First Labs on Real-Time Scheduling

Report of TP1

AKKAR KHADIJA

L2

Exercice 1:

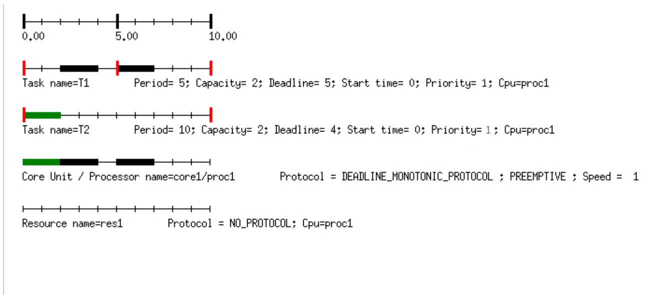
1. Simulate the following task configuration on one core using **deadline monotonic**.

	First Release	**WCET**	**D**	**P**
T_1	0	2	5	5
T_2	0	2	4	10

Tab. 1: Task Configuration - Exercice 1

(a) => *Yes, the configuration is schedulable because there are no deadline missed if we computed the scheduling on the feasibility interval.*

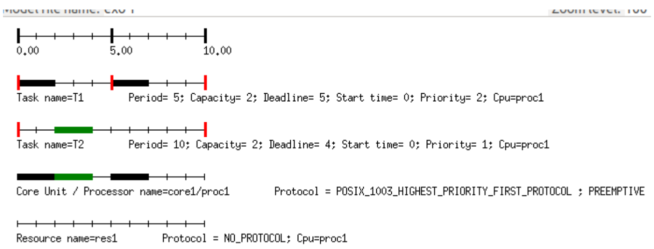
(b) => *The worst-case response time for T_1 is 4 time units, and for T_2 is 2 time units.*



```
Task name=T1        Period= 5; Capacity= 2; Deadline= 5; Start time= 0; Priority= 1; Cpu=proc1

Task name=T2        Period= 10; Capacity= 2; Deadline= 4; Start time= 0; Priority= 1; Cpu=proc1

Core Unit / Processor name=core1/proc1       Protocol = DEADLINE_MONOTONIC_PROTOCOL ; PREEMPTIVE ; Speed =  1

Resource name=res1       Protocol = NO_PROTOCOL; Cpu=proc1
```

Fig. 1: Simulation of the task configuration with the deadline monotonic

2. *Yes, the task set is schedulable if the scheduling is computed within the feasibility interval, given that T_1 has the highest priority (2):*



```
Task name=T1        Period= 5; Capacity= 2; Deadline= 5; Start time= 0; Priority= 2; Cpu=proc1
Task name=T2        Period= 10; Capacity= 2; Deadline= 4; Start time= 0; Priority= 1; Cpu=proc1
Core Unit / Processor name=core1/proc1        Protocol = POSIX_1003_HIGHEST_PRIORITY_FIRST_PROTOCOL ; PREEMPTIVE
Resource name=res1        Protocol = NO_PROTOCOL; Cpu=proc1
```

Fig. 2: Simulation of the task configuration with the a fixed priority

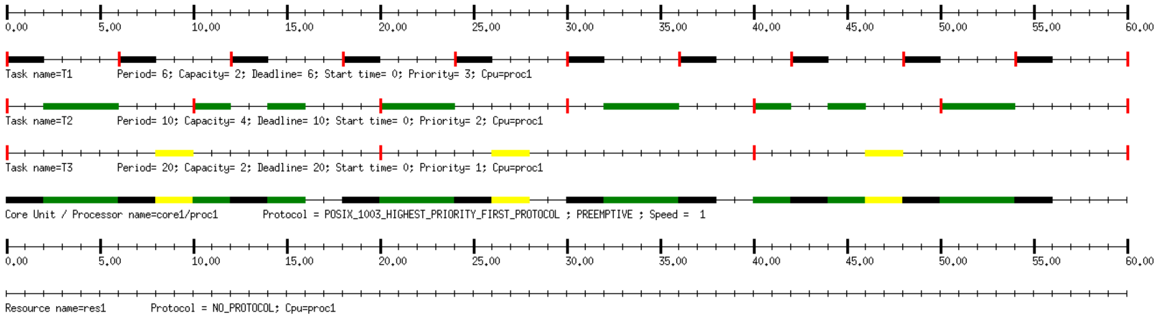
Exercice 2:

1. Simulate the following task configuration on one core using.

	First Release	WCET	D	P
T_1	0	2	6	6
T_2	0	4	10	10
T_3	0	2	20	20

Tab. 2: Task Configuration - Exercice 2

=> *Yes, the configuration is schedulable using a fixed priority, as the results show:*



```
Task name=T1        Period= 6; Capacity= 2; Deadline= 6; Start time= 0; Priority= 3; Cpu=proc1
Task name=T2        Period= 10; Capacity= 4; Deadline= 10; Start time= 0; Priority= 2; Cpu=proc1
Task name=T3        Period= 20; Capacity= 2; Deadline= 20; Start time= 0; Priority= 1; Cpu=proc1
Core Unit / Processor name=core1/proc1        Protocol = POSIX_1003_HIGHEST_PRIORITY_FIRST_PROTOCOL ; PREEMPTIVE ; Speed =  1
Resource name=res1        Protocol = NO_PROTOCOL; Cpu=proc1
```

