

# Exercice : Processus

---

## Exercice 1

- 1) Ouvrir le gestionnaire de tâches. Vous pouvez directement saisir la commande **taskmgr** dans la barre de tâches windows. Assurez-vous d'avoir
  - tous les détails d'affichage
  - La colonne PID affichée, sinon clic droit sur les entêtes de colonne puis cocher PID.
- 2) Quelles sont les ressources matérielles évaluées par le gestionnaire de tâches (onglet performance) ?
- 3) Quelle est la fréquence affichée du processeur ?
- 4) Quel est le nombre de coeurs du processeur ?
- 5) Quelle est la capacité de la RAM ? Utilisée ? Disponible ?
- 6) Quelle est la taille du disque dur ?
- 7) Donner trois caractéristiques du processeur graphique ?

## Exercice 2

Le bloc-notes peut-il bloquer votre ordinateur ?

- 1) Ouvrir le bloc notes et vérifier la présence du processus associé dans le gestionnaire de tâches puis relever :
  - a) la quantité de mémoire qui lui est allouée ?
  - b) le PID de ce processus et son statut.
- 2) Saisir la phrase "Le bloc notes utilise peu de ressources !" dans le bloc notes.
  - a) La quantité de mémoire allouée a-t-elle changé ?
  - b) Enregistrer votre fichier et contrôler les paramètres du processus. Que remarquez-vous ?
- 3) Copier votre phrase puis collez-la ? Sélectionnez vos 2 phrases puis collez-les ? Répéter ce copier-coller-enregistrer plusieurs fois.

La quantité de mémoire et de processeur utilisé varie-t-elle ?
- 4) Le bloc notes peut-il planter l'ordinateur ? Poursuivez vos copier-coller et observer l'évolution du processus ?

## Exercice 3

- 1) Lancer le navigateur firefox et vérifier la présence du processus dans le gestionnaire de tâches.
- 2) Combien de processus sont en cours d'exécution ?
- 3) Est-il possible de mettre fin à des processus sans tous les arrêter ?
- 4) Un des processus est le parent des autres. Lequel ? Vérifier que tous les autres processus s'arrêtent lorsqu'on met fin au processus parent.
- 5) Ajouter la colonne "type" et retrouver le processus parent ?

## Exercice 4

- 1) Trier les processus selon leur PID.
  - a) Quel est le processus qui a la plus petite valeur ?
  - b) Que signifie la valeur attribuée au processeur ?
- 2) Lancer le bloc-notes et noter son PID.
- 3) Dans une console windows, saisir la commande **tasklist**.
  - a) Quel est l'affichage obtenu ?
  - b) Saisir la commande suivi d'un slash et du point d'interrogation `/?` . Que renvoie la commande ?

- c) Quelle commande faut-il saisir pour afficher le processus associé au bloc-notes ?
  - d) Lancer firefox puis afficher dans la console la liste des processus associés.
  - e) Vérifier que le premier de la liste est le parent des autres processus (relancer plusieurs fois firefox).
  - f) Pour arrêter un processus, on utilise la commande **taskkill** en indiquant le PID du processus.
- Arrêter le bloc-notes et firefox en ligne de commandes.

## Exercice 5

Cet exercice se réalise sur une machine avec Linux installé.

- 1) Ouvrir la console et saisir la commande **ps -a -u -x**. Que renvoie-t-elle ?
- 2) Lancer le navigateur web puis vérifier les processus.
- 3) Pour affiner l'affichage, il est possible d'utiliser la commande **grep** en complément de la première. On appelle ça un tube ou pipe.  
Voici la syntaxe : **ps -a -u -x | grep "texte à chercher"**  
Afficher uniquement les processus du navigateur web.
- 4) Il existe une commande qui affiche l'arborescence des processus montrant ainsi la filiation. Cette commande est **pstree**.
  - a) Cette commande est-elle installée sur votre système ? Sinon l'installer.
  - b) Afficher l'arborescence des processus.
- 5) La commande **top** permet de contrôler les processus en temps réel d'exécution.
  - a) Afficher les processus actifs.
  - b) Lancer une application et contrôler sa présence sur la console.
- 6) La commande qui arrête un processus est **kill** avec le PID du processus. Arrêter le navigateur web avec cette commande.

