Second Labs on Real-Time Scheduling

Report of TP2

AKKAR KHADIJA
L2

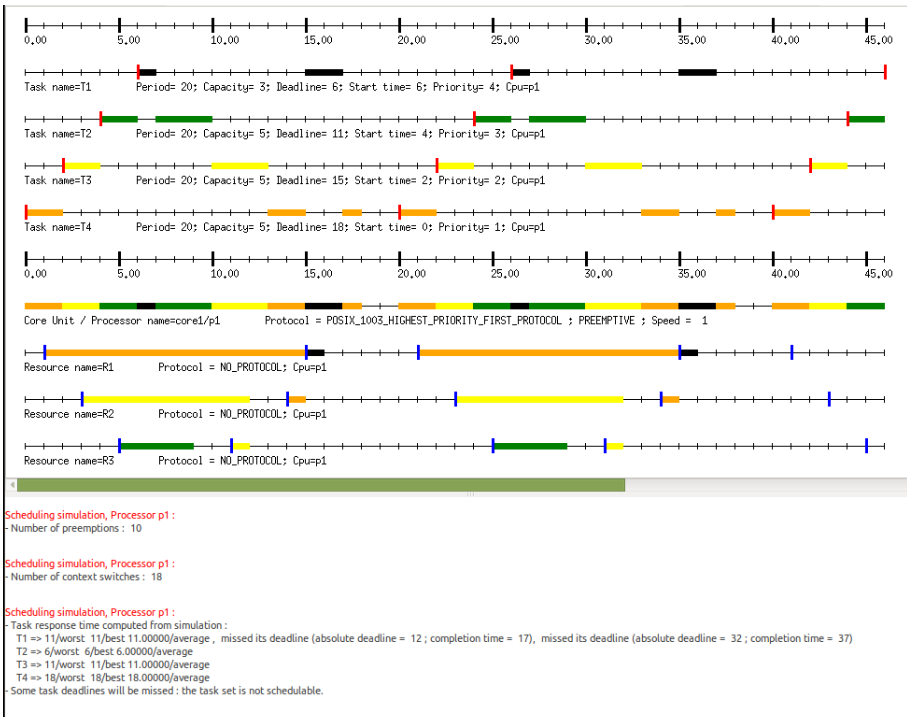
Exercice 1:

Let's assume the following task sharing resources R1, R2 and R3:

	First release	WCET	D	P	Priority
T_1	6	$3 : \boxed{}\boxed{R_1}\boxed{}$	6	20	4
T_2	4	$5 : \boxed{}\boxed{R_3}\boxed{R_3}\boxed{R_3}\boxed{}$	11	20	3
T_3	2	$5 : \boxed{}\boxed{R_2}\boxed{R_2}\boxed{R_2 R_3}\boxed{}$	15	20	2
T_4	0	$5 : \boxed{}\boxed{R_1}\boxed{R_1}\boxed{R_1 R_2}\boxed{}$	18	20	1

Fig. 1: Task Configuration - Exercice 1

1. The simulation without a specific protocol for resource allocation is **not schedulable**, as shown in this capture, where task T_1 missed its deadline twice.