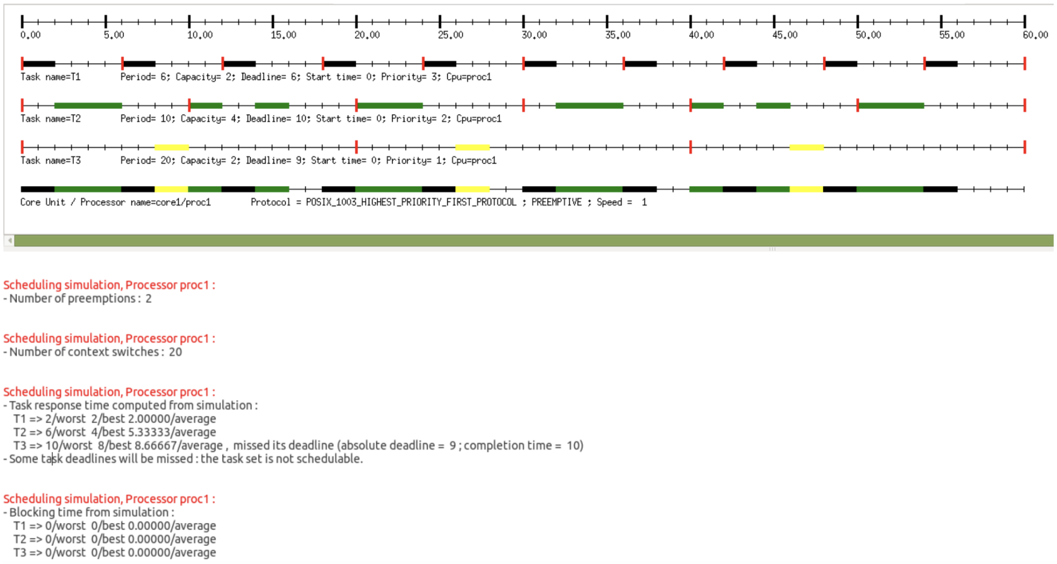
Fig. 3: Simulation of the task configuration with a fixed priority

2. Same question with the following task configuration.

	First Release	WCET	D	P
T_1	0	2	6	6
T_2	0	4	10	10
T_3	0	2	9	20

Tab. 3: Task Configuration - Exercice 2

=> *No, the configuration is not schedulable using a fixed priority, as the results show:*

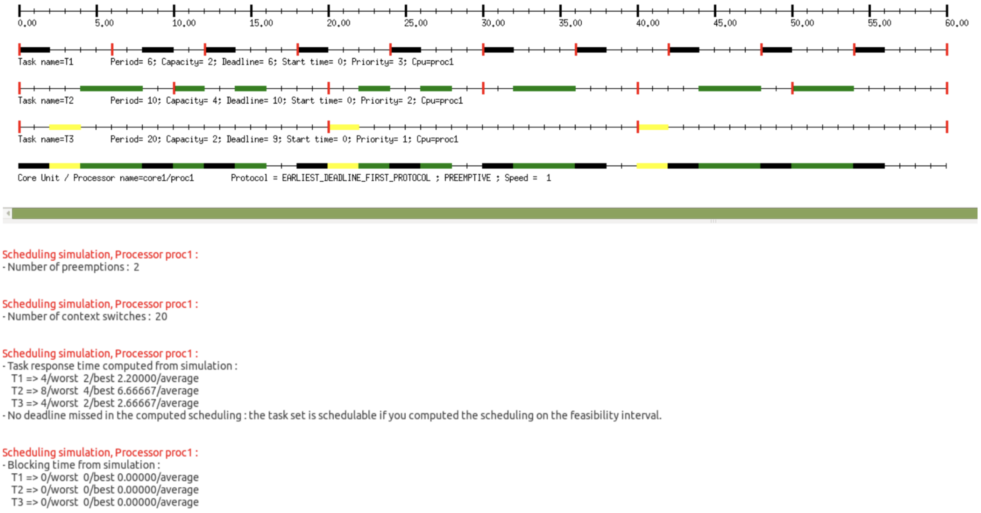


```
Task name=T1        Period= 6; Capacity= 2; Deadline= 6; Start time= 0; Priority= 3; Cpu=proc1
Task name=T2        Period= 10; Capacity= 4; Deadline= 10; Start time= 0; Priority= 2; Cpu=proc1
Task name=T3        Period= 20; Capacity= 2; Deadline= 9; Start time= 0; Priority= 1; Cpu=proc1
Core Unit / Processor name=core1/proc1        Protocol = POSIX_1003_HIGHEST_PRIORITY_FIRST_PROTOCOL ; PREEMPTIVE ; Speed =  1
```

```
Scheduling simulation, Processor proc1 :
- Number of preemptions :  2

Scheduling simulation, Processor proc1 :
- Number of context switches :  20

Scheduling simulation, Processor proc1 :
- Task response time computed from simulation :
   T1 => 2/worst  2/best 2.00000/average
   T2 => 6/worst  4/best 5.33333/average
   T3 => 10/worst  8/best 8.66667/average ,  missed its deadline (absolute deadline =  9 ; completion time =  10)
- Some task deadlines will be missed : the task set is not schedulable.

Scheduling simulation, Processor proc1 :
- Blocking time from simulation :
   T1 => 0/worst  0/best 0.00000/average
   T2 => 0/worst  0/best 0.00000/average
   T3 => 0/worst  0/best 0.00000/average
```

