

# EN3240: Embedded Systems Engineering

## Assignment 4 — Real Time Scheduling

Name: Thalagala B. P.

Index No: 180631J

September 16, 2022

**This is an individual assignment!**  
**Due Date: 18 September 2022 by 11.59 PM**

### Instructions

Please read the instructions and questions carefully. Write your answers directly in the space provided. Compile the tex document and submit the resulting PDF. This is an individual assignment. You are not allowed to take or give any help in completing this assignment.

When drawing images, you may draw them by hand, take a photo and add that as a Figure in LaTeX. Please avoid this option unless you have excellent handwriting and drawing capability!

Consider the tasks in the following table for scheduling questions in (a), (b) and (c). In the subsequent figures, the first arrival of tasks are shown using \* symbol. Use \* to indicate subsequent task arrivals. If a task violates its deadline, only abandon that task at the time it violates the deadline, show the violation using ‘X’ mark, and proceed with the next highest priority one. In other words, the ‘X’ mark should be at the same position of \* symbol, if violation happens. If the priority of a currently executing task is the same as the priority of a newly arrived task, continue to execute the current task. If the priorities of two tasks are the same and both are ready to execute, then always choose the one with the smaller execution time. The deadline and period values are relative to the arrival time. For example, if a task arrives in time step 2 with a deadline of 8, its actual deadline is timestep 10. Also, if a task arrives in time step 2 with a period of 8, it arrives next in timestep 10.

	First Arrival	Deadline	Period	Execution Time
<b>T1</b>	0	4	4	1
<b>T2</b>	1	4	4	2
<b>T3</b>	2	5	5	3

## Problem 1: Real Time Scheduling (2 Points)

Test whether the tasks are schedulable on a single processor using Rate-Monotonic (RM) scheduling by applying both necessary and sufficiency tests. Indicate whether the test passes or fails.

### Sufficiency Test

A set of  $n$  tasks is schedulable on a uniprocessor by the Rate Monotonic (RM) Scheduling algorithm, if the processor utilization ( $\mu$ ) obey the following inequality.

$$\mu \leq n \left( 2^{\frac{1}{n}} - 1 \right)$$

$$\mu = \sum_{i=1}^3 \frac{c_i}{p_i} = \frac{1}{4} + \frac{2}{4} + \frac{3}{5} = 1.35$$

$$n \left( 2^{\frac{1}{n}} - 1 \right) = 3 \times \left( 2^{\frac{1}{3}} - 1 \right) = 0.78$$

According to the obtained values, the sufficient condition for RMS algorithm does not hold for the given tasks. Therefore, the Sufficiency Test fails. This alone does not implies the tasks can not be schedulable. We need to check for the necessary test as well.

### Necessary Test

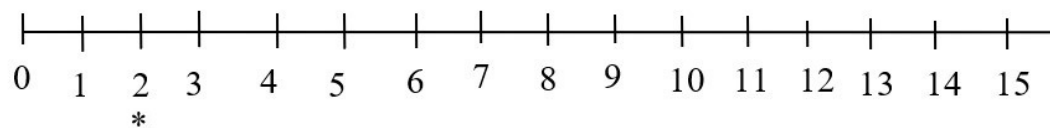
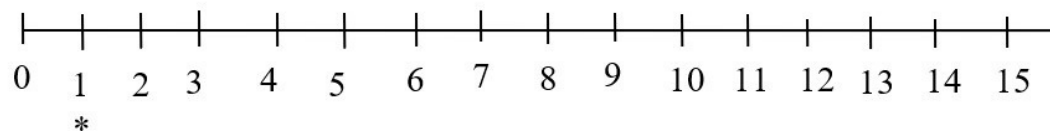
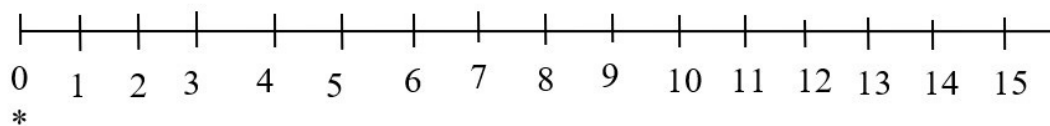
For uniprocessor, the necessary condition for schedulability,

$$\mu \leq 1$$

However, according to the obtained values for  $\mu = 1.35$  this necessary test also fails. Therefore it can be concluded that the given three periodic tasks can not be scheduled on a uni processor using the RMS algorithms.

## Problem 2: Rate Monotonic Scheduling (4 Points)

Schedule the tasks up to 15 time steps with Rate Monotonic (RM) scheduling.



**Problem 3: EDF Scheduling (4 Points)**

Schedule the tasks using EDF scheduling up to 15 timesteps.