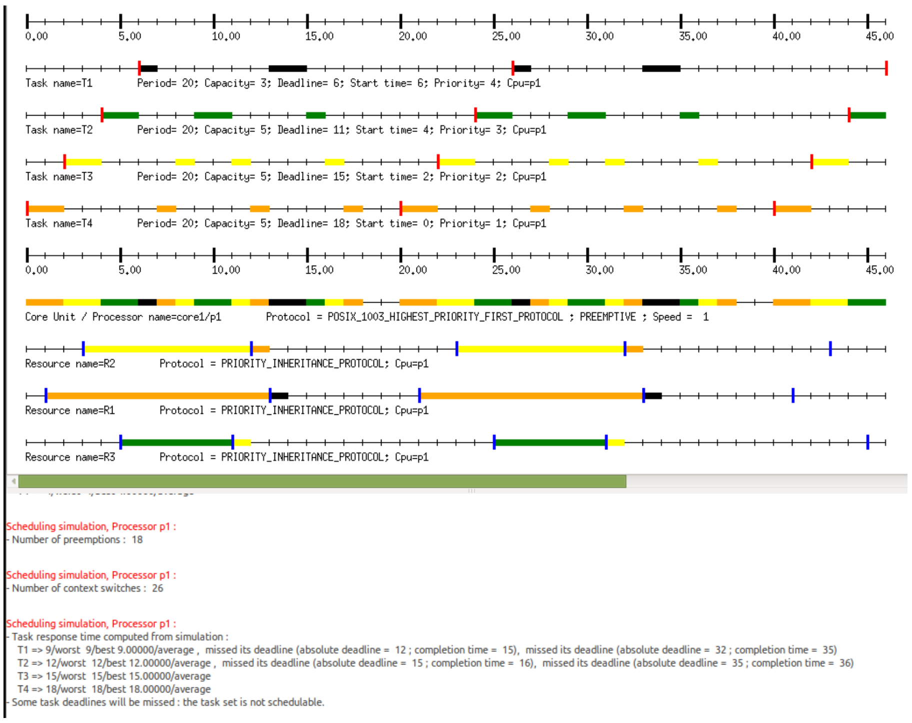


```
Task name=T1        Period= 20; Capacity= 3; Deadline= 6; Start time= 6; Priority= 4; Cpu=p1
Task name=T2        Period= 20; Capacity= 5; Deadline= 11; Start time= 4; Priority= 3; Cpu=p1
Task name=T3        Period= 20; Capacity= 5; Deadline= 15; Start time= 2; Priority= 2; Cpu=p1
Task name=T4        Period= 20; Capacity= 5; Deadline= 18; Start time= 0; Priority= 1; Cpu=p1
Core Unit / Processor name=core1/p1       Protocol = POSIX_1003_HIGHEST_PRIORITY_FIRST_PROTOCOL ; PREEMPTIVE ; Speed =  1
Resource name=R1        Protocol = NO_PROTOCOL; Cpu=p1
Resource name=R2        Protocol = NO_PROTOCOL; Cpu=p1
Resource name=R3        Protocol = NO_PROTOCOL; Cpu=p1
```

```
Scheduling simulation, Processor p1 :
- Number of preemptions :  10

Scheduling simulation, Processor p1 :
- Number of context switches :  18

Scheduling simulation, Processor p1 :
- Task response time computed from simulation :
  T1 => 11/worst  11/best 11.00000/average ,  missed its deadline (absolute deadline =  12 ; completion time =  17),  missed its deadline (absolute deadline =  32 ; completion time =  37)
  T2 => 6/worst  6/best 6.00000/average
  T3 => 11/worst  11/best 11.00000/average
  T4 => 18/worst  18/best 18.00000/average
- Some task deadlines will be missed : the task set is not schedulable.
```

Fig. 2: Simulation of the task configuration without a specific protocol

2. The simulation with the *Priority Inheritance* protocol is **not schedulable**, as both T_1 and T_2 missed their deadlines twice in this case.



