

JI3-2011

**3^{èmes} Journées Informatiques des CPGE
Agadir, du 12 au 14 mai 2011**

Travaux Pratiques Linux

Saad Agoujil, FST Errachidia

TP1: commandes de bases Linux

1 Introduction

Le but de ce TD est la prise en main des commandes de base de l'environnement Linux.

2 Le manuel

Une description de toute les commandes est disponible avec la commande `man` ou `help`. N'hésitez pas à l'utiliser. `man man`, `man ls`.

3 Premier contact

Ouvrir un terminal

Les commandes que vous tapez sont analysées et exécutées par un interprète de commandes appelé Shell .

Tapez `ls` pour obtenir la liste des fichiers.

4 Commandes

Une commande est l'exécution d'un programme dans l'interprète (Shell). Elle prend en entrée des options et/ou des paramètres. Elle peut renvoyer de l'information à l'écran ou dans un fichier, modifier un fichier, ou produire un message d'erreur.

Exercices

Tapez les commandes suivantes et utilisez le manuel `man` pour expliquer ce qu'elles font.

```
date
which date
cat /etc/passwd.
```

Le fichier `/etc/passwd` contient les informations liées aux utilisateurs de la machine.

```
wc -l /etc/passwd
touch essai
ls
ls -l
ls -al
```

Si une commande ne rend pas la main, on peut arrêter le programme correspondant en tapant "CTRL C".

Tapez la commande:

```
dd
```

Que se passe-t-il ? Après avoir repris la main, utilisez le manuel pour comprendre ce qui s'est passé.

5 Répertoire de travail

Certaines commandes, dont `pwd`, `ls`, `touch`, `mkdir`, `cd` et `rm`, permettent d'explorer l'espace de travail personnel et de le modifier.

A l'aide du manuel (`man`) ou `help` et en exécutant et en analysant le résultat de chacune des lignes de commande suivantes, préciser le rôle de chacune de ces commandes :

```
pwd
ls
ls -l
ls -a
ls -al
rm essai
ls -l
mkdir SEessais
ls -l
cd SEessais
pwd
ls
touch toto
ls -l
cd..
```

Que représentent les répertoires "." et ".." ?

```
pwd
ls -l
cd /etc
pwd
```

```
ls -l
which date
cd /bin
pwd
ls -l
cd
pwd
ls -l
```

6 Editeurs de texte

Un éditeur de texte permet de rentrer du texte dans un fichier afin de le conserver. La commande `cat` constitue un éditeur (très) simplifié.

```
cat > fich1 (Enter)
Entrer le texte (Enter)
A stocker (Enter)
Dans le fichier (Enter)
CTRL D
```

Tapez la commande:

```
ls -l
cat fich1
```

L'option `> fich1` indique à la commande `cat` que les informations produites par `cat` ne seront pas affichées à l'écran, mais seront stockées dans le fichier `fich1`.

Dans le deuxième appel à `cat`, quel le statut du mot `fich1` ? Expliquer.

Il existe bien sûr des éditeurs de texte plus sophistiqués comme `vi` ou `emacs`.

7 Manipulation de fichiers

Revenez à votre répertoire racine en tapant la commande `cd`.

Créez un sous-répertoire `unix`.

Dans ce répertoire, créez un fichier "texte.txt" contenant la phrase "Il fait beau".

Tapez `ls`.

Modifiez le nom du fichier `texte.txt` en `texte2.txt` en tapant la commande `mv texte.txt texte2.txt`.

Tapez `ls`.

Créez une deuxième répertoire `test` dans votre répertoire racine, et un sous-répertoire `tt` dans le répertoire `test`.

Déplacez le fichier `texte2.txt` précédent dans le répertoire `test/tt`.

En utilisant la commande `cp`, copiez le fichier `texte2.txt` dans le répertoire racine.

Supprimez le fichier `texte2.txt` du répertoire `test/tt` en utilisant la commande `rm`.

TP 02: le système de fichier UNIX

1 Introduction

Le but de ce TD est l'étude du système de fichier UNIX.

