University of Glasgow | School of Computing Science

Priority-driven Scheduling of Periodic Tasks

Advanced Operating Systems (M)
Tutorial 2

Tutorial Outline

- Review of lectured material
- Worked examples
- Question and answer

Review of Lectures

- Priority-Driven Scheduling of Periodic Tasks
  - Rate- and deadline-monotonic; earliest deadline first; least slack time
  - Maximum schedulable utilisation
    - Of fixed priority systems: non-optimal; behaviour when maximum schedulable utilisation exceeded
    - Of dynamic priority systems: optimality
  - More general schedulability tests
    - Critical instants and time-demand analysis
    - Conceptual understanding of the process; graphical visualisation of time demand
  - Outline of practical factors
    - Impact of blocking time, context switch overhead, etc., on schedulability
  - Outline of POSIX scheduling API for real-time tasks

Key Learning Outcomes

- 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 can be scheduled
  - Maximum schedulable utilisation for a range of algorithms
  - Time-demand analysis

Proving Schedulability: Example 1

- Can the system of five 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) be scheduled using the rate monotonic algorithm?

Proving Schedulability: Example 2

- Can the system of three independent preemptable periodic tasks $T_1=(8, 3)$, $T_2=(9, 3)$ and $T_3=(15, 3)$ be scheduled using the rate monotonic algorithm?

Proving Schedulability: Example 3

- Can the system of three independent preemptable periodic tasks $T_1=(8, 4)$, $T_2=(12, 4)$ and $T_3=(20, 4)$ be scheduled using the rate monotonic algorithm or the EDF algorithm?

Proving Schedulability: Examples

- Aim of the examples has been to demonstrate how to determine whether a system can be scheduled, show when it is necessary to simulate a system

Question and Answer