännvägen 6 B, on Tuesday, June 18, 2013 at 11:00–12:00. Inspection at another occasion could be arranged by contacting the course examiner.

Language:
Your solutions should be written in English.

IMPORTANT ISSUES

1. Use separate sheets for each answered problem, and mark each sheet with the problem number.
2. Justify all answers. Lack of justification can lead to loss of credit even if the answer might be correct.
3. Explain all calculations thoroughly. If justification and method is correct then simple calculation mistakes do not necessarily lead to loss of credit.
4. If some assumptions in a problem are missing or you consider that the made assumptions are unclear, then please state explicitly which assumptions you make in order to find a solution.
5. Write clearly! If I cannot read your solution, I will assume that it is wrong.

Good Luck!

PROBLEM 1

State whether the following propositions are TRUE or FALSE. Each correct statement will give 0.5 points; each erroneous statement will give -0.5 points; an omitted statement gives 0 points. Although a motivation for a correct answer is not required, a convincing one gives another 0.5 points, while an erroneous/weak one gives another -0.5 points. **Quality guarantee**: The total result for this problem cannot be less than 0 points. (6 points)

a) In token ring protocol, a node can grab a "free" token to transmit message even if the priority of its pending message is *lower* than the priority in PPP field of the token.

b) Recovery block for fault-tolerant scheduling is based on *active* redundancy.

c) The largest possible *competitive factor* of a scheduling algorithm is equal to 1.

d) If periodic tasks are never released simultaneously, then *hyper-period analysis* is currently the only way to perform exact feasibility tests on asynchronous task sets.

e) Global earliest-deadline first (EDF) scheduling on multiprocessors does not suffer from *Dhall's effect*.

f) The synthetic utilization guarantee bound for deadline-monotonic scheduling of *liquid tasks* on uniprocessor is larger than 50%.

PROBLEM 2

When designing a real-time system it is customary to view the work as consisting of the three major phases: specication, implementation and verification.

a) How do you measure/compare performance of different designs of real-time systems? (2 points)

b) Describe the fundamental design choices that must be made in the implementation phase (3 points)

c) Describe three different sources of uncertainty that exist in formal verification? (3 points)

PROBLEM 3

For multiprocessor scheduling, two different policies for assigning tasks to multiprocessors are Branch and Bound (B&B) and Simulated Annealing (SA).

a) What is the main idea for assigning tasks to processors using Branch and Bound (B&B) algorithm? State four mechanisms to avoid exhaustive search in B&B algorithm. (4 points)

b) Describe how the principles of Simulated Annealing (SA) algorithm can be adapted to solve the real-time multiprocessor scheduling problem. (2 points)

c) Let a DPPE task set be a set of tasks with deadline, precedence, preemption and exclusion constraints. The DPPE schedulability problem that is, the problem of determining (hard real-time) schedulability on a uni-processor for a DPPE task set is NP-complete. The B&B algorithm by Xu and Parnas (as studied in homework assignment #1) solves the optimization problem of finding the schedule (if one exists) with the minimum possible task lateness for a DPPE task set on a uni-processor. Given the NP-complete DPPE schedulability problem, show that the corresponding DPPE optimization problem (as solved by Xu and Parnas algorithm) is NP-hard. (4 points)

PROBLEM 4

Consider a task set Γ where each task $\tau_i \in \Gamma$ is characterized by (C_i, D_i, T_i) such that $C_i \leq D_i \leq T_i$. The following questions are related to scheduling synchronous periodic tasks on one processor.

