

Numérique et Sciences Informatiques
Chapitre IX - Gestion des processus et des ressources
par un système d'exploitation

I. Définition d'un processus

Définition

Un **processus** (process en anglais) un programme en cours d'exécution : il est composé d'une suite d'instructions à exécuter.

II. Ordonnancement de plusieurs processus

Exécuter l'ensemble des instructions d'un processus prend du temps. Plusieurs types d'exécutions sont possibles :

- l'exécution séquentielle : les processus s'exécutent les uns après les autres. Dans la pratique, il faudrait attendre longtemps avant de pouvoir exécuter un programme. De plus certains processus ne s'arrêtent jamais.
- l'exécution parallèle : les processus s'exécutent en même temps. dans la pratique, il faudrait autant de processeurs que de processus pour que cela soit possible.
- l'exécution entrelacée : des morceaux de processus différents sont exécutés en alternance.

Deux processus



Exécution séquentielle



Exécutions parallèles (deux processeurs)



Exécutions entrelacées



Autre entrelacement



Il faut donc que le système d'exploitation désigne quel processus s'exécute et pendant combien de temps. C'est l'**ordonnanceur** qui choisit l'ordre d'exécution des processus.

Chaque processus peut donc se trouver dans plusieurs **états** possibles selon s'il s'exécute ou non.

III. Etats d'un processus

Un processus est identifié par un numéro unique : le **PID** (Process IDentification).

Un processus est toujours créé par un processus. Donc chaque processus possède un unique **processus père**. Donc à chaque processus, on peut attribuer un numéro de processus père qui sera unique le **PPID** (Parent Process IDentification).

Seul le processus de PID=0 n'a pas de père. Il s'exécute à l'allumage de l'ordinateur.

Ce premier processus exécutera le processus `init`, qui, quant à lui, créera d'autres processus. Donc `init` a un PID=1 et un PPID=0.

Voici les différents états d'un processus :

- **Initialisation** (ou *Création*) : état dans lequel se trouve le processus à sa création. Il attend de l'ordonnateur de se placer à l'état *Prêt*.
- **Prêt** (ou *En attente*) : dans cet état, le processus attend de l'ordonnateur de pouvoir s'exécuter. Les processus dans cet état sont dans une File et attendent que l'ordonnateur leur permette de passer à l'état *Elu*.
- **Elu** (ou *Exécution*) : le processus est en cours d'exécution par le processeur.
- **Bloqué** (ou *Endormi*) : le processus a été interrompu ou attend un événement, comme des données par exemple.
- **Terminé** (ou *Terminaison*) : Le processus est terminé. Le résultat est connu ou le programme a été forcé de s'arrêter.

Un processus qui consomme trop de temps sur le processeur pourra être **préempté** et donc repasser à l'état *Prêt*. L'opération qui fait passer un processus de l'état *Prêt* à l'état *Elu* s'appelle l'**élection**.

L'opération qui fait passer un processus de l'état *Elu* à l'état *Prêt* s'appelle la **préemption**.

Quand un processus est dans l'état *Prêt* ou *Elu*, on dit qu'il est **actif**.

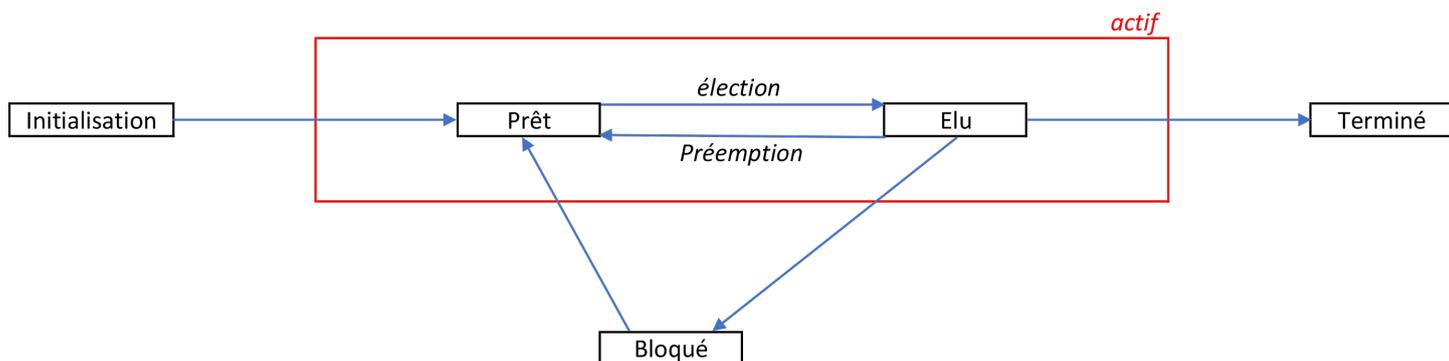


FIGURE 1 – Schéma récapitulatif des états d'un processus

IV. Interblocage (deadlock)

Un **interblocage** (ou deadlock) est un phénomène qui se produit lorsque des processus s'attendent mutuellement. Certains processus nécessitent un ensemble de ressources pour s'exécuter. L'utilisation d'une ressource passe par trois étapes :

- la demande de la ressource : si la demande n'est pas satisfaite, il faut attendre (état *Bloqué*).
- l'utilisation de la ressource : le processus peut utiliser la ressource (état *Elu*).
- la libération de la ressource : le processus libère la ressource demandée (état *Elu*).

Imaginons maintenant deux processus $P1$ et $P2$ ayant besoin de deux ressources $R1$ et $R2$ pour s'exécuter.

- $P1$ demande dans un premier temps la ressource $R1$, puis ensuite la ressource $R2$.
- $P2$ demande dans un premier temps la ressource $R2$, puis ensuite la ressource $R1$.

Que se passe-t-il ?

Processus	Etat	Commentaire
$P1$	Prêt	-
$P2$	Prêt	-
$P1$	Elu	demande $R1$
$P1$	Bloqué	en attente de $R1$
$P2$	Elu	demande $R2$
$P2$	Bloqué	en attente de $R2$
$P1$	Bloqué	$R1$ lui est alloué
$P2$	Bloqué	$R2$ lui est alloué
$P1$	Elu	demande $R2$
$P1$	Bloqué	en attente de $R2$
$P2$	Elu	demande $R1$
$P2$	Bloqué	en attente de $R1$

A la fin, les processus $P1$ et $P2$ sont bloqué et en attente respectivement des ressources $R2$ et $R1$ utilisées respectivement par $P2$ et $P1$. Il y a interblocage.