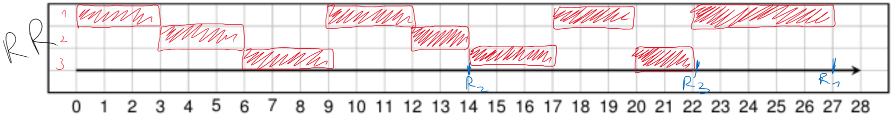
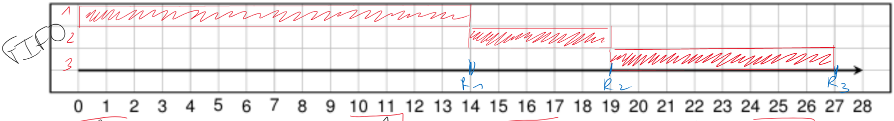
TD1

1. Suppose that three aperiodic jobs arrived at time instant 0 in the ready queue in the following order: J_1 (first), J_2 (second), J_3 (last). If the worst-case execution times of the jobs are respectively $C_1 = 14$, $C_2 = 5$, and $C_3 = 8$, find their response times and compute the system's average response time ($\frac{1}{n}\sum_{i=1}^{n} R_i$) for Round Robin (RR) with quantum time Q=3 and First In First Out (FIFO).

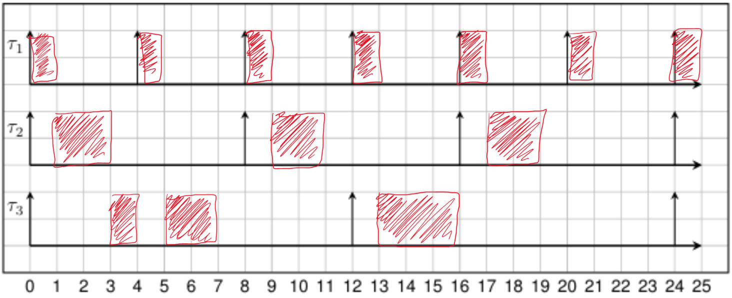




$$\boxed{R^{RR}_{AVG} = \frac{14+22+27}{3} = 21}$$

$$\boxed{R^{FIFO}_{AVG} = \frac{14+19+27}{3} = 20}$$

2. Verify the schedulability (use analytical methods) and construct the schedule according to the Rate Monotonic (RM) algorithm for the following set of periodic tasks:



	C_i	T_i
τ_1	1	4
τ_2	2	8
τ_3	3	12

n	U
1	1.000
2	0.828
3	0.779

$R^1_{\tau_3} = 7$ $R^2_{\tau_3} = 4$

What are the response times of the first and the second job of τ_3? Which one is longer and why?

3. Verify the schedulability under Rate Monotonic (RM) and Earliest Deadline First (EDF) of the following task sets (use analytical methods) and construct the corresponding schedules:

RM)

	C_i	T_i
τ_1	1	4
τ_2	2	6
τ_3	2	8

n	U_{max}		U	
1	1.000	$\geq$	0,25	✓
2	0.828	$\geq$	0,58	✓
3	0.779	$\leq$	0,83	× → RTA ⊛

EDF)

	C_i	T_i
τ_1	1	4
τ_2	2	6
τ_3	**3**	8

Car EDF ↓

$0,25 \leq 1$ ✓

$0,58 \leq 1$ ✓

$0,83 \leq 1$ ✓

Is it possible to schedule the second task set under RM?

Non, contre exemple

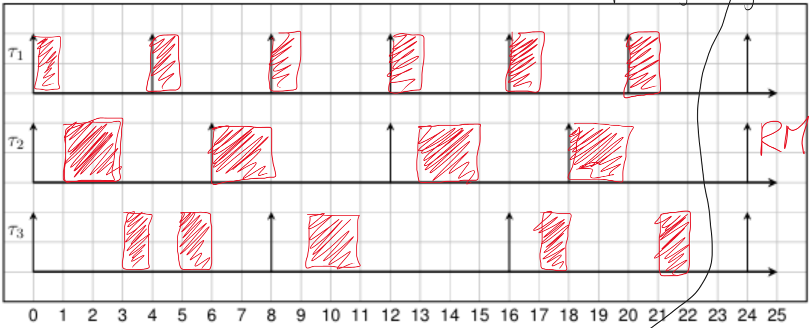
si non calculer RTA pour justifier

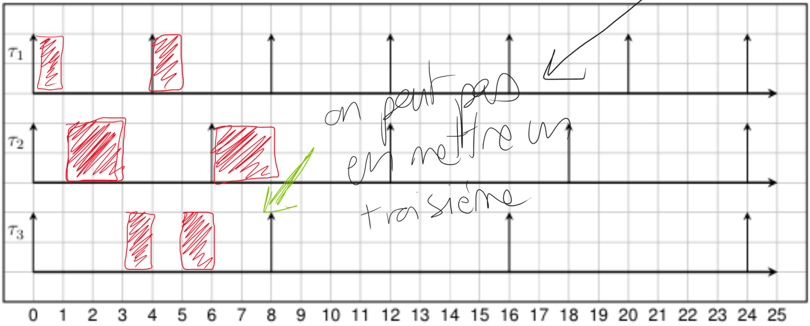
⊛ Response Time Analysis :

