

TD temps réel 1

CORRIGE

1 INTRODUCTION : RM, EDF, LLF	2
1.1 QUESTION 1	2
2 THEOREME DE LA ZONE CRITIQUE	5
2.1 CALCUL DE U	5
2.2 THEOREME DE LA ZONE CRITIQUE	5
3 RATE MONOTONIC OU EARLIEST DEADLINE	6
3.1 QUESTION 1	6
3.2 QUESTION 2	6
3.3 QUESTION 3	6
4 TEMPS DE REPONSE DE BOUT EN BOUT	7
4.1 RAPPELS	7
4.2 EXEMPLE	8
4.2.1 Initialisations	8
4.2.2 Itération 1	9
4.2.3 Itération 2	9
4.2.4 Itération 3	10
4.2.5 Résultat	10
4.3 EXEMPLE 2	11
4.3.1 Bus CAN	11
4.3.2 Initialisations	12
4.3.3 Itération 1	12
4.3.4 Itération 2	12
4.3.5 Itération 3	13
4.3.6 Itération 4	13
4.3.7 Itération 5	14
4.3.8 Résultat	14
5 SERVEUR DIFFERE	15
5.1 QUESTION 1	15
6 SERVEUR DE SCRUTATION	16
6.1 QUESTION 1	16
6.2 QUESTION 2	17

1 Introduction : RM, EDF, LLF

Trois tâches, prêtes à $t = 0$:

T1 : (C=1, D=P=3)

T2 : (C=1, D=P=4)

T3 : (C=2, D=P=6)

1.1 Question 1

Calcul de U

$$U = 1/3 + 1/4 + 2/6 = 0,33 + 0,25 + 0,33 = 0,91.$$

Le coefficient U est inférieur à 1 : l'ensemble des tâches est ordonnançable par EDF.

Il est supérieur à 0,78 : on ne peut **rien** dire pour RMS : en effet U inférieur ou égal à $N(2^{1/N}-1)$ est une condition **suffisante** pour RMS.

Remarques :

- RMS : priorité fixe,
- EDF : priorité dynamique, moins de « temps morts » que RMS, mais moins bon comportement en cas de surcharge. Plus coûteux à implémenter.
- LLF prend en compte le coefficient C, ce qui n'est fait ni par RMS, ni par EDF. Encore plus coûteux à implanter que EDF.

Rappel : dans tous les cas le calcul d'ordonnancement n'est fait que pour les tâches qui ont été activées ou réactivées...

Schémas d'ordonnancement

On rappelle la valeur du PPCM des périodes : 12.

1.1.1.1 Ordonnancement RMS

La décision d'ordonnancement est prise à chaque activation de tâche.

Schéma d'ordonnancement en appliquant RMS :

Date	1	2	3	4	5	6	7	8	9	10	11	12
T1	■			■			■			■		
T2		■			■			■				
T3			■			■		■			■	

Commentaire :

Bien que la condition suffisante ne soit pas satisfaite, on constate sur le schéma que l'ensemble des trois tâches est ordonnançable par RMS.

Il reste un créneau libre en 11 : aucune tâche n'a été réactivée, la prochaine date d'activation est 12 pour toutes les tâches.

1.1.1.2 Ordonnancement EDF (1)

La décision d'ordonnancement est prise à chaque activation de tâche.

Schéma d'ordonnancement en appliquant EDF :

Date	1	2	3	4	5	6	7	8	9	10	11	12
T1	■			■			■			■		
T2		■				■			■			
T3			■		■			■			■	

Commentaires :

Au temps 3, deux tâches ont leur échéance pour la date 6 :

- T1, qui a besoin d'une unité de temps pour sa deuxième période,
- T3, qui a besoin d'une seconde unité de temps pour sa première période.

On peut choisir :

- T1, parce qu'elle est la première dans l'ordre de la numérotation (sa période est plus courte),
- T3, pour éviter de faire un changement de contexte,

Ici, on choisit T1.

T3 est réordonnancée en 5 pour sa seconde unité de temps.

Au temps 9, deux tâches ont leur échéance pour la date 12 :

- T2, qui a besoin d'une unité de temps pour sa deuxième période,
- T3, qui a besoin d'une seconde unité de temps pour sa deuxième période.

Comme précédemment, on peut choisir :

- T2, parce qu'elle est la première dans l'ordre de la numérotation (sa période est plus courte),
- T3, pour éviter de faire un changement de contexte,

Ici, on choisit T2.