2 Manipulation de fichiers

Le système de fichier UNIX est organisé selon une arborescence dont la racine est `/`.

Le répertoire d'accueil (home directory) est le répertoire dans lequel on se trouve au début. Il est noté `~`. Par exemple, la commande `cd ~` ramène dans le répertoire d'accueil.

Le répertoire courant est celui dans lequel on se trouve à un moment donné.

Créez dans votre répertoire d'accueil un répertoire `rep`.

1) Quelle est la commande à utiliser ?

Allez dans ce sous-répertoire (commande `cd`).

Créez les sous-répertoires `work` et `play`.

En utilisant la commande `ls -l`, affichez la liste des fichiers du répertoire `rep` avec les détails.

En utilisant la commande `ls`, affichez la liste des fichiers qui commencent aussi par `."`.

2) Quelle est l'option à utiliser ?

3) Quels sont les fichiers contenus dans ce répertoire ?

Copiez le fichier `/etc/passwd` dans votre répertoire d'accueil.

4) Quelle est la commande à utiliser ?

Déplacez le dans votre sous-répertoire `play`.

5) Quelle est la commande à utiliser ?

On se place maintenant dans le répertoire `play`.

Créez dans ce répertoire un fichier `toto` contenant la phrase "hello", en utilisant l'éditeur `xemacs`.

Créez un lien symbolique du fichier `tata` vers le fichier `toto`.

6) Quelle est la commande à utiliser ?

Tapez `ls -l`.

7) Qu'observez-vous ?

En utilisant la commande `cat`, visualisez le contenu du fichier `tata`.

8) Qu'observez-vous ?

Modifiez le fichier `toto` en rajoutant le mot "world" après "hello". Visualisez le contenu du fichier `tata`.

9) Qu'observez-vous ?

Créez un répertoire `other`, et créez dans ce répertoire un fichier de nom ".fl".

Remontez d'un niveau dans l'arborescence et essayez de supprimer le répertoire `other`.

10) Que se passe-t'il ?

Affichez le contenu du répertoire `other`. Utilisez l'option `-a`.

11) Comment arriver à supprimer le répertoire `other` ?

Placez vous dans le répertoire `play`.

12) Expliquez les notions de chemin absolu et chemin relatif et donnez un exemple de leur utilisation respective pour passer du répertoire `play` au répertoire d'accueil.

■ Commandes de Base:

- « `ls` »: lister le contenu d'un répertoire
- « `rm` »: supprimer un fichier
- « `cp` »: copier un fichier
- « `ln` »: créer un lien sur un fichier
- « `man` »: afficher l'aide d'une commande
- « `mv` »: déplacer un fichier
- « `cd` »: changer de répertoire
- « `mkdir` »: créer un répertoire
- « `rmdir` »: supprimer un répertoire
- « `pwd` »: afficher le répertoire courant
- « `cat` »: afficher le contenu d'un fichier
- « `file` »: afficher le type de contenu du fichier
- « `locate` »: localiser un fichier sur le disque

TP 3: Editeur vi, autorisation et droits d'accès

`touch` : création d'un nouveau fichier (sert normalement à mettre à jour des fichiers).

`cat fichier` : affiche le contenu de fichier permet aussi de concaténer des fichiers.

```
cat fichier1 fichier2 > fichier3
```

`more fichier` : affiche le contenu de fichier **page par page**.

Editeur vi

`vi` (prononcez vie-äie) est l'**éditeur de texte de base** sous Linux. Ou Unix

Pour lancer vi, c'est simple :

```
prompt> vi XF86Config
```

vi s'ouvre avec le fichier XF86Config dedans.

Vous pouvez **donc créer** un nouveau fichier simplement en tapant [**vi nom_du_fichier**].

Se déplacer dans le texte

Utiliser les quatre flèches, mais aussi les touches **h, j, k & l**.

Les modes de vi

2 modes : le mode "**Insert**" et le mode **normal**.

En mode normal vous pouvez pas insérer de texte dans le fichier, mais les touches du clavier sont autant de touches de commandes.

