

Introduction aux commandes de base sous UNIX

--

Atelier DNS AFTLD Yaoundé 2004

- Exemple d'une session (se logger, exécuter des commandes, se délogger) - Les commandes de base

1. Une session avec un "shell" sous UNIX

Une session sous UNIX est composée de trois parties:

- Se logger (login: / password:)
- Exécuter des commandes (interaction avec l'interpréteur de commandes)
- Se délogger (logout / exit)

1.1 Exemple d'invite typique

```
MonOS (machine.reseau.local) ttyp0
login: robert
Password:

Last login: Tue Oct 14 06:03:14 on ttyp0

/utilisateurs/robert$ _
```

login:

L'invite de login demande à l'utilisateur de taper son identificateur

Password:

Après l'identificateur, l'utilisateur doit taper son mot de passe. Le mot de passe n'apparaît pas quand on le tape.

Une fois loggé, utiliser les commandes pour interagir avec l'interpréteur ("shell").

1.2 Exemple de commande simple

Une commande utile: la commande `date` pour connaître... la date et l'heure:

```
/utilisateurs/robert$ date
Tue Oct 14 14:07:06 PST 1997
/utilisateurs/robert$ _
```

Une fois le travail terminé, se délogger.

1.3 Fin de session

La commande `exit` ou `logout` met fin à la session en cours, et l'invite réapparaît, en attente d'une nouvelle session:

```
/utilisateurs/robert$ exit
MonOS (machine.reseau.local) ttyp0

login:
```

2. Les commandes de base

Les commandes suivantes constituent un minimum nécessaire à la navigation dans le système de fichiers (l'arborescence) UNIX.

- `cd` / `pwd`: changer de répertoire / afficher le répertoire courant
- `ls`: afficher la liste des fichiers et sous-répertoires d'un répertoire donné ou du répertoire courant par défaut
- `cp`: faire une *copie* d'un fichier vers un nouvel endroit
- `mv`: *déplacer* un ou plusieurs fichiers d'un endroit à un autre
- `rm`: effacer un ou plusieurs fichiers
- `mkdir` / `rmdir`: créer un nouveau répertoire ou en supprimer un existant

3. Droits et permissions

Notions d'*utilisateurs* et de *groupes*: UNIX distingue trois classes d'utilisateurs: le propriétaire du fichier (*owner*), le groupe d'utilisateurs auquel appartient le fichier (*group*), et le reste du monde (*others*). Rappel: un fichier ne peut appartenir à plus d'un utilisateur et d'un groupe à la fois.

Les commandes pour manipuler les droits:

- `chmod` *les droits* `le_fichier`
- `chown` *nouveau_propriétaire* `le_fichier` -- ne peut être utilisé que par le super-utilisateur
- `chgrp` *groupe_cible* `le_fichier` -- on doit appartenir au groupe cible.

Exercices:

1. Examiner les droits du fichier `.profile` dans votre répertoire -- en profiter pour le rendre accessible en écriture pour le groupe auquel appartient ce fichier.

Ensuite, essayer de changer le propriétaire de ce fichier.

2. Exécuter les commandes suivantes:

- créer un répertoire `dossier`
- passer super-utilisateur et changer les droits de ce répertoire pour qu'il appartienne à l'utilisateur `root` et au groupe `bin`
- toujours en tant que super-utilisateur, créer dans ce répertoire un fichier nommé `test`. Changez le propriétaire du fichier afin qu'il vous appartienne (votre compte).
- quitter la session super-utilisateur (`exit`) et tenter de supprimer ce fichier

3. Dans un répertoire quelconque, exécuter la commande:

```
touch un_nom_de_fichier
```

Afficher les permissions de ce fichier avec la commande:

```
ls -l un_nom_de_fichier
```

4. Enchaînement et redirections

La commande `wc` permet de compter le nombre de lignes, mots et caractères que contient un fichier texte. Utiliser la commande `ls -l`, le pipe (`|`) et `wc` pour connaître le nombre de fichier dans le répertoire etc.

Rediriger la *sortie standard* de la commande ci-dessus vers un fichier de votre choix.

Ajouter la date du jour à la fin de ce fichier en utilisant la commande `date`.

5. Le système de fichier UNIX -- arborescence, montage et liens

Le système de fichiers sous UNIX a une racine unique (`/`). Des partitions peuvent être montées à la place de répertoires. Des liens peuvent pointer sur des fichiers ou répertoires présents ailleurs dans l'arborescence.

1. Utiliser la commande `mount` pour examiner les partitions montées sur le système de fichier.
2. Déterminer la nature du répertoire `/home`
3. Examiner les répertoires `/var/tmp` et `/usr/tmp`. Remarquer dans quel répertoire on se trouve si l'on fait:

```
cd /usr/tmp
cd ..
pwd
```

4. Créer dans le répertoire `/tmp`, un lien symbolique pointant sur lui-même. Utiliser la commande `ln --` voir le man correspondant. Observer ce qui se passe lorsque l'on essaye de faire `cd le_lien_symbolique`.