T1 s'insère alors, puis T3 consomme sa seconde unité de temps.

Il reste un créneau libre en 11 : aucune tâche n'a été réactivée, la prochaine date d'activation est 12 pour toutes les tâches.

1.1.1.3 Ordonnancement EDF (2)

Reprenons le scénario, en choisissant maintenant de toujours réduire le nombre de changements de contexte :

Date	1	2	3	4	5	6	7	8	9	10	11	12
T1	■			■			■			■		
T2		■				■			■			
T3			■	■				■	■			

Au temps 4, on choisit T3.

Au temps 7, on peut choisir T1 ou T3, on choisit T1

Au temps 9, on peut choisir T2 ou T3, on choisit T3 qui est déjà active, pour éviter le changement de contexte.

Il reste un créneau libre en 11 : aucune tâche n'a été réactivée, la prochaine date d'activation est 12 pour toutes les tâches.

Remarque :

EDF, comme RMS, ne prend pas en compte le coefficient C pour calculer les priorités.

1.1.1.4 Ordonnancement LLF

On donne maintenant un schéma d'ordonnancement en appliquant l'algorithme LLF (*Least Laxity First*), algorithme qui prend en compte la capacité C demandée par une tâche à chacune de ses activations.

Rappel :

La tâche qui sera ordonnancée est celle dont la « laxité », ou marge, est la plus petite, sachant que :
 $\text{marge} = \text{échéance} - \text{temps de calcul restant} - \text{date courante}$.

Le calcul d'ordonnancement peut être fait à chaque incrément d'horloge (uniquement pour les tâches actives), ou bien à chaque activation de tâche. Dans le premier cas il y a plus de changements de contexte.

Date	0	1	2	3	4	5	6	7	8	9	10	11
T1	2			2			2			2		
T2		2				2			3			
T3			2		1			3			1	

Pour chaque date, on visualise les calculs pour toutes les tâches actives :

	date		
Tâche 1	marge	échéance	C restant
Tâche 2	marge	échéance	C restant
Tâche 3	marge	échéance	C restant

Ce qui donne :

Dat e	0	1	2	3	4	5	6
T1	2 3 1			2 6 1			2 9 1
T2	3 4 1	2 4 1			3 8 1	2 8 1	
T3	4 6 2	3 6 2	2 6 2	2 6 1	1 6 1		4 12 2

En 3, on peut choisir T1 ou T3, on choisit T1 qui vient de se réveiller et qui est la plus prioritaire
 choisir T3 entraînerait moins de changements de contexte. On pourrait introduire la notion de criticité.

Dat e	7	8	9	10	11
T1			2 12 1		
T2		3 12 1			
T3	3 12 2	3 12 1	2 12 1	1 12 1	

En 8, on peut choisir T2 ou T3, on choisit T2 qui vient de se réveiller et qui est la plus prioritaire,
 choisir T3 entraînerait moins de changements de contexte.
 De même, on choisit T1 en 9. T3 est de plus en plus retardée...

Il reste un créneau libre en 11 : aucune tâche n'a été réactivée, la prochaine date d'activation est 12 pour toutes les tâches.

2 Théorème de la zone critique

Trois tâches, prêtes à $t = 0$:

T1 : (C=25, D=P=100)

T2 : (C=50, D=P=200)

T3 : (C=100, D=P=300)

2.1 Calcul de U

Calcul de U :

$U = 25/100 + 50/200 + 100/300 = 0,25 + 0,25 + 0,33 = 0,83$.

U est donc $> 0,78$, on ne peut **rien** dire pour RMS.

2.2 Théorème de la zone critique

Rappel de la formule :

$$W_i(t) = \sum_{j=1}^i C_j \lceil t/T_j \rceil$$

- $W_i(t)$ est alors la consommation de temps cpu demandée à la date t par les i premiers processus, ceux-ci étant numérotés par ordre de priorité croissante : $\text{priorité}(P_n) > \text{priorité}(P_{n+1})$
- $\lceil t/T_j \rceil$ donne le nombre d'activations de la tâche j , de période T_j , dans la fenêtre de temps de taille t ,
- $\lceil t/T_j \rceil * C_j$ est donc le temps cpu consommé par T_j dans cette fenêtre,
- $\sum_{j=1}^{(i-1)} C_j \lceil t/T_j \rceil$ est le retard imposé à la tâche T_i par les tâches d'indices inférieurs, donc plus prioritaires.