En mode "Insert", les touches de commandes (notamment les lettres !) se transforment en vraies lettres que vous pouvez insérer dans le fichier.

Lorsque vi s'ouvre, il est en mode normal.

Pour passer en mode Insert :

tapez i pour insérer du texte à l'endroit où se trouve le curseur,

tapez A pour ajouter du texte à la fin d'une ligne.

En mode Insert, vous pouvez taper du texte, effacer avec la touche [Suppr] ou [Bkspace]. **Pour quitter le mode Insert, tapez [Esc]**.

Les commandes

Après avoir quitté le mode Insert, ou avant d'y être entré, les touches du clavier correspondent à des commandes.

- ◆ **:q!** [Entrée] pour quitter sans sauver,
- ◆ **:w** [Entrée] pour enregistrer,
- ◆ **:wq** [Entrée] pour enregistrer et quitter,
- ◆ **x** efface le caractère qui se trouve sous le curseur,
- ◆ **dd** efface la ligne sur laquelle se trouve le curseur,
- ◆ **:u**[Entrée] permet d'annuler (ou :undo).

Insérer et remplacer du texte

Ces commandes vous placent toutes en mode "Insert". Pour en sortir, tapez la touche [Esc].

- ◆ **A** Ajouter du texte à la fin de la ligne
- ◆ **a** Ajouter du texte après le curseur
- ◆ **i** Insérer du texte avant le curseur
- **ou O** Créer une ligne vierge sous ou au dessus du curseur
- ◆ **R** Remplacer le texte à partir du curseur
- ◆ **n s** Supprime n caractères et passe en mode insertion

- ◆ **n <<** ou **n >>** Indente n lignes vers la gauche ou vers la droite.
- ◆ **r c** Remplacer le caractère sous le curseur par c

Effacer du texte

- ◆ **n dd** Efface n lignes à partir de la ligne courante (**dd** efface la ligne courante).
- ◆ **n x** Efface n caractères à partir du caractère courant (**x** efface juste le caractère courant).
- ◆ **D** Efface le reste de la ligne à partir du caractère courant.

- ◆ **n dw** ou **n db** Efface les n mots suivants ou précédents, à partir du mot suivant.
- ◆ **d/str** Efface tout depuis le curseur jusqu'à la première occurrence de str.
- ◆ **:g/^\$/d** Efface toutes les lignes vides

Commandes sur les fichiers

- ◆ **:nom_fichier** Ouvre le fichier nom_fichier.
- ◆ **:sh** Lance un shell ; lorsque vous tapez exit dans ce shell, vous revenez à vi.
- ◆ **:file nouv_nom_fichier** Change le nom du fichier courant, équivalent à "enregistrer sous" lorsque vous enregistrerez.
- ◆ **:x,yw nouv_nom_fichier** Enregistre les lignes x à y dans le fichier nouv_nom_fichier.
- ◆ **:x,yw>>fichier** Ajoute les lignes x à y à la fin du fichier fichier.
- ◆ **:e!** Recharge le contenu du fichier dans l'état où il se trouvait à l'ouverture.
- ◆ **:r fichier** insère le contenu de fichier à l'emplacement du curseur.

Mouvements du curseur

- ◆ **h, l, k, j** ou les 4 flèches Déplace le curseur d'un caractère vers la gauche, la droite, le haut, le bas.

- ◆ \$ Déplace le curseur en fin de ligne.

Exercice :

Créer un fichier essai contenant 4 phrases de votre choix ensuite utiliser les commandes cités ci-dessus pour les manipuler.

