

Série 1 – Gestion du processeur central

Exercice 1

Soient 4 processus P1, P2, P3 et P4 qui effectuent du calcul et des entrées/sorties sur un disque selon les temps donnés ci-dessous:

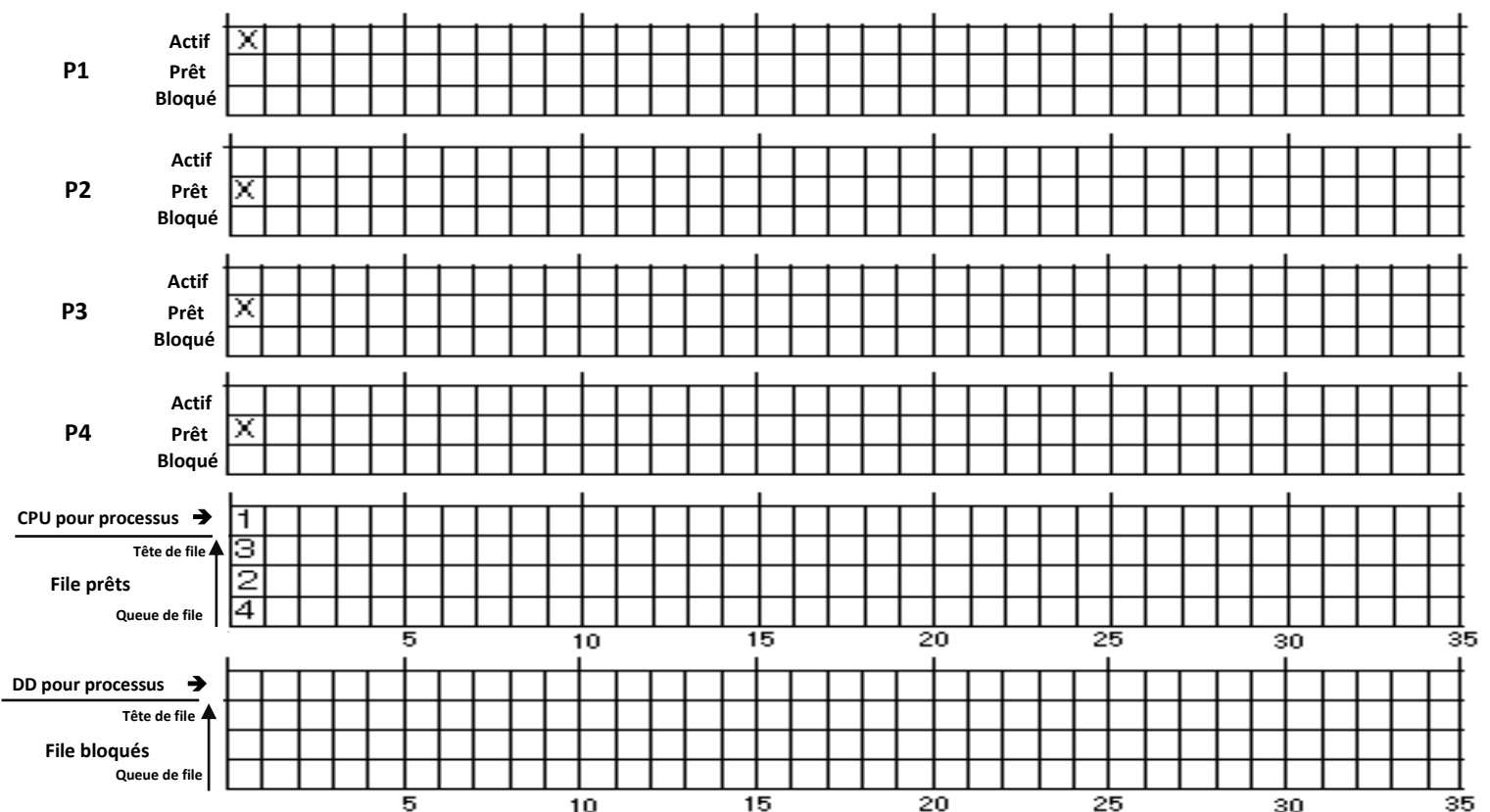
Processus P1	Processus P2	Processus P3	Processus P4
Calcul : 3 U	Calcul : 4 U	Calcul : 2 U	Calcul : 7 U
E/S : 7 U	E/S : 2 U	E/S : 3 U	
Calcul : 2 U	Calcul : 3 U	Calcul : 2 U	
E/S : 1 U	E/S : 1 U		
Calcul : 1 U	Calcul : 1 U		

On considère que l'ordonnancement sur le processeur se fait selon une politique à priorité préemptive : le processus élu à un instant t est celui qui est à l'état prêt avec la plus forte priorité.

