

TD – Ordonnancement des processus

Exercice 1

Simuler l'exécution, avec un ordonnancement SJF des 5 travaux suivants :

Processus	Temps d'exécution	Temps d'arrivée
P1	2	0
P2	4	0
P3	1	3
P4	1	3
P5	1	3

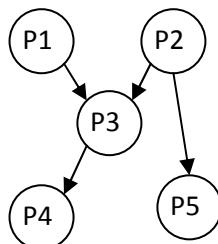
Calculer l'attente moyenne dans ce contexte, **puis** dans un contexte où tous les travaux sont soumis en même temps, **puis** dans le contexte où on considère que le temps de commutation n'est pas négligeable (=1 unité de temps), et conclure.

Exercice 2

Cinq travaux arrivent simultanément et demandent à être exécutés.

Processus	Durée estimée
P1	1
P2	3
P3	2
P4	2
P5	3

1. Représenter le diagramme de Gantt de ces processus en utilisant l'algorithme FCFS, puis SJF.
2. Calculer à chaque fois le temps de traitement moyen.
3. Dans ce qui suit, on suppose que le système dispose de deux processeurs et que les processus soient dépendants ;



- a. Représenter le diagramme de Gantt en utilisant l'algorithme d'ordonnancement FCFS puis SJF.
- b. Calculer le temps de traitement moyen et conclure.

Exercice 3

Soient 6 processus qui arrivent au système.

Processus	Temps d'exécution	Temps d'arrivée
P1	5	0
P2	3	1
P3	4	2
P4	5	3
P5	6	3
P6	2	5

Dressez le diagramme de Gantt en utilisant l'algorithme d'ordonnancement SRTF

EXERCICE 4

Soient 4 processus de priorités différentes qui arrivent au système.

☞ Une valeur de priorité élevée correspond à une priorité plus importante.

Représenter le diagramme de Gantt illustrant l'exécution des 4 processus en utilisant l'algorithme d'ordonnancement par priorité :

Processus	Temps d'exécution	Temps d'arrivée	Priorité
P1	4	0	3
P2	3	1	4
P3	4	2	6
P4	5	3	5

a. non préemptif,

b. préemptif.

Exercice 5

Simuler l'exécution de 2 processus P1 et P2 sur une machine dont le gestionnaire de processus possède les caractéristiques suivantes :

- les priorités des processus varient dynamiquement de 1 à 5, où 5 dénote la plus forte priorité. Quand un processus ne consomme pas entièrement son quantum, sa priorité est décrémentée (-1),
- les quantums varient dynamiquement de 1 à 5 unités de temps : quand un processus ne consomme pas entièrement son quantum, son quantum est incrémenté (+1),
⇒ les priorités varient de manière inversement proportionnelle aux quanta : quand la priorité d'un processus est décrémentée, son quantum est incrémenté.

Les processus ont les caractéristiques suivantes :

- ils sont soumis en même temps,
- les quanta initiaux sont de 2 pour P1 et de 5 pour P2,

- les priorités respectives initiales sont 2 et 3,
- chacun des deux processus requiert 16 unités de temps **de calcul**, E/S non inclus,
- P1 lance une opération d'E/S au 4^{ème} et 10^{ème} instant de son exécution. Chacune dure 1 unité de temps,
- P2 lance une opération d'E/S au 3^{ème} et 11^{ème} instant qui durent respectivement 7 et 3 unités de temps.

Remarque : L'ordonnanceur décrit n'est pas un "bon" ordonnanceur. Il vaudrait mieux, pour se rapprocher des critères d'un bon ordonnanceur, que le quantum d'un processus n'ayant pas consommé son dernier quantum soit décrémenté et non pas incrémenté.

Exercice 6

Un algorithme d'ordonnancement gère les priorités de la manière suivante :

- un processus qui entre dans la file d'attente des processus prêts, reçoit une valeur de priorité de base,
- toutes les secondes, la priorité est recalculée avec la formule :

$$\text{priorité} = (\text{temps de l'unité centrale utilisé}) + \text{priorité de base}$$
- toutes les secondes, un examen des priorités de tous les processus demandant l'unité centrale est effectué et le processus ayant la plus petite valeur de priorité est choisi. les processus ayant la même priorité sont ordonnancés selon leur date d'arrivée.

Construire l'assignation produite pour l'exemple suivant avec une priorité de base égale à 1.

Tâche	Temps d'exécution	Temps d'arrivée
T1	7	0
T2	4	$0+\epsilon$
T3	6	1
T4	1	$1+\epsilon$
T5	2	$1+2\epsilon$
T6	4	2
T7	1	$2+\epsilon$

Exercice 7

Un système utilise 3 files d'attente, la file n° 3 étant hiérarchiquement la plus élevée. Les processus ont un numéro de priorité fixé une fois pour toutes entre 1 et 3 et ils entrent directement dans la file d'attente correspondant à leur numéro. Chaque file est gérée par tourniquet avec une valeur du quantum égale à 1.