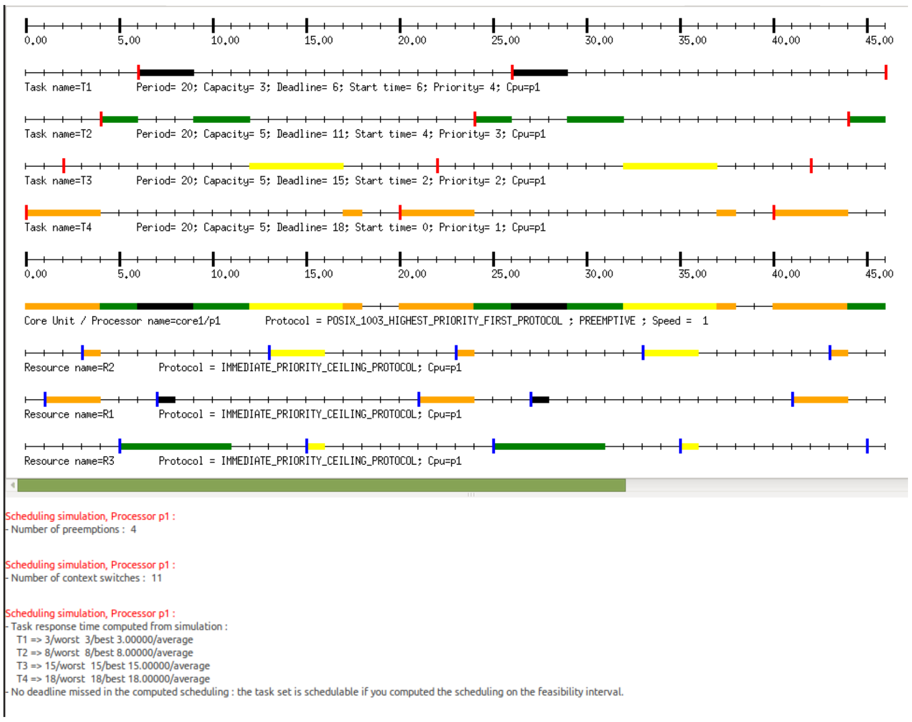
```
Task name=T1        Period= 20; Capacity= 3; Deadline= 6; Start time= 6; Priority= 4; Cpu=p1
Task name=T2        Period= 20; Capacity= 5; Deadline= 11; Start time= 4; Priority= 3; Cpu=p1
Task name=T3        Period= 20; Capacity= 5; Deadline= 15; Start time= 2; Priority= 2; Cpu=p1
Task name=T4        Period= 20; Capacity= 5; Deadline= 18; Start time= 0; Priority= 1; Cpu=p1
Core Unit / Processor name=core1/p1       Protocol = POSIX_1003_HIGHEST_PRIORITY_FIRST_PROTOCOL ; PREEMPTIVE ; Speed =  1
Resource name=R2        Protocol = PRIORITY_INHERITANCE_PROTOCOL; Cpu=p1
Resource name=R1        Protocol = PRIORITY_INHERITANCE_PROTOCOL; Cpu=p1
Resource name=R3        Protocol = PRIORITY_INHERITANCE_PROTOCOL; Cpu=p1
```

```
Scheduling simulation, Processor p1 :
- Number of preemptions :  18

Scheduling simulation, Processor p1 :
- Number of context switches :  26

Scheduling simulation, Processor p1 :
- Task response time computed from simulation :
  T1 => 9/worst  9/best 9.00000/average ,  missed its deadline (absolute deadline =  12 ; completion time =  15),  missed its deadline (absolute deadline =  32 ; completion time =  35)
  T2 => 12/worst  12/best 12.00000/average ,  missed its deadline (absolute deadline =  15 ; completion time =  16),  missed its deadline (absolute deadline =  35 ; completion time =  36)
  T3 => 15/worst  15/best 15.00000/average
  T4 => 18/worst  18/best 18.00000/average
- Some task deadlines will be missed : the task set is not schedulable.
```

Fig. 3: Simulation of the task configuration with the Priority Inheritance protocol

3. Simulation with the *Stack-based* Protocol (Immediate Ceiling Inheritance Protocol) is

schedulable.



```
Task name=T1        Period= 20; Capacity= 3; Deadline= 6; Start time= 6; Priority= 4; Cpu=p1
Task name=T2        Period= 20; Capacity= 5; Deadline= 11; Start time= 4; Priority= 3; Cpu=p1
Task name=T3        Period= 20; Capacity= 5; Deadline= 15; Start time= 2; Priority= 2; Cpu=p1
Task name=T4        Period= 20; Capacity= 5; Deadline= 18; Start time= 0; Priority= 1; Cpu=p1
Core Unit / Processor name=core1/p1       Protocol = POSIX_1003_HIGHEST_PRIORITY_FIRST_PROTOCOL ; PREEMPTIVE ; Speed =  1
Resource name=R2       Protocol = IMMEDIATE_PRIORITY_CEILING_PROTOCOL; Cpu=p1
Resource name=R1       Protocol = IMMEDIATE_PRIORITY_CEILING_PROTOCOL; Cpu=p1
Resource name=R3       Protocol = IMMEDIATE_PRIORITY_CEILING_PROTOCOL; Cpu=p1

Scheduling simulation, Processor p1 :
- Number of preemptions :  4

Scheduling simulation, Processor p1 :
- Number of context switches :  11

Scheduling simulation, Processor p1 :
- Task response time computed from simulation :
  T1 => 3/worst  3/best 3.00000/average
  T2 => 8/worst  8/best 8.00000/average
  T3 => 15/worst  15/best 15.00000/average
  T4 => 18/worst  18/best 18.00000/average
- No deadline missed in the computed scheduling : the task set is schedulable if you computed the scheduling on the feasibility interval.
```

