

University of Glasgow | School of Computing Science

Priority-driven Scheduling of Aperiodic and Sporadic Tasks

Advanced Operating Systems (M)
Tutorial 3

Tutorial Outline

- Review of exercise 1
- Review of lectured material
- Worked examples
- Question and answer

Review of Exercise 1

- Q. 1: A system consists of three independent, preemptable, periodic tasks: (3, 1), (5, 2), and (10,3) running on a single processor.
  - Construct an earliest deadline first schedule of this system in the interval [0, 30). Label any missed deadlines. [2 marks]

Review of Exercise 1

- Q. 1: A system consists of three independent, preemptable, periodic tasks: (3, 1), (5, 2), and (10,3) running on a single processor.
  - Construct a rate-monotonic schedule for this system in the interval [0, 30). Label any missed deadlines. [2 marks]

Review of Exercise 1

- Q. 1: A system consists of three independent, preemptable, periodic tasks: (3, 1), (5, 2), and (10,3) running on a single processor.
  - Suppose we want to reduce the execution time of the task with period 3 to make the task system schedulable according to the EDF algorithm. What is the minimum reduction necessary for the system to be schedulable by the EDF algorithm? [2 marks]

Review of Exercise 1

- Question 2: A system contains five independent preemptable periodic tasks with utilization of $u_1 = 0.8$ and $u_2 = u_3 = u_4 = u_5 = 0.01$. Can these tasks be scheduled using a rate monotonic algorithm? Explain your answer. [4 marks]

Review of Exercise 1

- Question 3: Briefly describe the features of the POSIX 1003.1b API that allow implementation of Rate Monotonic scheduling [3 marks]. Highlight a significant limitation of the API for implementing periodic tasks in a real-time system [2 marks].

Review of Lectures

- Aperiodic and sporadic jobs
  - Assumptions, definitions, and system model
- Simple approaches to scheduling aperiodic jobs
  - Background, interrupt-drive, and polled execution
  - Periodic servers
- Bandwidth-preserving servers
  - Deferrable server
  - Sporadic server
  - Constant utilisation and total bandwidth servers
- Scheduling sporadic jobs

Key Learning Outcomes

- Understanding of how aperiodic jobs can be scheduled using a bandwidth preserving server
- Understanding of various types of bandwidth preserving server and trade-offs in their design
- Know the scheduling guarantees for the various types of bandwidth preserving server
- Know how to accept and schedule sporadic jobs

Example 1(a)

- Consider a system with three periodic tasks:
  - $T_1 = (6, 1)$
  - $T_2 = (10, 1)$
  - $T_3 = (14, 3)$
- Questions:
  - What does an EDF schedule for this system look like?

Example 1(b)

- Consider a system with three periodic tasks:
  - $T_1 = (6, 1)$
  - $T_2 = (10, 1)$
  - $T_3 = (14, 3)$
- Questions:
  - What does an EDF schedule for this system look like?
  - A deferrable server with period 4 and budget 1 is added. Can the system be scheduled?

Example 1(b)

- Consider a system with three periodic tasks:
  - $T_1 = (6, 1)$
  - $T_2 = (10, 1)$
  - $T_3 = (14, 3)$
- Questions:
  - An aperiodic task arrives with $r_A = 6$ and $e_A = 2$. What does the schedule look like? How does the budget of the server vary?

Example 2

- Consider a system of three periodic tasks: $T_1 = (3, 1)$, $T_2 = (4, 0.5)$, $T_3 = (10, 2)$. The system must support three aperiodic jobs:
  - A_1 which is released at time 0.5
  - A_2 which is released at time 12.25
  - A_3 which is released at time 17
- The aperiodic jobs execute for 0.75 units of time. The system is scheduled using RM, with a simple sporadic server $T_s = (5, 0.5)$ supporting the aperiodic jobs.
- Simulate the system for sufficient time to show how the aperiodic jobs are scheduled. What is the response time for each of the aperiodic jobs?

Summary

- Should understand how to evaluate the schedules for various types of server
- Should know how to demonstrate correctness of a system with aperiodic or sporadic tasks scheduled using a bandwidth preserving server

- Any further questions?