Fig. 4: Simulation of the task configuration with a fixed priority

=> *But, it is schedulable using the Earliest Deadline First, as the results show:*



```
Task name=T1        Period= 6; Capacity= 2; Deadline= 6; Start time= 0; Priority= 3; Cpu=proc1
Task name=T2        Period= 10; Capacity= 4; Deadline= 10; Start time= 0; Priority= 2; Cpu=proc1
Task name=T3        Period= 20; Capacity= 2; Deadline= 9; Start time= 0; Priority= 1; Cpu=proc1
Core Unit / Processor name=core1/proc1        Protocol = EARLIEST_DEADLINE_FIRST_PROTOCOL ; PREEMPTIVE ; Speed =  1
```

```
Scheduling simulation, Processor proc1 :
- Number of preemptions :  2

Scheduling simulation, Processor proc1 :
- Number of context switches :  20

Scheduling simulation, Processor proc1 :
- Task response time computed from simulation :
   T1 => 4/worst  2/best 2.20000/average
   T2 => 8/worst  4/best 6.66667/average
   T3 => 4/worst  2/best 2.66667/average
- No deadline missed in the computed scheduling : the task set is schedulable if you computed the scheduling on the feasibility interval.

Scheduling simulation, Processor proc1 :
- Blocking time from simulation :
   T1 => 0/worst  0/best 0.00000/average
   T2 => 0/worst  0/best 0.00000/average
   T3 => 0/worst  0/best 0.00000/average
```

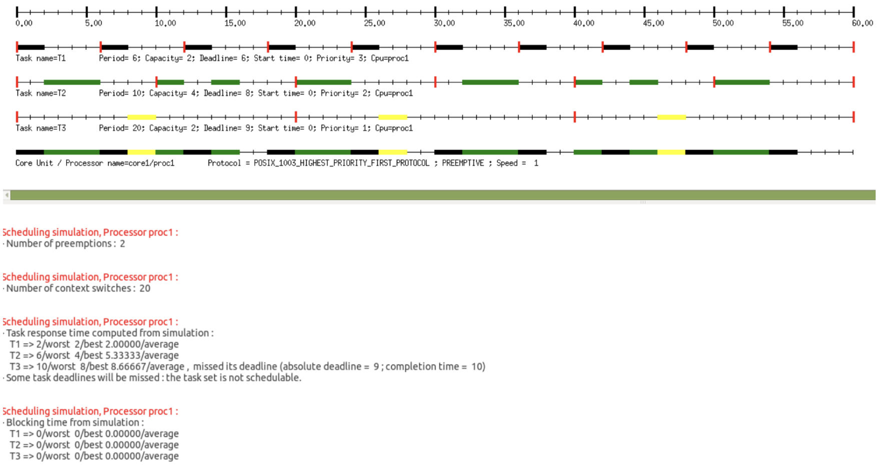
Fig. 5: Simulation of the task configuration with the Earliest Deadline First

3. Same question with the following task configuration.

	First Release	**WCET**	**D**	**P**
T_1	0	2	6	6
T_2	0	4	8	10
T_3	0	2	9	20

Tab. 4: Task Configuration - Exercice 2

=> *No, the configuration is not schedulable using a fixed priority, as the results show:*

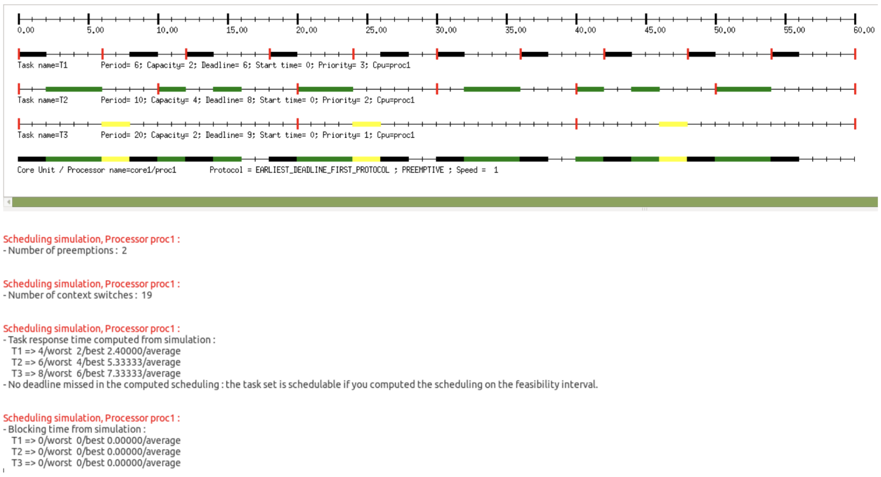


```
Task name=T1        Period= 6; Capacity= 2; Deadline= 6; Start time= 0; Priority= 3; Cpu=proc1
Task name=T2        Period= 10; Capacity= 4; Deadline= 8; Start time= 0; Priority= 2; Cpu=proc1
Task name=T3        Period= 20; Capacity= 2; Deadline= 9; Start time= 0; Priority= 1; Cpu=proc1
Core Unit / Processor name=core1/proc1        Protocol = POSIX_1003_HIGHEST_PRIORITY_FIRST_PROTOCOL ; PREEMPTIVE ; Speed =  1
```

```
Scheduling simulation, Processor proc1 :
- Number of preemptions : 2

Scheduling simulation, Processor proc1 :
- Number of context switches : 20

Scheduling simulation, Processor proc1 :
- Task response time computed from simulation :
  T1 => 2/worst  2/best 2.00000/average
  T2 => 6/worst  4/best 5.33333/average
  T3 => 10/worst  8/best 8.66667/average ,  missed its deadline (absolute deadline =  9 ; completion time =  10)
- Some task deadlines will be missed : the task set is not schedulable.

Scheduling simulation, Processor proc1 :
- Blocking time from simulation :
  T1 => 0/worst  0/best 0.00000/average
  T2 => 0/worst  0/best 0.00000/average
  T3 => 0/worst  0/best 0.00000/average
```

