

# TP 6 Linux : Processus et Redirection

## EXERCICE 1

---

1. Pour afficher seulement la liste des processus en cours, on utilise l'option U lors de l'exécution de top et on rentre l'identifiant de l'utilisateur.

### Etat d'un processus

La commande top permet de voir en temps réel le déroulement des processus. On voit bien les champs CPU, MEM, ... changent en permanence, selon l'avancement de l'exécution et l'ordonnancement exécuté par le SE.

Pour suspendre la commande en avant-plan : ctrl +z. On vérifie avec jobs. Pour la remettre à l'avant-plan on fait fg (foreground) et bg à l'arrière-plan (background). Lorsqu'on applique la commande jobs, le système nous retourne tous les processus qui tournent avec un numéro relatif. fg 1 permet de mettre en avant plan le processus qui a pour numéro de job 1. On peut très bien faire fg nom du processus comme fg firefox par exemple.

### Priorités

on relance la commande en arrière-plan avec (commande)&

Pour trouver l'aide on-line, on lance la commande top et ensuite pendant l'exécution de ce programme on tape ? on obtient l'aide. Pour restreindre l'affichage de nos processus seuls, il suffit de taper u lors de l'exécution du top suivi de votre login.

2. Vous avez lancé une commande qui monopolise la console sans rien afficher car le traitement est très long. Comment la passer en arrière-plan ?
3. Listez les processus appartenant à root.
4. Comment afficher une arborescence de tous les processus ?
5. Comment savoir très simplement si le processus firefox tourne ?
6. Isolez les numéros des processus correspondant aux divers shells bash lancés.
7. Exécutez une commande sleep 100&. Terminez ensuite proprement ce processus.
8. Que se passe-t-il si en tant que root vous "tuez" violemment le processus 1 ?
9. Un processus récalcitrant refuse de s'arrêter. Quelle est la seule action possible ?
10. Comment être certain qu'un processus important ne s'arrêtera pas lorsque vous fermerez votre console ou terminal ?

## EXERCICE 2

---

1. Quel est le résultat de la commande `>fic` sans rien devant ?
2. Que va afficher `ls -R / > liste` et que contiendra `liste` si vous exécutez cette commande avec un utilisateur ordinaire ?
3. En tant qu'utilisateur, pour lister l'intégralité des fichiers du système et placer tous les résultats quels qu'ils soient dans `liste`, quelle doit être la bonne redirection ?
  - (a) `2>dev/null >liste`
  - (b) `>liste 2>liste`
  - (c) `1>liste 2>&1`
  - (d) `2>1 >liste`
4. Comment via `echo` faire passer un message par la sortie d'erreur ?
5. Comment faire afficher le contenu d'un fichier avec `cat` et une redirection en entrée ?
6. Comment supprimer tous les messages d'erreur ?
  - (a) `2>/dev/null`
  - (b) `2>/dev/zero`
  - (c) `>/dev/null 2>&1`
  - (d) `2>fic ; rm fic`

## EXERCICE 3

---

Supposant que vous avez un programme appelé `testRedirect` qui affiche "ici Marrakech" en `stdout` et "Non ce n'est pas ici" en `stderr`.

Pour chacun des exemples suivants, précisez quel est l'output sur l'écran, et quel est l'output dans les fichiers référencés dans la redirection

1. `./testRedirect > file.out`
2. `./testRedirect 2> file.out 1> file.err`
3. `./testRedirect 2>&1 1>/dev/null`
4. `./testRedirect 1>/dev/null 2>&1`
5. `./testRedirect 2>&1`

## EXERCICE 4

---

Votre programme favori semble ne plus répondre. La seule chose que vous avez à votre disposition, est votre invite de commandes shell.

Donnez deux différentes méthodes permettant de tuer ce programme?