On donne priorité (P1) > priorité (P3) > priorité (P2) > priorité (P4).

On considère que l'accès au disque pour l'exécution des requêtes d'E/S se fait toujours selon une politique FIFO.

1. Compléter le chronogramme d'états des 4 processus P1, P2, P3 et P4, ainsi que les chronogrammes donnant les états des files d'attente des processus à l'état prêts et bloqués.
2. Donner le temps de réponse moyen et d'attente obtenus.



Exercice 2

Soit un ordonnanceur de type Linux dans lequel les processus partagent un disque comme seule ressource autre que le processeur. Le disque n'est accessible qu'en accès exclusif et non préemptif et les demandes d'E/S sont gérées à l'ancienneté.

Dans ce système, on considère 4 processus P1, P2, P3, P4 pour lesquels :

- P1 et P2 sont des processus appartenant à la classe **SCHED_FIFO**. Dans cette classe, le processeur est donné au processus de plus haute priorité. Ce processus peut être préempté par un processus de la même classe ayant une priorité supérieure ;
- P3 et P4 sont des processus appartenant à la classe **SCHED_RR**. Dans cette classe, le processeur est donné au processus de plus haute priorité pour un quantum de temps égal à 10 ms. La politique appliquée est celle du tourniquet pour les processus de même priorité.

Les processus de la classe **SCHED_FIFO** sont toujours plus prioritaires que les processus de la classe **SCHED_RR**.
Les 4 processus ont les comportements suivants:

P1 (priorité 50)	P2 (priorité 49)	P3 (priorité 49)	P4 (priorité 49)
Calcul (40 ms)	Calcul (30 ms)	Calcul (40 ms)	Calcul (100 ms)
Lecture disque (50 ms)	Lecture disque (80 ms)	Lecture disque (40 ms)	
Calcul (30 ms)	Calcul (70 ms)	Calcul (10 ms)	
Lecture disque (40 ms)	Lecture disque (20 ms)		
Calcul (10 ms)	Calcul (10 ms)		

1. Représenter le chronogramme d'exécution des 4 processus.

Exercice 3

On considère un système dans lequel les seules ressources partagées sont un disque géré par un canal et un processeur. Les requêtes disques sont gérées à l'ancienneté (FIFO).

L'ordonnanceur est activé à chaque début et fin de traitement d'une E/S disque ou après un quantum Q si une E/S ne s'est pas produite dans le dernier quantum.

L'ordonnanceur alloue le processeur au processus P_i en attente qui a le plus fort rapport T_i/T_{cpu} .

- T_i représente la durée totale écoulée depuis le début de l'exécution du processus P_i (c'est-à-dire le temps écoulé depuis l'instant de première requête du processeur par le processus P_i).
- T_{cpu} est le cumul des durées pendant lesquelles le processeur a été alloué au processus i .

Lorsque le rapport T_i/T_{cpu} est égal à 0/0, celui-ci est interprété comme $+\infty$.

Lorsque plusieurs processus ont la même priorité, c'est le processus d'indice le plus fort qui obtient le processeur.

L'ordonnanceur utilise un délai de garde (temporisation) de durée q. Il est activé:

- soit lorsque le délai de garde est écoulé;
- soit lorsque le processus actif fait une requête d'E/S;
- soit lorsque l'exécution d'une telle requête (E/S) se termine.

Dans tous les cas, il commence par réarmer le délai de garde puis procède à l'allocation du processeur quand celle-ci est possible.

A l'instant initial trois processus sont présents dans le système et commencent à s'exécuter en faisant leur première requête à l'ordonnanceur.

