TD1 - ORDONNANCEMENT

1. Supposez 5 processus (P0 → P4) suivants sont prêts pour l'exécution. Leur temps d'arrivage et leur durée sont spécifiés par la table ci-après :

<u>Processus</u>	<u>Arrivé</u>	<u>Durée</u>
<i>P</i> 0	1	10
P1	1	8
P2	2	3
P3	4	4
P4	5	2

a) Faire l'ordonnancement de ces processus avec les algorithmes FIFO, tourniquet (quantum du temps = 2 et quantum du temps = 5), SJF.

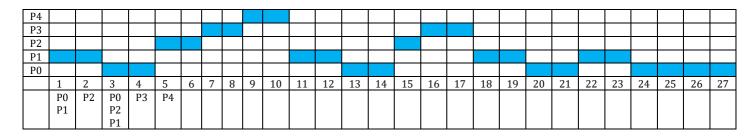
FIFO

P4																											
Р3																											
P2																											
P1																											
P0																											
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
	P0 P1	P2		Р3	P4																						

temps d'attente de P0 : 8, temps d'attente de P1 : 0, temps d'attente de P2 : 18, temps d'attente de P3 : 21, temps d'attente de P4 : 25,

⇒ temps d'attente moyen : 14.4

Tourniquet avec quantum du temps = 2



temps d'attente de P0 : 2, temps d'attente de P1 : 0 temps d'attente de P2 : 4 temps d'attente de P3 : 6 temps d'attente de P4 : 8

⇒ temps d'attente moyen : 4

Tourniquet avec quantum du temps = 5

P4																											
Р3																											
P2																											
P1																											
P0																											
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
	P0	P2		Р3	P4																						
	P1																										1

temps d'attente de P0 : 5 temps d'attente de P1 : 0 temps d'attente de P2 : 10 temps d'attente de P3 : 13 temps d'attente de P4 : 17

⇒ temps d'attente moyen : 9

SJF (préemtive)

P4																											
Р3																											
P2																											
P1																											
P0																											
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
	P1	P2		Р3	P4																						
	P0																										

t1: P1(8) - P0(10)

t2: P2(2) - P1(7) - P0(10)

t3: P2(1) - P1(7) - P0(10)

t4: P2(0) - P3(4) - P1(7) - P0(10)

t5: P4(1) - P3(4) - P1(7) - P0(10)

t6: P4(0) - P3(4) - P1(7) - P0(10)

t7: P3(3) - P1(7) - P0(10) 8

t8: P3(2) - P1(7) - P0(10)

t9: P3(1) - P1(7) - P0(10)

t10: P3(0) - P1(7) - P0(10)

t17 : P0(10)

temps d'attente de P0 : 17 temps d'attente de P1 : 0 temps d'attente de P2 : 1 temps d'attente de P3 : 6 temps d'attente de P4 : 4

⇒ temps d'attente moyen : 5.6

SJF (non préemtive)

P4																											
Р3																											
P2																											
P1																											
P0																											
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
	P1																										
	P0																										

t(1): P1(8) - P0(10)

t(8): P1(0) - P4(2)-P2(3) - P3(4) - P0(10)

t(10): P4(0)-P2(3) - P3(4) - P0(10)

t(13) : P2(0) - P3(4) - P0(10)

t(17): P3(0) - P0(10)

temps d'attente de P0 : 17 temps d'attente de P1 : 0 temps d'attente de P2 : 10 temps d'attente de P3 : 13 temps d'attente de P4 : 8

⇒ temps d'attente moyen : 9.6

b) Avec chaque algorithme, donner le temps d'exécution pour chaque processus, temps d'exécution moyen et temps d'attente moyen.

temps d'exécution : 10 temps d'exécution : 8 temps d'exécution :3 temps d'exécution :4 temps d'exécution :2

⇒ temps d'exécution moyen : 5.4

FIFO: temps d'attente moyen: 14.4

Tourniquet avec quantum du temps = 2 : temps d'attente moyen : 4

Tourniquet avec quantum du temps = 5 : temps d'attente moyen : 9

SJF (préemtive) : temps d'attente moyen : 5.6 **SJF (non préemtive) :** temps d'attente moyen : 9.6