Fig. 6: Simulation of the task configuration with a fixed priority

=> *But, it is schedulable using the Earliest Deadline First, as the results show:*



```
Task name=T1        Period= 6; Capacity= 2; Deadline= 6; Start time= 0; Priority= 3; Cpu=proc1
Task name=T2        Period= 10; Capacity= 4; Deadline= 8; Start time= 0; Priority= 2; Cpu=proc1
Task name=T3        Period= 20; Capacity= 2; Deadline= 9; Start time= 0; Priority= 1; Cpu=proc1
Core Unit / Processor name=core1/proc1        Protocol = EARLIEST_DEADLINE_FIRST_PROTOCOL ; PREEMPTIVE ; Speed =  1
```

```
Scheduling simulation, Processor proc1 :
- Number of preemptions : 2

Scheduling simulation, Processor proc1 :
- Number of context switches : 19

Scheduling simulation, Processor proc1 :
- Task response time computed from simulation :
  T1 => 4/worst  2/best 2.40000/average
  T2 => 6/worst  4/best 5.33333/average
  T3 => 8/worst  6/best 7.33333/average
- No deadline missed in the computed scheduling : the task set is schedulable if you computed the scheduling on the feasibility interval.

Scheduling simulation, Processor proc1 :
- Blocking time from simulation :
  T1 => 0/worst  0/best 0.00000/average
  T2 => 0/worst  0/best 0.00000/average
  T3 => 0/worst  0/best 0.00000/average
```

Fig. 7: Simulation of the task configuration with the Earliest Deadline First

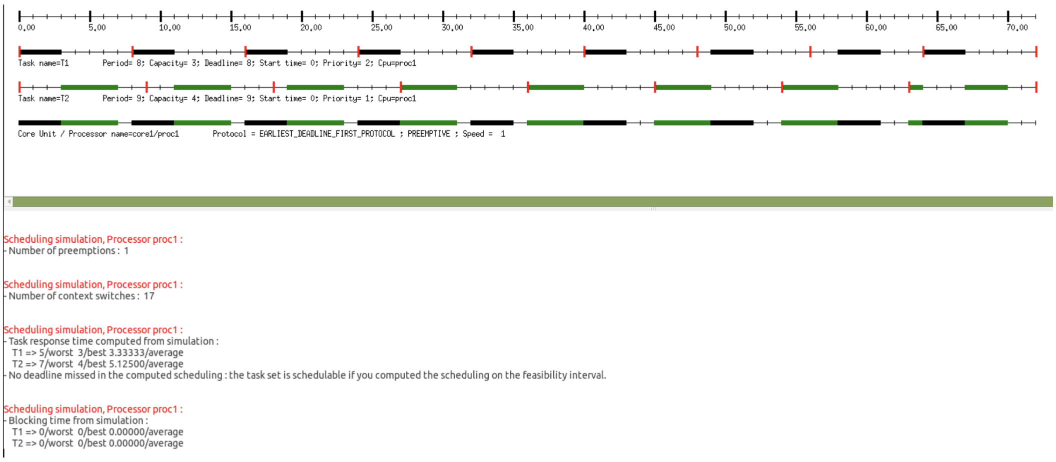
Exercice 3:

Simulate the following task configuration on one core using.