Le tableau suivant donne (en nombre de quanta) la durée totale de CPU nécessaire à leur exécution, le nombre d'E/S disque réalisé par chaque processus et la date, mesurée en temps CPU écoulé depuis le début d'exécution, des requêtes d'E/S.

N° processus	Temps total CPU (quantum)	Nombre d'accès disque	Dates des accès (tps CPU relatif)
P1	5,5	1	1,5
P2	5	0	--
P3	1,5	2	0,5 puis 1

La durée d'une E/S sur disque est 2 quanta.

1. Compléter le chronogramme de la page 3 qui donne l'état des différents processus à chaque instant ainsi que la valeur des rapports T_i/T_{cpu} lors de chaque changement d'état.
2. Commenter le chronogramme en donnant les avantages et inconvénients de cette gestion du processeur.

Exercice 4

Les processus demandeurs du processeur central sont rangés dans N files F_1, F_2, \dots, F_N .

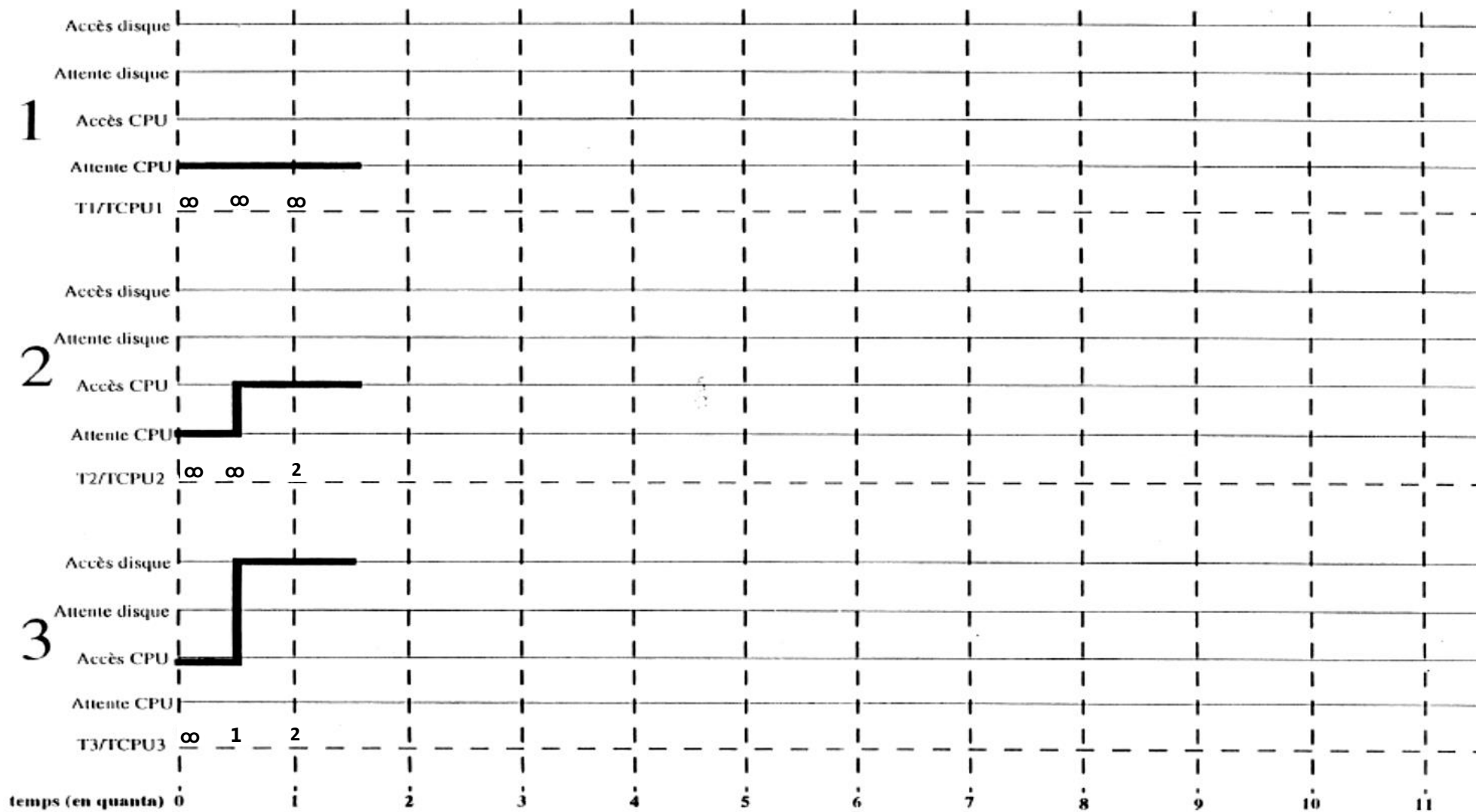
A chaque file est associé un quantum de temps Q_i et les processus arrivent par la file F_1 .