Fig. 4: Simulation of the task configuration with the Stack-based protocol

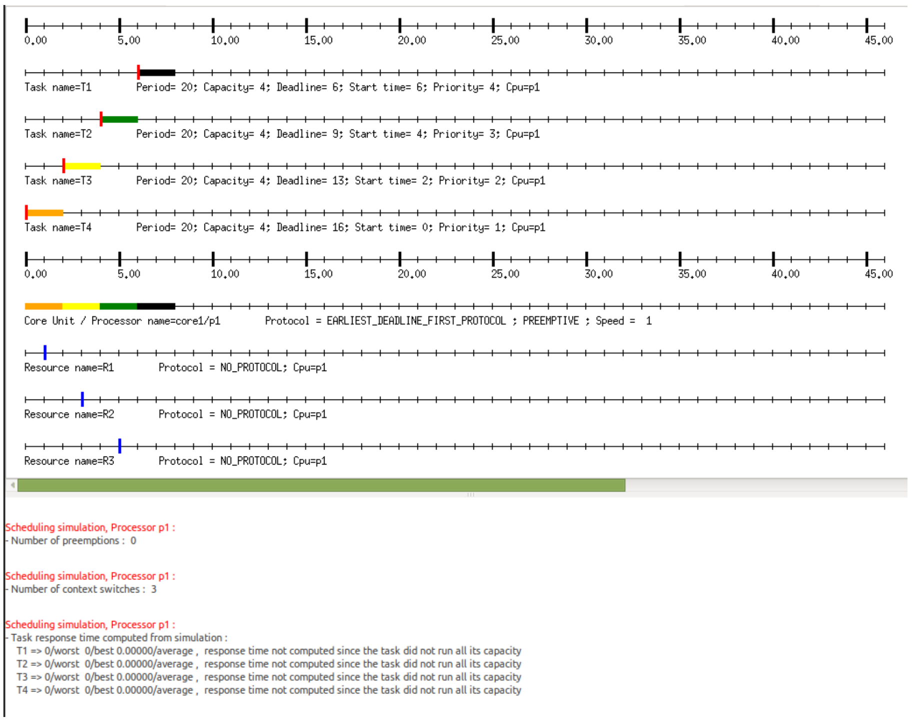
Exercice 2:

Let's assume the following task sharing resources R1, R2, R3 and R4:

	First release	WCET	D	P	Priority
T_1	6	4 : $\boxed{\ }\boxed{R_4}\boxed{R_4R_3}\boxed{\ }$	6	20	4
T_2	4	4 : $\boxed{\ }\boxed{R_3}\boxed{R_3R_4}\boxed{\ }$	9	20	3
T_3	2	4 : $\boxed{\ }\boxed{R_2}\boxed{R_2R_1}\boxed{\ }$	13	20	2
T_4	0	4 : $\boxed{\ }\boxed{R_1}\boxed{R_1R_2}\boxed{\ }$	16	20	1

Fig. 5: Task Configuration - Exercice 2

=> The simulation is **not schedulable** due to the capacities of the tasks, as shown in the figure below:



```
Task name=T1        Period= 20; Capacity= 4; Deadline= 6; Start time= 6; Priority= 4; Cpu=p1
Task name=T2        Period= 20; Capacity= 4; Deadline= 9; Start time= 4; Priority= 3; Cpu=p1
Task name=T3        Period= 20; Capacity= 4; Deadline= 13; Start time= 2; Priority= 2; Cpu=p1
Task name=T4        Period= 20; Capacity= 4; Deadline= 16; Start time= 0; Priority= 1; Cpu=p1
Core Unit / Processor name=core1/p1        Protocol = EARLIEST_DEADLINE_FIRST_PROTOCOL ; PREEMPTIVE ; Speed =  1
Resource name=R1        Protocol = NO_PROTOCOL; Cpu=p1
Resource name=R2        Protocol = NO_PROTOCOL; Cpu=p1
Resource name=R3        Protocol = NO_PROTOCOL; Cpu=p1

Scheduling simulation, Processor p1 :
- Number of preemptions :  0

Scheduling simulation, Processor p1 :
- Number of context switches :  3

Scheduling simulation, Processor p1 :
- Task response time computed from simulation :
  T1 => 0/worst  0/best 0.00000/average ,  response time not computed since the task did not run all its capacity
  T2 => 0/worst  0/best 0.00000/average ,  response time not computed since the task did not run all its capacity
  T3 => 0/worst  0/best 0.00000/average ,  response time not computed since the task did not run all its capacity
  T4 => 0/worst  0/best 0.00000/average ,  response time not computed since the task did not run all its capacity
```

