

# 17.1

## Les processus

NSI TERMINALE - JB DUTHOIT

### 17.1.1 Introduction

Afin de bien fonctionner, de nombreuses tâches ou application doivent être exécuté en même temps , par le système d'exploitation et les utilisateurs.

Pour permettre cela, et gérer les problèmes qui en découlent, le système d'exploitation va générer des processus, et les gère ensuite.

### 17.1.2 Définition

#### Définition

| Un **processus** est un programme en cours d'exécution sur un ordinateur.

⚠ La notion de programmes et de processus est différente. Le même programme exécuté plusieurs fois génère plusieurs processus.

#### Création d'un processus

La création d'un processus peut intervenir

- au démarrage du système
- par un appel d'un autre processus
- par une action d'un utilisateur (lancement d'application par exemple)

### 17.1.3 Visualisation des processus

#### Sur Windows

- Avec l'interface windows, il suffit d'appuyer simultanément sur les touches ctrl, alt et suppr.
- En ligne de commande : Cliquer sur "Windows"+"R", puis entre cmd puis Entrée.
  - tasklist liste les processus sur l' ordinateur
  - taskkill permet d'arrêter un processus.

#### Sur Linux

Il est possible de les visualiser grâce à la commande `ps -a -u -x`.

La colonne USER indique le nom de l'utilisateur qui a lancé le processus.

La colonne PID donne l'identifiant numérique du processus.

Les colonne % CPU et % MEM donnent respectivement le taux d'occupation du processeur et de la mémoire.

Un caractère "?" indique que le processus n'a pas été lancé depuis un terminale.

La colonne STAT indique l'état du processus :

- R pour running (prêt ou en exécution)
  - S pour sleeping (en attente)
- ☛ La commande `top` est très pratique, avec un affichage qui s'actualise toutes les secondes.

### 17.1.4 Terminer un processus

#### Sur Windows

Il suffit de cliquer sur fin de tâches, ou bien d'utiliser `taskkill` en ligne de commande.

### 17.1.5 Sur Linux

Pour tuer un processus, on lui envoie un signal de terminaison. On en utilise principalement deux :

- SIGTERM (15) : demande la terminaison d'un processus. Cela permet au processus de se terminer proprement en libérant les ressources allouées.
- SIGKILL (9) : demande la terminaison immédiate et inconditionnelle d'un processus. C'est une terminaison violente à n'appliquer que sur les processus récalcitrants qui ne répondent pas au signal SIGTERM.

#### Exercice 17.167

Lancer une application et visualiser sur votre ordinateur les différents processus. Retrouver le processus correspondant à l'application que vous avez lancée. Puis supprimer-le en utilisant les lignes de commandes.

Processus Performance Historique des applications Démarrage Utilisateurs Détails Services						
Nom	Statut	5%	87%	0%	0%	2%
		Processe...	Mémoire	Disque	Réseau	Processe...
Applications (12)						
> Adobe Acrobat Reader DC (32 ...		0,1%	5,1 Mo	0 Mo/s	0 Mb/s	0%
> Explorateur Windows		0%	26,0 Mo	0 Mo/s	0 Mb/s	0%
> Gestionnaire des tâches		0,4%	28,4 Mo	0 Mo/s	0 Mb/s	0%
> GNU Image Manipulation Progr...		0%	41,7 Mo	0 Mo/s	0 Mb/s	0%
> Google Chrome (24)		0,1%	545,9 Mo	0,1 Mo/s	0 Mb/s	0%

```
gimp-2.10.exe 5660 Console 1 110 844 Ko
```

```
C:\Users\DUTHOIT>taskkill /IM gimp-2.10.exe
Opération réussie : un signal de fin a été envoyé au processus "gimp-2.10.exe" de PID 5660.
```

### Remarque

Vous remarquerez qu'une grande partie de la mémoire, dans mon exemple précédent, est utilisée par le navigateur Google Chrome et ses 24 onglets ouverts !

### 17.1.6 PID et PPID

Un processus est caractérisé par un identifiant unique : son PID (Process Identifier). Lorsqu'un processus engendre un fils, l'OS génère un nouveau numéro de processus pour le fils. Le fils connaît aussi le numéro de son père : le PPID (Parent Process Identifier).

⚠ Comme une ressource (le processeur ou un périphérique) ne peut pas être partagée, c'est son temps d'utilisation qui va l'être : le temps d'utilisation d'une ressource est partagé en intervalles très courts, pendant lesquels l'ordonnanceur l'alloue à un seul utilisateur.