Ce tourniquet n'est activé que si les files de niveau supérieur sont toutes vides et que la file à laquelle il s'applique n'est pas elle-même vide.

1. Un processus peut-il être victime de phénomène de famine (jamais servis) ?
2. Donner l'assignation produite par l'exemple ci-dessous.

Processus	Temps d'exécution	Temps d'arrivée	Priorité
P1	7	0	2
P2	4	0	3
P3	6	1	1
P4	1	1	2
P5	2	$1+\epsilon$	3
P6	4	2	1
P7	1	2	2

Exercice 8

Soient A, B, C, D et E cinq processus de durée d'exécution respectives 9, 5, 2, 7 et X. Dans quel ordre doivent-ils être lancés pour minimiser le TTM ? La réponse dépendra de X

Exercice 9

Les lectures ou écritures d'un bloc disque durent un temps constant de 20 ms, mais une seule opération peut avoir lieu à un instant donné. Lors de la fin d'une entrée-sortie pour un processus, celui-ci est mis en bout de la file des processus prêts. On considère deux processus dont les actions sont les suivantes :

Processus P1

Calcul 10 ms
lecture B1
calcul 40 ms
écriture B1
calcul 10 ms
lecture B2
calcul 10 ms

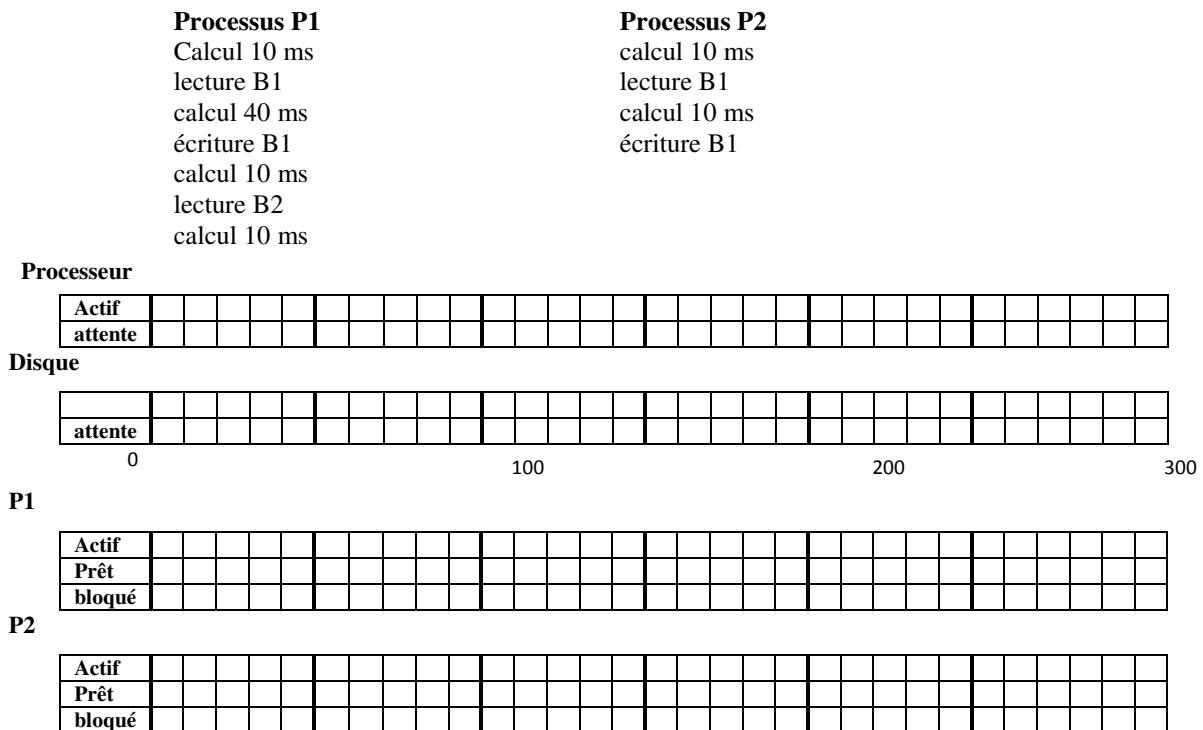
Processus P2

calcul 10 ms
lecture B1
calcul 10 ms
écriture B1

Le processus P1 est lancé au temps 0, et le processus P2 est lancé 10 ms après.

Donner le diagramme correspondant sur le dessin ci-dessous, dans le cas où :

1. **Processeur** géré par **FCFS** et le **Disque** géré par **FCFS**
2. **Processeur** géré par **SJF** et le **Disque** géré par **FCFS**
3. **Processeur** géré par **SRTF** et le **Disque** géré par **FCFS**



Exercice 10

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

On fait les hypothèses suivantes :

- * H1: 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.
- * H2 : 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.
- * H3 : 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 le scheduling «Round-Robin» avec un quantum égal à 2 unités de temps :

- Dessinez le diagramme de Gantt.
- En déduire les temps d'arrivée des processus P4 et P5 ainsi que les temps de réponse de chaque processus.