

## Examen de systèmes d'exploitation 1

(Documentation non autorisée)

---

### Exercice1 : Questions de Cours

Q1) la stratégie d'ordonnement de processus la plus appropriée pour un système d'exploitation es temps partagé est :

- (a) Le Court-travail Premier (SJF).
- (b) Priorité.
- (c) Rond-Robin.
- (d) Premier arrivée Premier Servi (FCFS).
- (e) tout ce qui précède.

Q2) Un processus était observé de commuter depuis l'état actif vers l'état prêt. L'ordonnement (ou le scheduling) doit être :

- a) Le plus court job le premier (SJF)
- b) Préemptive
- c) Non-préemptive.
- d) Round Robin
- e) Aucune de ce qui précède

Q3) Les processus peuvent être dans un de trois états : Actif, Prêt, ou Bloqué. Dans quel état est le processus pour chacun des deux cas suivants ?

- (a) Attente des données d'être lues à partir d'un disque.
- (b) Avoir juste accompli une E/S et attendre d'être ordonnancé encore sur le processeur.

Q4) un programme d'éditeur de liens

- (a) place le programme dans la mémoire afin de l'exécution.
- (b) traduit un programme source en un programme objet.
- (c) lie le programme avec d'autres programmes nécessaires pour son exécution.
- (d) Est une interface d'un programme avec les entités produisant ses données d'entrée.

Q5) Vrai /faux. Entourer le choix adéquat. Une bonne réponse vaut 1 point, une mauvaise réponse vaut -0.5 points et pas de réponse vaut 0.

- (a) Un quantum court dans un ordonnancement Round Robin donne un meilleur temps de réponse pour les utilisateurs interactifs mais une utilisation peu efficace du processeur qu'un plus long quantum.
- (b) Un processus est une version exécutable d'un programme.

### Exercice 2:

Un système fait appel à l'algorithme d'ordonnancement avec priorité préemptif (les processus au numéro de priorité élevé ont une priorité plus importante). Les processus sont introduits dans le système avec une priorité de 0. Lors de l'attente dans la file des processus prêt, la priorité d'un processus change au rythme  $\alpha$ . Lors de l'exécution du processus, sa priorité est modifiée au rythme  $\beta$ .

- a) Quel algorithme résulte de  $\beta > \alpha > 0$
- b) Quel algorithme résulte de  $\beta < \alpha < 0$

### Exercice 3 :

Soient quatre processus dont les temps d'arrivés et d'exécutions estimé sont données dans la table suivante :

processus	Temps d'arrivés	Temps d'exécution
P1	0	8
P2	3	6
P3	5	2
P4	6	1

- 1) Donner le diagramme de GANTT illustrant l'ordonnancement des processus en utilisant la méthode du plus court temps restant d'abord « SRTF » (short remained time first)
- 2) Si les processus p1, p2, p3, et p4 sont arrivés au instant 2, 5, 7, 8 respectivement, donner le diagramme de GANTT illustrant l'ordonnancement des processus en utilisant la méthode du Round Robin (RR) avec un quantum =2.
- 3) Une commutation de mot d'état prend  $c$  unités de temps et la durée moyenne de la phase de calcul d'un processus est de  $p$  unités de temps. Calculer le rendement (temps passé pour l'exécution des processus sur temps total pendant lequel le CPU est occupé) du CPU en fonction de  $c$ ,  $p$  et la valeur  $q$  du quantum lorsqu'un Round Robin est utilisé. Étudier les cas particuliers où  $q$  tend vers l'infini  $\infty$  et vers 0.

## Corrigé :

### Exercice1 :

Q1) réponse : (c) Rond-Robin.

Q2) réponse : ( b) Préemptive

Q3) réponse :

(a) Attente des données d'être lues à partir d'un disque. : **bloqué**

(b) Avoir juste accompli une E/S et attendre d'être ordonnancé encore sur le processeur. : **prêt**

Q4) réponse : (c) lie le programme avec d'autres programmes nécessaires pour son exécution.