Notions corrélées : période d'activité

On va chercher si on peut trouver $W_i(t) = t$

On initialise avec $W_i(0) = \sum_{j=1}^i C_j$, c'est à dire $t_0 = \varepsilon$

Calcul pour $i = 1$:

$$W_1(0) = 25$$

$$W_1(25) = 25 \lceil 25/100 \rceil = 25 \Rightarrow \text{OK}$$

Calcul pour $i = 2$:

$$W_2(0) = 25 + 50 = 75$$

$$W_2(75) = 25 \lceil 75/100 \rceil + 50 \lceil 75/200 \rceil = 75 \Rightarrow \text{OK}$$

Calcul pour $i = 3$:

$$W_3(0) = 25 + 50 + 100 = 175$$

$$W_3(175) = 25 \lceil 175/100 \rceil + 50 \lceil 175/200 \rceil + 100 \lceil 175/300 \rceil = 25*2 + 50*1 + 100*1 = 200$$

$$W_3(200) = 25 \lceil 200/100 \rceil + 50 \lceil 200/200 \rceil + 100 \lceil 200/300 \rceil = 25*2 + 50*1 + 100*1 = 200 \Rightarrow \text{OK}$$

3 Rate monotonic ou Earliest deadline

3.1 Question 1

3.1.1.1 Liste des tâches

Liste des tâches :

T1 : (C=10, T=40)

T2 : (C=20, T=60)

T3 : (C=20, T=80)

$$U = 1/4 + 2/6 + 2/8 = 0,25 + 0,33 + 0,25 = 0,83$$

U est donc supérieur à 0,78 : on ne peut **rien** dire pour RM, l'ordonnancement EDF est possible.

PPCM = 240

3.1.1.2 Diagramme RMS

Diagramme d'ordonnancement obtenu en appliquant RMS (par 10 unités de temps) :

Date	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
T1	■				■				■				■				■				■			
T2		■	■				■	■						■	■				■	■				
T3				■		■				■	■							■				■		

3.1.1.3 Diagramme EDF

Diagramme d'ordonnancement obtenu en appliquant EDF :

Date	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
T1	■					■			■				■				■					■		
T2		■	■				■	■						■	■			■	■					
T3				■	■					■	■									■	■			

3.2 Question 2

Tâche nettoyage T4 : (C=10, T=60)

Calcul de la nouvelle valeur de U, en prenant en compte T4:

$$U = (1/4 + 2/6 + 2/8) + 1/6 = 0,83 + 0,16 = 0,99$$

La prise en compte de l'arrosage dans ces conditions est possible avec EDF.

Elle n'est pas possible avec RMS, en effet les seuls créneaux libres en RMS sont :

[11,12], [15,16] et [19,20].

3.3 Question 3

Début en 90, il faut donc finir avant 150, ce n'est pas possible : un seul créneau libre avant 150, c'est le 120 pendant 10 unités de temps.

Début en 90, il faut donc finir avant 190, c'est possible parce qu'en 120 et 160 il y a des créneaux de 10.

4 Temps de réponse de bout en bout

4.1 Rappels

Charge due aux tâches de priorité supérieure à la tâche T_i :

$$W_i(t) = C_i + \sum_{(j > \text{prio}(i))} \left\lceil \frac{t}{P_j} \right\rceil * C_j$$

Le calcul se fait par recherche itérative du point fixe : $W_i(t) = t$

Dans le cas de tâches communiquant par messages (cf. hypothèses sur les émissions et les réceptions), on ajoute la gigue qui, a priori, n'est pas connue :

$$W_i(t) = C_i + \sum_{(j > \text{prio}(i))} \left\lceil \frac{t + J_j}{P_j} \right\rceil * C_j$$

Le pire temps de réponse pour une tâche T_i est donné par :

$$R_i = J_i + W_i(t) \text{ avec } W_i(t) = t$$

Arrêt : $R_i(k) = R_i(k-1)$

Principe de l'analyse holistique : itérer jusqu'au point fixe $W(R_i) = R_i$

1. initialiser les giges et les temps de réponse :

- $J_i = 0$
- $W_i = C_i + \sum_{(j > \text{prio}(i))} \left\lceil \frac{J_j + \varepsilon}{P_j} \right\rceil * C_j = C_i + \sum_{(j > \text{prio}(i))} C_j$
- $R_i^{\text{courant}} =$
 - pour les tâches : $R_i^{\text{courant}} = J_i + W_i = C_i + \sum_{(j > \text{prio}(i))} C_j$
 - pour les messages : $R_i^{\text{courant}} = J_i + M_i$
- $R_i^{\text{ancien}} = 0$

