

## CHAPITRE V: GESTION DE PROCESSUS ET DE PROCESSEUR

### I/ Introduction :

Dans l'exécution des différents processus, il faut disposer de plusieurs ressources machine contribuant au long du cycle de vie d'un processus, la plus importante c'est la ressource qui effectue l'exécution elle-même, c'est le processeur. Alors, pour cette ressource, le système d'exploitation doit doter d'un ensemble de mécanismes gérant d'une manière efficace et optimale le travail du processeur et contrôlant l'exécution des processus.

### II/ Le processus :

#### 1/ Définition :

Un processus est un programme en cours d'exécution, il se trouve au niveau de la mémoire centrale. Le processus est créé le moment d'exécution et tué après sa fin.

#### 2/ ce qui se passe après le lancement d'un programme :

Quand l'utilisateur lance l'exécution d'un programme, et quand la machine utilise le principe de la mémoire virtuelle et le multitâche, alors ce programme passe par les étapes suivantes :

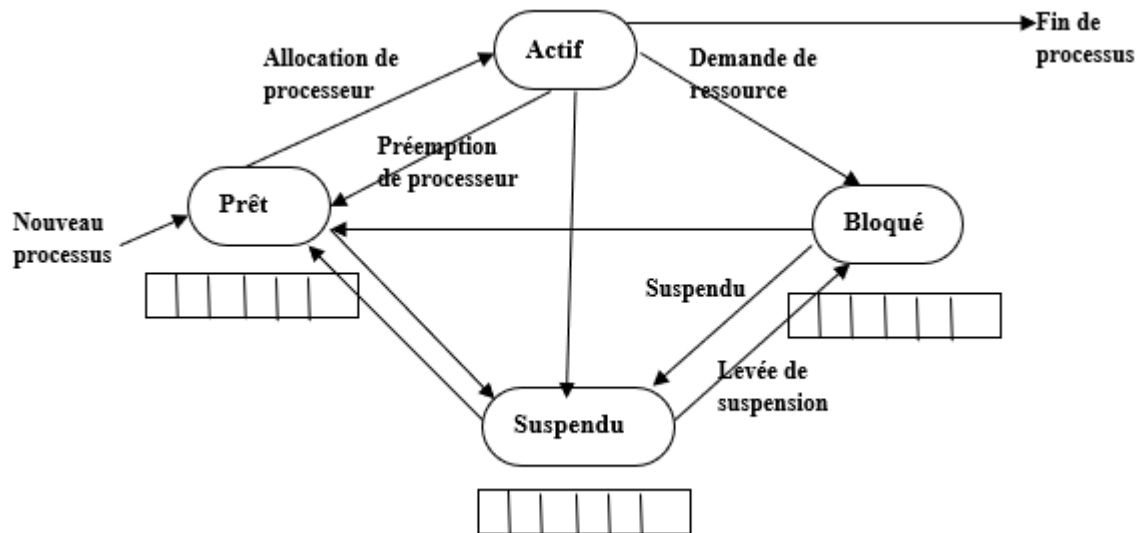
- Le système crée un **Job** au niveau de la mémoire virtuelle tout en chargeant le programme.
- Après, selon un critère ou autre ce job est admis dans le système tout en le chargeant vers la mémoire centrale dans sa totalité ou une partie, ce chargement crée ce qu'on appelle un **processus**.
- Le processus est dirigé, après, vers l'exécution dans le processeur selon des conditions spécifiques, où ce processus peut changer son état, durant cette exécution, plusieurs fois jusqu'à sa terminaison.

#### 3/ Le changement d'état d'un processus :

Durant son exécution le processus peut être :

- Actif : (Elu) c'est-à-dire, le processus dispose de toutes les ressources nécessaires et il est en train de s'exécuter au niveau du processeur.
- Prêt : (Eligible) c'est-à-dire, le processus dispose de toutes les ressources nécessaires sauf le processeur → processus en attente du processeur.
- Bloqué : le processus est en attente d'une autre ressource que le processeur ou d'un évènement.
- Suspendu : le processus est arrêté temporairement par l'utilisateur ou le SE pour ne plus être en compétition pour le processeur.

Pour chacun des états prêt, bloqué et suspendu, on associe une file d'attente pour les processus dans cet état et elle est gérée par un outil du SE.



**4/ Le descripteur d'un processus :**

A la création d'un processus (idem pour le job), son image est créée dans la mémoire contenant toutes les informations nécessaires pour ce processus au long de son exécution, cette image représente aussi le processus au niveau des différentes files d'attente des différents états. Cette image est nommée descripteur ou bloc de contrôle de processus (job) PCB (JCB) qui contient :

<b>Identité :</b>	
Nom externe	nom interne
<b>Priorité (dynamique, statique)</b>	
<b>Contexte :</b> CO, PSW, RG, Acc, ...	
<b>Etat</b>	
<b>Limites mémoires :</b> Table des pages, les zones occupées, les adresses, ...	
<b>Ressource :</b> Types, quantité	
<b>Informations de comptabilité</b>	

Alors, les files d'attente utilisées dans chaque état contiennent les PCBs des processus, tel que :

- Le PCB d'un nouveau processus est mis dans la file d'attente des processus prêts.
- Attente jusqu'à sélection pour entrer dans le processeur → le processus devient actif, et on utilise un pointeur vers son PCB dans la mémoire.
- Si le processus attend une ressource ou un évènement, son PCB entre dans la file d'attente des processus bloqués.
- S'il est suspendu, son PCB se mis dans la file d'attente des processus suspendu.

