

Processus

1. Principe

De manière informelle, un processus peut être considéré comme un programme en cours d'exécution. Celui-ci progresse de manière séquentielle.

Un processus est aussi caractérisé par :

- Un compteur ordinal et les registres processeur donnant son activité courante.
- Une pile contenant les valeurs temporaires.
- Un segment de données contenant des variables globales.
- Une table sur les descripteurs des fichiers ouverts.

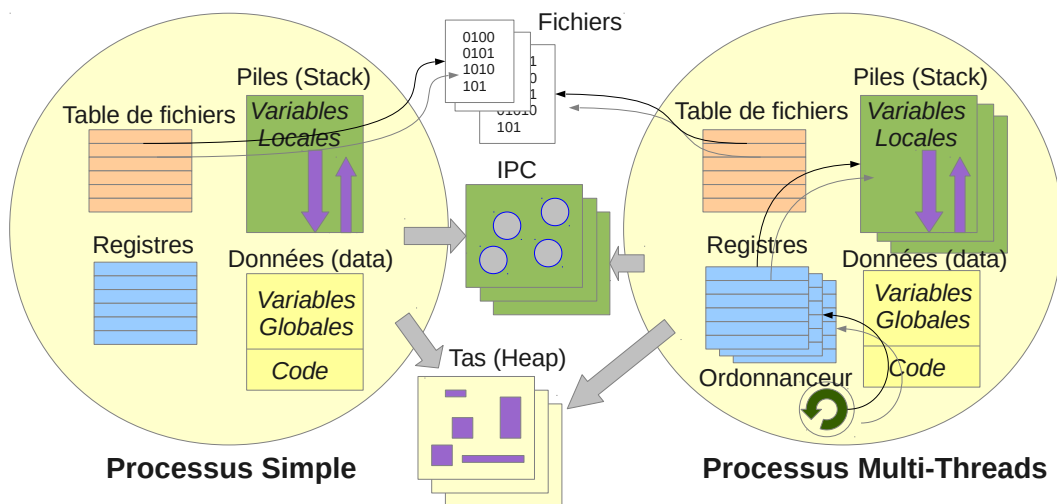


Illustration 1: Principe interne d'un processus

2. État d'un processus

À tout instant donné il n'y a qu'un seul processus en exécution par processeur. Ce processus est appelé le processus courant. Le système maintient une liste de processus prêts qu'il pourrait exécuter et procède périodiquement à un ordonnancement.

Comme le montre le schéma suivant un processus est actif ou inactif :

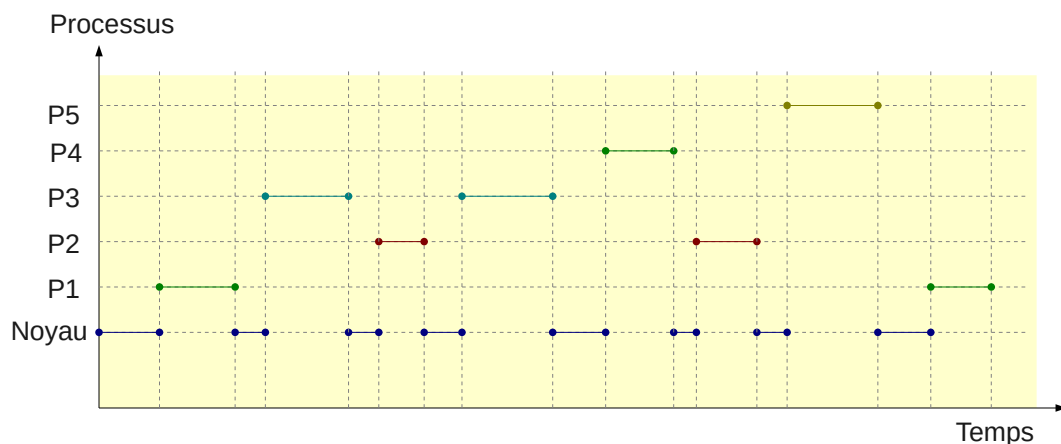


Illustration 2: Chronogramme activation de processus

À chaque processus est attribué un quantum de temps. Le quantum de temps est très court et l'utilisateur a l'illusion que les processus s'exécutent en parallèle. C'est ce que l'on appelle du pseudo-parallélisme. Au cours de son exécution, un processus change d'état.

L'état d'un processus est défini par son activité courante. Les différents états d'un processus sont les

suivant :

- En Exécution ; Le processus est exécuté par le processeur.
- Prêt ; Le processus est en attente d'exécution.
- Suspendu ; Le processus est en attente d'une ressource.
- Stoppé ; Le processus est suspendu par une intervention extérieure.
- Zombie ; Le processus est terminé, mais il est toujours référencé par le système.

