Parallel and Distributed Real-Time Systems
EDA421/DIT171

Final Exam, June 02, 2014 at 14:00 – 18:00

Examiner:
Professor Jan Jonsson

Questions:
Behrooz Sangchoolie, phone: 073 571 7799

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 Monday, June 23, 2014 at 09:00.

Inspection:
Room 4128, Rännvägen 6 B, on Monday, June 23, 2014 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) Measurement-based approach to determine the WCET of a function is often used in the industry.

b) Pandya and Malek proposed fault-tolerant scheduling algorithm to tolerate a single fault where a task is re-executed at higher priority level than the priority of the original task.

c) A *necessary* feasibility test for Timed-Token Protocol is that the deadline of each message transmission must be less than the *target token-rotation time*.

d) There exists a *dynamic-priority* based global scheduling algorithm with utilization bound of 50%.

e) Switched Ethernet provides *contention-free* communication.

f) Priority Inheritance Protocol is not a deadlock-free protocol.

PROBLEM 2

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

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

b) Do you agree or not that the deadlines of some tasks may be allowed to be missed in some real-time applications? Motivate your answer using an example. (3 points)

PROBLEM 3

Consider the Multiprocessor Scheduling decision problem — that is, the problem of determining schedulability for a set of independent tasks with a common deadline using a set of identical processors.

a) State the steps used for proving that a decision problem, such as the Multiprocessor Scheduling problem, is NP-complete. (2 points)

b) The Multiprocessor Scheduling problem is NP-complete in the strong sense for an arbitrary number of processors. Describe the meaning of strong NP-completeness. (4 points)

c) The Multiprocessor Scheduling problem is solvable in pseudo-polynomial time for any fixed number of processors. Describe the meaning of pseudo-polynomial time algorithm. (4 points)

PROBLEM 4

Consider 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 uniprocessor.

a) Assume that all the deadlines of some arbitrary task set Γ are met using *non-preemptive* earliest-deadline-first (EDF) scheduling on uniprocessor. Can you always guarantee that all the deadlines of this task set Γ are also met using preemptive EDF scheduling on uniprocessor? Why or why not? (4 points)

b) Assume that all the deadlines of some arbitrary task set Γ are met using *preemptive* EDF scheduling on uniprocessor. Can you always guarantee that all the deadlines of this task set Γ are also met using non-preemptive EDF scheduling on uniprocessor? Why or why not? (4 points)

c) Assume that all the deadlines of some arbitrary task set Γ are met using *preemptive* EDF scheduling on uniprocessor. Can you always guarantee that all the deadlines of this task set Γ are also met using preemptive deadline-monotonic (DM) scheduling on uniprocessor? Why or why not?

(4 points)

PROBLEM 5

The following questions are related to scheduling tasks on multiprocessors.

a) Consider a task set Γ having $n = 50$ periodic tasks that are to be scheduled on a multiprocessor platform having $m = 3$ processors. The deadline of each task is equal to its period. The total utilization of the task set is 1.25. You have three following alternatives to select the scheduling algorithm:

- Option 1: RM-US
- Option 2: RM-FF
- Option 3: P-fair

Which algorithm you will select for the task set? Why? What advantages and disadvantages your selected algorithm has over the other two alternative algorithms? (6 points)

b) State the response-time test proposed by Andersson and Jonsson for global fixed-priority scheduling. Explain why this test is not an exact test. (3 points)

c) For static (off-line) multiprocessor scheduling, a useful method is the branch-and-bound (B&B) algorithm. Describe briefly *three* methods that can be used to avoid examining all possible schedules in search for the optimal schedule. (3 points)

PROBLEM 6

This question is related to scheduling both periodic and aperiodic tasks on uniprocessor using *server-based approach.* 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 first arrives at time 0.

	τ_1	τ_2
C_i	3	2
T_i	9	6

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 aperiodic tasks are given as follows:

	τ_x	τ_y
a_i	2	8
C_i	3	3

a) Draw the schedule from time $t = 0$ to $t = 18$ assuming *background scheduling*? (3 points)

b) The aperiodic tasks are serviced using *deferrable server* approach where the server task τ_s has period $T_s = 7$ and capacity $C_s = 2$. Draw the schedule from time $t = 0$ to $t = 18$? (3 points)

c) Based on the solution of subproblems a) and b), which approach is better? Why? (2 points)

PROBLEM 7

This question is related to uniprocessor scheduling of hard aperiodic tasks based on *server-less approach*. Consider the following set of four aperiodic tasks where the arrival time (a_i), worst-case execution time (C_i) and the deadline (D_i) of the tasks are given as follows. The ratio between C_i and D_i of each task is given in the last row of the following table. These tasks shares no resources and uniprocessor *deadline-monotonic* (DM) scheduling will be used to schedule the tasks.

	τ_1	τ_2	τ_3	τ_4
a_i	2	2	3	5
C_i	1	2	$2(\sqrt{2}-1)$	1
D_i	10	5	2	10
C_i/D_i	0.1	0.40	0.4142	0.1

The generalized synthetic guarantee bound U^{ζ}_{LB} for time-independent uniprocessor scheduling of aperiodic tasks is given as follows:

$$U^{\zeta}_{LB} = 1 + \alpha - \sqrt{1 + \gamma + \alpha^2}$$

where α is the preemptable-deadline ratio and γ is the resource-blocking ratio.

a) What is the value of α and γ for the above task set where the time-independent scheduling algorithm is the deadline-monotonic scheduling? (2 point)

b) Can you accept (i.e., guarantee schedulability) each of the four tasks as it arrives based on the above guarantee bound? Why or why not? (4 point)
