

Fonctionnement d'un ordinateur

Notion de processus

Systèmes d'exploitation

Le **système d'exploitation (OS)** constitue l'interface entre la machine et l'utilisateur. Il est constitué d'un ensemble de programmes pour gérer les ressources et les partager :

- la gestion des processus,
- la gestion des fichiers,
- la gestion de la mémoire,
- la gestion des périphériques,
- le traitement des entrées-sorties,
- la sécurité.

Une machine peut posséder plusieurs système d'exploitation. L'utilisateur peut choisir l'OS au démarrage (système **LINUX** et **Windows** par exemple). Le système d'exploitation peut se trouver sur le disque dur ou sur un autre support (clé USB par exemple).

Processus

Un **processus** est un programme en cours d'exécution. Un processus utilise des ressources physiques (processeur, mémoire) et logiques (données). Contrairement à un programme, qui est statique, un processus est dynamique (il a un début et une fin).

Un processus peut créer d'autres processus.

Des identificateurs sont associés à chaque processus :

- Le **PID** (Process Identification) est le numéro du processus,
- Le **PPID** est le numéro du processus père,
- l'**UID** (User Identification) est l'identifiant de l'utilisateur qui a démarré le processus,
- le **GID** (Group Identification) est l'identifiant du groupe de l'utilisateur qui a démarré le processus.

On peut distinguer trois types de processus :

- les applications,
- les processus qui fonctionnent en permanence en arrière plan (services sous Windows, daemons sous Linux), et s'exécutent dès le démarrage de la machine, comme les antivirus,
- les processus du système d'exploitation.

Dans la plupart des cas les processus sont créés par d'autres processus (processus père/fils).

Sous Windows :

Le **gestionnaire des tâches** permet d'observer en détail les processus en cours et de les interrompre.

Sous Linux :

- La commande **top** permet de lister tous les processus en cours d'exécution.
- La commande **htop** propose d'exécuter différentes actions sur les processus affichés.
- La commande **ps** liste les processus.
- La commande **ps aux** liste les processus par ordre de PID.
- La commande **ps aux | grep pi** liste les processus qui contiennent pi dans leur nom.
- La commande **ps tree -p** affiche les processus sous forme d'arbre (pères et les fils).
- La commande **ps -aef** affiche en plus le PPID
- La commande **kill 1522** permet d'arrêter le processus dont le PID est 1522

Démarrage d'une machine

Le démarrage d'une machine commence par l'exécution du **BIOS** (Basic Input/Output System) par le **CPU** (Central Processing Unit). Le BIOS est un petit programme stocké dans une puce **ROM** (Read Only Memory). Au démarrage de l'ordinateur, le BIOS vient lire des informations dans une mémoire **CMOS** (Complementary Metal Oxide Semiconducteur). Les informations contenues dans la CMOS est maintenu par un faible courant électrique fourni par une pile.

Ensuite il y a le chargement du **MBR** (Master Boot Record), premier secteur du disque où se trouve le **boot** (fichier de démarrage).

Pour Windows NT le fichier de boot est nommé **NTLDR** (NT Loader).

Quelques processus lancés sur Windows NT au démarrage :

- system avec le PID 4,
- registry avec le PID 96 (ntoskrnl.exe),
- applications Services et Contrôleur (services.exe),
- applications d'ouverture de session (winlogon.exe),
- application de démarrage de windows (wininit.exe),
- gestionnaire des fenêtre du bureau (dwm.exe)
- démarrage du bureau (explorer.exe)
- ...

Un gestionnaire des tâches permet d'observer les différents processus (taskmgr.exe)

Certains processus sont critiques (ils ne peuvent pas être interrompus).

Les processus lancés sous Linux sont :

- init (ancêtre de tous les processus),
- le daemon kthread (ancêtre de tous les threads).

Un daemon sous Linus correspond à un service sous Windows. C'est un processus qui s'exécute en arrière plan.

Etats d'un processus

Un processus peut être :

- Bloqué (en attente d'un événement)
- Prêt
- Actif

Au moment de sa création, un processus est prêt.

Le système dispose d'une table des processus (avec des files d'attente pour les processus prêts et les processus bloqués).

C'est le système d'exploitation qui décide quel processus est actif et désactive un processus pour en réactiver un autre.

Python pour gérer des processus

Gestion processus avec le module os de Python

```
import os
```

lancer notepad :

```
os.system("C:/windows/notepad.exe")
```

ouverture dans le programme par défaut du fichier test.txt qui se trouve dans le répertoire courant :

```
os.system("test.txt")
```

ou bien :

```
os.startfile("test.txt")
```

ouverture dans le programme par défaut du fichier test.csv qui se trouve dans le répertoire courant :

```
os.startfile("test.csv")
```

ouverture du fichier test2.txt qui se trouve dans le sous-répertoire nsi du répertoire courant :

```
os.startfile("nsi\\test2.txt")
```

lancer gimp :

```
os.startfile("C:/Program Files/GIMP 2/bin/gimp-2.10.exe")
```

ouverture dans le programme par défaut d'une image test.jpg qui se trouve dans le répertoire nsi :

```
os.startfile("nsi\\test.jpeg")
```