Fig. 6: Simulation of the task configuration

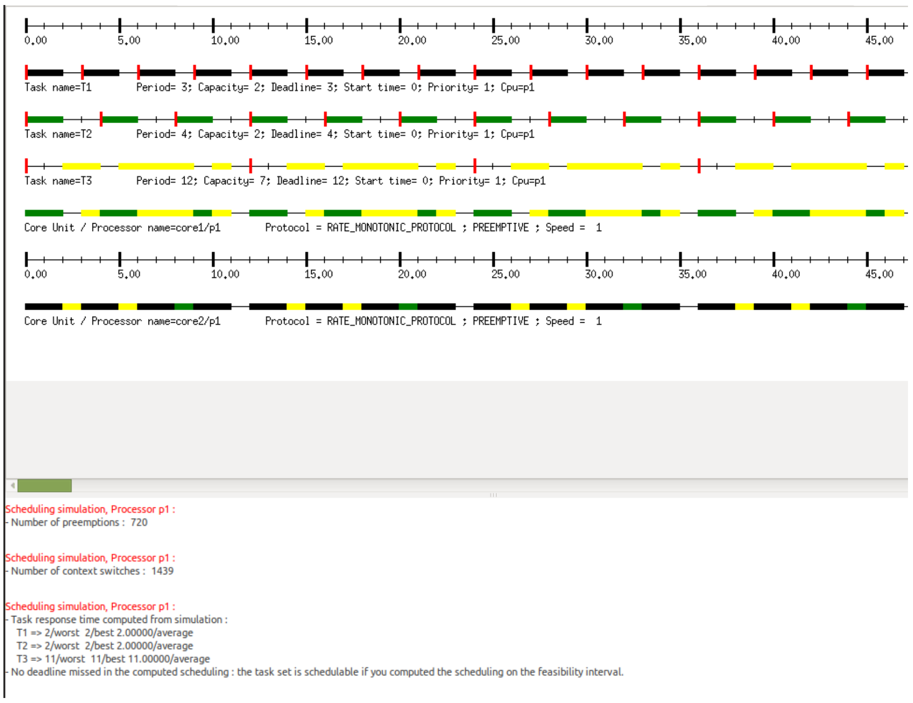
Exercice 3:

1. Simulate the following task configuration on one processor with two cores using a fully global Rate Monotonic scheduler:

	First release	WCET	D	P
T_1	0	2	3	3
T_2	0	2	4	4
T_3	0	7	12	12

Fig. 7: Task Configuration - Exercice 3-1

=> The simulation is **schedulable** because there are no deadline missed if we computed the scheduling on the feasibility interval.



```
Task name=T1        Period= 3; Capacity= 2; Deadline= 3; Start time= 0; Priority= 1; Cpu=p1
Task name=T2        Period= 4; Capacity= 2; Deadline= 4; Start time= 0; Priority= 1; Cpu=p1
Task name=T3        Period= 12; Capacity= 7; Deadline= 12; Start time= 0; Priority= 1; Cpu=p1
Core Unit / Processor name=core1/p1        Protocol = RATE_MONOTONIC_PROTOCOL ; PREEMPTIVE ; Speed =  1
Core Unit / Processor name=core2/p1        Protocol = RATE_MONOTONIC_PROTOCOL ; PREEMPTIVE ; Speed =  1

Scheduling simulation, Processor p1 :
- Number of preemptions :  720

Scheduling simulation, Processor p1 :
- Number of context switches :  1439

Scheduling simulation, Processor p1 :
- Task response time computed from simulation :
   T1 => 2/worst  2/best 2.00000/average
   T2 => 2/worst  2/best 2.00000/average
   T3 => 11/worst  11/best 11.00000/average
- No deadline missed in the computed scheduling : the task set is schedulable if you computed the scheduling on the feasibility interval.
```