2. itérations, tant qu'il existe un R_i^{ancien} différent de R_i^{courant} :

- pour i de 1 à n : on va mettre à jour le gigue en utilisant les temps de réponse. On dit que T_j précède la T_i , si j est la précédente de i dans le graphe de communication :
 - $J_i = \max(J_i, \max_{j \text{ précède } i} (R_j^{\text{courant}}))$
- pour tous les i : calculer les temps de réponses par tâche, par site et par réseau,
 - $R_i^{\text{ancien}} = R_i^{\text{courant}}$
 - $R_i^{\text{courant}} = \text{CalculTempsRep}(i)$

3. aller à 2

Calcul du temps de réponse

CalculTempsRep(i, k)	
Tâche	$R_i = J_i + W_i$ avec : $W_i = C_i + \sum_{(j > \text{prio}(i))} \left\lceil \frac{J_j + W_i}{P_j} \right\rceil * C_j$ On prend en compte les tâches qui ont une priorité supérieure à la tâche courante
Message	$R_i = J_i + M_i$ M_i est le délai de transmission du message entre la tâche émettrice et la tâche destinataire.

$$J_i = \max(J_i, \max_{j \text{ précède } i} (R_j^{\text{courant}}))$$

$$W_i = C_i + \sum_{(j > \text{prio}(i))} \left\lceil \frac{J_j + W_i}{P_j} \right\rceil * C_j$$

Si tâche : $R_i = J_i + W_i$

Si message : $R_i = J_i + M_i$

initialisation	Pour tous messages et tâches : $J_j = 0, R_i = 0$ Pour tous messages et tâches : calcul du temps de réponse r_i .
Calcul itératif	Tant que : $(\exists i : r_i \neq r'_i) \{$
	$\forall i :$
	$J_i = \max(J_i, \forall j \text{ tel que } j < i : r_j)$
	$r'_i = r_i$
	CalculTempsRep(r_i)
	$\}$

$\forall j$ avec $j < i : r_j$ (r_j pour tous les j tel que T_j précède T_i)

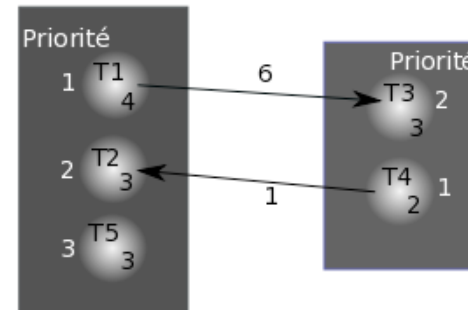
4.2 Exemple

Caractéristiques des tâches:

Tâche	Période	Capacité	Priorité	Processeur
T1	100	4	1	A
T2	60	5	2	A
T3	100	3	2	B
T4	60	2	1	B
T5	90	3	3	A

Caractéristiques des messages :

Message	Sens	Délai de comm.
M1	De T1 vers T3	6
M2	De T4 vers T2	1



4.2.1 Initialisations

T3 dépend de T1 par M1

T2 dépend de T4 par M2

$$J_i = 0$$

$$W_i = C_i + \sum_{(j > \text{prio}(i))} \left\lceil \frac{J_j + \varepsilon}{P_j} \right\rceil * C_j = C_i + \sum_{(j > \text{prio}(i))} C_j$$

$$R_i^{\text{ancien}} = 0$$

$$\text{Si tâche : } R_i^{\text{courant}} = J_i + W_i = C_i + \sum_{(j > \text{prio}(i))} C_j$$

Si message : $R_i^{\text{courant}} = M_i$

	T1(A)	T2(A)	T3(B)	T4(B)	T5(A)	M1	M2
Gigue J_i	0	0	0	0	0	0	0
W_i	4	5+4=9	3+2=5	2	3+5+4=12	6	1
Coeff. R_i	4	5+4=9	3+2=5	2	3+5+4=12	6	1

4.2.2 Itération 1

M1 précède T3

M2 précède T2

Gigue : Temps de réponse du message attendu ($J_{T3} = R_{M1}$ et $J_{T2} = R_{M2}$)

$J_i = \max (\max_{j \text{ préc } i} (R_j^{\text{courant}}))$
 $W_i = C_i + \sum_{(j > \text{prio}(i))} \lceil (J_j + W_j) / P_j \rceil * C_j$
 Si tâche : $R_i = J_i + W_i$
 Si message : $R_i = J_i + M_i$

T1 et T4 sont les plus prioritaires sur leurs sites respectifs et ne dépendent de personne, donc pas de changement.