affiche le répertoire courant

```
print(os.getcwd())
```

affiche l'os

```
print(os.name)
```

liste le contenu du répertoire courant

```
print(os.listdir())
```

créer un répertoire dans le répertoire courant

```
os.mkdir("nsi2")
```

créer un répertoire nsi3 dans le répertoire nsi2

```
os.mkdir("nsi2\\nsi3")
```

changer de répertoire courant

```
os.chdir("nsi2")
```

supprimer le fichier test.jpeg dans le répertoire nsi

```
os.remove("nsi/test.jpeg")
```

supprimer le répertoire nsi3

```
os.rmdir("nsi2/nsi3")
```

obtenir le PID et le PPID du processus python.exe

```
print(os.getpid())
```

```
print(os.getppid())
```

obtenir le nom d'utilisateur du processus python.exe

```
print(os.getlogin())
```

ouvrir le gestionnaire de tâches

```
os.system("taskmgr")
```

Gestion processus avec le module subprocess de Python

```
import subprocess
```

ouvrir le fichier test.txt du répertoire courant dans notepad

```
subprocess.call("C:/windows/notepad.exe test.txt")
```

ou (à partir de la version 3.5 de Python)

```
subprocess.run("C:/windows/notepad.exe test.txt")
```

ouverture du fichier test.jpeg du répertoire nsi dans gimp

```
subprocess.call("C:/Program Files/GIMP 2/bin/gimp-2.10.exe nsi/test.jpeg")
```

Le gestionnaire de tâches sous Windows

Sous Windows, faire un clic droit sur la barre des tâches et sélectionner « gestionnaire des tâches ».

Dans l'onglet détail les processus sont affichés dans l'ordre alphabétique. On peut voir les PID.

Dans l'onglet Fichier on peut lancer un processus (**firefox.exe** par exemple). Avec un clic droit, on peut arrêter un processus.

L'invite de commande sous Windows

L'invite de commande permet de contourner l'interface graphique Windows (taper `cmd` dans la zone de recherche, sélectionner « Exécuter en tant qu'administrateur »).

```
C:\windows\system32>tasklist
```

Entrer la commande `tasklist`. Tous les processus sont listés.

Entrer la commande `tasklist | findstr "firefox"`. Tous les processus firefox sont listés.

```
C:\windows\system32>tasklist | findstr "firefox"
firefox.exe           14480 Console           1    289 816 Ko
firefox.exe           5628 Console           1     50 772 Ko
firefox.exe           5140 Console           1    116 068 Ko
firefox.exe          12236 Console           1     49 080 Ko
firefox.exe          10936 Console           1     73 816 Ko
```

Pour arrêter firefox, il faut tuer tous les processus.

Avec le PID :

```
C:\WINDOWS\system32>taskkill /PID 14480 /T
```

L'option `/T` sert à tuer tous les processus enfants.

Avec le nom :

```
C:\WINDOWS\system32>taskkill /f /im firefox.exe
```

L'option `/f` sert à forcer l'arrêt et l'option `/im` tuer tous les processus enfants.

Les processus sous Linux

On ouvre un terminal. La commande `top` permet de lister tous les processus en cours d'exécution.

La commande `htop` permet d'effectuer des actions supplémentaires.

Dans un terminal, la commande `ps` liste les processus associés au terminal.

La commande `ps aux | grep fire` liste les processus qui contiennent fire dans leur nom

`pstree -p` affiche l'arborescence des processus.

`ps -aef` affiche le PPID en plus.

Pour arrêter un processus il faut connaître son PID : `kill PID`

Thread et multithreading

Un processus est une suite d'instructions traitées par le processeur. Ces instructions sont regroupées en séquences appelées **thread**. Le processeur n'exécute qu'un seul thread à la fois.

Le **multithreading** permet de traiter simultanément deux threads.

Un processeur ne peut pas exécuter plus d'instructions simultanées qu'il n'a de cœurs. La simultanéité n'est qu'apparente. En réalité, le processeur exécute successivement et alternativement des thread de chaque processus.

L'**ordonnanceur** (ou **scheduler**) est le module du système d'exploitation qui se charge de l'organisation de l'exécution des processus. On appelle **quantum de temps** le plus petit temps d'exécution que l'OS peut attribuer à un thread. L'ordre de grandeur d'un quantum de temps est 10 ms. Il existe de nombreux algorithmes d'ordonnement :

- first-come, first-served (FCFS)
- shortest job first (SJF)
- shortest remaining time (SRT)
- round robin (RR, algorithme du tourniquet)
- ordonnancement avec priorité

Gestion des ressources

Plusieurs types de ressources sont à la disposition d'un processus : RAM, disques, clés USB, fichiers, périphériques.

L'utilisation d'une ressource se fait en trois temps :

- demande d'accès,
- utilisation de la ressource,
- libération de la ressource.

Notion d'interblocage

L'interblocage (**deadlock** en anglais) se produit lorsque des processus concurrents s'attendent mutuellement. Un processus peut aussi s'attendre lui-même. Les processus bloqués dans cet état le sont définitivement, il s'agit donc d'une situation d'impasse à éviter. La solution consiste à arrêter au moins un des processus concernés.