

## Exercise: RMS with deadline violation detection (ca. 1.5h)

### 1. Introduction

In the last exercise we wrote a scheduler that allows to simulate the scheduling behavior of Rate Monotonic Scheduling (RMS).

### 2. RMS simulator with LL-Test

Now augment your code, such that first a LL-Test is done automatically and the user will be informed about the result of the LL-Test.

### 3. Computation of maximum needed simulation time

Then compute how long T the simulation has to run such that we can decide whether the task set can be scheduled successfully, i.e., such that no deadlines are violated, or not. Let the simulation then run for T time steps.

Your code should automatically detect any deadline violation and inform the user during the simulation. After the simulation print the result whether the tasks are schedulable or not.

### Simulation run WITHOUT a deadline violation:

```
Rate Monotonic Scheduler (RMS) generated.

Please enter number of tasks to schedule: 3

    Enter execution time for task 0 in ms : 10
    Enter period for task 0 in ms : 30

    Enter execution time for task 1 in ms : 20
    Enter period for task 1 in ms : 50

    Enter execution time for task 2 in ms : 30
    Enter period for task 2 in ms : 150

Schedulability test:
-----
    CPU utilization u           = 0.933333
    limit for u for secure schedulability = 0.779763

=> the specified set of 3 tasks ...
... cannot be scheduled with a guarantee that all deadlines will be met =(

LCD of all tasks period lengths is 150. So I will simulate up to time=150 ms...

Press a key to start the simulation!

[...]
```

  

```
Simulation time : 140
Quantum = 10
Periodic task #0 : exec_time=10 ms, period=30 ms
Periodic task #1 : exec_time=20 ms, period=50 ms
Periodic task #2 : exec_time=30 ms, period=150 ms
-----
```

```
Time 0 : choosed task #0 (tasks that waited were: 0 1 2 )
Time 10 : choosed task #1 (tasks that waited were: 1 2 )
Time 20 : choosed task #1 (tasks that waited were: 1 2 )
Time 30 : choosed task #0 (tasks that waited were: 2 0 )
Time 40 : choosed task #2 (tasks that waited were: 2 )
Time 50 : choosed task #1 (tasks that waited were: 2 1 )
Time 60 : choosed task #0 (tasks that waited were: 2 1 0 )
Time 70 : choosed task #1 (tasks that waited were: 2 1 )
Time 80 : choosed task #2 (tasks that waited were: 2 )
Time 90 : choosed task #0 (tasks that waited were: 2 0 )
Time 100 : choosed task #1 (tasks that waited were: 2 1 )
Time 110 : choosed task #1 (tasks that waited were: 2 1 )
Time 120 : choosed task #0 (tasks that waited were: 2 0 )
Time 130 : choosed task #2 (tasks that waited were: 2 )
Time 140 : no tasks in task queue!
Simulation finished at time step 150
Result: :) No deadline was violated
Press any key to exit.
```

## Simulation run WITH a deadline violation:

Rate Monotonic Scheduler (RMS) generated.

Please enter number of tasks to schedule: 3

```
Enter execution time for task 0 in ms : 10
Enter period for task 0 in ms : 30
```

```
Enter execution time for task 1 in ms : 20
Enter period for task 1 in ms : 50
```

```
Enter execution time for task 2 in ms : 30
Enter period for task 2 in ms : 120
```

Schedulability test:

-----

```
CPU utilization u          = 0.983333
limit for u for secure schedulability = 0.779763
```

=> the specified set of 3 tasks ...

... cannot be scheduled with a guarantee that all deadlines will be met =(

LCD of all tasks period lengths is 600. So I will simulate up to time=600 ms...

Press a key to start the simulation!

[...]

Simulation time : 120

Quantum = 10

Periodic task #0 : exec\_time=10 ms, period=30 ms

Periodic task #1 : exec\_time=20 ms, period=50 ms

Periodic task #2 : exec\_time=30 ms, period=120 ms

-----

```
added task with id=0 that will have to compute for 10 msecs. Period time: 30 ms
added task with id=2 that will have to compute for 30 msecs. Period time: 120 ms
```

```
*** Scheduler choosed task #0 ***
```

There are 3 tasks that have not yet finished their computation.

```
Task #2 : Time computed=20, Time needed: 30
```

```
Task #0 : Time computed=10, Time needed: 10
```

```
Task #2 : Time computed=0, Time needed: 30
```

Task #0 has now finished its computation!

```
Time 0 : choosed task #0 (tasks that waited were: 0 1 2 )
Time 10 : choosed task #1 (tasks that waited were: 1 2 )
Time 20 : choosed task #1 (tasks that waited were: 1 2 )
Time 30 : choosed task #0 (tasks that waited were: 2 0 )
```

```
Time 40 : choosed task #2 (tasks that waited were: 2 )
Time 50 : choosed task #1 (tasks that waited were: 2 1 )
Time 60 : choosed task #0 (tasks that waited were: 2 1 0 )
Time 70 : choosed task #1 (tasks that waited were: 2 1 )
Time 80 : choosed task #2 (tasks that waited were: 2 )
Time 90 : choosed task #0 (tasks that waited were: 2 0 )
Time 100 : choosed task #1 (tasks that waited were: 2 1 )
Time 110 : choosed task #1 (tasks that waited were: 2 1 )
Time 120 : choosed task #0 (tasks that waited were: 2 0 2 )
```

Deadline of task 2 was violated! --> simulation will cancel.