	First Release	**WCET**	**D**	**P**
T_1	0	3	8	8
T_2	0	4	9	9

Tab. 5: Task Configuration - Exercice 3

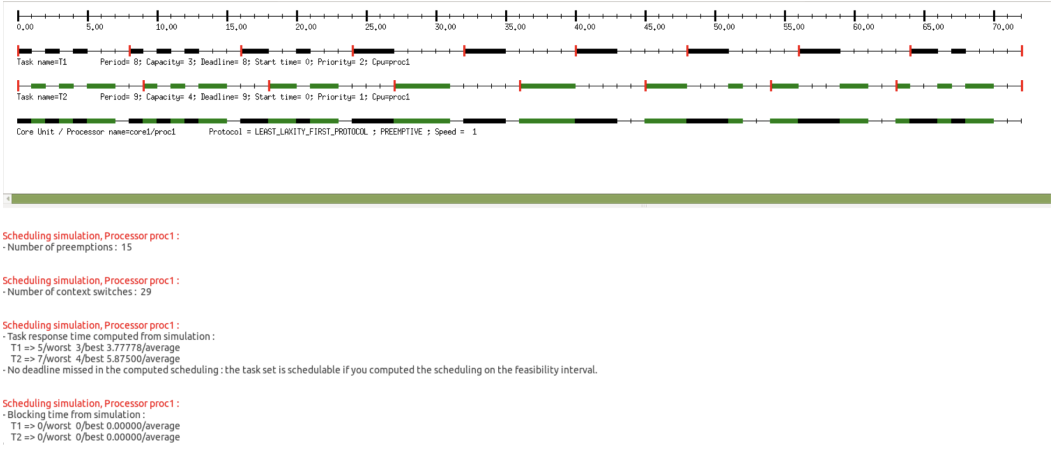
(a) The result of scheduling simulation with the Earliest Deadline First:



```
Task name=T1        Period= 8; Capacity= 3; Deadline= 8; Start time= 0; Priority= 2; Cpu=proc1
Task name=T2        Period= 9; Capacity= 4; Deadline= 9; Start time= 0; Priority= 1; Cpu=proc1
Core Unit / Processor name=core1/proc1        Protocol = EARLIEST_DEADLINE_FIRST_PROTOCOL ; PREEMPTIVE ; Speed =  1

Scheduling simulation, Processor proc1 :
- Number of preemptions : 1

Scheduling simulation, Processor proc1 :
- Number of context switches : 17

Scheduling simulation, Processor proc1 :
- Task response time computed from simulation :
   T1 => 5/worst  3/best 3.33333/average
   T2 => 7/worst  4/best 5.12500/average
- No deadline missed in the computed scheduling : the task set is schedulable if you computed the scheduling on the feasibility interval.

Scheduling simulation, Processor proc1 :
- Blocking time from simulation :
   T1 => 0/worst  0/best 0.00000/average
   T2 => 0/worst  0/best 0.00000/average
```

Fig. 8: Simulation of the task configuration with the Earliest Deadline First

(b) The result of scheduling simulation with the Least Laxity First:



```
Task name=T1        Period= 8; Capacity= 3; Deadline= 8; Start time= 0; Priority= 2; Cpu=proc1
Task name=T2        Period= 9; Capacity= 4; Deadline= 9; Start time= 0; Priority= 1; Cpu=proc1
Core Unit / Processor name=core1/proc1        Protocol = LEAST_LAXITY_FIRST_PROTOCOL ; PREEMPTIVE ; Speed =  1

Scheduling simulation, Processor proc1 :
- Number of preemptions : 15

Scheduling simulation, Processor proc1 :
- Number of context switches : 29

Scheduling simulation, Processor proc1 :
- Task response time computed from simulation :
   T1 => 5/worst  3/best 3.77778/average
   T2 => 7/worst  4/best 5.87500/average
- No deadline missed in the computed scheduling : the task set is schedulable if you computed the scheduling on the feasibility interval.

Scheduling simulation, Processor proc1 :
- Blocking time from simulation :
   T1 => 0/worst  0/best 0.00000/average
   T2 => 0/worst  0/best 0.00000/average
```

Fig. 9: Simulation of the task configuration with the Least Laxity First

=> *Based on the results from the two scheduling simulations, the Earliest Deadline First algorithm has fewer context switches and preemptions for this task configuration. This means it uses fewer resources and performs better.*

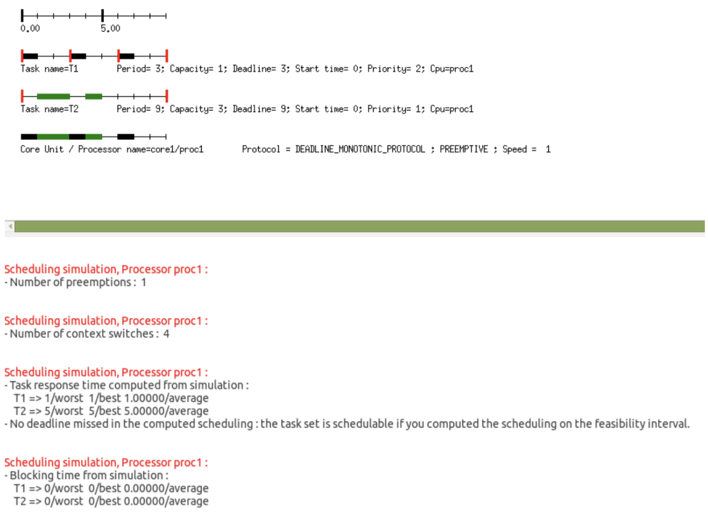
Exercice 4:

1. Simulate the following task configuration on one core using.

	First Release	**WCET**	**D**	**P**
T_1	0	1	3	3
T_2	0	3	9	9

Tab. 6: Task Configuration - Exercice 4

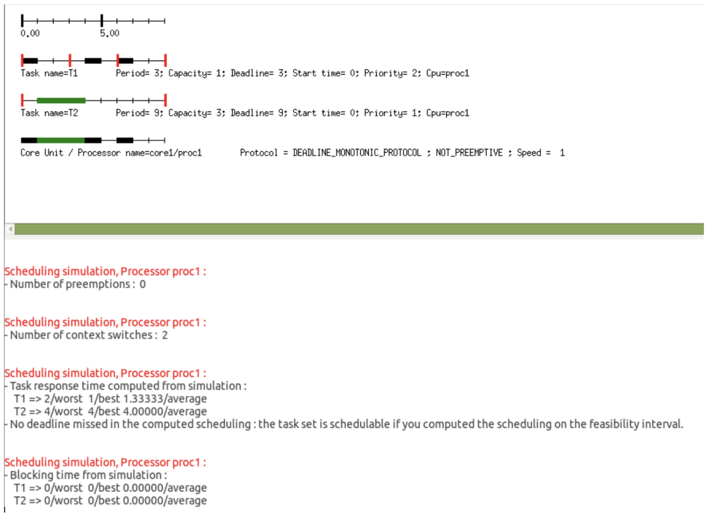
(a) The result of scheduling simulation with the Preemptive Rate Monotonic:



```
Task name=T1        Period= 3; Capacity= 1; Deadline= 3; Start time= 0; Priority= 2; Cpu=proc1

Task name=T2        Period= 9; Capacity= 3; Deadline= 9; Start time= 0; Priority= 1; Cpu=proc1

Core Unit / Processor name=core1/proc1        Protocol = DEADLINE_MONOTONIC_PROTOCOL ; PREEMPTIVE ; Speed =  1

Scheduling simulation, Processor proc1 :
- Number of preemptions : 1

Scheduling simulation, Processor proc1 :
- Number of context switches : 4

Scheduling simulation, Processor proc1 :
- Task response time computed from simulation :
   T1 => 1/worst  1/best 1.00000/average
   T2 => 5/worst  5/best 5.00000/average
- No deadline missed in the computed scheduling : the task set is schedulable if you computed the scheduling on the feasibility interval.

Scheduling simulation, Processor proc1 :
- Blocking time from simulation :
   T1 => 0/worst  0/best 0.00000/average
   T2 => 0/worst  0/best 0.00000/average
```

Fig. 10: Simulation of the task configuration with the Preemptive Rate Monotonic

(b) The result of scheduling simulation with the Non Preemptive Rate Monotonic:



```
Task name=T1        Period= 3; Capacity= 1; Deadline= 3; Start time= 0; Priority= 2; Cpu=proc1

Task name=T2        Period= 9; Capacity= 3; Deadline= 9; Start time= 0; Priority= 1; Cpu=proc1

Core Unit / Processor name=core1/proc1        Protocol = DEADLINE_MONOTONIC_PROTOCOL ; NOT_PREEMPTIVE ; Speed =  1

Scheduling simulation, Processor proc1 :
- Number of preemptions : 0

Scheduling simulation, Processor proc1 :
- Number of context switches : 2

Scheduling simulation, Processor proc1 :
- Task response time computed from simulation :
   T1 => 2/worst  1/best 1.33333/average
   T2 => 4/worst  4/best 4.00000/average
- No deadline missed in the computed scheduling : the task set is schedulable if you computed the scheduling on the feasibility interval.

Scheduling simulation, Processor proc1 :
- Blocking time from simulation :
   T1 => 0/worst  0/best 0.00000/average
   T2 => 0/worst  0/best 0.00000/average
```

Fig. 11: Simulation of the task configuration with the Non Preemptive Rate Monotonic

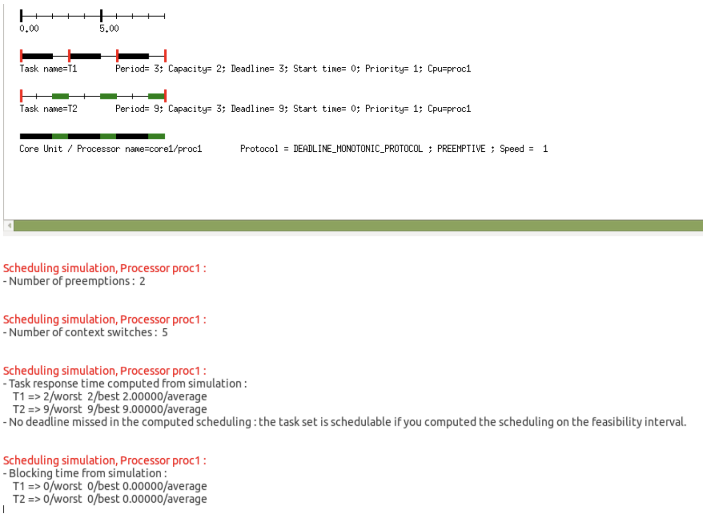
=> *In preemptive scheduling, whenever a task needs to be interrupted, the CPU performs a context switch, leading to more frequent switches. In contrast, in non-preemptive scheduling, the CPU doesn't allow interruptions, resulting in fewer context switches.*

2. Same question with the following task configuration.

(a) The result of scheduling simulation with the Preemptive Rate Monotonic:

	First Release	**WCET**	**D**	**P**
T_1	0	2	3	3
T_2	0	3	9	9

Tab. 7: Task Configuration - Exercice 4

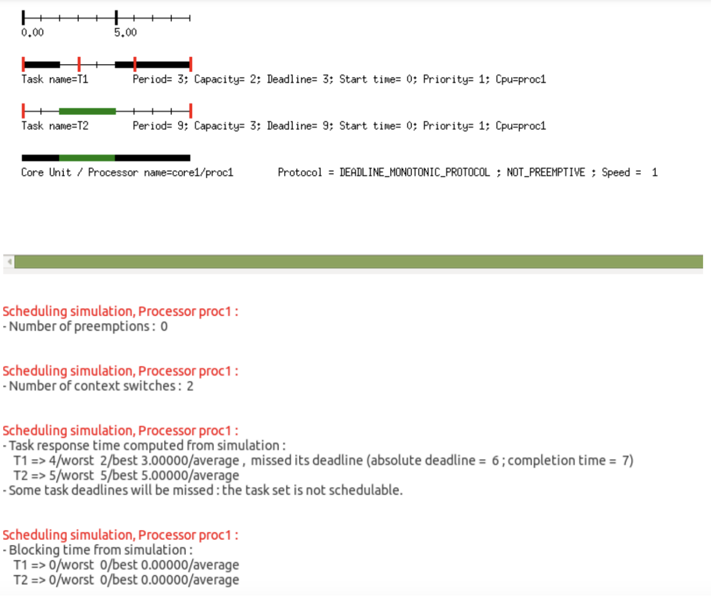


```
0.00          5.00

Task name=T1        Period= 3; Capacity= 2; Deadline= 3; Start time= 0; Priority= 1; Cpu=proc1

Task name=T2        Period= 9; Capacity= 3; Deadline= 9; Start time= 0; Priority= 1; Cpu=proc1

Core Unit / Processor name=core1/proc1       Protocol = DEADLINE_MONOTONIC_PROTOCOL ; PREEMPTIVE ; Speed =  1
```

```
Scheduling simulation, Processor proc1 :
- Number of preemptions : 2

Scheduling simulation, Processor proc1 :
- Number of context switches : 5

Scheduling simulation, Processor proc1 :
- Task response time computed from simulation :
  T1 => 2/worst  2/best 2.00000/average
  T2 => 9/worst  9/best 9.00000/average
- No deadline missed in the computed scheduling : the task set is schedulable if you computed the scheduling on the feasibility interval.

Scheduling simulation, Processor proc1 :
- Blocking time from simulation :
  T1 => 0/worst  0/best 0.00000/average
  T2 => 0/worst  0/best 0.00000/average
```

Fig. 12: Simulation of the task configuration with the Preemptive Rate Monotonic

(b) The result of scheduling simulation with the Non Preemptive Rate Monotonic:



```
0.00          5.00

Task name=T1        Period= 3; Capacity= 2; Deadline= 3; Start time= 0; Priority= 1; Cpu=proc1

Task name=T2        Period= 9; Capacity= 3; Deadline= 9; Start time= 0; Priority= 1; Cpu=proc1

Core Unit / Processor name=core1/proc1       Protocol = DEADLINE_MONOTONIC_PROTOCOL ; NOT_PREEMPTIVE ; Speed =  1
```

```
Scheduling simulation, Processor proc1 :
- Number of preemptions : 0

Scheduling simulation, Processor proc1 :
- Number of context switches : 2

Scheduling simulation, Processor proc1 :
- Task response time computed from simulation :
  T1 => 4/worst  2/best 3.00000/average , missed its deadline (absolute deadline =  6 ; completion time =  7)
  T2 => 5/worst  5/best 5.00000/average
- Some task deadlines will be missed : the task set is not schedulable.

Scheduling simulation, Processor proc1 :
- Blocking time from simulation :
  T1 => 0/worst  0/best 0.00000/average
  T2 => 0/worst  0/best 0.00000/average
```

Fig. 13: Simulation of the task configuration with the Non Preemptive Rate Monotonic

=> *In this task configuration, the task set is schedulable only in preemptive mode. In non-preemptive mode, the task missed its deadline (absolute deadline = 6, completion time = 7).*

Exercice 5:

Let's simulate the following task configuration.

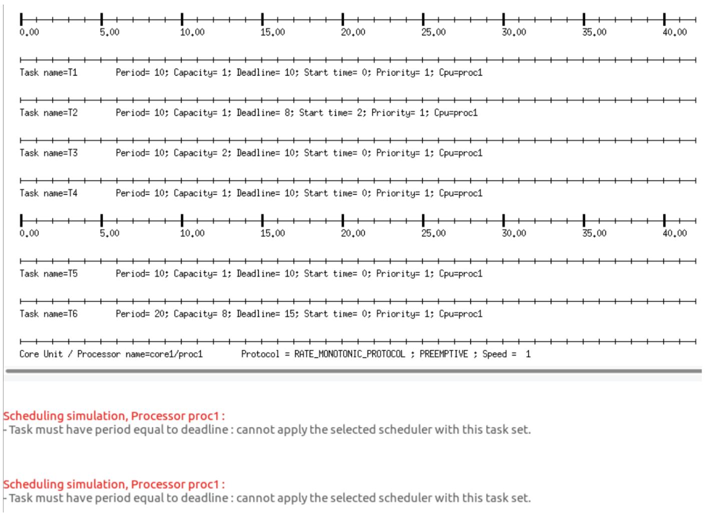
	First Release	WCET	D	P
T_1	0	1	10	10
T_2	2	1	8	10
T_3	0	2	10	10
T_4	0	1	10	10
T_5	0	1	10	10
T_6	0	8	15	20

Tab. 8: Task Configuration - Exercice 5

We have the following precedence constraints:

- T_1 and T_2 must complete execution before T_3 starts.
- T_3 must complete execution before T_4 and T_5.

