

# Chapitre 3: gestion du processeur central

## 1 Concepts de processus

Un processus est une instance d'un programme en execution. Il est compose d'un code, de donnees et d'attributs.

### 1.1 État d'un processus

Quand un processus s'exécute il change d'état. Chaque processus peut se trouver dans l'un des états suivants:

- **Nouveau:** le processus est en cours de création;
- **Prêt:** le processus attend d'être affecté au processeur;
- **Exécution:** les instructions sont en cours d'exécution;
- **Attente:** le processus attend qu'un évènement se produise (exemple: fin d'opération d'entrée / sortie);
- **Terminé:** le processus termine son exécution.

## 2 Scheduling (Ordonnancement)

### 2.1 Définition d'un Scheduler (Ordonnanceur)

C'est un module du système d'exploitation qui sélectionne le prochain processus à exécuter.

### 2.2 Objectifs du scheduling

La politique du scheduling vise à optimiser les performances du système, à savoir:

- Équité;
- Utilisation des ressources: maximiser l'utilisation des ressources;
- Rendement: taux d'occupation de l'UC.

### 2.3 Critères de scheduling

- Utilisation de l'UC: est le taux en pourcentage de l'utilisation de l'UC;
- Capacité de traitement: c'est le nombre de processus par unité de temps;
- Temps d'attente: la somme des temps passé par le processus dans la file des processus prêts;
- Temps de réponse: la différence entre le temps début d'exécution et le temps fin d'exécution.

### 3 Les techniques de scheduling (Algorithme d'ordonnancement)

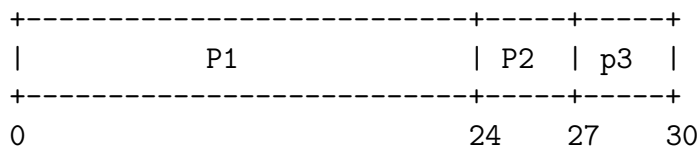
#### 3.1 FIFO (First In First Out)

Cette technique consiste à servir le premier processus arrivé dans la file d'attente des processus prêts.

**Exemple 1** Soit le tableau des processus prêts suivant:

Processus	Temps d'exécution (ut)
P1	24
P2	3
P3	3

Les processus sont arrivés au même moment à  $t = 0$  dans l'ordre  $P_1, P_2, P_3$ .  
Le diagramme d'exécution de ces trois processus selon la technique FIFO est comme suit:



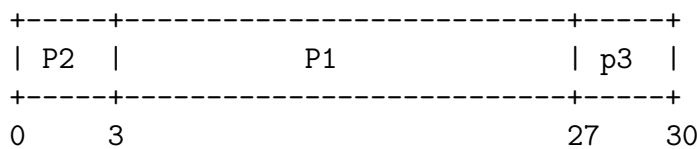
#### Temps d'attente de chaque processus

- *Le temps d'attente de  $P_1 = 0$  ut (unité de temps).* Le processus  $P_1$  est arrivé à l'instant  $t = 0$  et il a commencé son exécution à  $t = 0$ , donc le processus  $P_1$  n'a pas attendu le processeur. Son temps d'attente est égale à 0;
- *Le temps d'attente de  $P_2 = 24$  ut.* Le processus  $P_2$  est arrivé à l'instant  $t = 0$  et il a commencé son exécution à  $t = 24$ . Le temps d'attente du processus  $P_2 = 24$ ;
- *Le temps d'attente de  $P_3 = 27$  ut.* Le processus  $P_3$  est arrivé à l'instant  $t = 0$  et il a commencé son exécution à  $t = 27$ . Le temps d'attente du processus  $P_3 = 27$ ;

**Temps d'attente moyen** est égale à la moyenne des temps d'attente.  
 temps d'attente moyen =  $(0 + 24 + 27)/3 = 17$

**Exemple 2** On reprend le premier exemple mais l'ordre d'arrivé cette fois est:  $P_2, P_1, P_3$

#### Diagramme d'exécution



#### Temps d'attente de chaque processus

- $P_1 = 3$  ;
- $P_2 = 0$  ;
- $P_3 = 27$  .