Nouveaux calculs pour les tâches T2, qui dépend de T4, et T3, qui dépend de T1 :

$J_2 = \max_{j \text{ préc } i} (R_j^{\text{courant}}) = R_{M2} = 1$ (M2 précède T2)

$J_3 = \max_{j \text{ préc } i} (R_j^{\text{courant}}) = R_{M1} = 6$ (M1 précède T3)

$J_5 = 0$

Pour les messages M1 et M2 :

$J_i = \max (\max_{j \text{ préc } i} (R_j^{\text{courant}}))$ donc, ici : $J_{M1} = \max(4, R_{T1}) = \max(4, 4)$ et $J_{M2} = \max(2, R_{T4}) = \max(2, 2)$,

$W_2 = C_2 + \lceil (J_1 + W_2) / P_1 \rceil * C_1 = 5 + \lceil (0 + 5) / 100 \rceil * 4 = 5 + 4 = 9$

$W_3 = C_3 + \lceil (J_4 + W_3) / P_4 \rceil * C_4 = 3 + \lceil (0 + 3) / 100 \rceil * 2 = 3 + 2 = 5$

T5 ne dépend de personne et les tâches plus prioritaires qu'elle ont de coefficients fixes.

$W_5 = C_5 + \lceil (J_1 + W_5) / P_1 \rceil * C_1 + \lceil (J_2 + W_5) / P_2 \rceil * C_2$
 $= 3 + \lceil (0 + 12) / 100 \rceil * 4 + \lceil (1 + 12) / 60 \rceil * 5$
 $= 3 + 4 + 5 = 12$

J_i ne variera plus, seul J_2 peut changer.

Temps de réponse : $R_i = J_i + W_j$

Temps de réponse pour les messages M1 et M2 : $R_i = J_i + M_i$

	T1(A)	T2(A)	T3(B)	T4(B)	T5(A)	M1	M2
J_i	0	1	6	0	0	4	2
W_i	4	9 (fixe)	5 (fixe)	2	3+5+4=12	6	1
R_i	4	10	6+5=11	2	12	6+4=10	1+2=3

4.2.3 Itération 2

On arrive avec :

$W_2 = 9$

$W_3 = 5$

$W_5 = 12$

Nouveaux calculs pour les tâches T2, qui dépend de T4, et T3, qui dépend de T1 :

$J_2 = \max_{j \text{ préc } i} (R_j^{\text{courant}}) = R_{M2} = 3$ (M2 précède T2)

$J_3 = \max_{j \text{ préc } i} (R_j^{\text{courant}}) = R_{M1} = 10$ (M1 précède T3)

Pour les messages M1 et M2 :

$J_i = \max (\max_{j \text{ préc } i} (R_j^{\text{courant}}))$ donc, pas de changement puisque aucun coeff. n'a été modifié depuis l'itération précédente ni T1 qui émet M1 ni pour T4 qui émet M2.

Les W_i de T2 et T3 ne varient plus (les giges des tâches plus prioritaires sont fixées), leurs R_i ne dépendant que de leur propre gigue :

$W_2 = C_2 + \lceil (J_1 + W_2) / P_1 \rceil * C_1 = 5 + \lceil (0 + 9) / 100 \rceil * 4 = 5 + 4 = 9$

$W_3 = C_3 + \lceil (J_4 + W_3) / P_4 \rceil * C_4 = 3 + \lceil (0 + 5) / 100 \rceil * 2 = 3 + 2 = 5$

Pour T5 :

$W_5 = C_5 + \lceil (J_1 + W_5) / P_1 \rceil * C_1 + \lceil (J_2 + W_5) / P_2 \rceil * C_2$
 $= 3 + \lceil (0 + 12) / 100 \rceil * 4 + \lceil (3 + 12) / 60 \rceil * 5 = 12$

Temps de réponse : $R_i = J_i + W_j$, donc R_2 et R_3 changent puisque J_{M1} et J_{M2} ont été modifiées.

Pour les messages M1 et M2 :

Les de M1 et M2 dépendent de R_1 et R_4 qui sont fixes.