Autorisations ; modifier les droits d'un fichier

Seul le propriétaire d'un fichier peut modifier ses droits d'accès. Pour cela, il utilise la commande **chmod**.

chmod mode nom_fichier

mode indique de quelle façon les droits d'accès doivent être modifiés . Il se décompose en (qui) op permissions
 qui (optionnel) indique quelles classes sont concernées par la commande **chmod** et est composé de 1 ou plusieurs lettres parmi **u,g et o** . (user,group ou other)

si aucune classe n'est spécifiée toutes les classes sont concernées .

op peut être :

- + pour ajouter des droits d'accès
- pour enlever des droits d'accès



Exemple :

chmod ug-w fichier1

chmod utilise le codage linéaire , fondé sur l'association de valeurs numériques aux différentes permissions : **lecture : 4 , écriture : 2, exécution : 1, pas de permission : 0**
 chaque triplet se code par l'addition de 4,2,1 ou 0, pour un **rwX** il faudra ajouter 4+2+1=7

Droits	Valeur octale	Valeur binaire
---	0	000
--X	1	001
-W-	2	010
-WX	3	011
r--	4	100
r-X	5	101
rw-	6	110
rwX	7	111

Chmod 700 fichier 1 ⇔ chmod u+rwX fichier 1

Autre exemple

- chmod +r fichier.txt »: positionne l'accès en lecture

- « `chmod -r fichier.txt` »: enlève l'accès en lecture
- « **chmod o-x fichier** »: enlève l'accès en exécution pour les autres
- « `chmod go-x fichier` »: enlève l'accès en exécution pour les membres du groupe et les autres
- Mode absolu: convertir les permissions en bits
Ex: « **chmod 444 fichier** » ⇔ **Droits r--r--r --** »
« **chmod 421 fichier** » ⇔ **Droits r-- -w-r--** »

DEUXIEME PARTIE : DROITS D'ACCES AUX FICHIERS ET REPERTOIRES

Exercices 2 – Création des droits

1. Dans votre répertoire courant, créez un répertoire courant **essai_droit**, par défaut ce répertoire est à 755 (rwxr-xr-x),

Quelles sont les commandes (en notation symbolique et en base 8) pour lui donner les droits suivant (on suppose qu'après chaque commande on remet le répertoire à 755:

	propriétaire			groupe			les autres		
	droit en lecture	droit en écriture	droit d'accès	droit en lecture	droit en écriture	droit d'accès	droit en lecture	droit en écriture	droit d'accès
commande 1	oui	oui	oui	oui	non	oui	Non	non	oui
commande 2	oui	non	oui	non	oui	non	Non	non	oui
commande 3	non	oui	non	non	non	oui	Oui	non	non
commande 4	non	non	oui	oui	non	oui	Non	non	non

2. Créez un fichier **droit** dans le répertoire **essai_droit**, par défaut ce fichier est à 644 (rw-r--r--). En partant du répertoire courant, pour chaque commande de l'exercice précédent, essayez d'accéder au répertoire **essai_droit** (commande `cd`), de faire un `ls` dans **essai_droit** et de modifier le fichier avec un éditeur quelconque (`vi` par exemple).

L'éditeur vi

A la base de l'administration sous Unix on trouve **l'éditeur vi**. Cet éditeur de texte en mode terminal d'apparence très basique a une multitude d'avantages.

Le but des manipulations suivantes, est de vous familiariser avec les commandes de base de l'éditeur vi.

■ vi <nom du fichier a éditer>

Pour insérer du texte il suffit de déplacer **le curseur la ou on le désire d'appuyer** sur **Esc** puis sur **i** et on peut insérer du texte, à la fin de l'insertion on appuie sur **Esc**.

De même :

- on peut rajouter du texte avec **a**,
- supprimer un caractère avec **x**,
- supprimer un mot avec **dm** (delete word)
- ou encore remplacer un mot **cw** (change word).

Si l'on tape un nombre **x** avant la commande celle-ci sera répétée x fois.

Quand on **désire quitter** l'application il faut taper :

- ◆ **:q** Si **aucune modification** n'a été apportée
- ◆ **:q!** pour **ne pas enregistrer** les modifications en quittant,
- ◆ **:wq** pour quitter en enregistrant
- ◆ **:wq!** Pour quitter et forcer l'enregistrement du fichier.

Un recherche de mot se fait avec **/<mot recherché>**, pour répéter la dernière commande il suffit de taper **.**

Et pour intégrer un fichier il suffit de faire : **r <nom du fichier>**