Les changements d'état d'un processus :

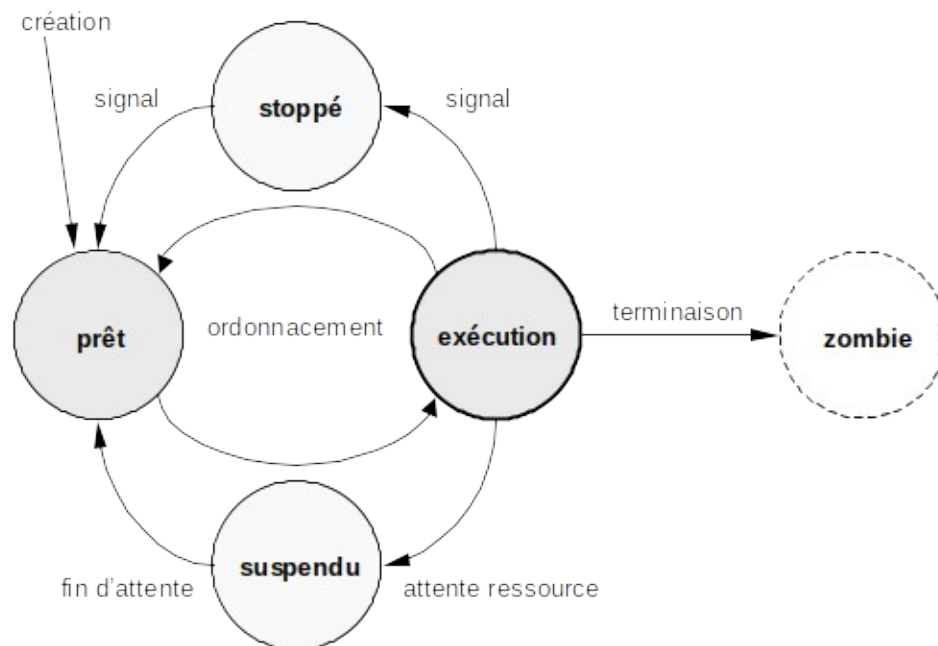


Illustration 3: Graphe d'état d'un processus

Attributs d'un processus

Durant son exécution, un processus est caractérisé par plusieurs attributs maintenus par le système :

- son état,
- son identificateur (**pid**)
- les valeurs des registres et du compteur ordinal,
- l'identité de l'utilisateur sous le nom duquel il s'exécute,
- les informations utilisées par le noyau pour l'ordonnancement (priorité...),
- les informations sur l'espace d'adressage (segment de code, de données, de pile),
- les informations sur les entrées/sorties effectuées par le processus (descripteurs de fichiers ouverts, répertoire courant...),
- des informations de comptabilité résumant les ressources utilisées par le processus.

3. Identificateurs d'un processus

Plusieurs identificateurs d'utilisateurs et de groupes sont associés à un processus :

- l'identificateur d'utilisateur réel ; celui qui exécute le processus,
- l'identificateur d'utilisateur effectif ; l'utilisateur utilisé pour les contrôles d'accès par bit **setuid**,
- l'identificateur de groupe réel ; le groupe primaire de celui qui exécute l'application,
- l'identificateur de groupe effectif ; le groupe utilisé pour les contrôles d'accès par bit **setgid**,
- une liste d'identificateur de groupe ; tout utilisateur peut appartenir à plusieurs groupes.

4. Filiation

La création d'un processus est effectuée en dupliquant le processus courant par l'appel système `fork`. Un processus père crée un processus fils identique à lui-même, à l'exception de son `pid`.

Il existe un processus initial `init` avec un `pid = 1`, l'ancêtre de tous les processus.

Groupe de processus

Un groupe de processus est un ensemble contenant un ou plusieurs processus. Tout processus fait partie d'un groupe et ses descendants appartiennent par défaut au même groupe. Un processus peut choisir de créer un nouveau groupe, et ainsi devenir chef (leader) du groupe. Un groupe est identifié par un numéro de groupe égale au numéro du processus leader.

Cette notion de groupe permet d'envoyer des signaux à tous les processus membre du groupe. Les interpréteurs de commande (`bash`, `csh`, `ksh`...) utilisent ces groupes pour permettre à l'utilisateur de suspendre l'exécution de processus et de la reprendre, en avant plan ou en arrière plan.

5. Session

Une session est un ensemble contenant un ou plusieurs groupe de processus. À chaque session est associé un terminal de contrôle unique.

Une session est identifiée par le numéro de son processus leader. Généralement, une nouvelle session est créée par le programme `login`, lorsqu'un utilisateur se connecte.

6. Quelques commandes liées aux Processus

La commande `ps` permet de voir l'état des processus.

- voir tous les processus et informations

```
$ ps aux
```

- filtrer les informations

```
$ ps aux | egrep 'PID|root'
```

- choisir le format d'affichage

```
$ ps axo "%p %y %x %c %C"
```

La commande `uptime` donne l'information du taux de charge à 1, 5 et 15 minutes.

```
$ uptime
```

La commande `ulimit` permet de gérer les ressources système.

```
$ ulimit -a
```

La commande `vmstat`, pour avoir un aperçu rapide de la consommation cpu.

```
$ vmstat 5 4
```

Les commandes `nice` et `renice` permettent de donner une priorité forte (-20) à faible (19) à un processus.

```
$ nice 3 ./prog
# renice -15 1234
```

Envoyer un signal à un processus

```
$ kill 1234
$ kill -9 1234
$ kill -1 1234
$ killall prog
```

Trouver les processus occupant une ressource.

```
$ fuser -mu /mnt/essai
```

Voir les fichiers ouverts

```
$ lsof
```

Voir les appels systèmes fait par un processus

```
$ strace prog
```