

SYSTÈME D'EXPLOITATION OPERATING SYSTEM – CORRECTION I

Systeme d'exploitation

Exercice 1

1. La commande « `uname -o` » affiche : GNU/Linux
La commande « `uname -r` » affiche : 5.10.60-v7l+
2. Le serveur sur lequel on est utilise un système d'exploitation Linux (GNU = GNU is Not Unix)
5.10.60-v7l+ pourrait être la version.
3. L'option « `-o` » affiche le système d'exploitation utilisé sur le serveur : on rappelle que c'est ce qui fait la liaison entre le matériel, l'utilisateur et les différentes applications.
L'option « `-r` » affiche la version du noyau du système d'exploitation : le noyau est la brique principale du système d'exploitation.

Remarque : lorsqu'on cherche quelque chose de précis dans le manuel, on peut taper « /ce qu'on cherche » pour faire une recherche de caractère. Par exemple, vous pouvez taper `/-o`.

Exercice 2

La distribution du système d'exploitation utilisée par le serveur est Raspbian CGN : c'est le nom de l'organisme qui a configuré « à sa façon » et redistribué le système d'exploitation.

Remarque : on peut retrouver cette distribution dans [ce graphique](#) qui répertorie toutes les distributions Linux qui existent (ou ont existé) depuis 1992. On remarquera tout en bas les distributions Android !

Systeme de fichiers

Exercice 3 : consulter le contenu d'un répertoire (ls)

Le répertoire « `/home/lfc1ppartage` » contient 4 répertoires (nommés `tpRx1`, `tpRx2`, `tpSE1` et `tpSE1`).

```
lfc1pxx@SshSzpiegServer:~ $ ls /home/lfc1ppartage/  
tpRx1  tpRx2  tpSE1  tpSE2
```

Exercice 4 : afficher une arborescence de fichiers 1 (tree)

1. La commande `tree` permet d'afficher l'arborescence des répertoires et dossiers
2. Le symbole « `/` » en fin de ligne indique la racine : on demande à afficher le système de fichiers à partir de la racine.
3. L'option « `-L 1` » permet d'afficher l'arbre avec une profondeur de 1 depuis la racine.
L'option `-n` permet de ne pas mettre de système de couleur dans l'affichage : c'est pour vous éviter de prendre des habitudes avec ce code couleur, qui ne sera peut-être pas le même avec un autre shell.
L'option `-F` permet de mettre un « `/` » à la fin du nom d'un répertoire : pour mieux les repérer par rapport aux fichiers.

Exercice 5 : afficher une arborescence de fichiers 2 (tree)

1. Commande à taper : `tree -nF /home/lfc1ppartage/`
2. « 9 directories, 4 files » signifie « 9 répertoire et 4 fichiers » : c'est-à-dire qu'il y a en tout 9 répertoires (tout ce qui se finit par « `/` ») et 4 fichiers (le reste) dans cet arbre.

Exercice 6 : créer un répertoire (mkdir)

```
lfc1pxx@SshSzpiegServer:~ $ pwd  
/home/lfc1pxx  
lfc1pxx@SshSzpiegServer:~ $ ls  
lfc1pxx@SshSzpiegServer:~ $ mkdir tpSE1  
lfc1pxx@SshSzpiegServer:~ $ ls  
tpSE1
```

Exercice 7 : se déplacer dans un répertoire 1 (cd)

```
lfclpxx@SshSzpiegServer:~ $ cd ./tpSE1/  
lfclpxx@SshSzpiegServer:~/tpSE1 $ pwd  
/home/lfclpxx/tpSE1
```

Exercice 8 : se déplacer dans n'importe quel répertoire du système de fichier (cd)

1.

```
lfclpxx@SshSzpiegServer:~/tpSE1 $ cd ..  
lfclpxx@SshSzpiegServer:~ $ pwd  
/home/lfclpxx
```

On constate qu'on est remonté dans le répertoire père de celui où on se trouvait. Le paramètre « .. » désigne donc bien le répertoire père du répertoire courant.