6. Un éditeur de texte: vi

`vi` est un éditeur dit modal: il existe deux modes d'exécution:

- le mode *entrée*: on peut taper du texte comme on le ferait sur un éditeur quelconque.
- le mode *commande*: permet d'exécuter des commandes de manipulation du texte, de déplacement, de recherche et de sauvegarde / chargement du texte.

6.1 Commandes de base

Pour passer du mode *entrée* au mode *commande* -- touche [ESC] Pour passer du mode *commande* au mode *entrée* -- plusieurs touches:

- `i`: insérer du texte à l'endroit du curseur (INSERT)
- `a`: insérer du texte *après* le curseur (AFTER)
- `I`: insérer du texte en *début* de ligne

- A: insérer du texte en *fin* de ligne
- o: commencer une nouvelle ligne *après* la ligne courante
- O: commencer une nouvelle ligne *avant* la ligne courante

6.2 Se déplacer dans vi:

On peut utiliser les touches fléchées pour se déplacer. De plus, on peut utiliser certaines commandes pour aller plus vite.

- w: un mot vers l'avant (WORD)
- b: un mot vers l'arrière

La plupart des commandes peuvent prendre un argument numérique avant leur exécution. Par exemple, pour avancer de 10 mots, taper: 10w.

6.3 Copier/couper/coller du texte:

- yy: copier la ligne courante dans le tampon (ex: 5yy -> copier 5 lignes dans le tampon)
- p: coller le contenu du tampon *après* la ligne courante.
- P: coller le contenu du tampon *avant* la ligne courante.
- D: détruire le reste de la ligne depuis le curseur (et coupe dans le tampon)
- dd: détruit entièrement la ligne courante (et coupe dans le tampon)
- dw: effacer le mot à droite du curseur
- db: effacer le mot à gauche du curseur

6.4 Charger / sauver

- :r fichier_à_insérer -- insère le fichier indiqué à l'endroit du curseur
- :e fichier_à_éditer -- charge un nouveau fichier afin de l'éditer
- :w -- sauve le fichier sous le nom courant, ou sous un autre nom si il est précisé
- :q -- quitte vi, à la condition que le texte n'a pas été modifié depuis son chargement
- ZZ -- sauve ET quitter vi.

Pour les commandes précédentes, il est possible de forcer certaines commandes (ex: :q! pour quitter en abandonnant les changements), ou de les combiner (ex: :wq, qui est équivalent à ZZ).

1. S'entraîner à manipuler des fichiers textes avec vi (couper / copier des lignes et des mots, insérer un autre texte au milieu du texte courant, sauver sous le même nom, un autre nom, etc...)

7. Détail des commandes de base

cd / pwd

Syntaxe: cd *nouveau_répertoire*

Changer de répertoire courant. Par exemple, pour se positionner dans le répertoire /usr/local/bin, il suffit de taper:

```
cd /usr/local/bin
```

Syntaxe: pwd

Afficher le répertoire courant. Exemple:

```
$ pwd
/utilisateurs/norbert
```

ls

Syntaxe: `ls [répertoire | fichier]`

Afficher la liste des fichiers et des sous-répertoires d'un répertoire donné, ou du répertoire courant si aucun argument n'est précisé. Exemple:

```
ls /etc/
ls /etc/passwd
```

cp

Syntaxe: `cp source [source2] destination`

Copier un ou plusieurs fichiers vers un autre répertoire, ou copier un fichier vers un autre nom. Exemple:

```
cp lettre.txt lettre_2.txt
cp lettre.txt /utilisateurs/robert/
cp lettre_1.txt lettre_2.txt lettre_3.txt /utilisateurs/public/
```

mv

Syntaxe: `mv fichier1 [fichier2] ... destination`

Déplacer un, ou plusieurs, fichiers, vers un répertoire cible. Exemple:

```
mv lettre.txt /tmp/
mv lettre_1.txt lettre_2.txt /tmp/
```

rm

Syntaxe: `rm fichier1 [fichier2] ...`

Effacer un ou plusieurs fichiers sur le disque. Exemple:

```
rm lettre.txt
rm /tmp/lettre_1.txt
rm lettre_2.txt lettre_3.txt
```

mkdir / rmdir

Syntaxe: `mkdir nom_du_répertoire`

Crée un répertoire du nom précisé. Exemple:

```
mkdir Lettres
mkdir /utilisateurs/robert/Travail
```

Syntaxe: `rmdir nom_du_répertoire`

Efface le répertoire du nom précisé. `rmdir` n'efface pas les répertoires qui contiennent des fichiers. Exemple:

```
rmdir Lettres
rmdir /utilisateurs/robert/Travail
```


CONTENTS

1. Une session avec un "shell" sous UNIX	1
1.1 Exemple d'invite typique	1
1.2 Exemple de commande simple	1
1.3 Fin de session	2
2. Les commandes de base	2
3. Droits et permissions	2
4. Enchaînement et redirections	3
5. Le système de fichier UNIX -- arborescence, montage et liens	3
6. Un éditeur de texte: vi	3
6.1 Commandes de base	3
6.2 Se déplacer dans vi:	4
6.3 Copier/couper/coller du texte:	4
6.4 Charger / sauver	4
7. Détail des commandes de base	4