- c) Selon votre avis, quelle est la meilleure solution? Justifier
- ⇒ La meilleure solution est Tourniquet avec quantum du temps = 2 parce-que les processus n'ont pas à attendre trop longtemps pour être exécutés (le plus petit temps d'attente moyen des solutions ci-dessus (=4))

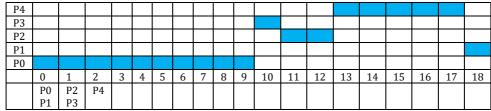
- 2. Proposer les algorithmes suivants pour ordonnancer les processus:
 - FIFO
 - SJF
 - Tourniquet
 - Ordonnancement avec priorité
 - a) Avec quel algorithme, y a-t-il un risque de famine?
 - ⇒ SJF est algorithme a un risque de famine
 - b) Quel processus peuvent-t-ils être les victimes?
 - c) Comment peut-t-on éviter ce problème? Donner votre avis.
 - **⇒** coordonner plusieurs algorithmes pour coordonner le processus
- **3.** Il y a 5 processus (P0 → P4) qui sont prêts pour l'exécution avec la durée et priorité comme dans la table ci-après :

<u>Processus</u>	<u>Arrivé</u>	<u>Durée</u>	<u>Priorité</u>
P0	0	10	3
P1	0	1	1
P2	1	2	3
Р3	1	1	4
P4	2	5	2

Supposez qu'après une fois d'exécution, la priorité du processus se diminue 1 unité.

- a) Faire l'ordonnancement de ces processus non préemptif avec priorité.
- b) Faire l'ordonnancement de ces processus préemptif avec priorité.
- c) Donner le temps d'exécution moyen et temps d'attente moyen pour chaque ordonnancement.

Priorité (non préemptif)



t(0): P0(10)(3) P1(1)(1)

t(10): P1(1) P2 P3(1) P4

t13:P1(1) P4(5)

t13: P4(5) P1

temps d'attente de P0 :0 temps d'attente de P1 : 18 temps d'attente de P2 : 11 temps d'attente de P3 : 10 temps d'attente de P4 : 13

 \Rightarrow temps d'attente moyen : 10.4

Priorité (préemptif)

P4																			
Р3																			
P2																			
P1																			
P0																			
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
	P0	P2	P4				_	_	_	,									
	P1	Р3																	

t0: P0(9)-2, P1(1)-1

t1: P3(0)-3, P2(2)-3, P0(9)-2, P1(1)-1

t2: P2(1)-2, P0(9)-2, P1(1)-1, P4(5)-2

t3: P2(0)-1, P0(9)-2, P1(1)-1, P4(5)-2

t4: P4(4)-1, P0(9)-2, P1(1)-1,

t5: P0(8)-1, P4(4)-1, P1(1)-1,

t6: P0(8)-1, P4(4)-1

t7: P0(8)-1, P4(3)-0

t8: P0(7)-0, P4(3)-0

t9: P0(7)-0, P4(2)--1

t10: P0(6)--1, P4(2)--1

t11: P0(6)--1, P4(1)--2

t12: P0(5)--2, P4(1)--2

t13: P0(5)--2, P4(0)

temps d'attente de P0 :0 temps d'attente de P1 : 6 temps d'attente de P2 : 2 temps d'attente de P3 : 1 temps d'attente de P4 : 4

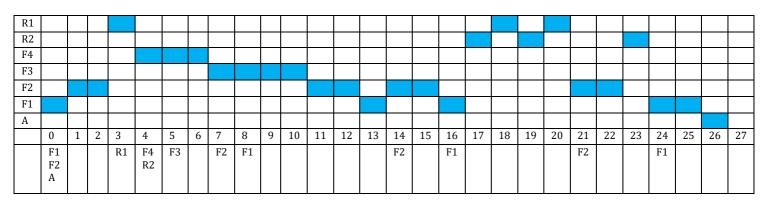
⇒ temps d'attente moyen : 2.6

	Université des Sciences de HCMV – Département d'Informatique
4.	On considère une implémentation de l'extension temps-réel de POSIX sur un système

d'exploitation avec 30 niveaux de priorité (de 0 à 29, 29 étant la plus forte priorité). Par conséquent, le système offre trois politiques d'ordonnancement: SCHED_FIFO (pour les processus FIFO temps-réel), SCHED_RR (pour les processus tourniquet temps-réel) et SCHED_OTHER (pour les processus temps partagé). Le quantum d'exécution pour la politique SCHED_RR est d'une unité de temps. L'ordonnancement est préemptif.