Un processus situé en tête de file F_i ($i > 1$) ne sera pris en compte que si toutes les files F_j ($0 < j < i$) sont vides.

Si un processus pris de la file F_i ne s'est pas terminé à la fin de son quantum, il sera placé dans la file F_{i+1} , sauf pour la file F_N où il y restera jusqu'à sa terminaison.

Réaliser cette politique dans le cas où :

- $N=4$
- $Q_{i+1}=Q_i*2$ et que $Q_1=20ms$
- Interruption horloge à chaque 20ms



Exercice 5

L'algorithme d'ordonnancement des processus du système UNIX est de la classe des ordonnanceurs du type RR avec plusieurs niveaux dépendants. L'ordonnanceur alloue le processeur à un processus pour un quantum de temps. Il préempte le processus qui a consommé son quantum de temps et il le met dans l'un des niveaux de priorités (file). La priorité d'un processus est fonction de l'utilisation récente (temps) du processeur central.

1. Quels sont les informations nécessaires pour implanter cet algorithme d'ordonnancement ?
2. Soient **Pri** la priorité d'un processus et **Cpu** l'utilisation récente (temps) du processeur central. Après un quantum de temps, l'ordonnanceur calcule pour chaque processus la nouvelle valeur **Cpu** en utilisant la formule $Cpu = Cpu/2$. Après cela, il recalcule **Pri** en utilisant la formule $Pri = (Cpu/2 + k)$; où **k** est une constante connue par le système.
Ecrire l'algorithme d'ordonnancement dans le cas où le système gère **m** niveaux de priorité et le processus élu est celui qui possède la plus basse priorité.
3. Soient trois processus **A**, **B** et **C** dans le système ayant une priorité initiale de **60**. La valeur de la constante **k** est 60, le quantum de temps est fixé à 1 seconde. L'horloge interrompt le processeur central 60 fois à la seconde. A chaque interruption, on incrémente de 1 la valeur de **Cpu**. Après chaque quantum, on recalcule les valeurs de **Cpu** et **Pri** selon le processus décrit dans l'algorithme d'ordonnancement.
Appliquer l'algorithme précédent en utilisant les valeurs données pour une période de 5 secondes.

Exercice 6

Le principe de l'ordonnanceur de partage équitable (UNIX version V) est de diviser la communauté des utilisateurs en un ensemble de groupes de partage équitable.

Cependant, le système alloue son temps d'unité centrale proportionnellement à chaque groupe, indifféremment du nombre de processus qu'il y a dans les groupes.

Soient **Pri** la priorité d'un processus et **Cpu** l'utilisation récente (temps) du processeur central et **Gpe** le groupe du processus.

L'horloge interrompt le processus en exécution 60 fois pendant son quantum de temps. La procédure de traitement de l'interruption horloge incrémente de 1 la valeur **Cpu** du processus en exécution et le champ **Gpe** de tous les processus du même groupe.

Après chaque quantum, on recalcule les champs **Pri** et **Gpe** de chaque processus selon les formules suivantes :

$$Cpu = Cpu / \text{nombre de groupes}$$

$$Gpe = Gpe / \text{nombre de groupes}$$

$$Pri = \text{priorité de base} + Cpu / \text{nombre de groupes} + Gpe / \text{nombre de groupes}$$

Considérons les trois processus A, B et C de priorité initiale 60.

Supposons que le processus A soit dans un groupe et que les processus B et C soient dans un autre groupe. La priorité de base est 60 (la plus haute priorité) et le quantum de temps est égal à 1 seconde.

1. Appliquer l'algorithme de partage équitable pour les processus A, B et C pour une période de 5 secondes.