INFO-F-404 : Operating Systems II

1 Exercises

Exercise 1a : Rate Monotonic

Let's consider the system represented by Table 1. These are all *periodic*, *synchronous* tasks with *implicit deadline*. We will consider the case where these tasks (and jobs) are *independent* and *preemptible*.

Task index	Release time	WCET	Deadline	Period
Task τ_1	0	10	50	50
Task τ_2	0	20	80	80
Task τ_3	0	10	100	100
Task τ_4	0	50	200	200

Table 1: System of 4 periodic, synchronous tasks with implicit deadline.

a) Verify that this system could be scheduled using RM algorithm.

b) Plot the scheduling of these 4 tasks using RM. Each job takes its worst case execution time (WCET) to end. Use Figure 1.

c) Try to find a periodic system such as its utilization (U_{total}) is in range $[0.7; 1]$ and which could not be scheduled using RM algorithm.

Exercise 1b : Rate Monotonic

Let's consider the system represented by Table 2. These are all *periodic*, *synchronous* tasks with *implicit deadline*. We will consider the case where these tasks (and jobs) are *independent* and *preemptible*.

a) Verify that this system could be scheduled using RM algorithm.

1. Try to use the technique based on the processor *utilization*.
2. Try to use the technique based on the *worst response time*.

Task index	Release time	WCET	Deadline	Period
Task τ_1	0	4	10	10
Task τ_2	0	3	15	15
Task τ_3	0	4	20	20

Table 2: System of 3 periodic, synchronous tasks with implicit deadline.

Exercise 2a : Deadline Monotonic

Let's consider the system represented by Table 3. These are all *periodic*, *synchronous* tasks with *constrained deadline*. We will consider the case where these tasks (and jobs) are *independent* and *preemptible*.

Task index	Release time	WCET	Deadline	Period
Task τ_1	0	10	50	50
Task τ_2	0	20	40	80
Task τ_3	0	10	30	100
Task τ_4	0	50	150	200

Table 3: System of 4 periodic, synchronous tasks with constrained deadline.

a) Verify that this system could be scheduled using DM algorithm.

b) Plot the scheduling of these 4 tasks using DM. Each job takes its worst case execution time (WCET) to end. Use Figure 1.

Note: notice that if a FTP algorithm RM (defined only for systems with implicit deadline), it will assign priorities as follows : $\tau_1 > \tau_2 > \tau_3 > \tau_4$. Once τ_1 and τ_2 are executed (instant $t = 30$) τ_3 misses its deadline.

Exercise 2b : Deadline Monotonic

Let's consider the system represented by Table 4. These are all *periodic*, *synchronous* tasks with *constrained deadline*. We will consider the case where these tasks (and jobs) are *independent* and *preemptible*.

Task index	Release time	WCET	Deadline	Period
Task τ_1	0	3	5	5
Task τ_2	0	2	8	9
Task τ_3	0	1	4	11

Table 4: System of 3 periodic, synchronous tasks with constrained deadline.

a) Verify that this system could be scheduled using DM algorithm.

b) Is there any other algorithm that can successfully schedule τ by assigning fix priorities to each task?

Exercise 3 : Systems with arbitrary deadline

Let's consider the system represented by Table 5. These are all *periodic*, *synchronous* tasks with *arbitrary deadline*. We will consider the case where these tasks (and jobs) are *independent* and *preemptible*.

Task index	Release time	WCET	Deadline	Period
Taske τ_1	0	10	50	50
Taske τ_2	0	20	40	80
Taske τ_3	0	10	150	100
Taske τ_4	0	50	220	200

Table 5: System of 3 periodic, synchronous tasks with arbitrary deadline.

a) Find the feasibility interval for this system.

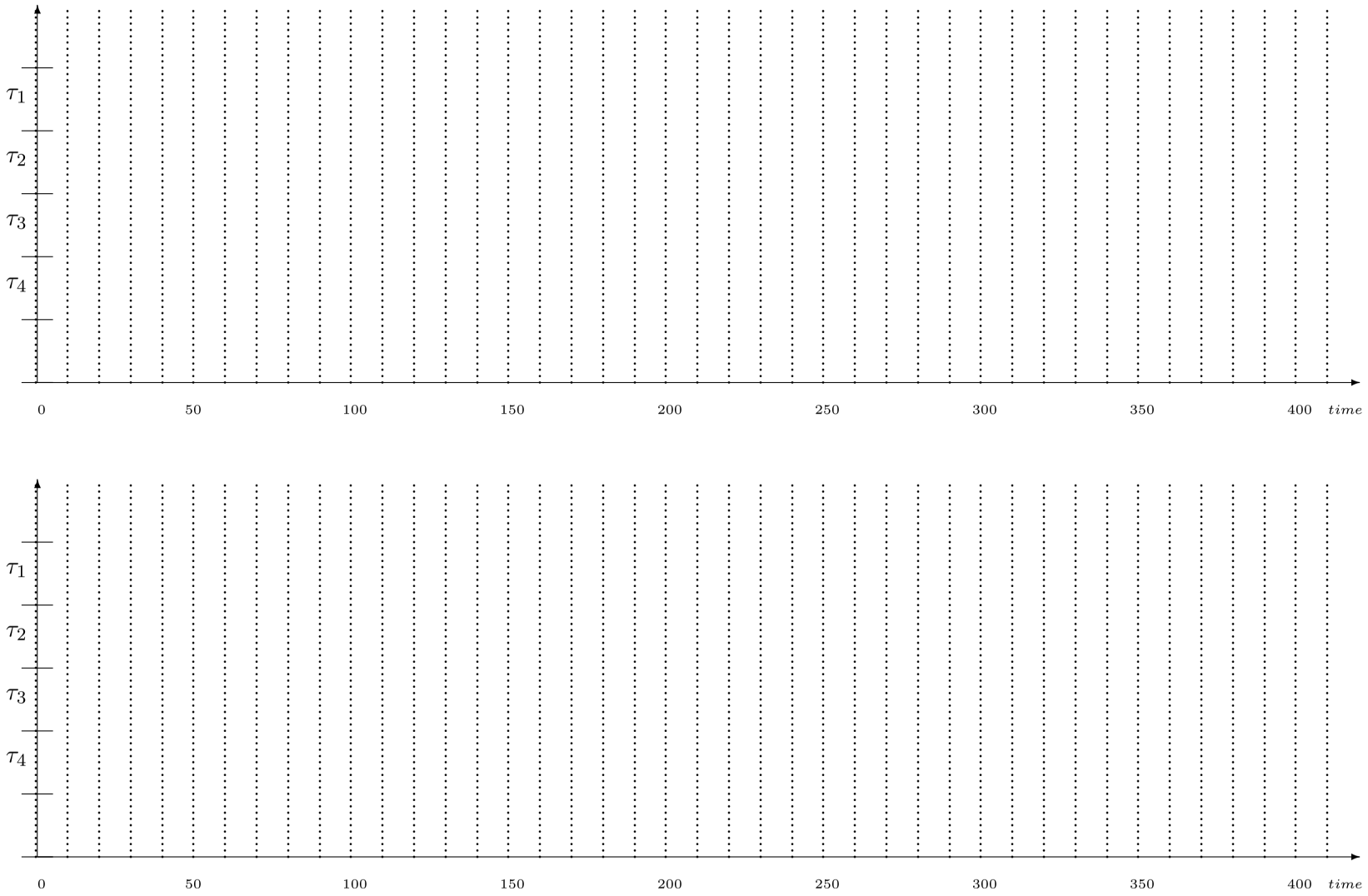

Figure 1: *Scheduling.*