CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

Real-Time Systems

Exercise #5

Course Assistant Elena Marzi

Department of Computer Science and Engineering
Chalmers University of Technology

Exercise #5: scheduling

Today:

- Repetition on relevant scheduling concepts
- Exercise on cyclic executive
- Exercise on pseudo-parallel execution
- Exercise on feasibility test: processor utilization analysis

The examples are based on some old exam problems.

Scheduling

- **Schedule**: reservation of spatial and temporal resources for a given set of tasks
- **How** is scheduling implemented?
  - Cyclic executive/ Pseudo-parallel execution
- **When** are scheduling decisions taken?
  - Preemptive/ Non-preemptive
- **Which** task should excecute?
  - RM/ EDF

Scheduling

"With cyclic executives the schedule is generated off-line, and stored in a time table. The schedule can be generated by simulating a run-time system with pseudo-parallel execution."

"With pseudo-parallel execution the schedule is generated on-line, as a side-effect of tasks being executed. Ready tasks are sorted in a queue and receive access to the processor based on priority."

Example 1: Cyclic executive

Problem: Consider a real-time system with two periodic tasks that should be scheduled using a time table. The parameters for the two tasks are given below. Both tasks arrive the first time at time 0.

a) Construct a time table for the execution of the two tasks. The tasks are allowed to preempt each other.

	C_i	D_i	T_i
τ_1	2	5	5
τ_2	4	7	7

Example 1: Cyclic executive

Problem: Consider a real-time system with two periodic tasks that should be scheduled using a time table. The parameters for the two tasks are given below. Both tasks arrive the first time at time 0.

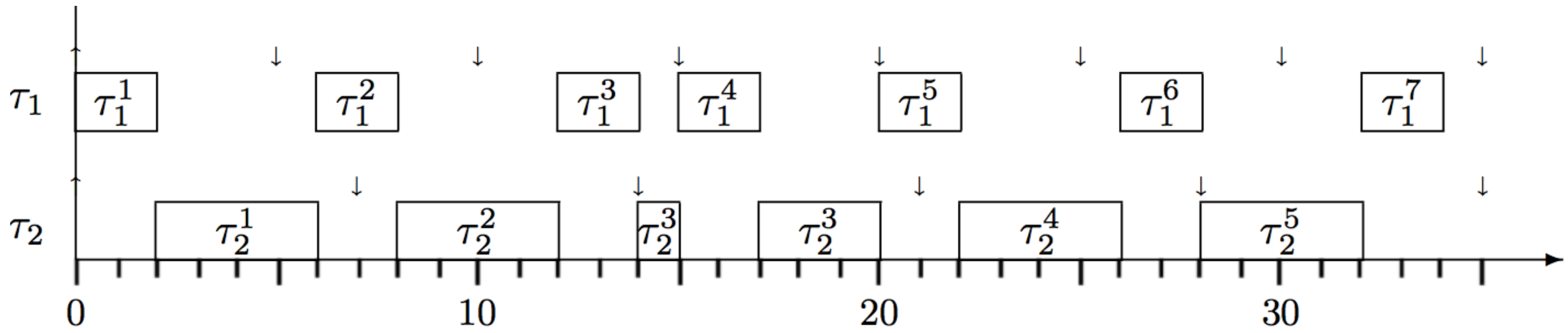
a) Construct a time table for the execution of the two tasks. The tasks are allowed to preempt each other.

b) Does your schedule constitute the best possible schedule, or does there exist a superior one?

	C_i	D_i	T_i
τ_1	2	5	5
τ_2	4	7	7

Example 1: Cyclic executive

Simulate the execution of the tasks using EDF scheduling within the hyper-period (LCM = 35)



$(\tau_1^1,0,2)$, $(\tau_2^1,2,6)$, $(\tau_1^2,6,8)$, $(\tau_2^2,8,12)$, $(\tau_1^3,12,14)$, $(\tau_2^3,14,15)$, $(\tau_1^4,15,17)$, $(\tau_2^3,17,20)$, $(\tau_1^5,20,22)$, $(\tau_2^4,22,26)$, $(\tau_1^6,26,28)$, $(\tau_2^5,28,32)$, $(\tau_1^7,32,34)$

Since EDF scheduling is known to be optimal for the given assumptions, this is the best possible schedule!

Example 1: Cyclic executive

Q: Why would simulation of RM scheduling not generate a better schedule? RM is also known to be optimal.

Q: Consider a task set with $D_i = T_i$ that is schedulable using RM. Is the task set also schedulable using EDF?

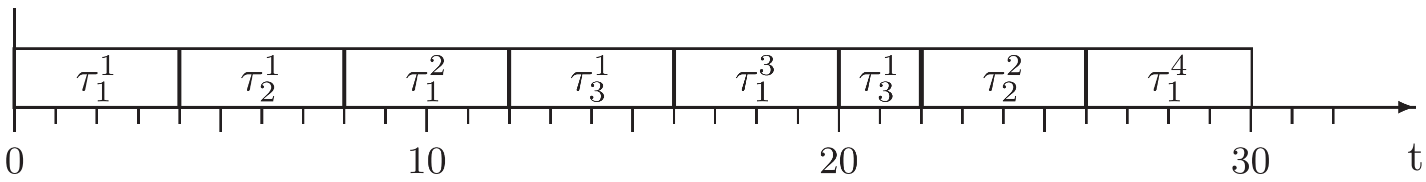
Q: Consider a task set with $D_i = T_i$ that is schedulable using EDF. Is the task set also schedulable using RM?