2.

```
lfclpxx@SshSzpiegServer:~ $ cd ./tpSE1/  
lfclpxx@SshSzpiegServer:~/tpSE1 $ pwd  
/home/lfclpxx/tpSE1
```

3.

```
lfclpxx@SshSzpiegServer:~/tpSE1 $ cd /home/lfclpxx/  
lfclpxx@SshSzpiegServer:~ $ pwd  
/home/lfclpxx
```

4. a) C'est un **chemin relatif** : puisqu'on remonte dans le répertoire père à partir du répertoire courant.
b) C'est un **chemin absolu** : on pourrait taper cette commande de n'importe quel autre répertoire, on arriverait toujours au même endroit.

Exercice 9 : créer sa propre arborescence de répertoires (mkdir)

1.

```
lfclpxx@SshSzpiegServer:~ $ cd /home/lfclpxx/tpSE1/  
lfclpxx@SshSzpiegServer:~/tpSE1 $ pwd  
/home/lfclpxx/tpSE1
```

2.

```
lfclpxx@SshSzpiegServer:~/tpSE1 $ mkdir documents  
lfclpxx@SshSzpiegServer:~/tpSE1 $ ls  
documents
```

3.

```
lfclpxx@SshSzpiegServer:~/tpSE1 $ cd ./documents/  
lfclpxx@SshSzpiegServer:~/tpSE1/documents $ pwd  
/home/lfclpxx/tpSE1/documents
```

4. a) On obtient :

```
lfclpxx@SshSzpiegServer:~/tpSE1/documents $ tree ./../.. -nF  
./../..  
├── tpSE1/  
│   └── documents/
```

On remarque que les réponses affichées sont exactement les mêmes.

```
2 directories, 0 files
```

```
lfclpxx@SshSzpiegServer:~/tpSE1/documents $ tree /home/lfclpxx/ -nF  
/home/lfclpxx/  
├── tpSE1/  
│   └── documents/
```

```
2 directories, 0 files
```

b) Avec la première commande, on affiche l'arborescence de fichiers du répertoire père du répertoire père du répertoire courant (**chemin relatif!**), qui est le répertoire /home/lfclxx/

Avec la deuxième commande, on affiche l'arborescence de fichiers du même répertoire mais en le nommant avec son **chemin absolu** /home/lfclxx/

5.

```
lfclpxx@SshSzpiegServer:~/tpSE1/documents $ mkdir boulot
lfclpxx@SshSzpiegServer:~/tpSE1/documents $ ls
boulot
```
6.

```
lfclpxx@SshSzpiegServer:~/tpSE1/documents $ cd ./boulot/
lfclpxx@SshSzpiegServer:~/tpSE1/documents/boulot $ pwd
/home/lfclpxx/tpSE1/documents/boulot
```

Exercice 10 : copier un fichier existant (cp)

1. Avec le chemin relatif:

```
lfclpxx@SshSzpiegServer:~/tpSE1/documents/boulot $ cd ../
lfclpxx@SshSzpiegServer:~/tpSE1/documents $ pwd
/home/lfclpxx/tpSE1/documents
```

Avec le chemin absolu :

```
lfclpxx@SshSzpiegServer:~/tpSE1/documents/boulot $ cd /home/lfclpxx/tpSE1/documents/
lfclpxx@SshSzpiegServer:~/tpSE1/documents $ pwd
/home/lfclpxx/tpSE1/documents
```

2. Avec le chemin absolu :

```
lfclpxx@SshSzpiegServer:~/tpSE1/documents $ cp /home/lfclppartage/tpSE1/
documents/fiche.txt /home/lfclpxx/tpSE1/documents/
lfclpxx@SshSzpiegServer:~/tpSE1/documents $ ls
boulot  fiche.txt
```

OU

Avec le chemin relatif :

```
lfclpxx@SshSzpiegServer:~/tpSE1/documents $ cp /home/lfclppartage/tpSE1/
documents/fiche.txt .
lfclpxx@SshSzpiegServer:~/tpSE1/documents $ ls
boulot  fiche.txt
```