$R_3^0 = 5$ // $R_3^{k+1} = C_3 + \left\lceil \frac{R_3^k}{T_1} \right\rceil C_1 + \left\lceil \frac{R_3^k}{T_2} \right\rceil C_2$

$R_3^1 = 2 + \left\lceil \frac{5}{4} \right\rceil 1 + \left\lceil \frac{5}{6} \right\rceil 2 = 6$

$R_3^2 = 2 + \left\lceil \frac{6}{4} \right\rceil + \left\lceil \frac{6}{6} \right\rceil 2 = 6 < 8 \Rightarrow$ ✓





on peut pas en mettre un troisième

4. Verify the schedulability of each task from the following set under Rate Monotonic (RM):

	C_i	T_i
τ_1	2	7
τ_2	11	16
τ_3	1	50

$U_1 = 0,29 < 1$ ✓

$U_2 = 0,97 \geq 0,828$ ✗

$U_3 = 0,99 \geq 0,779$ ✗

$R^0 = 13$

$R^0 = 14$

$R_2^1 = 11 + \lceil \frac{13}{7} \rceil \cdot 2 = 15$

$R_2^2 = 11 + \lceil \frac{15}{7} \rceil 2 = 17$

$R_2^3 = 11 + \lceil \frac{17}{7} \rceil 2 = 17 > 16$ ✗

non ordonnançable par RM.

$R_3^1 = 1 + \lceil \frac{14}{7} \rceil \cdot 2 + \lceil \frac{14}{76} \rceil 11 = 16$

$R_3^2 = 1 + \lceil \frac{16}{7} \rceil 2 + \lceil \frac{16}{76} \rceil 11 = 18$

$R_3^3 = 1 + \lceil \frac{18}{7} \rceil 2 + \lceil \frac{18}{76} \rceil 11 = 29$

$R_3^4 = 1 + \lceil \frac{29}{7} \rceil 2 + \lceil \frac{29}{76} \rceil 11 = 33$

$R_3^5 = 1 + \lceil \frac{33}{7} \rceil 2 + \lceil \frac{33}{76} \rceil 11 = 44$

$R_3^6 = 1 + \lceil \frac{44}{7} \rceil 2 + \lceil \frac{44}{76} \rceil 11 = 48$

$R_3^7 = 1 + \lceil \frac{48}{7} \rceil 2 + \lceil \frac{48}{76} \rceil 11 = 48 < 50$

$\Rightarrow$ ordonnançable avec RM

5. Consider the task sets scheduled by a fixed-priority scheduler:

a)

	C_i	T_i
τ_1	6	10
τ_2	10	25

b)

	C_i	T_i
τ_1	60	150
τ_2	80	200
τ_2	30	300

Assign task priorities and check task set schedulability. If the tasks are not schedulable, propose a task transformation or another technique to improve system schedulability. *Hint:* Rate Monotonic (RM) is able to schedule task sets with harmonic periods up to 100% utilization.

ⓐ ∀ prio, pas schedulable

↳ les T_i doivent être multiple l'un de l'autre si $T_2 = 30 = 3T_1$ et si $U \leq 1$ ça passe

$U_2 = \frac{6}{10} + \frac{10}{30} = \frac{28}{30} \leq 1 \Rightarrow$ schedulable.

ⓑ $T_2 = 300$ donc $U = \frac{60}{150} + \frac{80}{300} + \frac{30}{300} = 0,767 \leq 1$

donc schedulable car T sont harmonique

6. Given two processors, P_1 and P_2, and the following task set, assign tasks to processors (*i.e.*, partitioned scheduling) such that the tasks running on each processor are schedulable by Rate Monotonic (RM).

	C_i	T_i	U_i	P_i
τ_1	1	4		
τ_2	3	10		
τ_3	9	20		
τ_4	35	100		
τ_5	12	120		

n	1	2	3	4	5	6	7	8
U	1.000	0.828	0.779	0.756	0.743	0.734	0.728	0.724

$\tau_2, \tau_3 \rightarrow \frac{3}{10} + \frac{9}{20} = U = 975 \leq 0,828$ ou ≤ 1 car harmonique

$\tau_1, \tau_4, \tau_5 \rightarrow U = \frac{12}{120} + \frac{1}{4} + \frac{35}{100} = 0,7 \leq 0,779$ ✓

TD2

1. Verify whether the following task set is schedulable by the Rate Monotonic (RM) algorithm.

Apply the processor utilization approach first, and then the Response Time Analysis if the utilization test fails.