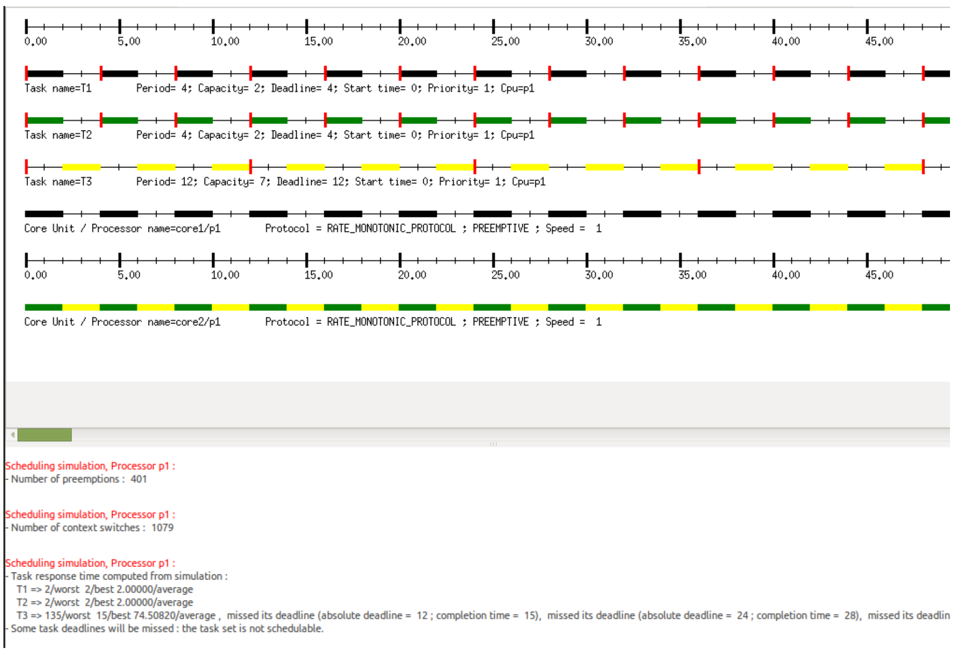
Fig. 8: Simulation of the task configuration with a fully global Rate Monotonic scheduler

2. Same question with the following configuration:

	First release	WCET	D	P
T_1	0	2	4	4
T_2	0	2	4	4
T_3	0	7	12	12

Fig. 9: Task Configuration - Exercice 3-2

=> The simulation is **not schedulable** as shown in this capture, where task T_3 missed its deadline three times.



```
Task name=T1        Period= 4; Capacity= 2; Deadline= 4; Start time= 0; Priority= 1; Cpu=p1
Task name=T2        Period= 4; Capacity= 2; Deadline= 4; Start time= 0; Priority= 1; Cpu=p1
Task name=T3        Period= 12; Capacity= 7; Deadline= 12; Start time= 0; Priority= 1; Cpu=p1
Core Unit / Processor name=core1/p1        Protocol = RATE_MONOTONIC_PROTOCOL ; PREEMPTIVE ; Speed =  1
Core Unit / Processor name=core2/p1        Protocol = RATE_MONOTONIC_PROTOCOL ; PREEMPTIVE ; Speed =  1

Scheduling simulation, Processor p1 :
- Number of preemptions :  401

Scheduling simulation, Processor p1 :
- Number of context switches :  1079

Scheduling simulation, Processor p1 :
- Task response time computed from simulation :
   T1 => 2/worst  2/best 2.00000/average
   T2 => 2/worst  2/best 2.00000/average
   T3 => 135/worst  15/best 74.50820/average ,  missed its deadline (absolute deadline =  12 ; completion time =  15),  missed its deadline (absolute deadline =  24 ; completion time =  28),  missed its deadlin
- Some task deadlines will be missed : the task set is not schedulable.
```

Fig. 10: Simulation of the task configuration with a fully global Rate Monotonic scheduler

