Les commandes de base UNIX

TP 5 corrigé : Droits, Processus, Liens

1. **Affichez la liste des processus associés à votre terminal. Affichez la liste des processus dont vous êtes propriétaire. Recommencez en utilisant l'option -l. À quoi correspondent les colonnes PID et PPID ?**

```
nacre ~ $ ps
 PID TT       S  TIME COMMAND
16970 pts/2    S  0:00 /usr/local/util/bin/zsh4
16992 pts/2    S  0:07 vim systeme-sol.tml
17382 pts/3    R  0:00 /usr/local/util/bin/zsh4
 nacre ~ $ ps -l
 F   UID   PID  PPID %C PRI NI   SZ  RSS    WCHAN S TT       TIME COMMAND
 8  9625 16970 16967  0  48 20 4032 3152 mutex_ex S pts/2     0:00 /usr/local/u
 8  9625 16992 16970  1  58 20 4312 3760 mutex_ex S pts/2     0:07 vim systeme-
8  9625 17382 17379  0  48 20 3912 2840 mutex_ex S pts/3     0:00 /usr/local/u
```

Les numéros apparaissant dans la colonne PID sont les numéros des différents processus. Le PPID désigne lui le numéro du processus père du processus en question. Par exemple, ici, le processus 16992 (vim) est un fils du processus 16970 (zsh).

2. **Lancez une commande longue en arrière plan. Quel est le comportement du processus associé lorsqu'il reçoit les signaux suivants :**

Pour envoyer un signal à un processus, on peut faire : nacre ~ $ kill -23 17382

17382 désignant un numéro de processus (PID) et 23 le numéro d'un signal (SIGSTOP).

- sigkill (9)
  Le processus est tué sans pouvoir de défendre.
- sigstop (23)
  Le processus s'interrompt.
- sigcont (25)

Le processus interrompu redémarre.

3. **Utilisez la commande nice pour lancer des commandes ayant une faible priorité.**

```
nacre ~ $ nice -19 ./calcul
```

4. **Interprétez la hiérarchie des processus qui vous appartiennent.**

Pour vous aider à comprendre, vous pouvez afficher un arbre en lançant la commande pstree (FreeBSD) ou ptree (Solaris).

5. **La commande ps | wc compte deux processus en plus de ceux qui existent réellement lorsqu'on lance la commande. Pourquoi ?**

   Ce sont les processus associés aux commandes ps et wc que vous venez de lancer !

6. **Donner deux commandes pour reprendre l'exécution d'une instruction interrompue par un ^Z.**

   Vous disposez des commandes fg et bg.

   fg permet de poursuivre le déroulement du programme.

   bg le fait aussi mais il relance le programme *en tâche de fond*.

nom de login : whoami affiche l' « *User ID* »

Les droits

1. **Changez les droits d'un fichier fic1 pour que tous ceux de votre groupe puissent écrire dedans.**

chmod g+w fic1

2. **Donnez en une seule ligne le droit d'exécution à tous les utilisateurs d'un fichier script qui n'a jusqu'alors que des droits standards (-rw-r--r--).**

chmod uog+x script

3. **Le fichier toto a les droits suivants : -rwxr--r--. Modifiez-en les droits en une ligne de commande de sorte que le propriétaire n'ait plus que le droit de lecture.**

chmod u-wx toto

4. **Modifier les droits du fichier toto (-rwxr--r--) de sorte que le groupe et les autres utilisateurs aient les mêmes droits que le propriétaire.**

chmod og+wx toto

5. **Quelle option permet de modifier récursivement les droits d'un répertoire et des fichiers qu'il contient ?**

   C'est l'option -R. Par exemple, pour interdire tous les droits aux fichiers présents dans une branche du répertoire confidentiel, on peut faire :

   chmod -R og-rwx confidentiel

6. **Quelle option de mkdir permet de créer un répertoire en spécifiant les droits sur ce répertoire ?**

   C'est l'option -m qui permet de le faire. Par exemple, pour créer un répertoire en lecture interdite aux autres :

   mkdir -m og-r repertoire

7. **Affichez et interprétez les droits de /usr/sbin/mount.**

nacre ~ $ ll /usr/sbin/mount

-r-xr-xr-x 1 root bin 27208 Jan 9 2000 /usr/sbin/mount*

   Il s'agit d'un fichier éxécutable par tout utilisateur.

8. **Umask**

Pour un fichier :

Si vous tapez **umask 022**, vous partez des droits maximum **666** et vous retranchez **022**, on obtient donc **644**, par défaut les fichiers auront comme droit **644** (**-rw-r-r--**).

Si vous tapez **umask 244**, vous partez des droits maximum **666** et vous retranchez **244**, on obtient donc **422**, par défaut les fichiers auront comme droit **422** (**-rw--w--w-**).

Pour un répertoire :

Si vous tapez **umask 022**, vous partez des droits maximum **777** et vous retranchez **022**, on obtient donc **755**, par défaut les fichiers auront comme droit **644** (**-rwxr-xr-x**).

Si vous tapez **umask 244**, vous partez des droits maximum **777** et vous retranchez **244**, on obtient donc **533**, par défaut les fichiers auront comme droit **422** (**-rwx-wx-wx**).

umask n'est utilisatable que si on est propriétaire du fichier.

Information supplémentaire : les liens

1. **Vous avez chez vous un répertoire tmp/ qui contient un fichier bidon. Créez un lien physique sur tmp/bidon appelé blo, dans votre répertoire d'accueil (HOME). Comparez les contenus de tmp/bidon et de blo. Que contient blo ?**

ln tmp/bidon blo

Les contenus sont identiques. Les noms de fichiers tmp/bidon et blo renvoient au même endroit sur le disque dur.

2. **Même question avec un lien symbolique.**

ln -s tmp/bidon blo

Le contenu *semble* être identique, cependant, à propos du blo, le disque dur ne contient comme information que le nom du fichier vers lequel le lien pointe (tmp/bidon).

3. **Quelles sont les différences entre les liens durs et les liens symboliques ?**

On vient d'en voir une. On en déduit que le temps d'accès au contenu d'un lien dur est plus rapide que pour un lien symbolique. Une autre différence notable est que, contrairement aux liens symboliques, un lien dur ne peut pas pointer vers un répertoire.

4. **Dans quel cas ne peut-on pas faire de lien physique ? Que faut-il faire ?**

On ne peut pas faire de lien dur vers un répertoire. Dans ce cas, il faut donc utiliser un lien symbolique.

5. **Quel est l'effet de chmod sur un lien ?**

S'il s'agit d'un lien dur, les droits des *deux* fichiers liés sont modifiés. En revanche, s'il s'agit d'un lien symbolique, les deux fichiers peuvent avoir des droits différents.