

⇒ temps d'attente moyen : 14.4

⇒ temps d'attente moyen : 4

### Tourniquet avec quantum du temps = 5

[illegible]

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 (préemptive)

[illegible]
$$t1 : P1(8) - P0(10)$$
$$t2 : P2(2) - P1(7) - P0(10)$$
$$t3 : P2(1) - P1(7) - P0(10)$$
$$t4 : P2(0) - P3(4) - P1(7) - P0(10)$$
$$t5 : P4(1) - P3(4) - P1(7) - P0(10)$$
$$t6 : P4(0) - P3(4) - P1(7) - P0(10)$$
$$t7 : P3(3) - P1(7) - P0(10) \quad 8$$
$$t8 : P3(2) - P1(7) - P0(10)$$
$$t9 : P3(1) - P1(7) - P0(10)$$
$$t_{10} : P_3(0) - P_1(7) - P_0(10)$$

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 préemptive)

[illegible]
$$t(1) : P1(8) - P0(10)$$
$$t(8) : P1(0) - P4(2) - P2(3) - P3(4) - P0(10)$$
$$t(10) : P4(0) - P2(3) - P3(4) - P0(10)$$
$$t(13) : P2(0) - P3(4) - P0(10)$$
$$t(17) : P3(0) - P0(10)$$

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 (préemptive) :** temps d'attente moyen : 5.6

**SJF (non préemptive) :** 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 à un risque de famine**

b) Quel processus peuvent-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
P3	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)**

P4																			
P3																			
P2																			
P1																			
P0																			
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
	P0 P1	P2 P3	P4																

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

⇒ temps d'attente moyen : 10.4

**Priorité (préemptif)**

P4																			
P3																			
P2																			
P1																			
P0																			
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
	P0 P1	P2 P3	P4																

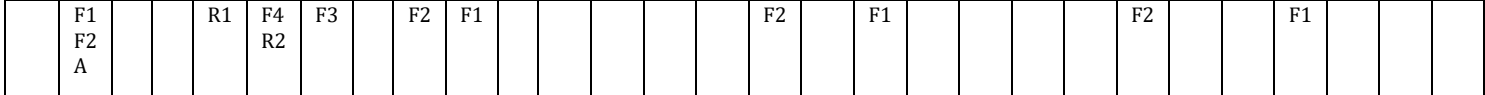
t0 : P0(9)-2, P1(1)-1  
 t1 : P3(0)-3, P2(2)-3, P0(9)-2, P1(1)-1  
 t2 : P2(1)-2, P0(9)-2, P1(1)-1, P4(5)-2  
 t3 : P2(0)-1, P0(9)-2, P1(1)-1, P4(5)-2  
 t4 : P4(4)-1, P0(9)-2, P1(1)-1,  
 t5 : P0(8)-1, P4(4)-1, P1(1)-1,  
 t6 : P0(8)-1, P4(4)-1  
 t7 : P0(8)-1, P4(3)-0  
 t8 : P0(7)-0, P4(3)-0  
 t9 : 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

4. On considère une implémentation de l'extension temps-réel de POSIX sur un système

**unité de temps.** L'ordonnancement est préemptif.

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.

Processus	Temps d'arrivée	Priorité*	Durée	Pemarque
A	0, 6, 12, 18, 24, 30	10	1	Périodique (quay lại tại time nhất định)
B	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)
C	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:

a) D(7)

☐

b) Les temps de réponse de chaque processus est :

- A : 0  
B : 1  
C : 1  
D : 5