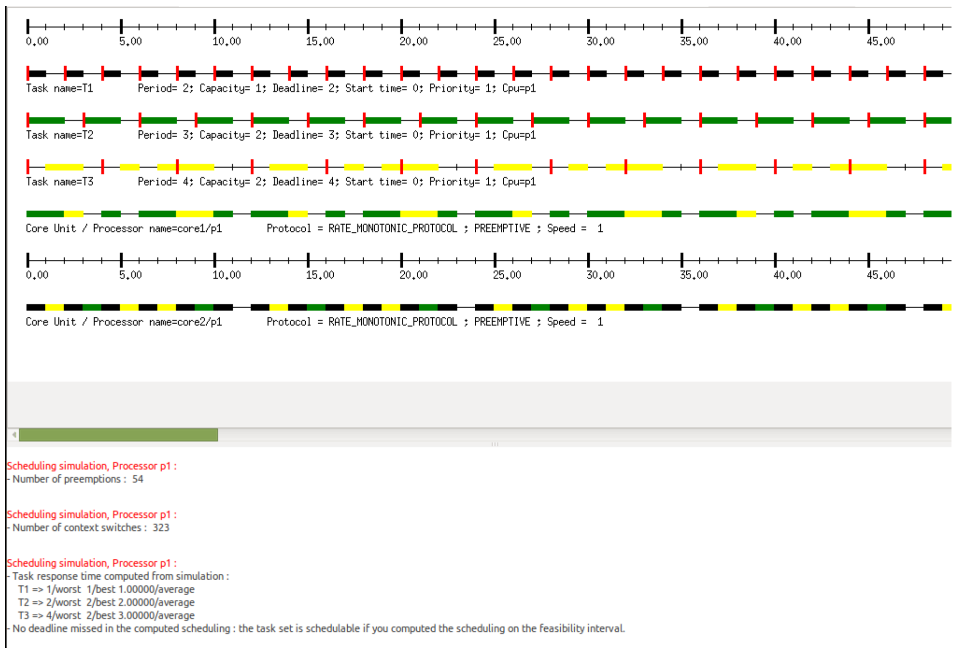
3. *From the two scheduling simulations, we see that when two tasks have the same periods and are synchronized, both cores of the processor run them at the same time, which affects the performance of the third task, T_3. Even though T_1 has a lighter load in the second simulation, the tasks need to be different to better alternate and share the work on T_3.*

Exercice 4:

Let's assume the following configuration of tasks:

	First release	WCET	D	P
T_1	0	1	2	2
T_2	0	2	3	3
T_3	0	2	4	4

Fig. 11: Task Configuration - Exercice 4



```
Task name=T1        Period= 2; Capacity= 1; Deadline= 2; Start time= 0; Priority= 1; Cpu=p1
Task name=T2        Period= 3; Capacity= 2; Deadline= 3; Start time= 0; Priority= 1; Cpu=p1
Task name=T3        Period= 4; Capacity= 2; Deadline= 4; Start time= 0; Priority= 1; Cpu=p1
Core Unit / Processor name=core1/p1        Protocol = RATE_MONOTONIC_PROTOCOL ; PREEMPTIVE ; Speed =  1
Core Unit / Processor name=core2/p1        Protocol = RATE_MONOTONIC_PROTOCOL ; PREEMPTIVE ; Speed =  1
```

```
Scheduling simulation, Processor p1 :
- Number of preemptions :  54

Scheduling simulation, Processor p1 :
- Number of context switches :  323

Scheduling simulation, Processor p1 :
- Task response time computed from simulation :
   T1 => 1/worst  1/best 1.00000/average
   T2 => 2/worst  2/best 2.00000/average
   T3 => 4/worst  2/best 3.00000/average
- No deadline missed in the computed scheduling : the task set is schedulable if you computed the scheduling on the feasibility interval.
```

Fig. 12: Simulation of the task configuration

=> The worst response time doesn't occur at time 0, meaning the worst-case scenario isn't when all tasks start simultaneously. This issue arises from synchronized idle times, which lead to wasted processing time. Essentially, when two tasks finish simultaneously, T3 cannot be divided to fully utilize both cores, which contributes to an increase in its response time.

