

SR02 : TD 9

TD Exercices sur une séance de 2h

Objectifs

- Analyser les techniques d'ordonnancement de processus

Exercice 1. (Ordonnancement de processus)

Soient les processus suivants :

Processus	Date d'arrivée	Temps d'exécution
A	0	3
B	1,001	6
C	4,001	4
D	6,001	2

Processus

- Elaborer le diagramme d'exécution des processus selon l'algorithme d'ordonnancement :
 1. FCFS (First Come First Served)
 2. SJF (Shortest Job First)
 3. SRT (Shortest Remaining Time) avec un quantum = 1
 4. Tourniquet (quantum=2)
 5. Tourniquet (quantum=1)
- Pour chaque cas calculer (arrondir au centième) le temps de traitement moyen

Exercice 2. (Ordonnancement avec plusieurs files)

Un ordonnanceur 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 un algorithme d'ordonnancement. Cet algorithme n'est activé que si les files des niveaux supérieurs sont toutes vides et que la file à laquelle il s'applique n'est pas elle-même vide.

On considère le système de tâches défini par le tableau suivant :

Processus	Temps Exec	Arrivée	Priorité
P1	7	0	2
P2	4	0	3
P3	6	1	1
P4	1	1	2
P5	2	1	3
P6	4	2	1
P7	1	2	2

Système de tâches

- Un processus peut-il être victime d'un phénomène de famine ?
- Quel est le temps de traitement moyen (à un centième près) si on utilisait l'algorithme d'ordonnancement "Plus Court d'abord : SJF (Shortest Job First)" au niveau de chacune des trois files ?
- Quel est le temps de traitement moyen (à un centième près) si on utilisait l'algorithme d'ordonnancement "Tourniquet : RR (Round Robin)" avec quantum=2, au niveau de chacune des trois files ?
- Quel est le temps de traitement moyen (à un centième près) si on utilisait l'algorithme d'ordonnancement "Temps Restant Plus court d'abord : SRT (Shortest Remaining Time)" avec quantum=1, au niveau de chacune des trois fils ?

Exercice 3. (Ordonnancement avec priorité)

Un algorithme d'ordonnancement avec priorité fonctionne de la façon suivante :

1. Chaque processus reçoit une priorité de base quand il rejoint la file d'attente ;
2. Toutes les secondes, la priorité est recalculée avec la formule :

$$\text{Priorité} = (\text{temps de CPU utilisé} / 2) + \text{priorité de base}$$

3. Toutes les secondes, le CPU est attribué au processus ayant le plus grand numéro de priorité (les processus ayant la même priorité sont départagés selon l'algorithme FCFS)
- Donner le schéma d'exécution correspondant au système suivant :

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

Processus