Processu s	Politique	Pattern d'arrivage	Période	Instant d'arriv	Temps d'exécutio	Priorité
				é	n	
Fifo1	SCHED_FIFO	Périodique	8	0	1	28
Fifo2	SCHED_FIFO	Périodique	7	0	2	26
Fifo3	SCHED_FIFO	Apériodique	-	5	4	20
Fifo4	SCHED_FIFO	Apériodique	-	4	3	20
Rr1	SCHED_RR	Apériodique	-	3	3	10
Rr2	SCHED_RR	Apériodique	-	4	3	10
Autre	SCHED_OTHER	Apériodique	-	0	1	0

Un processus périodique est un processus qui est activé régulièrement, aux temps 0, T, 2T, 3T, ... où T est la période du processus. Donnez le diagramme de Gantt pour l'ordonnancement des processus de la table 1 entre les instants 0 et 26.



- t(0) F1(1), F2(2), A(1)
- t(1) F2(1), A(1)
- t(2) F2(0), A(1)
- t(3) R1(1), A(1)
- t(4) F4(2) R1(1) R2(3) A(1)
- t(6) F4(0) R1(1) R2(3) A(1)
- t(7) R1(1) R2(3) A(1)F3(3)
- t(10) R1(1) R2(3) A(1)F3(0)
- t(11) R1(1) R2(3) A(1)F2(2)F1(0)
- t(13) R1(1) R2(3) A(1)F2(0)
- t(14) R1(1) R2(3) A(1)F2(1)
- t(15) R1(1) R2(3) A(1)F2(0)
- t(16) F1(0)R1(1) R2(3) A(1)
- t(17) R2(2)R1(2) A(1)

5. Nous utilisons un ordonnancement préemptif avec priorité. Nous allons utiliser un jeu de processus qui mélange des processus périodiques et des processus ponctuels. De plus, deux processus partagent une ressource R.

Processu s	Temps d'arrivage	Priorité*	Durée	Pemarque
A	0, 6, 12, 18, 24, 30	10	1	Périodique (quay lại tại time nhất định)
В	0, 10, 20, 30	8	4	Périodique, à chaque itération, le processus doit acquérir R à la fin du temps 1 et la libérer à la fin du temps 3. (tại thời điểm 2 và 3 phải có R thì B mới dc thực hiện tiếp, đang đợi tài nguyên thì ko dc vào hàng chờ)
С	18	5	6	Ponctuel (ko cần quay lại)
D	0	1	8	Ponctuel, à la fin du temps 6, le processus acquière R et la conserve jusqu'à la fin du temps 8. (cần R tại thời điểm 7-8

^{*}Plus la valeur de priorité est importance plus le processus est prioritaire.

a. Faire l'ordonnancement de ces processus sur 32 unités de temps.

```
t(0): A(1), B(4), D(8)
```

t(1) : D(7), B(4),

t(2) : D(6), B(4),

t(3): D(5), B(4),

t(4): D(4), B(4),

t(5) : D(3), B(4),

t(6): A(1), D(3), B(4),

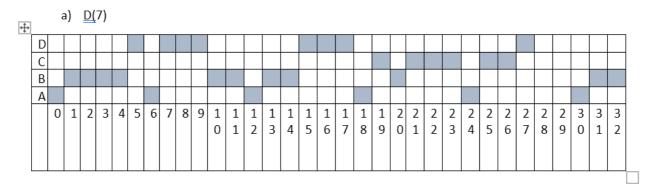
t(7) : D(2), B(4),

t(8) : D(1), B(4),

t(9) : D(0), B(4),

t(10) : B(4), B(4),

Question 5:



- b) Les temps de <u>réponse</u> de <u>chaque processus est :</u>
 - <u>A</u>:0
 - <u>B:</u>1
 - <u>C:</u>1
 - <u>D:</u>5