Scheduling Algorithms (1)

Real-Time and Embedded Systems (M)

Tutorial 2

UNIVERSITY
of
GLASGOW

Tutorial Outline

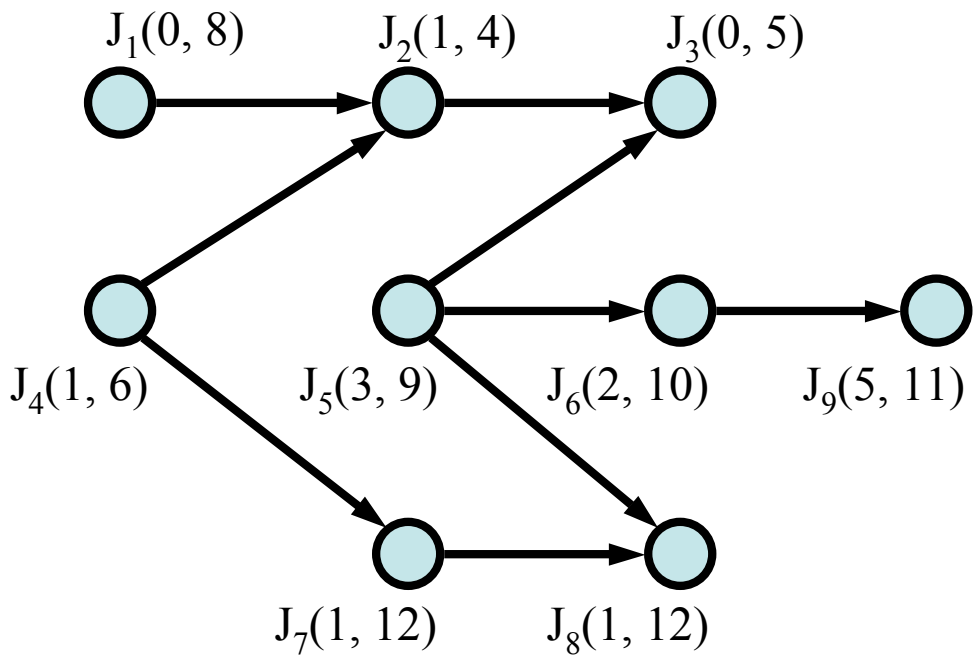
- Review of problem set 1
- Review of lectures
- Worked examples
- Question & answer

Review of Problem Set 1

- Question 1: Briefly explain what is a real time system, and discuss the difference between hard and soft real time systems. Discuss, with examples, whether the distinction between hard and soft real time systems is always clear cut. [5 marks]

Review of Problem Set 1

- Question 2: The task diagram below shows a set of jobs with the feasible interval for each listed next to the job name. The arrows in the diagram indicate the precedence relations between the jobs. Calculate the effective release times and deadlines of the jobs, and demonstrate whether they can be scheduled using the EDF algorithm. Each job executes for 1 time unit. The system has one processor. [4 marks]



Review of Problem Set 1

- Question 3: Can the set of jobs from Question 2 be scheduled using the least slack time algorithm? Justify your answer. [2 marks]

Review of Problem Set 1

- Question 4: In systems with multiple processors, job scheduling can be either *static* or *dynamic* which regards to the mapping of jobs to processors. Discuss the trade-off between performance and ease of validation for these different approaches to scheduling hard real time systems. [4 marks]

Review of Problem Set 1

- Any other questions about the problem set?

Review of Lectures

- Lecture 4: Clock-Driven Scheduling
  - Assumption: fixed set of periodic tasks
  - Cyclic scheduling; computing frame size
  - Slack stealing for aperiodic jobs
  - Acceptance test for sporadic jobs
  - Advantages and disadvantages
- Lecture 5: Priority-Driven Scheduling of Periodic Tasks (1)
  - Rate- and deadline-monotonic
  - Earliest deadline first; least slack time
  - Maximum schedulable utilization
- Lecture 6: Priority-Driven Scheduling of Periodic Tasks (2)
  - More general schedulability tests
  - Critical instants and time-demand analysis
  - Outline of practical factors

Key Learning Outcomes

- Understanding of when and how to use clock-driven scheduling
- Understanding of when and how to use priority-scheduling
  - Rate monotonic
  - Deadline monotonic
  - Earliest deadline first
  - Least slack time
- Understanding how to prove that a system is schedulable
  - Maximum schedulable utilization for a range of algorithms
  - Time-demand analysis

Proving Schedulability: Example 1

- Is the system of 5 independent, preemptable, tasks T_1=(1.0, 0.25), T_2=(1.25, 0.1), T_3=(1.5, 0.3), T_4=(1.75, 0.07) and T_5=(2.0, 0.1) schedulable using the rate monotonic algorithm?

Proving Schedulability: Example 2

- Is the system of 3 independent, preemptable, tasks T_1=(8, 3), T_2=(9, 3) and T_3=(15, 3) schedulable using the rate monotonic algorithm?

Proving Schedulability: Example 3

- Is the system of 3 independent, preemptable, tasks T_1=(8, 4), T_2=(12, 4) and T_3=(20, 4) schedulable using the rate monotonic algorithm or the EDF algorithm?

Proving Schedulability: Examples

- Aim of the examples has been to demonstrate how to perform schedulability analysis, show when it is necessary to simulate a system

Any Further Questions?