	C_i	T_i	B_i	U
τ_1	3	10	6	0,9 ✓
τ_2	7	25	5	0,78 ✓
τ_3	8	50	6	0,86 ✗ → RTA
τ_4	10	1000	0	0,75 ✓

n	1	2	3	4
U	1.000	0.828	0.779	0.756

$$U = \frac{B_i + C_i}{T_i} + \sum_j \frac{C_j}{T_j}$$

$$R_3^0 = 18$$

$$R_3^1 = 6+8+\left\lceil\frac{18}{25}\right\rceil 7 + \left\lceil\frac{18}{10}\right\rceil 3 = 27$$

$$R_3^2 = 14+\left\lceil\frac{27}{25}\right\rceil 7 + \left\lceil\frac{27}{10}\right\rceil 3 = 37$$

$$R_3^3 = 14+\left\lceil\frac{37}{25}\right\rceil 7 + \left\lceil\frac{37}{10}\right\rceil 3 = 40$$

$$R_3^4 = 14+\left\lceil\frac{40}{25}\right\rceil 7 + \left\lceil\frac{40}{10}\right\rceil 3 = 40 \leq 50 \checkmark \Rightarrow \text{schedulable par RM}$$

2. Consider four periodic tasks τ_1, τ_2, τ_3 and τ_4 (in decreasing priority order, *i.e.*, τ_1 has the highest priority) that share five resources guarded by semaphores, S_A, S_B, S_C, S_D, and S_E. Compute the maximum blocking time B_i for each task if the resources are accessed using:
   a) Priority Inheritance Protocol (PIP)
   b) Priority Ceiling Protocol (PCP)

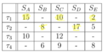
	S_A	S_B	S_C	S_D	S_E
τ_1	15	-	10	-	2
τ_2	-	8	-	17	5
τ_3	10	-	12	-	-
τ_4	-	6	9	-	8

The longest durations of critical sections for each task and for each semaphore are given in the table (no nested critical sections). Use the fact that task τ_i:

- can be blocked only once by each lower-priority task τ_j
- can be blocked only once on each semaphore S_k

PCP $\quad B_i = \max \{ D_{j,k} \mid j > i,\ C(S_k) \leq i \}$

Priority Ceilling



	S_A	S_B	S_C	S_D	S_E
τ_1	15	-	10	-	2
τ_2	-	8	-	17	5
τ_3	10	-	12	-	-
τ_4	-	6	9	-	8

$$\Rightarrow \quad C(S_A) = C(S_C) = C(S_E) = 1$$
$$C(S_B) = C(S_D) = 2$$

$B_4 = 0$; $B_3 = 9$; $B_2 = 12$; $B_1 = 12$

PIP : $B_4 = 0$; $B_3 = \max(T_4) = 9$; $B_2 = \max(12+8, 10+9) = 20$; $B_1 = 5+10+9 = 24$

3. Verify the schedulability under Deadline Monotonic (DM) of the following task set if the shared resources are accessed according to:
   a) Priority Inheritance Protocol (PIP)
   b) Priority Ceiling Protocol (PCP)

	S_A	S_B	S_C
τ_1	3	3	-
τ_2	12	-	12
τ_3	8	10	-

	C_i	T_i	D_i	B_i^{PIP}	B_i^{PCP}
τ_1	8	50	25	22	12
τ_2	45	80	80	10	10
τ_3	20	150	140	0	0

U ne sert à rien pour savoir si c'est schedulable

PIP

$R_3^0 = 73$

$R_3^1 = \underline{0} + 20 + \left\lceil \frac{73}{80} \right\rceil 45 + \left\lceil \frac{73}{50} \right\rceil 8 = 81$

$R_3^2 = 20 + \left\lceil \frac{81}{80} \right\rceil 45 + \left\lceil \frac{81}{50} \right\rceil 8 = 126$

$R_3^3 = 20 + \left\lceil \frac{126}{80} \right\rceil 45 + \left\lceil \frac{126}{50} \right\rceil 8 = 134$

$R_3^4 = 20 + \left\lceil \frac{134}{80} \right\rceil 45 + \left\lceil \frac{134}{50} \right\rceil 8 = 134$ ✓

PCP m̂ resultat car $B_3^{PCP} = B_3^{PIP} = 0$

4. Find the largest worst-case execution time of task τ_2 such that the task set remains schedulable under Rate Monotonic (use analytical methods):

	C_i	T_i
τ_1	10	16
τ_2	$\mathbf{C_2}$	50

U puis par itération sur RTA.

par dichotomie