1. The result of scheduling simulation with the Rate Monotonic approach:



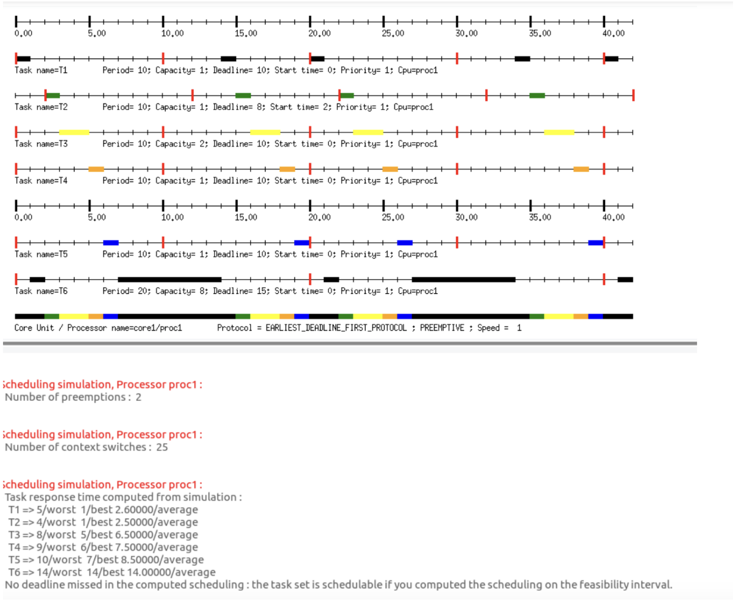
```
Task name=T1        Period= 10; Capacity= 1; Deadline= 10; Start time= 0; Priority= 1; Cpu=proc1
Task name=T2        Period= 10; Capacity= 1; Deadline= 8; Start time= 2; Priority= 1; Cpu=proc1
Task name=T3        Period= 10; Capacity= 2; Deadline= 10; Start time= 0; Priority= 1; Cpu=proc1
Task name=T4        Period= 10; Capacity= 1; Deadline= 10; Start time= 0; Priority= 1; Cpu=proc1
Task name=T5        Period= 10; Capacity= 1; Deadline= 10; Start time= 0; Priority= 1; Cpu=proc1
Task name=T6        Period= 20; Capacity= 8; Deadline= 15; Start time= 0; Priority= 1; Cpu=proc1
Core Unit / Processor name=core1/proc1       Protocol = RATE_MONOTONIC_PROTOCOL ; PREEMPTIVE ; Speed =  1
```

```
Scheduling simulation, Processor proc1 :
- Task must have period equal to deadline : cannot apply the selected scheduler with this task set.

Scheduling simulation, Processor proc1 :
- Task must have period equal to deadline : cannot apply the selected scheduler with this task set.
```

Fig. 14: Simulation of the task configuration with the Rate Monotonic

=> *It is not schedulable using the rate monotonic approach discussed in the lectures, because the task's period must be equal to its deadline.*

2. The result of scheduling simulation with the Earliest Deadline First approach:



```
Task name=T1        Period= 10; Capacity= 1; Deadline= 10; Start time= 0; Priority= 1; Cpu=proc1
Task name=T2        Period= 10; Capacity= 1; Deadline= 8; Start time= 2; Priority= 1; Cpu=proc1
Task name=T3        Period= 10; Capacity= 2; Deadline= 10; Start time= 0; Priority= 1; Cpu=proc1
Task name=T4        Period= 10; Capacity= 1; Deadline= 10; Start time= 0; Priority= 1; Cpu=proc1
Task name=T5        Period= 10; Capacity= 1; Deadline= 10; Start time= 0; Priority= 1; Cpu=proc1
Task name=T6        Period= 20; Capacity= 8; Deadline= 15; Start time= 0; Priority= 1; Cpu=proc1
Core Unit / Processor name=core1/proc1        Protocol = EARLIEST_DEADLINE_FIRST_PROTOCOL ; PREEMPTIVE ; Speed =  1
```

```
Scheduling simulation, Processor proc1 :
 Number of preemptions :  2

Scheduling simulation, Processor proc1 :
 Number of context switches :  25

Scheduling simulation, Processor proc1 :
 Task response time computed from simulation :
  T1 => 5/worst  1/best 2.60000/average
  T2 => 4/worst  1/best 2.50000/average
  T3 => 8/worst  5/best 6.50000/average
  T4 => 9/worst  6/best 7.50000/average
  T5 => 10/worst  7/best 8.50000/average
  T6 => 14/worst  14/best 14.00000/average
 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 the Earliest Deadline First

=> *It is schedulable using the Earliest Deadline First (EDF) approach discussed in the lectures.*