Parallel and Distributed Real-Time Systems
EDA421/DIT171

Final exam, December 17, 2008 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 Monday, January 5, 2009 at 09:00.

Inspection:

Room 4128, Rännvägen 6 B, on Monday, January 5, 2009 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) The *jitter* of a periodic task in a schedule reflects the amount of deviation from the expected periodicity of the task's completions.

b) In the Token Ring communication network, it is impossible to guarantee messages with hard deadlines.

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

d) An optimization problem can never be classified as an NP-complete problem.

e) When there are mutual exclusion constraints in a system, it is impossible to find an optimal on-line scheduling algorithm (unless it is clairvoyant).

f) In the Fault-Tolerant Average algorithm for clock synchronization, the new clock value is the mean of all collected clock readings excluding the t fastest and t slowest 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) Give a short description of how the IPET (Implicit Path Enumeration Technique) for WCET analysis works. (3 points)

c) Explain why WCET analysis for a multiprocessor system will be more complicated than for a single processor. (2 points)

PROBLEM 3

There are several different approaches to preemptiveness and priority rules during scheduling.

a) Describe the difference between *greedy* and *p-fair* scheduling. For each method, list some advantages and disadvantages. (4 points)

b) Describe the concept of *time-independent* scheduling. Also give an example of a known scheduling policy that does not fall into this category. (2 points)

c) Describe the concept of *priority-driven* scheduling. Also give an example of a known scheduling policy that does not fall into this category. (2 points)

PROBLEM 4

For static (off-line) multiprocessor scheduling, a useful method is the *branch-and-bound* (B&B) algorithm.

a) In many B&B algorithms, the search for an optimal schedule begins with an empty schedule. Describe the general approach used for extending an empty or partial schedule towards a complete schedule (that is, how child vertices are generated.) (3 points)

b) For the type of B&B algorithm mentioned in subproblem a), describe briefly four methods that can be used to avoid examining all possible schedules in search for the optimal schedule. (4 points)

c) In the B&B algorithm by Xu and Parnas, the search for an optimal schedule instead begins with an initial complete schedule. Explain briefly how new improved candidate schedules are generated in that algorithm. In your answer it is not necessary to describe all variables involved in the generation of child vertices, but rather your answer should give the intuition behind the choice of candidates. (3 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 MULTIPROCESSOR SCHEDULING problem is NP-complete in the strong sense for an arbitrary number of processors. Describe the meaning of strong NP-completeness. (2 points)

c) The MULTIPROCESSOR SCHEDULING problem is a so-called number problem. Describe the formal characteristics of a number problem, and how it relates to strong NP-completeness. (2 points)

PROBLEM 6

One method for analyzing the performance of a system that experiences overload is to use the concept of *competitive factor*.

a) Give the formal definition of competitive factor. (4 points)

b) What is the best achievable competitive factor for an on-line scheduler according to Baruah *et al.*? Under what assumptions does this result apply? (2 points)

c) Give the competitive factor for the earliest-deadline-first (EDF) scheduling policy. Also comment on the implications of this result. (2 points)

PROBLEM 7

Consider the following multiprocessor scheduling policies:

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

RM, RM-US and RM-FF are used for periodic task scheduling, while DM-LT, EDF-US and EDF-FF are used for aperiodic task scheduling.

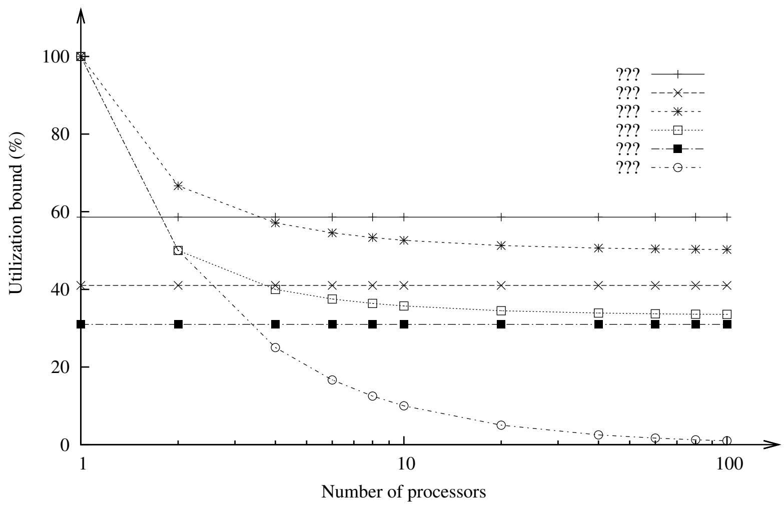
a) Explain the idea behind *utilization separation* (as used in RM-US and EDF-US.) (2 points)

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

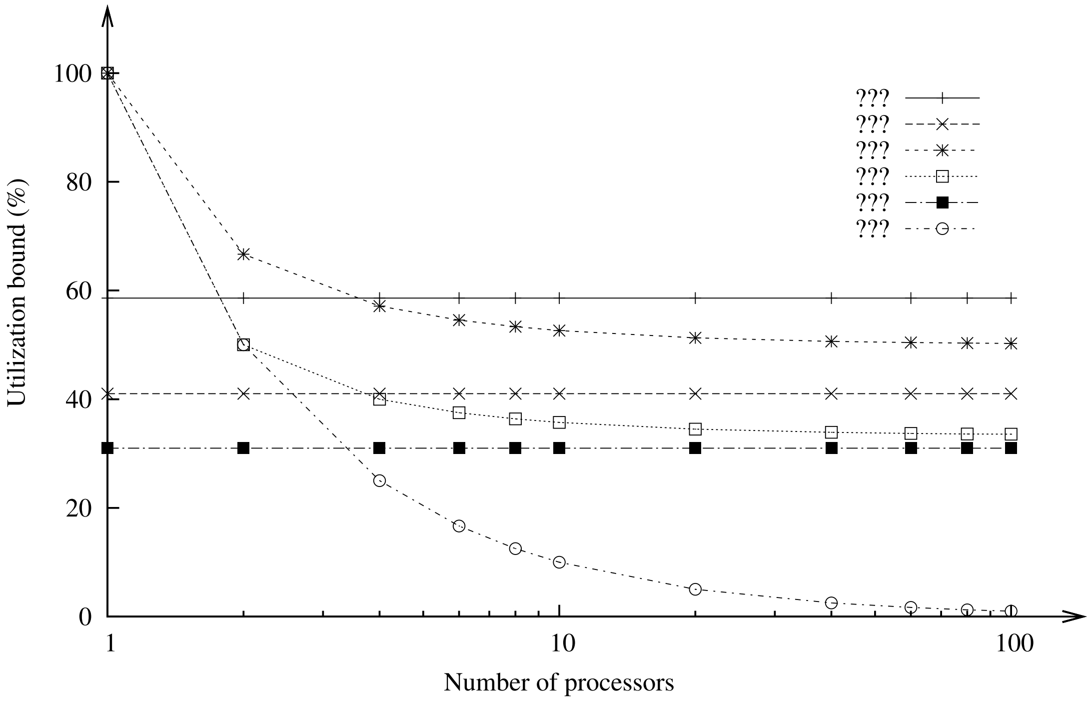
c) Explain the general *first-fit* assignment algorithm (as used in RM-FF and EDF-FF.) (3 points)

d) 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: 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. (6 points)



Extra copy of the diagramme in Problem 7 d).