A: A scheduler that is optimal among all schedulers that use dynamic priorities can never be worse than a scheduler for static priorities. This is because static priorities is a special case of dynamic priorities.

Example 2: Pseudo-parallel execution

Problem: Decide, based on the table and timing diagram below, whether the schedule was generated by an RM or an EDF scheduler. In the timing diagram τ_i^k is used for representing instance *k* of periodic task *i*.

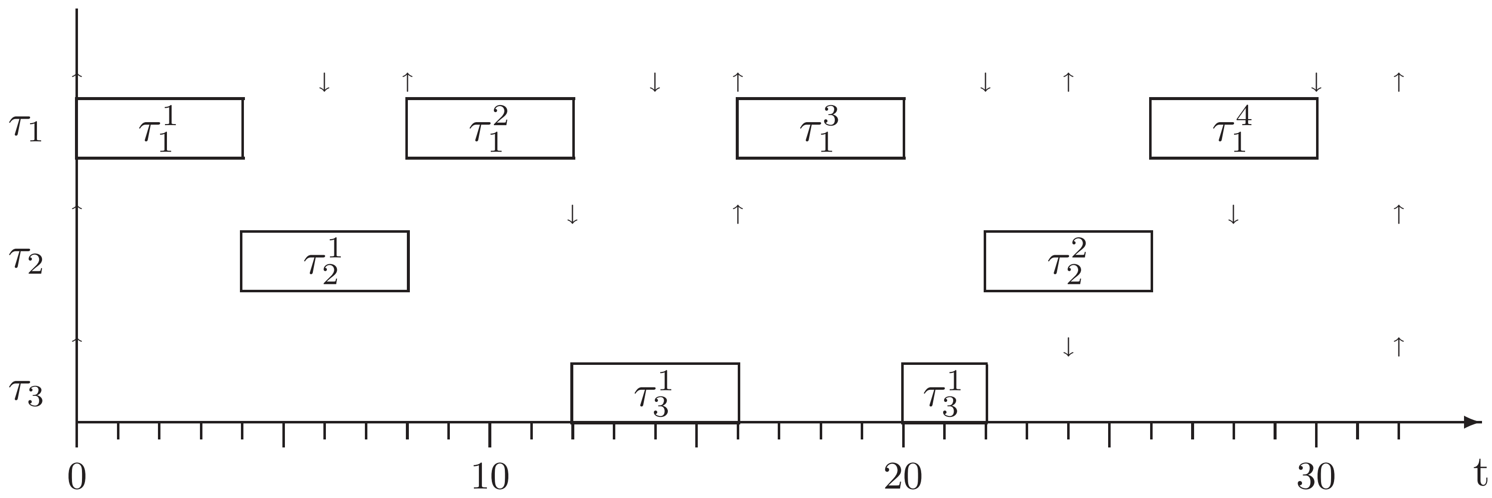
	C_i	D_i	T_i
τ_1	4	6	8
τ_2	4	12	16
τ_3	6	24	32



Example 2: Pseudo-parallel execution

If the tasks' arrival times and deadlines are indicated in a timing diagram, it is clear that we have a scheduler with dynamic priorities, that is, EDF. For example, τ_1 takes priority over τ_2 at t = 0 while the converse applies at t = 24.

A simulation of the tasks with EDF gives the same execution scenario.



Example 3: Pseudo-parallel execution

Problem: Consider a real-time system with three periodic tasks. The parameters for the three tasks are given below. All tasks arrive the first time at time 0.

a) Can you guarantee the schedulability of the task set using the RM scheduling algorithm?

Task	C_i	T_i	D_i
A	1	7	7
B	1	14	14
C	4	18	18

Example 3: Pseudo-parallel execution

a) The utilization U of the system is

$$U = \sum_{i=1}^{n} \frac{C_i}{T_i} = \frac{1}{7} + \frac{1}{14} + \frac{4}{18} \approx 0.4365$$

The utilization bound U_{RM} is:

$$U_{RM} = n\left(2^{1/n} - 1\right) = 3\left(2^{1/3} - 1\right) \approx 0.780 \qquad U < U_{RM}$$

The test succeeds! Schedulability is thus guaranteed.

Example 3: Pseudo-parallel execution

Problem: Consider a real-time system with three periodic tasks. The parameters for the three tasks are given below. All tasks arrive the first time at time 0.

a) Can you guarantee the schedulability of the task set using the RM scheduling algorithm?

b) Add a task Z with $T_Z = D_Z = 100$ and $C_Z = x$ to the task set. What is the maximum value of x such that the new task set is schedulable for RM scheduling based on Liu and Layland's utilization test?

Task	C_i	T_i	D_i
A	1	7	7
B	1	14	14
C	4	18	18

Example 3: Pseudo-parallel execution

b) The utilization U of the system is

$$U = \sum_{i=1}^{n} \frac{C_i}{T_i} = \frac{1}{7} + \frac{1}{14} + \frac{4}{18} + \frac{x}{100} = 0.4365 + 0.01x$$

The utilization bound U_{RM} is:

$$U_{RM} = n\left(2^{1/n} - 1\right) = 4\left(2^{1/4} - 1\right) \approx 0.7568$$

The test succeeds if:

$$U \le U_{RM} \Rightarrow 0.4365 + 0.01x \le 0.7568 \Rightarrow$$

$$\Rightarrow 0.01x \le 0.7568 - 0.4365 \Rightarrow x \le 100 \cdot 0.3203 = 32.03$$