**Temps d'attente moyen** =  $(3 + 0 + 27)/3 = 10$

**Remarque** La politique FIFO désavantage les processus courts s'ils ne sont pas exécutés en premier.

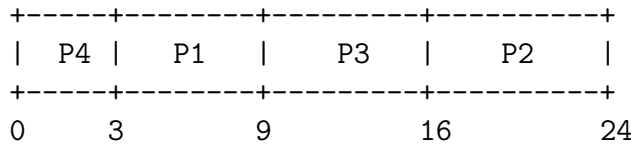
### 3.2 SJF (Short test Job First)

Le plus court d'abord. Consiste à exécuter le processus qui a le plus court temps d'exécution en premier.

**Exemple**

Processus	Temps d'exécution (ut)
P1	6
P2	8
P3	7
P4	3

**Diagramme d'exécution** On commence par exécuter le processus qui le temps d'exécution le plus court, c'est-à-dire  $P_4$  ensuite  $P_1, P_3, P_2$ .



**Temps d'attente de chaque processus**

- $P_1 = 3$  ;
- $P_2 = 16$  ;
- $P_3 = 9$  ;
- $P_4 = 0$  .

**Temps d'attente moyen** =  $(3 + 16 + 9 + 0)/4 = 7$

**Remarque** La difficulté avec l'algorithme SJF est dans l'estimation avec exactitude du temps d'exécution des processus

### 3.3 SRTF (Short test Remained Time First)

Le plus court temps restant d'abord. Quand un processus  $p$  s'exécute et il lui reste  $t$  unités de temps pour terminer son exécution, ensuite un processus  $q$  dont le temps d'exécution est inférieur à  $t$ , dans ce cas là, on interrompt l'exécution du processus  $p$  et on donne la main au processus  $q$ .

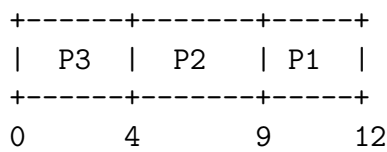
### 3.4 Scheduling avec priorité

**Exemple**

Processus	Temps d'exécution (ut)	Priorité
P1	3	3
P2	5	2
P3	4	1

- Valeur de priorité plus basse  $\Rightarrow$  processus plus prioritaire;
- Arrivée à  $t = 0$  dans l'ordre suivant  $P_1, P_2, P_3$ .

**Diagramme d'exécution**



**Temps d'attente de chaque processus**

- $P_1 = 9$  ;
- $P_2 = 4$  ;
- $P_3 = 0$  .

**Temps d'attente moyen** =  $(9 + 4 + 0)/3 = 4.33$

**Remarque** La politique avec priorité peut être avec réquisition (préemption).

### 3.5 Politique Tourniquet (Round Robin)

Consiste à définir une tranche de temps appelée quantum et allouer le processeur aux processeur pendant un quntum de temps.

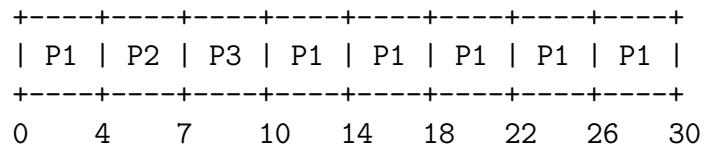
**Remarque** Il faut bien choisir la valeur du quantum avec soin. Une valeur du quantum trop petite causera une fréquence de commutation de contexte importante, et une valeur de quantum trop grande désavantage les processus les plus courts.

**Exemple**

Processus	Temps d'exécution (ut)
P1	24
P2	3
P3	3

Le quantum  $q = 4$  ut

### Diagramme d'exécution



### Temps d'attente de chaque processus

- $P_1 = 6$  ;
- $P_2 = 4$  ;
- $P_3 = 7$  .

**Temps d'attente moyen** =  $(6 + 4 + 7)/3 = 5.66$