	T1(A)	T2(A)	T3(B)	T4(B)	T5(A)	M1	M2
J_i	0	3	10	0	0	4	2
W_i	4	9	5	2	12	6	1
R_i	4	3+9=10	10+5=15	2	3+5+4=12	6+4=10	1+2=3

4.2.4 Itération 3

On arrive avec :

$W_2 = 9$

$W_3 = 5$

$W_5 = 12$

Nouveaux calculs pour les tâches T2, qui dépend de T4, et T3, qui dépend de T1 :

$J_2 = \max_{j \text{ préc } i} (R_j^{\text{courant}}) = R_{M2} = 3$ (M2 précède T2)

$J_3 = \max_{j \text{ préc } i} (R_j^{\text{courant}}) = R_{M1} = 10$ (M1 précède T3)

Les giges calculées pendant cette itération sont égales à celles de la précédente, arrêt des calculs.

$W_2 = C_2 + \lceil (J_1 + W_2) / P_1 \rceil * C_1 = 5 + \lceil (0 + 9) / 100 \rceil * 4 = 5 + 4 = 9$

$W_3 = C_3 + \lceil (J_4 + W_3) / P_4 \rceil * C_4 = 3 + \lceil (0 + 5) / 100 \rceil * 2 = 3 + 2 = 5$

Temps de réponse : $R_i = J_i + W_j$

	T1(A)	T2(A)	T3(B)	T4(B)	T5(A)	M1	M2
J_i	0	3	10	0	0	4	2
W_i	4	9	5	2	12	6	1
R_i	4	3+9=12	10+5=15	2	12	6+4=10	1+2=3

4.2.5 Résultat

Processeur A (T1 envoie M1 vers T3, T2 activée par T4 via M2) :

	1	2	3	4	5	6	7	8	9	10	11	12
T1												
T2	gigue	gigue	gigue									
T5												

Processeur B (T4 envoie M1 vers T2, T3 activée par T1 via M1) :

	1	2	3	4	5	6	7	8	9	10	11	12
T4												
T3	gigue	gigue	gigue	gigue	gigue	gigue	gigue	gigue	gigue	gigue		

Bus :

	1	2	3	4	5	6	7	8	9	10	11	12
M2	gigue	gigue										
M1	gigue	gigue	gigue	gigue								

4.3 Exemple 2

Caractéristiques des tâches:

Tâche	Capacité	Période	Priorité	Processeur
T1	5	20	1	A
T2	10	35	2	C
T3	10	25	2	A
T4	5	50	3	C
T5	15	60	3	B

Caractéristiques des messages :

Message	Période	Sens	Délai de comm.
M1		T1 vers T4	10
M2		T3 vers T5	10
M3		T4 vers T5	2

4.3.1 Bus CAN

Pour n bits, la taille maximale de la trame est :

$$L_{\max} = 47 + (8*n) + \lfloor ((34 + (8*n) - 1) / 4) \rfloor \text{ bits}$$

Bits de données	0	8	16	24	32	40	48	56	64
L_{\max}	55	65	75	85	95	105	115	125	135
Ou délai, en μ s, pour un débit de 1 Mbits/s									
Efficacité (%)	0	12	21	28	34	38	42	45	47

Donc 8 octets en 135 μ s, Si la base de temps (tic) est de 1ms, pendant un tic on peut transmettre $10^{-3} / (135*10^{-6}) = 10^3/135 = 7$ paquets de 8 octets

4.3.2 Initialisations

$$J_i = 0$$

$$W_i = C_i + \sum_{(j > \text{prio}(i))} C_j$$

$$\text{Si tâche : } R_i = J_i + W_i = C_i + \sum_{(j > \text{prio}(i))} C_j$$

$$\text{Si message : } R_i = M_i$$

	M1	M2	M3	T1(A)	T2(C)	T3(A)	T4(C)	T5(B)
J_i	0	0	0	0	0	0	0	0
W_i				5	10	10+5=15	5+10=15	15
R_i	10	10	2	5	10	15	15	15

4.3.3 Itération 1

M1 dépend de T1

M2 dépend de T3

M3 dépend de T4

T4 dépend de M1

T5 dépend de M2 et M3

Les giques de T1, T2 et T3 resteront à 0.

De plus T1 est la plus prioritaire sur A, donc aucun coefficient ne changera pour elle.

