

Systèmes d'Exploitation**FICHE TD N°2**
Ordonnancement des processus**Exercice 0:**

On considère 4 programmes P1, P2, P3 et P4 dont le comportement est défini comme suit :

Programme	Comportement
P1	Calcul pendant 40 ms Lecture disque pendant 50 ms Calcul pendant 30 ms Lecture disque pendant 40 ms Calcul pendant 20 ms
P2	Calcul pendant 30 ms Lecture disque pendant 80 ms Calcul pendant 80 ms Lecture disque pendant 20 ms Calcul pendant 10 ms
P3	Calcul pendant 40 ms Lecture disque pendant 40 ms Calcul pendant 10 ms
P4	Calcul pendant 80 ms

Les 4 programmes sont lancés en même temps dans un système de multiprogrammation. On considère que l'ordonnancement sur le processeur se fait selon une stratégie FCFS.

1. Remplir le diagramme d'exécution de ces 4 programmes selon le modèle ci-joint (Pour vous aider on a coché les 4 premières cases du programme P1).
2. Calculer les temps suivants :
 - Temps d'exécution de chaque programme
 - Temps moyen d'exécution
 - Temps d'attente pour chaque programme
 - Temps moyen d'attente

Notations :

- **Actif** : utilise le processeur.

- **Prêt** : attente du processeur.
- **E/S** : phase d'entrée/sortie.
- **Attente** : attente d'un événement autre que la libération du processeur.

P1

[illegible]

P2

[illegible]

P3

[illegible]

P4

[illegible]

Note : Chaque case de ces diagrammes correspond à 10 ms

Exercice 1 :

Cinq travaux A, B, C, D et E arrivent pratiquement en même temps dans un centre de calcul. Leur temps d'exécution respectif est estimé à 10, 6, 2, 4 et 8 secondes.

Tracez le digramme de Gantt et déterminez le temps moyen de rotation pour chacun des algorithmes d'ordonnancement suivants. Ne tenez pas compte du temps perdu lors de la commutation des processus.

- Premier arrivé, premier servi FCFS (exécution dans l'ordre 10, 6, 2, 4, 8) ;
- Plus court d'abord SJF ;
- Tourniquet (quantum $q = 4$ s).

Exercice 2 :

On considère l'ensemble des processus suivants :

N° Processus	Date d'arrivée	Temps CPU	Priorité
1	7h 00	10 mn	2
2	7h 00	15 mn	3
3	7h 03	8 mn	4
4	7h 10	18 mn	5

A- On suppose qu'on utilise un algorithme d'ordonnancement basé sur la priorité (les priorités sont croissantes : 5 est le plus prioritaire). Donnez le diagramme de Gantt pour les priorités données dans le tableau.

B- On voudrait que la priorité des processus soit dynamique au cours du temps. Ainsi, pour calculer la priorité d'un processus, on utilise la formule suivante :

$$\text{Priorité} = \frac{\text{Temps d'attente} + \text{Temps CPU Restant}}{\text{Temps CPU}}$$

Remarque : Lors des calculs, on arrondira suivant l'exemple suivant : 3.5 ou 3.6 -> 4, 3.1 ou 3.4 -> 3.

1. Donnez le diagramme de Gantt sachant que la priorité est recalculée toutes les 5 minutes.
2. Calculez le temps d'attente moyen ainsi que le temps de rotation moyen.
3. Comparez les résultats obtenus par rapport à ceux obtenus avec l'algorithme de priorité classique.

Exercice 3 :

On considère une architecture monoprocesseur sur laquelle on désire exécuter un ensemble de 4 programmes dont le comportement est défini comme suit :

- Programme P1 : Date d'arrivée (0)
 Calcul pendant 6 unités de temps
 E/S pendant 3 unités de temps
 Calcul pendant 3 unités de temps
 E/S pendant 4 unités de temps
 Calcul pendant 2 unités de temps
- Programme P2 : Date d'arrivée (3)
 Calcul pendant 2 unités de temps
 E/S pendant 5 unités de temps
 Calcul pendant 2 unités de temps
 E/S pendant 2 unités de temps
 Calcul pendant 1 unité de temps
- Programme P3 : Date d'arrivée (5)
 Calcul pendant 2 unités de temps
 E/S pendant 4 unités de temps
 Calcul pendant 1 unité de temps
- Programme P4 : Date d'arrivée (8)

Calcul pendant 1 unité de temps

E/S pendant 1 unité de temps

Calcul pendant 1 unité de temps

On considère que l'on dispose d'un seul canal simple pour gérer un disque, et que l'ordre de services des requêtes pour ce disque se fait selon une politique FCFS.

On considère que l'ordonnancement sur le processeur se fait selon une stratégie SRT.

1. Remplissez le diagramme de Gantt ci-dessous.
2. Calculez le nombre de commutations de contexte.
3. Calculez les :
 - a. Temps d'attente du processeur pour chaque programme ;
 - b. Temps d'exécution total de chaque programme.

P1															
E/S															
Attente															
Prêt															
Actif															

P2															
E/S															
Attente															
Prêt															
Actif															

P3															
E/S															
Attente															
Prêt															
Actif															

P4															
E/S															
Attente															
Prêt															
Actif															

Exercice 4 :

On considère 4 processus, A, B, C, D. On suppose que l'exécution des processus nécessite :

- Pour A : 7 unités de temps CPU, 3 unités de temps d'E/S et 5 unités de temps CPU.
- Pour B : 6 unités de temps CPU, 4 unités de temps d'E/S, 4 unités de temps CPU.
- Pour C : 5 unités de temps CPU.
- Pour D : 1 unité de temps CPU, 4 unités de temps d'E/S et 2 unités de temps CPU.

On suppose que :

- A se présente en premier, à l'instant 0,
- B se présente à l'instant 1,
- C se présente à l'instant 9,
- D se présente à l'instant 12.

Montrez comment les 4 processus vont utiliser le processeur dans chacun des cas suivants :

- 1) Chaque processus a son propre périphérique d'E/S et l'ordonnanceur fonctionne selon Premier Arrivé Premier Servi FCFS (sans préemption).
- 2) Chaque processus a son propre périphérique d'E/S et l'ordonnanceur utilise l'algorithme du tourniquet, avec un quantum de 5. Le temps de commutation est égal à 0. Donnez, dans ce cas, les temps de rotation des processus A, B, C et D.
- 3) Les trois processus utilisent le même périphérique d'E/S dont la file d'attente est gérée par l'algorithme FCFS. L'ordonnanceur du processeur utilise l'algorithme du tourniquet, avec un quantum de 5. Le temps de commutation est supposé égal à 0.

Exercice 5 :

On considère trois (3) processus P1, P2, P3 dont les durées d'exécution sont respectivement 6, 4 et 8 unités de temps.

On fait l'hypothèse suivante : après 1 unité de temps d'exécution, le processus P2 crée un processus fils (qu'on appellera P4) dont la durée d'exécution est de 3 unités de temps. Le processus P4 après 2 unités de temps d'exécution crée à son tour un nouveau processus fils P5, dont la durée d'exécution est de 2 unités de temps. On admet qu'un processus ayant créé un fils doit se bloquer jusqu'à la terminaison de son processus fils.

En supposant que tous les processus sont gérés en utilisant l'ordonnancement «Round-Robin» avec un quantum égal à 2 unités de temps, dessinez le diagramme de Gantt .