a) Assume that all the deadlines of some arbitrary task set Γ are met using *non-preemptive* deadline-monotonic scheduling on one processor. Can you always guarantee that all the deadlines of the task set Γ are also met using fully preemptive deadline-monotonic scheduling on one processor? Why or why not? Motivate your answer! (3 points)

a) Assume that all the deadlines of some arbitrary task set Γ are met using fully *preemptive* deadline-monotonic scheduling on one processor. Can you always guarantee that all the deadlines of the task set Γ are also met using non-preemptive deadline-monotonic scheduling on one processor? Why or why not? Motivate your answer! (3 points)

b) Consider a task set Γ that is *feasible* using fully-preemptive deadline-monotonic scheduling on one processor where $D_i = T_i$ for each task. Now consider a *new* schedulability test which is defined as follows: the Audsley's *Optimal Priority Assignment* (OPA) algorithm is applied using the Liu and Layland's utilization-bound-based schedulability condition. Can you guarantee that Γ also satisfies this new test? Why or why not? Motivate your answer! (4 points)

c) There exists a pseudo-polynomial time schedulability test to decide if a synchronous periodic task set can be preemptively scheduled using fixed priorities on one processor in such a way that all task instances will complete by their deadlines. State the mathematical expression of the test and also explain why does this test has pseudo-polynomial time complexity. (2 points)

PROBLEM 5

The following questions are related to scheduling periodic tasks on multiprocessor.

a) What does it mean by scheduling anomalies? Describe the different kinds of scheduling anomalies in multiprocessor systems? (4 points)

b) Consider a task set Γ where each task $\tau_i \in \Gamma$ is characterized by (C_i, D_i, T_i) such that $C_i \leq D_i \leq T_i$. The response time R_i of task τ_i for global fixed-priority scheduling can be computed using the following recursive definition:

$$R_i^{n+1} = C_i + \frac{1}{m} \sum_{j \in hp(i)} \left(\left\lceil \frac{R_i^n}{T_j} \right\rceil \cdot C_j + C_j \right)$$

where $hp(i)$ is the set of all the higher priority tasks of τ_i. If $R_i \leq D_i$ for all the tasks in set Γ, then the task set is schedulable using global fixed-priority scheduling. However, this test is a *sufficient* test rather than an exact test. Describe two reasons why this test is not an exact test. (3 points)

c) Describe the main idea of distributed Priority Ceiling Protocol (PCP) for handing shared resources in multiprocessor systems. (3 points)

PROBLEM 6

This question is related to scheduling both period and aperiodic tasks on uniprocessor using server-based approach for servicing the aperiodic requests. Consider the following two periodic tasks τ_1 and τ_2 where the deadline of each task is equal to its period. Each periodic task arrives at time 0.

	τ_1	τ_2
C_i	1	3
T_i	2	5

Also consider two aperiodic tasks τ_x and τ_y where the arrival time (a_i) and worst-case execution time (C_i) of these two aperiodic tasks are given as follows:

	τ_x	τ_y
a_i	2	9
C_i	2	2

The aperiodic tasks are serviced using a server task τ_s with period $T_s = 4$ and capacity $C_s = 1$. The periodic tasks including the server tasks are scheduled using uniprocessor rate-monotonic scheduling.

a) Draw the schedule from time $t = 0$ to $t = 20$ where the server task is a polling server? (3 points)

b) Draw the schedule from time $t = 0$ to $t = 20$ where the server task is a deferrable server? (3 points)

c) Based on the solution of subproblems a) and b), which server-based approach is better for reducing the response time of the aperiodic tasks? Why? (2 points)

PROBLEM 7

This question is related to scheduling hard aperiodic tasks on multiprocessors based on server-less approach. Consider the following set of seven aperiodic tasks where the arrival time (a_i), worst-case execution time (C_i) and the deadline (D_i) of the seven tasks are given as follows:

	τ_1	τ_2	τ_3	τ_4	τ_5	τ_6	τ_7
a_i	1	2	3	4	8	8	8
C_i	3	6	1	5	2	4	4
D_i	6	7	6	7	20	20	7

a) These aperiodic tasks are scheduled using EDF-FF algorithm. Show how the tasks are assigned to $m = 3$ processors using EDF-FF algorithm so that each task meets its deadline. (4 points)

b) Describe the main principle of focused addressing and bidding used for task forwarding. (2 points)
