

7.2

Découvrir les lignes de commandes

NSI 1ÈRE - JB DUTHOIT

7.2.1 weblinux

Afin d'éviter toute installation, tout en disposant toujours du même environnement de base, disponible depuis toute machine connectée à Internet, vous travaillerez à distance sur une **machine « Weblinux »** depuis un navigateur :

<http://weblinux.univ-reunion.fr>

Après lancement, vous êtes connecté automatiquement, en tant qu'utilisatrice **alice** et vous vous trouvez dans votre répertoire personnel.

```
VFS: Mounted root (9p filesystem) readonly on device 0:12.
devtmpfs: mounted
Freeing unused kernel memory: 136K (c054e000 - c0570000)
Préparation de la weblinux (patientez svp)
*****
* Votre Weblinux est maintenant opérationnelle. *
*****
$
```

Définition

Une façon d'interagir avec le système est d'utiliser un terminal (ou console) dans lequel vous pourrez taper des commandes (qui ne sont rien d'autre que des programmes). Le programme avec lequel vous interagissez pour exécuter les commandes s'appelle le **shell**.

En bas du **shell**, les trois icônes permettent de

- Sauvegarder sur PC une archive (**.tar**) qui reprend une grande partie du dossier personnel d'**alice**.
- D'envoyer un fichier de son PC vers la WebLinux.
- De coller dans le shell du texte copié sur votre PC.

Par défaut, l'**invite de commande** y reste réduite à un simple \$.

Ajouter PS1="\\[\\033[01;32m\\] [\\u@\\h\\[\\033[00m\\]]:\\\\[\\033[01;35m\\]\\w]\\[\\033[00m\\]\$\\\" comme ligne de commandes pour avoir une présentation plus habituelle :

```
[alice@weblinux:~]$ cd ..
[alice@weblinux:/home]$
```

7.2.2 Aide sur les commandes

Le plus important à retenir, surtout quand on connaît une commande mais pas ses options, ou si l'on découvre une commande inconnue dans un exemple, est de savoir accéder à l'**information intégrée** au shell :

- **man la_commande** renvoie une aide complète (souvent plus longue qu'une page, on peut ajouter à la suite **less** pour se déplacer dedans avec les flèches, puis quitter avec q, voir à la suite pourquoi). **info la_commande** offre un service comparable.

- Plus simplement, la plupart des commandes ont une option **-help** (en général, double tiret pour une option qui s'écrit sur plus d'un caractère) ou **-h** qui décrit leur utilisation.
☞ Avec **ls -help**, trouver comment lister le contenu d'un répertoire en triant les éléments par ordre décroissant de taille.

7.2.3 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 :

- 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?

A RETENIR :

- /** : Répertoire racine
- ~** : Répertoire d'accueil (home)
- .** : Répertoire courant
- ..** : Répertoire parent

- Pour lister le contenu du répertoire courant, utiliser **ls**. Combien d'éléments contient-il?
- 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 1 en début de ligne...)

☞ Vérifier avec **pwd** que vous êtes toujours dans le même dossier.

- 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/`.

7.2.4 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).

Exercice 7.143

On considère l'arborescence suivante :

```
/  
|---r  
|   |---a  
|   |   |---d  
|   |   |   |f3  
|   |   |   |f4  
|  
|   |---b  
|   |   |f2  
|   |   |---e  
|   |   |   |f5  
|  
|   |f1  
|---c  
|   |---f  
|   |   |f6  
|   |   |---g  
|   |   |   |f7  
|   |   |   |f8
```

1. On se place dans le dossier `b`, et on utilisera ici des **références relatives**.
 - a) Quel est le résultat de la commande `ls` ?
 - b) Quelle commande faut-il utiliser pour lister le contenu du répertoire `a` ?
 - c) Quelle commande faut-il exécuter pour que la commande `ls` affiche `f7 f8` ?
2. On se place dans le répertoire `b`, et on utilisera ici des **références relatives**.
 - a) Quelles commandes faut-il exécuter pour copier le fichier `f1` dans le répertoire `d` et afficher le contenu de ce répertoire ?
 - b) Quelle(s) commande(s) faut-il ensuite effectuer pour que la commande `pwd` affiche `/r/c/f` ?
3. Dans l'arbre de fichier (peu importe l'endroit, donc on utilisera des **références absolues**), quelles commandes faut-il exécuter pour :
 - a)
 - Créer un répertoire `h` dont le répertoire parent est `b`.

- Déplacer les fichiers qui sont dans le répertoire **c** (éventuellement ses sous-répertoires) dans le nouveau répertoire **h**.
- b) Dessiner l'arborescence après ces commandes
4. Dans l'arbre de fichier (peu importe l'endroit, donc on utilisera des **références absolues**), quelles commandes faut-il exécuter pour supprimer la branche **a** de l'arbre des fichiers ?
 5. En utilisant un émulateur linux :
 - a) Construire dans le répertoire **/** le répertoire **r** et ses sous-répertoires.
 - b) Vérifier les résultats aux questions précédentes.

Exercice 7.144

Une suite de commande permet à l'utilisateur d'automatiser certaines tâches. On parle alors de **script**, stocké dans un fichier d'extension **.sh**.

1. Se placer dans le répertoire racine **/**
2. Créer un fichier **auto.sh**
3. Ajouter au fichier les lignes suivantes :

```
mkdir ./nv_dossier
touch ./nv_dossier/nv_fichier
echo "Hello World !" > ./nv_dossier/nv_fichier
```
4. exécuter le fichier **auto.sh** et vérifier le résultat.