Calculs :

$$J_i = \max(J_i, \max_{j \text{ prec } i}(R_j))$$

$$R_i \text{ pour les messages : } R_i = J_i + M_i$$

R_i pour les tâches :

$$W_2 = C_2 = 10$$

$$R_2 = J_2 + 10 = 10$$

$$W_3 = C_3 + \lceil (J_1 + W_3) / P_1 \rceil * C_1 = 10 + \lceil (0 + 15) / 20 \rceil * 5 = 10 + 5 = 15$$

$$R_3 = J_3 + W_3 = 0 + 15 = 15$$

$$W_4 = C_4 + \lceil (J_2 + W_4) / P_2 \rceil * C_2 = 5 + \lceil (0 + 15) / 35 \rceil * 10 = 15$$

$$R_4 = J_4 + 15 = 10 + 15 = 25$$

$$W_5 = C_5 = 15$$

$$R_5 = J_5 + W_5 = 10 + 15 = 25$$

	M1	M2	M3	T1(A)	T2(C)	T3(A)	T4(C)	T5(B)
J_i	5	15	15	0	0	0	10	10
W_i				5	10	10+5=15	15	15
R_i	15	25	17	5	10	15	25	25

4.3.4 Itération 2

M1 dépend de T1

M2 dépend de T3

M3 dépend de T4

T4 dépend de M1

T5 dépend de M2 et M3

Les giques de T1, T2 et T3 resteront à 0.

De plus T1 est la plus prioritaire sur A, donc aucun coefficient ne changera pour elle.

Calculs :

$$J_i = \max(J_i, \max_{j \text{ prec } i}(R_j))$$

$$R_i \text{ pour les messages : } R_i = J_i + M_i$$

R_i pour les tâches :

$$\begin{aligned}
W_2 &= C_2 = 10 \\
R_2 &= J_2 + W_2 = 0 + 10 = 10 \\
W_3 &= C_3 + \lceil (J_1 + W_3) / P_1 \rceil * C_1 = 10 + \lceil (0 + 15) / 20 \rceil * 5 = 10 + 5 = 15 \\
R_3 &= J_3 + W_3 = 0 + 15 = 15 \\
W_4 &= C_4 + \lceil (J_2 + W_4) / P_2 \rceil * C_2 = 5 + \lceil 15 / 35 \rceil * 10 = 5 + 10 = 15 \\
R_4 &= J_4 + 15 = 15 + 15 = 30 \\
W_5 &= C_5 = 15 \\
R_5 &= J_5 + W_5 = J_5 + 15 = 25 + 15 = 40
\end{aligned}$$

	M1	M2	M3	T1(A)	T2(C)	T3(A)	T4(C)	T5(B)
J_i	5	15	25	0	0	0	15	25
W_i				5	10	10+5=15	15	15
R_i	15	25	27	5	10	15	30	40

4.3.5 Itération 3

M1 dépend de T1

M2 dépend de T3

M3 dépend de T4

T4 dépend de M1

T5 dépend de M2 et M3

Les gîgues de T1, T2 et T3 resteront à 0.

De plus T1 est la plus prioritaire sur A, donc aucun coefficient ne changera pour elle.

Calculs :

$$J_i = \max(J_i, \max_{j \text{ prec } i}(R_j))$$

R_i pour les messages : $R_i = J_i + M_i$

R_i pour les tâches :

$$\begin{aligned}
W_2 &= C_2 = 10 \\
R_2 &= J_2 + W_2 = 0 + 10 = 10 \\
W_3 &= C_3 + \lceil (J_1 + W_3) / P_1 \rceil * C_1 = 10 + \lceil (0 + 15) / 20 \rceil * 5 = 10 + 5 = 15 \\
R_3 &= J_3 + W_3 = 0 + 15 = 15 \\
W_4 &= C_4 + \lceil (J_2 + W_4) / P_2 \rceil * C_2 = 5 + \lceil 15 / 35 \rceil * 10 = 5 + 10 = 15 \\
R_4 &= J_4 + 15 = 15 + 15 = 30 \\
W_5 &= C_5 = 15 \\
R_5 &= J_5 + W_5 = J_5 + 15 = 25 + 15 = 40
\end{aligned}$$

	M1	M2	M3	T1(A)	T2(C)	T3(A)	T4(C)	T5(B)
J_i	5	15	30	0	0	0	15	27
W_i				5	10	10+5=15	15	15
R_i	15	25	32	5	10	15	30	42

4.3.6 Itération 4

M1 dépend de T1

M2 dépend de T3

M3 dépend de T4

T4 dépend de M1

T5 dépend de M2 et M3

Les gîgues de T1, T2 et T3 resteront à 0.

De plus T1 est la plus prioritaire sur A, donc aucun coefficient ne changera pour elle.