Q5)

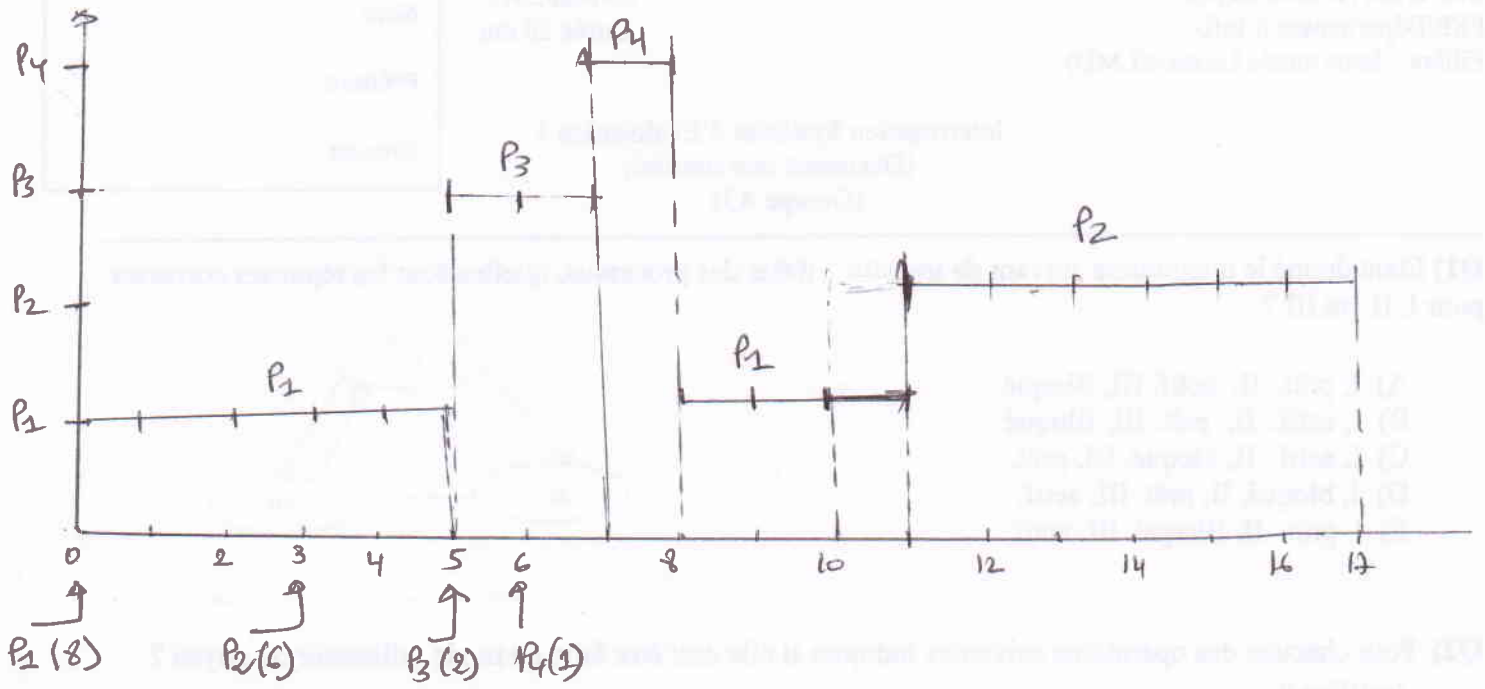
a) Un quantum court dans un ordonnancement Round Robin donne un meilleur temps de réponse pour les utilisateurs interactifs mais une utilisation peu efficace du processeur qu'un plus long quantum. (**Vrai**)

b) Un processus est une version exécutable d'un programme. (**Faux**)