## Exercice 6

Cet exercice pourra être réalisé sur Windows et sur Linux.

Python dispose d'un module qui permet d'exécuter des commandes au niveau du système d'exploitation. Ce module est **os**.

La méthode **system** de ce module permet d'exécuter une commande.

Par exemple **os.system("C :/windows/notepad.exe")** ouvre le bloc notes. On récupère le prompt de l'interpréteur python seulement en fermant l'application ouverte et l'interpréteur affiche alors la valeur 0.

La méthode **startfile** permet l'ouverture d'un fichier.

Elle prend en argument le chemin complet du fichier à ouvrir.

La méthode **getpid** renvoie le pid du processus python en cours.

La méthode **getppid** renvoie le pid du parent du processus python en cours.

D'autres méthodes utiles sont données ci-après (attention aux arguments) :

- **name** renvoie le nom de l'OS.
- **getcwd** renvoie le répertoire courant.
- **listdir** renvoie le contenu du répertoire courant.
- **makedirs** crée un répertoire.
- **chdir** change de répertoire courant
- **remove** supprime un fichier.
- **rmdir** supprime un répertoire.
- **name** renvoie le nom de l'OS.

Ouvrir l'interpréteur python et importer le module **os**. Toutes les actions demandées ci-après se font **exclusivement** en python.

- 1) Afficher le nom du système d'exploitation.
- 2) Quel est le répertoire courant ?
- 3) Lister le contenu du répertoire courant.
  - a) Combien y a-t-il de dossiers et fichiers dans le répertoire courant ?
  - b) Un fichier et un dossier sont cachés si le nom commence par un point. Pouvez-vous trouver combien il y en a ?
- 4) Créer un répertoire **titi** dans le répertoire courant puis placez-vous dans le répertoire **titi**.
- 5) Afficher le PID du processus python en cours d'utilisation. Celui du parent aussi.
- 6) Ouvrir le gestionnaire de tâches. Vérifier les PID de python et de son parent.
- 7) Arrêter le processus python en cours d'exécution avec une commande python.

## Exercice 7

Il existe une autre manière de procéder avec le module **subprocess**.

- 1) Dans un environnement windows, saisir les commandes suivantes et relever ce qui se passe :

```
import subprocess

subprocess.run('notepad.exe C:/users/ychistel/titi/test.txt')

import subprocess

subprocess.call('notepad.exe C:/users/ychistel/titi/test.txt')

import subprocess

p=subprocess.Popen('notepad.exe C:/users/yannick/titi/test.txt')
```

- 2) Quelle méthode semble la plus intéressante ? Pourquoi ?
- 3) On utilise la troisième méthode et notre objet p qui est un processus en cours d'exécution.
  - a) Quelle méthode renvoie le pid du processus en cours ?
  - b) Quelle méthode permet d'arrêter le processus en cours ?
- 4) Écrire un script python :
  - a) Qui permet d'exécuter un navigateur comme Firefox ou Chrome en affichant la page d'accueil du site du lycée.
  - b) Affiche le PID du processus du navigateur
  - c) Termine le processus.
- 5) Tester ce module dans un environnement linux.

## Exercice 8

On considère trois processus  $P_1$ ,  $P_2$  et  $P_3$  et trois ressources  $R_1$ ,  $R_2$  et  $R_3$  qui s'exécutent les uns après les autres selon :

$P_1$  : demande  $R_1$ , demande  $R_2$ , libère  $R_1$ , libère  $R_2$ .

$P_2$  : demande  $R_2$ , demande  $R_3$ , libère  $R_2$ , libère  $R_3$ .

$P_3$  : demande  $R_3$ , demande  $R_1$ , libère  $R_3$ , libère  $R_1$ .

- 1) Y a-t-il interblocage ?
- 2) Décrire une exécution de trois processus qui conduit à une situation d'interblocage.
- 3) Représenter cette situation par un schéma.