

 <b>Module : LPI 101 102</b>	<b>Enseignante : OUESLATI Héra</b>
<p style="text-align: center;"><b>TP 05</b></p> <p style="text-align: center;"><b>Gestion des processus</b></p>	<b>Niveau : RSI 21</b>
	<b>Année universitaire : 2024/2025</b>

### Travail Demandé

Vous rédigerez un compte rendu, sur lequel vous indiquerez la réponse à chaque question, vos explications et commentaires (interprétation du résultat), et le cas échéant la ou les commandes utilisées. Pour avoir plus d'information sur une commande, consulter en premier lieu les pages de manuels correspondantes à l'aide de la commande man.

### Volume Horaire 6 heures

#### 1. Redirections, enchainement des processus

- a. Ecrire une seule ligne de commande permettant d'afficher la liste des utilisateurs, votre répertoire courant et la liste des processus qui s'exécutent.
- b. Reprendre la question précédente en envoyant le résultat dans un fichier **temp**.
- c. Ajouter au fichier **temp** la liste des i-nodes de fichiers et de répertoires de votre répertoire de courant.
- d. Proposer une commande qui affiche à l'écran un message d'erreur. Modifier cette commande de telle sorte que vous générez un fichier **erreurs** qui contiendra le message d'erreur affiché (Indication : la sortie standard d'erreurs).
- e. Créez deux fichiers **f1** et **f2** contenant chacun la liste des inodes des répertoires et fichiers, votre nom de connexion (**whoami**) des deux manières suivantes :
  - i. Pour créer le fichier **f1**, créez trois fichiers séparés en utilisant des redirections simples (>), puis les concaténer tous avec la commande cat en utilisant aussi une redirection simple pour créer enfin le fichier **f1**.
  - ii. Pour créer le fichier **f2**, utilisez les redirections doubles (>>) pour créer le fichier et ajouter dans le fichier au fur et à mesure le résultat des commandes.

### Récapitulatif des redirections

<b>nom_commande&lt;nom_fic</b>	Nom_commande prend comme entrée le fichier nom_fic
<b>nom_commande &gt;fichier</b>	La commande envoie la sortie dans fichier (si fichier existe, il est écrasé).
<b>nom_commande &gt;&gt; fichier</b>	Rajoute la sortie à la fin du fichier (si fichier n'existe pas, il est créé).
<b>nom_commande 2&gt; fichier</b>	La commande envoie les erreurs dans fichier(le fichier existe,il est écrasé).
<b>nom_commande 2&gt;&gt; fichier</b>	La commande rajoute les erreurs dans fichier (si le fichier n'existe pas, il est créé).

## 2. Visualisation et terminaison des processus

- a. Lancer la commande **ps**. Interpréter les résultats.
- b. Lancer la commande **ps -l**. Interpréter les résultats.

Les informations relatives aux processus sont :

- UID : l'identifiant de l'utilisateur propriétaire de processus.
- PPID : le numéro de processus père (Processus qui à lancé le processus courant) (Processus parent IDentifiant).
- PID : le numéro de processus. (Process IDentifiant).
- %CPU : le pourcentage de temps machine consommé par le processus.
- %MEM : le pourcentage de temps mémoire consommé par le processus.
- TTY : le terminal (la session) depuis le quel le processus été lancé.
- S : l'état de processus au moment du lancement de la commande (voir ci-après).
- WCHAN : l'événement que le processus attend (l'événement pour lequel le processus est endormi).
- START : le jour de lancement de processus.
- TIME : le temps d'exécution de la commande.
- STIME : heure de départ de processus.
- CMD : le libellé de la commande lancée.
- PRI : facteur de priorité.

- NI : valeur utilisée pour le calcul de la priorité.
- ADDR : adresse mémoire du processus.

Pour l'état de processus donnée par la valeur de S(State).

- R : le processus est prêt (Runnable).
- CPU : le processus est élu(en cour d'exécution).
- T : le processus est stoppé.
- P : le processus attend une page(typiquement récupération des données du swap vers la mémoire réelle ).
- D :le processus est en attente d'une E/S disque.
- S : le processus « dort » depuis moins de 20secondes.
- L : le processus est « idle », il « dort » depuis plus de 20 secondes (Idle).
- Z : processus en attente d'un message du noyau (processus Zombie).

- c. Essayer les commandes **ps -e** et **ps -f**, combiner les deux options **ps -ef** et interpréter les résultats.

L'option **-e** permet d'afficher tous les processus de système y compris les processus systèmes (les démons) est le premier processus init a un PID=1.

La commande **ps -f** permet d'afficher les processus actifs mais avec différents informations y compris STIME et USER au lieu du UID.