Calculs :

$$J_i = \max(J_i, \max_{j \text{ prec } i}(R_j))$$

R_i pour les messages : $R_i = J_i + M_i$

R_i pour les tâches :

$$\begin{aligned}
W_2 &= C_2 + \lceil (J_4 + W_2) / P_4 \rceil * C_4 = 10 + \lceil (15 + 15) / 50 \rceil * 5 = 10 + 5 = 15 \\
R_2 &= J_2 + W_2 = 0 + 15 = 15 \\
W_3 &= C_3 + \lceil (J_1 + W_3) / P_1 \rceil * C_1 = 10 + \lceil (0 + 15) / 20 \rceil * 5 = 10 + 5 = 15 \\
R_3 &= J_3 + W_3 = 0 + 15 = 15 \\
W_4 &= C_4 + \lceil (J_2 + W_4) / P_2 \rceil * C_2 = 15 \\
R_4 &= J_4 + 15 = 15 + 15 = 30 \\
W_5 &= C_5 = 15 \\
R_5 &= J_5 + W_5 = J_5 + 15 = 32 + 15 = 47
\end{aligned}$$

	M1	M2	M3	T1(A)	T2(C)	T3(A)	T4(C)	T5(B)
J_i	5	15	30	0	0	0	15	32
W_i				5	10	10+5=15	15	15
R_i	15	25	32	5	10	15	30	47

4.3.7 Itération 5

Les gîgues de T1, T2 et T3 resteront à 0.

De plus T1 est la plus prioritaire sur A, donc aucun coefficient ne changera pour elle.

Calculs :

$$J_i = \max(J_i, \max_{j \text{ prec } i}(R_j))$$

R_i pour les messages : $R_i = J_i + M_i$

R_i pour les tâches :

$$\begin{aligned}
W_2 &= C_2 + \lceil (J_4 + W_2) / P_4 \rceil * C_4 = 10 + \lceil (15 + 15) / 50 \rceil * 5 = 10 + 5 = 15 \\
R_2 &= J_2 + W_2 = 0 + 15 = 15 \\
W_3 &= C_3 + \lceil (J_1 + W_3) / P_1 \rceil * C_1 = 10 + \lceil (0 + 15) / 20 \rceil * 5 = 10 + 5 = 15 \\
R_3 &= J_3 + W_3 = 0 + 15 = 15 \\
W_4 &= C_4 + \lceil (J_2 + W_4) / P_2 \rceil * C_2 = 15 \\
R_4 &= J_4 + 15 = 15 + 15 = 30 \\
W_5 &= C_5 = 15 \\
R_5 &= J_5 + W_5 = J_5 + 15 = 32 + 15 = 47
\end{aligned}$$

	M1	M2	M3	T1(A)	T2(C)	T3(A)	T4(C)	T5(B)
J_i	5	15	30	0	0	0	15	32
W_i								
R_i								

Les gîgues sont égaux à celles de l'itération précédente, arrêt des calculs.

4.3.8 Résultat

Processeur A (T1, prioritaire, envoi M1 vers T4, T3 envoi M1 vers T5) :

	1	2	3	4	5	6	7	8	9	10	...	15
T1												
T3											...	

Processeur B (T5 activé par T3 via M2 et par T4 via M3) :

	1	2	32	47
--	---	---	-----	-----	----	-----	-----	----

Tscrut : (C=5, T =25), il va consommer TOUT SON BUDGET à chaque activation

T1 : (C=5, T =30)

T2 : (C=10, T =50)

T3 : (C=25, T =75)

Date	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100	105	110	115	120	125
Ts																									
T1																									
T2																									
T3																									

6.2 Question 2

Diagramme d'ordonnancement dans le cas où le serveur doit traiter les deux tâches apériodiques:

TA1 : (C=8, réveil en 6)

TA2 : (C=15, réveil en 20)

Les tâches T1 à T3 sont toujours présentes :

T1 : (C=5, T =30)

T2 : (C=10, T =50)

T3 : (C=25, T =75)

Date	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100	105	110	115	120	125
Ts																									
T1																									
T2																									
T3																									

Commentaires :

Au temps 0, le serveur de scrutation n'a rien à faire, donc il rend la main, son budget est perdu.

Au temps 25, il consomme 5 unités de temps pour TA1.

Au temps 50, il consomme 3 unités de temps pour TA1 et 2 unités de temps pour TA2.

Au temps 75, il consomme 5 unités de temps pour TA2.

Au temps 100, il consomme 5 unités de temps pour TA2.

Au temps 125, il consomme 2 unités de temps pour TA2.