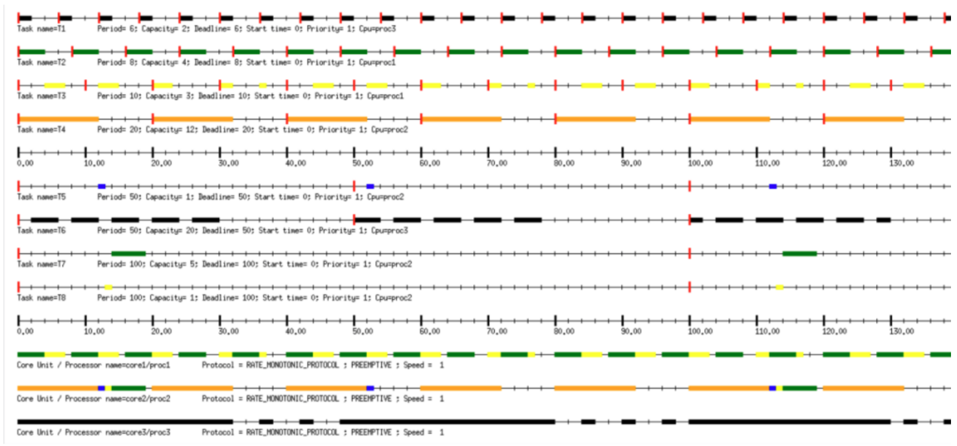
Exercice 5:

Let's assume the following configuration of tasks:

	WCET	D	P
T_1	2	6	6
T_2	4	8	8
T_3	3	10	10
T_4	12	20	20
T_5	1	50	50
T_6	20	50	50
T_7	5	100	100
T_8	1	100	100

Fig. 13: Task Configuration - Exercice 5

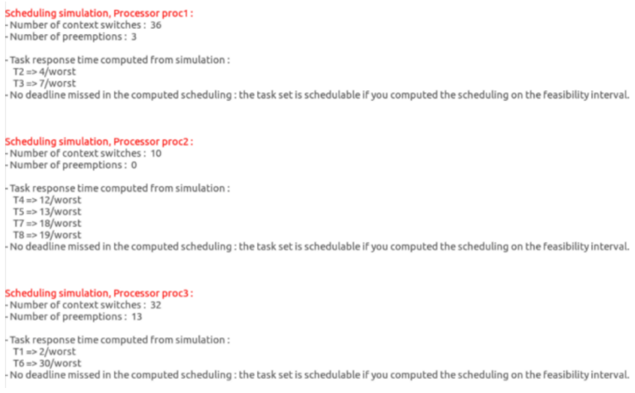
1. The configuration in this case is **schedulable** because there are no deadline missed if we computed the scheduling on the feasibility interval.



```
Scheduling simulation, Processor proc1 :
- Number of context switches : 36
- Number of preemptions : 3

- Task response time computed from simulation :
   T2 => 4/worst
   T3 => 7/worst
- No deadline missed in the computed scheduling : the task set is schedulable if you computed the scheduling on the feasibility interval.


Scheduling simulation, Processor proc2 :
- Number of context switches : 10
- Number of preemptions : 0

- Task response time computed from simulation :
   T4 => 12/worst
   T5 => 13/worst
   T7 => 19/worst
   T8 => 14/worst
- No deadline missed in the computed scheduling : the task set is schedulable if you computed the scheduling on the feasibility interval.


Scheduling simulation, Processor proc3 :
- Number of context switches : 32
- Number of preemptions : 13

- Task response time computed from simulation :
   T1 => 2/worst
   T6 => 30/worst
- No deadline missed in the computed scheduling : the task set is schedulable if you computed the scheduling on the feasibility interval.
```

Fig. 14: Simulation of the task configuration with a partitioned First-Fit Rate Monotonic scheduler

2. The configuration in this case is also **schedulable** because there are no deadline missed if we computed the scheduling on the feasibility interval.



```
Scheduling simulation, Processor proc1 :
- Number of context switches : 36
- Number of preemptions : 3

- Task response time computed from simulation :
   T2 => 4/worst
   T3 => 7/worst
- No deadline missed in the computed scheduling : the task set is schedulable if you computed the scheduling on the feasibility interval.


Scheduling simulation, Processor proc2 :
- Number of context switches : 10
- Number of preemptions : 0

- Task response time computed from simulation :
   T4 => 12/worst
   T5 => 13/worst
   T7 => 18/worst
   T8 => 19/worst
- No deadline missed in the computed scheduling : the task set is schedulable if you computed the scheduling on the feasibility interval.


Scheduling simulation, Processor proc3 :
- Number of context switches : 32
- Number of preemptions : 13

- Task response time computed from simulation :
   T1 => 2/worst
   T6 => 30/worst
- No deadline missed in the computed scheduling : the task set is schedulable if you computed the scheduling on the feasibility interval.
```

Fig. 15: Simulation of the task configuration with a partitioned First-Fit Earliest Deadline First scheduler