Parallel and Distributed Real-Time Systems
EDA421/DIT171

Final exam, December 14, 2011 at 14:00 – 18:00 in the V building

---

Examiner:

Docent Jan Jonsson
Phone: 031–772 5220

Content:

The written exam consists of 5 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 Wednesday, December 28, 2011 at 09:00.

Inspection:

Room 4128, Rännvägen 6 B, on Tuesday, January 3, 2012 at 13:00–15:00. 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, it is wrong.

Good Luck!

PROBLEM 1

State whether the following propositions are TRUE or FALSE. For each correct statement, you will be given 1 point; for each erroneous statement you will be given -1 point; if you make no statement at all, you will be given 0 points. **Quality guarantee**: The total result for this problem cannot be less than 0 points. (6 points)

a) One of the so-called *Richard's anomalies* states that task completion times may increase as a result of increasing the number of processors.

b) The Token Ring (IEEE 802.5) message format allows for 32 different priority levels.

c) The *competitive factor* of a scheduling algorithm is a measure of the number of tasks that meet their deadlines in a schedule generated by the algorithm.

d) Generating a time-table for static scheduling in a real-time system is so time consuming that it can never be done at run-time.

e) If a given task set is known to be not schedulable, a *sufficient* feasibility test will always report the answer "no" when applied to that task set.

f) By *high accuracy* in a clock synchronization context, we mean a small deviation between clocks.

PROBLEM 2

The following questions are related to estimation of worst-case execution times (WCET) of tasks.

a) Explain what the term *infeasible path* refers to in the context of WCET analysis and provide an example of such a path. Also explain why it is useful to identify infeasible paths. (3 points)

b) In the case that preemptive scheduling is used in a system, is it appropriate to include the cost for task preemptions in the worst-case execution time of a task? Justify your answer! (3 points)

PROBLEM 3

The following questions are related to the CAN (Controller Area Network) technique.

a) State whether communication medium access in CAN is token-based or contention-based. Motivate your answer. (1 points)

b) Describe the message format used in CAN. (1 points)

c) Describe the *binary countdown* algorithm as used in CAN. (3 points)

d) Describe how response-time-based schedulability analysis can be adapted to message scheduling in CAN. (3 points)

PROBLEM 4

For static (offline) multiprocessor scheduling, two useful algorithms are Simulated Annealing (SA) and Branch-and-Bound (B&B).

a) Describe the major functional differences between the SA and B&B algorithms. (4 points)

b) Describe the advantages and disadvantages of each algorithm. (2 points)

c) Describe the mechanisms used by each of the algorithms to reduce the average time complexity for finding a feasible solution. (2 points)

PROBLEM 5

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) Describe the procedure used for proving that a decision problem, such as the MULTIPROCESSOR SCHEDULING problem, is NP-complete. (4 points)

b) The corresponding MULTIPROCESSOR SCHEDULING optimization problem is said to be NP-hard. Describe the formal meaning of NP-hardness. (2 points)

c) Describe the problem classes P and NP. (4 points)

PROBLEM 6

The following questions are related to the management of on-line changes in a real-time system.

a) Give three underlying causes for on-line changes in a real-time system. (3 points)

b) Describe the main differences between a *guarantee scheme* and a *robust scheme* for task admission in a real-time system. (3 points)

c) Describe how the Deferrable Server by Lehoczky *et al.* handles the arrivals of aperiodic tasks in a system with rate-monotonic scheduling. Use a figure to illustrate how the server capacity is administrated. (3 points)

d) With the Deferrable Server, it is possible to achieve a utilization guarantee bound, $U_{\mathrm{RM+DS}}$, that is higher than the bound by Liu and Layland, $U_{\mathrm{RM}} = n(2^{1/n} - 1)$, for rate-monotonic scheduling. Indicate for what values of the server utilization U_S this is possible. (1 points)

PROBLEM 7

Consider the following policies for server-less scheduling of aperiodic tasks on multiprocessor systems:

- Deadline-Monotonic with Liquid Tasks (DM-LT)
- Earliest-Deadline-First Utilization Separation (EDF-US)
- Earliest-Deadline-First First-Fit (EDF-FF)

a) Explain the concept of *liquid tasks* (as used in DM-LT.) (1 points)

b) Explain the idea behind *utilization separation* (as used in EDF-US.) (2 points)

