

1.2

Les lignes de commandes

NSI 1ÈRE - JB DUTHOIT

1.2.1 Explorer l'arborescence

Utiliser les commandes *shell* qui permettent de se déplacer dans l'arborescence, de visualiser les répertoires, fichiers, ainsi que des informations sur eux :

1. Exécuter **pwd** qui donne l'adresse du répertoire dans lequel on se trouve. La noter. Est-ce une *adresse absolue* ou une *adresse relative* ? Comment le voit-on ?
2. Pour *lister* le contenu du répertoire courant, utiliser **ls**. Combien d'éléments contient-il ?
3. On peut ajouter des *options* à la commande :
 - **ls -a** permet de lister également les fichiers cachés, à savoir ceux qui ont un nom commençant par **.** et qui sont invisibles par défaut.
On voit apparaître notamment le **.** qui représente le répertoire courant, ainsi que **..** qui désigne le dossier parent situé au niveau juste au dessus dans l'arborescence.
 - **ls -l** donne une description plus complète.
Le premier caractère de chaque ligne indique s'il s'agit d'un fichier classique **-** ou d'un répertoire (**d** pour *directory*). On peut trouver également l'utilisateur propriétaire (*celui qui l'a créé, à moins qu'on ne l'ait changé ensuite*) et son groupe, la taille du fichier, sa date de dernière modification...
 - On peut combiner les options : **ls -a -l** ou **ls -al**. Donner le nom d'un répertoire, celui d'un fichier ordinaire caché et celui d'un fichier ordinaire non caché.
 - On peut aussi donner en *argument* le dossier à lister.
 - ☞ Exécuter **ls /** ou **ls -l /** et indiquer le nombre de répertoires immédiatement présents à la racine de l'arborescence (*Ignorer le **linuxrc** qui est un lien symbolique comme l'indique le **l** en début de ligne...*)
 - ☞ Vérifier avec **pwd** que vous êtes toujours dans le même dossier.
4. **cd** permet de se déplacer dans les répertoires. On lui donne en argument (*écrit après un espace*) l'adresse relative ou absolue où l'on veut se rendre.
 - **cd Devoirs** permet ainsi de se placer dans le sous-répertoire ainsi nommé. On peut commencer à écrire le nom du dossier puis utiliser l'*autocomplétion* en appuyant sur la touche de tabulation).
 - ☞ Combien de fichiers ordinaires et combien de dossiers contient ce répertoire ?
 - Revenir dans le dossier parent puis remonte d'un niveau encore en exécutant à nouveau la même commande (il suffit d'utiliser la flèche "vers le haut", on peut le faire plusieurs fois pour remonter dans l'historique des commandes).
 - ☛ Le nom des utilisateurs identifiés sur la Weblinux : ce sont ceux des sous-dossiers de **/home**.
 - On peut toujours revenir à la base de son espace personnel avec **cd**

5. À l'aide des commandes précédentes, représenter l'arborescence (*partielle*) des fichiers en partant de la racine /. Il est possible de l'écrire sous la forme suivante :

```

/
|--- bin
|--- dev
|--- home
|   |--- alice
|   |   |--- ...
...   ...   ...

```

Se limiter à *un* niveau de profondeur, sauf pour /home/ où vous illustrerez les sous-répertoires, et /home/alice où vous montrerez les fichiers ordinaires et quelques dossiers.

6. Se rendre dans le dossier /home/bob/vide/. Écrire une commande *unique* qui permet alors de se rendre directement dans /home/alice/Documents/.

1.2.2 Créer, copier, déplacer, supprimer

Créer un dossier, créer un fichier

1. Créer dans le répertoire `alice` le dossier `interros` en utilisant la commande `mkdir`
2. Créer dans le répertoire `interro` le fichier `interro1.txt` en utilisant la commande `touch`

Lire un fichier

Se placer dans le dossier /bob. Il est possible de lire le fichier `a.txt` avec la commande `cat`.

Déplacer, copier

1. Déplacer ce fichier à la base de ton espace personnel :
`mv Documents/detail-alice.txt ~`
 Le `~` final peut être remplacé par `.` si on est déjà situé au bon endroit.
 ☞ S'entraîner à déplacer les fichiers d'un endroit à un autre, en utilisant soit des références absolues, soit des références relatives.
2. Créer un nouveau dossier local `exo` `mkdir exo` (*make directory*). Copier le fichier `interro1.txt` dans ce dossier avec la commande `cp` qui fonctionne comme `mv` mais conserve le fichier d'origine.

Écrire dans un fichier

De façon générale, il est possible d'utiliser la commande `echo` pour écrire dans un fichier.

1. Se placer dans le répertoire `interro` et entrer `echo "exercice 1:">interro1.txt`
 ☞ Lire le fichier avec la commande `cat` et observer le résultat.
2. Entrer `echo "exercice 2:">interro1.txt`
 ☞ Lire le fichier avec la commande `cat` et observer le résultat.
3. Afin de ne pas écraser le fichier, il est possible d'écrire : `echo "exercice 3:">>interro1.txt`
 ☞ Lire le fichier avec la commande `cat` et observer le résultat.

4. Tester `echo "exercice 1:">interro2.txt`
☛ Que se passe-t-il ?

Supprimer un fichier

Détruire `interro2.txt` avec `rm interro2.txt` (*remove*).