**ps -ef** permet de combiner les deux options

**ps -u** permet d'afficher les processus d'un utilisateur.

**Exemple :**

```

hela@localhost:~
File Edit View Search Terminal Help
[hela@localhost ~]$ ps -fu root
UID          PID  PPID  C  STIME TTY          TIME CMD
root          1      0  0  10:21 ?        00:00:03 /sbin/init
root          2      0  0  10:21 ?        00:00:00 [kthreadd]
root          3      2  0  10:21 ?        00:00:00 [ksoftirqd/0]
root          4      2  0  10:21 ?        00:00:00 [migration/0]
root          5      2  0  10:21 ?        00:00:00 [watchdog/0]
root          6      2  0  10:21 ?        00:00:00 [events/0]
root          7      2  0  10:21 ?        00:00:00 [cpuset]
root          8      2  0  10:21 ?        00:00:00 [khelper]
root          9      2  0  10:21 ?        00:00:00 [netns]
root         10      2  0  10:21 ?        00:00:00 [async/mgr]
root         11      2  0  10:21 ?        00:00:00 [pm]
root         12      2  0  10:21 ?        00:00:00 [sync_supers]
root         13      2  0  10:21 ?        00:00:00 [bdi-default]

```

- d. Décrire ce qui se passe si on lance la fréquence de commandes qui suivent et déduire la différence entre une exécution en avant plan et une exécution en arrière plan (qu'on appelle tâche de fond).

Note : La commande **sleep n** permet de rien faire pendant n secondes :

**sleep 10**

**ps**

**sleep 10 &**

**ps**

[hela@localhost ~]\$ sleep 10

```

ps
sleep [hela@localhost ~]$ ps
  PID TTY          TIME CMD
 1925 pts/0    00:00:00 bash
 1949 pts/0    00:00:00 ps

```

Cette commande permet d'exécuter un processus endormi pendant 10s. le prompt est affiché après l'exécution de cette commande (après 10 seconde).

Ps affiche la liste des processus en cours, le processus sleep est terminé déjà il ne s'affiche pas.

Sleep 10& permet de lancer un processus en arrière plan. Le numéro est affiché en crochets c'est le numéro de la tâche job ID, le numéro 1950 numéro du processus en cours.

```

[hela@localhost ~]$ ps
  PID TTY          TIME CMD
 1925 pts/0    00:00:00 bash
 1970 pts/0    00:00:00 ps
[1]+  Done

```

**sleep 10**

Ps réaffiche la liste des processus

e. Quelle est la différence entre les séquences de commandes suivantes **sleep 10 ; ls** et **sleep 5 & ; ls** ? Expliquer.

f. Lancer la commande **sleep 1000** en arrière plan (en utilisant le **&**). Quel est le P ID du processus lancé ? Donner la commande qui permet d'afficher le PPID (parent PID) du processus lancé.

g. Exécuter la commande **kill -19** en donnant en paramètre le pid du processus.

Essayer de nouveau la commande **ps**. Quel est le rôle de la commande **kill -19** ?

h. Exécuter la commande **kill -9** en donnant en paramètre le pid du processus.

Essayer de nouveau la commande **ps**. Quel est le rôle de la commande **kill -9** ?

i. Détruisez le processus correspondant à votre interpréteur de commandes courant. Quel se passe-t-il ?

#### e. Les commandes du job control : **jobs**, **fg**, **bg**

a. Taper les commandes suivantes et interpréter les résultats obtenus sur écran pour chaque commande.

```
sleep 2000&
```

```
jobs -l
```

```
sleep 1000&
```

```
jobs -l
```

```
fg %2
```

```
CTRL^Z
```

```
jobs -l
```

```
bg %2
```

```
jobs -l
```

```
kill %2
```

```
jobs -l
```

b. Pour conclure, que permettent les commandes **fg** et **bg** ?

c. Donner le rôle de la commande **jobs -l** ? Quelle est la différence entre les deux commandes **ps** et **jobs** ?

- d. **(conclusion)** Dessiner dans un digramme, 4 états de processus en arrière plan, processus en avant plan, processus suspendu, processus terminé, et construire les liaisons entre les différents états en utilisant les différentes commandes qu'on a étudiées, en particulier, **kill**, **bg**, **fg**.

#### 4. Commandes diverses (**ps**, **top**, **time**)

- a. Essayer les commandes suivantes : **ps**, **top**
- b. Que font ces commandes ?
- c. Taper la commande : **time** who
- d. Interpréter les résultats de la commande précédente, vous pouvez consulter l'aide en ligne `man` pour voir le rôle de la commande `time`.

Time : affiche le temps nécessaire pour l'exécution de la commande (real time).

User : CPU user time.

Sys : CPU système time.

#### 5. Redirections et Pipes

- a. A l'aide d'une redirection, créer un fichier contenant la liste des processus tournant sur votre machine.
- b. A l'aide d'une pipe, donner la commande permettant de compter le nombre de processus en utilisant commande **wc**. (Indication : la commande **wc -l fichier** permet de compter le nombre de lignes passé en paramètre).
- c. A l'aide de la commande **sort**, et en utilisant aussi une pipe, donner la commande qui affiche la liste des processus triés par ordre alphabétique.