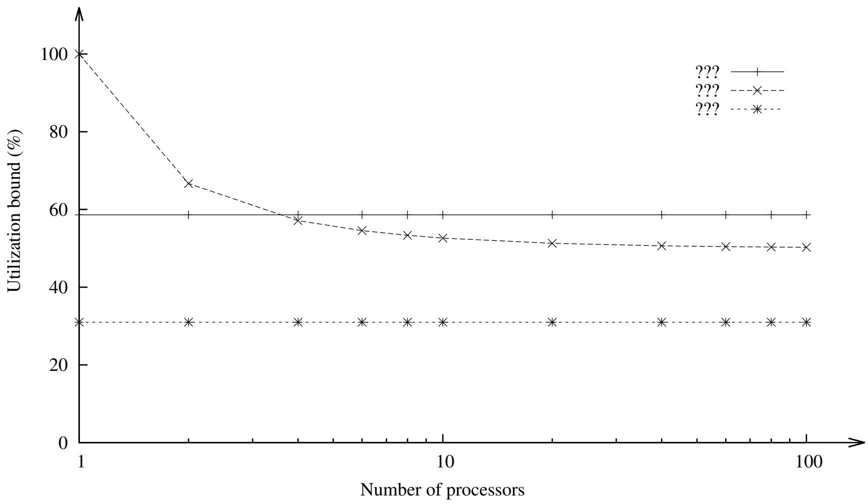
c) Explain the general *first-fit* assignment algorithm (as used in EDF-FF.) (2 points)

d) Explain the *time-independent* scheduling policy (as used in DM-LT.) Also give an example of a well-known scheduling policy that is <u>not</u> time-independent. (2 points)

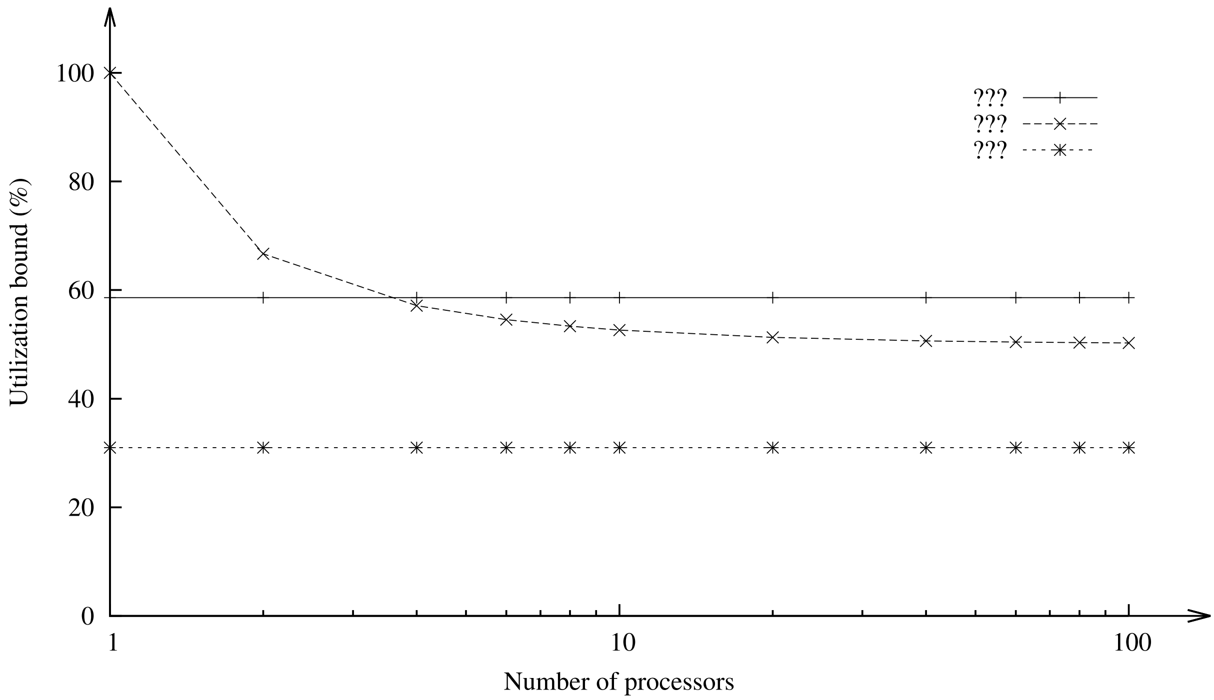
e) Explain the *priority-driven* scheduling policy (as used in EDF-US.) Also give an example of a well-known scheduling policy that is <u>not</u> priority-driven. (2 points)

f) For each of the listed scheduling policies, there is a guarantee *utilization bound*, which the per-processor utilization of the task set should not exceed in order to guarantee that the tasks will meet their deadlines. Note that for DM-LT, EDF-US and EDF-FF the bound refers to the *synthetic* utilization of the aperiodic tasks.

The diagramme below shows the guarantee utilization bound as a function of the number of processors for the different scheduling policies.



Complete the diagramme by associating each utilization-bound plot with the correct scheduling policy, and give a motivation for your choice. The diagramme is also available on a separate sheet at the end of the exam. Indicate your choices on that sheet and hand it in for grading together with the rest of your solutions. (3 points)



Extra copy of the diagramme in Problem 7 f).