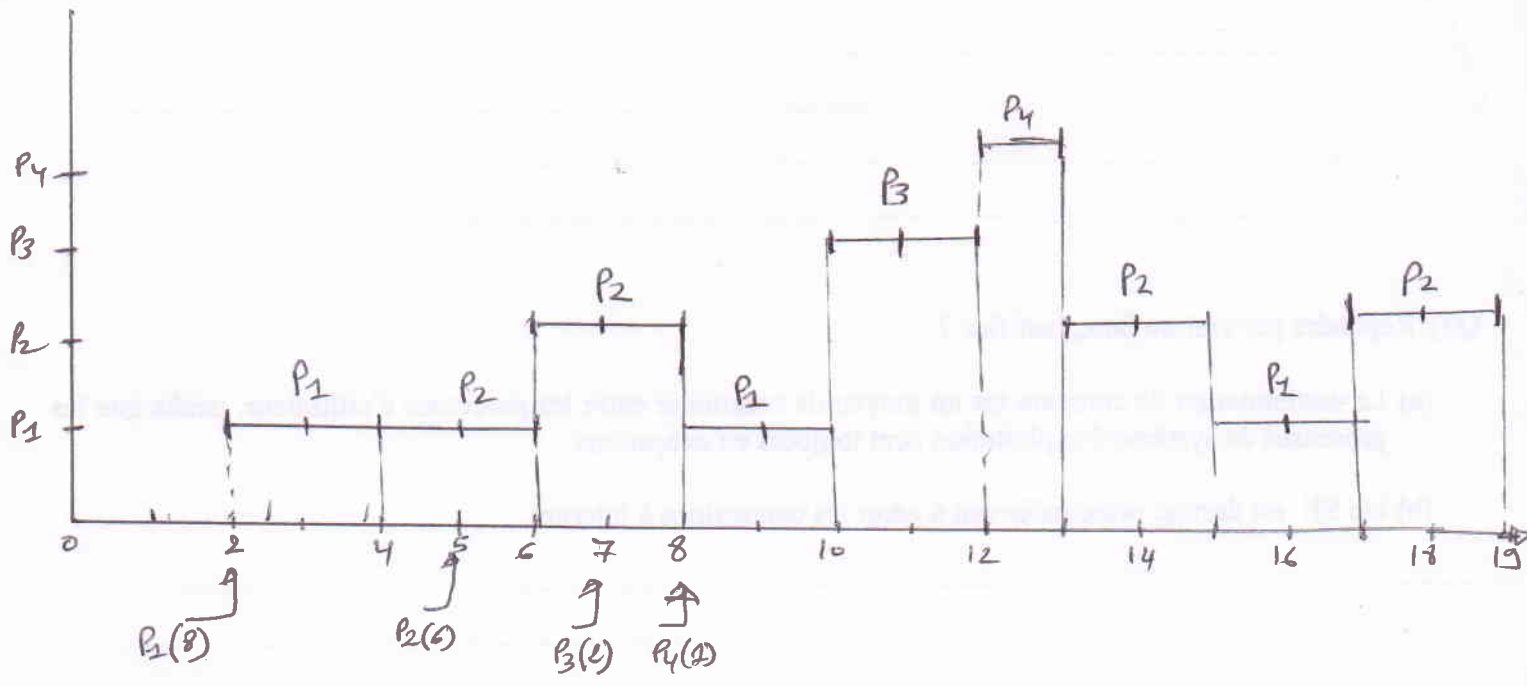
Exercice 2 : <pas disponible>

Exo3:

91/



92/



Suite de La question 2 :

Explication du diagramme ( Cette partie n'est pas demandée )

- A l'instant  $t=0$ , La file des processus prêts est vide

- à l'instant  $t=2$ , arrivé de  $P_1$  et  $P_1$  exécute pendant 2 unités CPU (1 quantum)

-  $t=4$ ,  $P_1$  exécute encore pendant 2 unités car aucun processus n'est arrivé.

-  $t=6$ ,  $P_2$  arrive dans La file

-  $t=6$ ,  $P_2$  passe dans La file et  $P_2$  exécute

-  $t=7$ ,  $P_3$  arrive dans La file

-  $t=8$ ,  $P_4$  arrive dans La file

•  $P_2$  termine son quantum et passe dans La file

•  $P_1$  entête de file est extrait pour être exécuté

-  $t=10$ ,  $P_1$  passe dans La file et  $P_3$  exécute

-  $t=12$ ,  $P_3$  termine son exécution et quitte et  $P_4$  exécute

-  $t=13$ ,  $P_4$  termine et quitte et  $P_2$  exécute

-  $t=15$ ,  $P_2$  passe dans La file,  $P_1$  exécute

-  $t=17$ ,  $P_1$  termine et  $P_2$  exécute  et la file est vide

-  $t=19$ ,  $P_2$  termine.

3) On note  $T_{\text{tot}}$  le temps total pendant lequel le CPU est occupé et  $R$  le rendement. D'après l'énoncé,

$$R = \text{Temps passé pour l'exécution des processus} / T_{\text{tot}} = p / T_{\text{tot}}$$

**On a** : le temps passé pour l'exécution des processus =  $p$

Il faut qu'on calcule le temps total  $T_{\text{tot}}$  en fonction de  $c$ ,  $p$  et  $q$ .

Soit une variable,  $n$ , qui correspond au nombre de commutations qui seront nécessaires pour exécuter un processus de durée  $p$ . En effet, il faudra ajouter la durée  $c$  d'une commutation à chaque fois qu'une "tranche" de durée  $q$  aura été consommée par le processeur. Alors le temps passé à faire des commutations est égale  $n*c = p/q*c$ .

Donc la durée  $T_{\text{tot}}$  se calcule en faisant la somme du temps  $p$  et du temps passé à faire des commutations, soit  $T_{\text{tot}} = p + p/q *c$ . par conséquent :  $R = p / (p + p/q *c)$

- lorsque  $q \rightarrow +\infty$ ,  $\lim_{q \rightarrow \infty} R = 1$ . Ce résultat est cohérent, puisqu'il indique que lorsque la durée du quantum est infinie, alors le rendement est maximal, du fait qu'il n'y a jamais besoin de commutations.
- Inversement, lorsque  $q \rightarrow 0$ ,  $\lim_{q \rightarrow 0} R = 0$ .