**5/ Les primitives de manipulation de processus :**

La manipulation d'un processus se fait par des primitives nécessaires qui utilisent son PCB et se sont des procédures systèmes. Ces primitives s'exécutent d'une manière indivisible :

- La création d'un processus.
- Activation d'un processus prêt.
- Suspension d'un processus.
- Destruction d'un processus.

### III/ Allocation de processeur à un processus :

Chaque fois, le processeur devient inactif, le système doit sélectionner un processus de la file d'attente des processus prêt et il le passe le contrôle, deux routines système sont concernées :

#### 1/ Le dispatcher :

Il prend en charge l'allocation du processeur à un processus sélectionné au préalable, dont il assure :

- La commutation de contexte.
- Branchement vers la première instruction du processus pour l'exécution.

#### 2/ Le scheduler :

Il s'occupe de la sélection du processus qui aura le prochain cycle d'exécution à partir de la file d'attente des processus prêts, c'est l'ordonnanceur des processus.

L'opération de scheduling est faite dans les situations suivantes :

- Lorsqu'un nouveau processus est créé, il faut décider s'il faut exécuter d'abord le processus parent ou le processus fils.
- Il faut prendre une décision d'ordonnancement lorsqu'un processus se termine → un autre processus doit être choisi dans le jeu de processus prêts.
- Lorsqu'un processus bloque sur des E/S, un sémaphore ou autre (à voir en 3LMD), un autre processus doit être sélectionné pour être exécuté.
- Lorsqu'une interruption d'E/S se produit, il faut également prendre une décision d'ordonnancement.
- Si l'horloge matérielle fournit des interruptions périodiques à une fréquence de 50 ou 60 Hz, par exemple, une décision d'ordonnancement peut être prise à chaque interruption d'horloge (système à temps partagé).

Il existe deux niveaux de scheduling :

- Scheduling des jobs pour décider quels processus à créer.
- Scheduling des processus pour décider quel processus prêt à exécuter.

#### ➤ Le scheduling de job :

Il permet de :

- Suivre l'état du job.
- Allouer les ressources nécessaires.
- Récupérer les ressources après exécution.

➤ Le scheduling de processus :

Il permet de :

- Suivre l'état du processus en créant le PCB.
- Décide la politique d'allocation du processeur.
- Allouer le processeur.
- Reprendre le processeur du processus terminé ou interrompu.

### 3/ Les différentes politiques de scheduling :

➤ Le scheduling non préemptif :

Un algorithme d'ordonnancement non préemptif sélectionne un processus, puis le laisse s'exécuter jusqu'à ce qu'il bloque (soit sur une E/S, soit en attente d'un autre processus) ou qu'il libère volontairement le processeur. Même s'il s'exécute pendant des heures, il ne sera pas suspendu de force. En effet, aucune décision d'ordonnancement n'intervient pendant les interruptions d'horloge. Une fois le traitement de l'interruption est terminé, le processus qui était en cours d'exécution avant l'interruption est toujours relancé. On peut citer comme politiques :

- Politique FCFS : premier arrivé premier servi.
- Politique SJF : Short Job First → le job qui a le temps d'exécution le plus petit c'est le premier qui s'exécute.
- Politique par priorité : le processus avec la priorité la plus élevée soit le premier.

➤ Le scheduling préemptif :

Un algorithme de scheduling préemptif sélectionne un processus et le laisse s'exécuter pendant un délai déterminé. Si le processus est toujours en cours à l'issue de ce délai, il est suspendu (mis dans la file d'attente des processus prêts), et l'ordonnanceur sélectionne un autre processus à exécuter. L'ordonnancement préemptif nécessite une interruption à la fin du délai afin de redonner le contrôle de l'UC à l'ordonnanceur. On peut citer :

- Politique SRTF : Short Remaining Time First → le processus avec le temps restant le plus court d'abord.
- Politique tourniquet (Round Robin) : cette politique consiste à allouer le processeur suivant une durée limitée nommée **le quantum de temps**, où chaque processus entre pendant cette durée de temps et il sort et attend un autre tour d'exécution. Cette politique est réalisée à l'aide d'une file d'attente circulaire des processus prêts avec une gestion FIFO.

Politique à plusieurs niveaux de queue : la file d'attente des processus prêts est subdivisée en plusieurs files suivant la classe des processus (Batch, interactifs, temps réel, ...), chacune est gérée indépendamment.