

# TP n°3

## Architecture des systèmes informatiques INFO3 - S5

### UNIX : Commandes utilisateur

L'objectif de ce TP est de se familiariser avec les commandes de bases du système d'exploitation UNIX en vue d'une utilisation en tant qu'administrateur système.

## 1 Récapitulatif des commandes utiles

Lors de votre connexion, vous accédez à votre répertoire de travail et vous pouvez exécuter des commandes (programmes) qui seront analysées par l'interpréteur de commande de votre terminal (dans le cas des TP sous Linux dans l'école, le programme **bash**).

Toute commande UNIX respecte le format suivant :

```
nom_commande [-options] [-options] <argument_1> <argument_2> ...
```

Les résultats des commandes sont écrits sur la sortie standard (par défaut, le terminal) et en cas de besoin, les données sont lues sur l'entrée standard (par défaut, le clavier).

D'une manière générale, on peut dire que les commandes disent *quoi faire*, les options *comment le faire* et les arguments *sur quoi le faire*.

### 1.1 Commandes usuelles

Les principales commandes que vous devez savoir manipuler sont listées dans le tableau 1. Vous devez les connaître suffisamment bien pour les manipuler sans avoir à regarder systématiquement dans le **man**.

Certaines de ces commandes permettent une exécution récursive (ex : **cp**, **rm**). Tester ces commandes. à quoi sert l'option **-f** de certaines commandes ?

### 1.2 La commande *alias*

Plutôt que de taper de longues commandes, ou bien parce que vous préférez vous rappeler d'un nom plutôt que du vrai nom Unix, vous pouvez définir des *alias*. Pour ce faire, utilisez la commande **alias** comme suit :

```
alias md=mkdir
alias ls='ls --color'
```

Pour lister les *alias* actuellement configurés pour votre session, utiliser la commande **alias** sans paramètres. Pour annuler les effets d'un alias utilisez la commande **unalias nom\_alias**.

**Attention, lorsque vous voulez tester les effets d'une commande, n'oubliez pas de supprimer l'alias potentiellement configuré sur votre système !**

<code>pwd</code>	affiche le répertoire courant
<code>whoami</code>	affiche le nom de l'utilisateur
<code>who am i</code>	idem avec insertion du numéro de terminal
<code>tty</code>	affiche le numéro de terminal
<code>who</code>	indique qui est connecté sur la machine
<code>echo [-n]</code>	affiche l'argument [sans retour chariot]
<code>ls [-algtui]</code>	liste le contenu du répertoire
<code>cat</code>	affiche le contenu de l'argument
<code>more nom_fichier</code>	affiche <code>nom_fichier</code> , formaté à l'écran
<code>grep [-v] "chaîne" nom_fichier</code>	chercher l'occurrence [la non occurrence] de <code>chaîne</code> dans chaque ligne de <code>nom_fichier</code>
<code>cp</code>	copie un fichier
<code>mv</code>	changement de nom
<code>rm</code>	destruction du fichier
<code>wc</code>	compte le nombre de lignes, de mots et de caractères
<code>du</code>	affiche la taille occupée par un répertoire
<code>mkdir</code>	crée un répertoire
<code>rmdir</code>	destruction d'un répertoire vide
<code>su</code>	changement d'utilisateur

TABLE 1 – Principales commandes utilisateur.

## 2 Traitement des fichiers

### 2.1 Édition

Il est particulièrement intéressant en tant qu'administrateur système de pouvoir manipuler des fichiers en ligne de commande. Généralement, lorsque l'on se connecte à distance sur un serveur, il n'y a pas d'interface graphique disponible. Sous UNIX, il existe de nombreux éditeurs de textes dans un terminal, nous allons manipuler les plus utilisés.

#### 2.1.1 *vi*

L'éditeur *vi* est l'éditeur standard du système d'exploitation UNIX. Pour utiliser cet éditeur, vous devez exécuter la commande `vi` dans votre terminal suivi du nom du fichier que vous voulez éditer. Si le fichier n'existe pas, *vi* le créera automatiquement.

*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.

Selon le mode dans lequel il se trouve, l'éditeur interprète différemment les caractères qu'il reçoit du clavier. Lorsque vous le lancez, *vi* se place initialement dans le mode de commandes.

**Commandes de base** Pour passer du mode entrée au mode commande, il suffit d'appuyer sur la touche `[ESC]`. Pour passer du mode commande au mode entrée, plusieurs touches sont disponibles :

- `'i'` : insérer du texte à l'endroit du curseur (mode INSERT)
- `'a'` : insérer du texte après le curseur (mode 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

**Se déplacer dans *vi*** On peut utiliser les touches fléchées pour se déplacer, mais également 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.

### 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

### 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'ait pas été modifié depuis son chargement
- 'ZZ' : sauver 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).

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, sauve sous le même nom, un autre nom, ...).

Il existe beaucoup d'autres options sur cette commande, explorez le **man** et testez les plus utiles.

#### 2.1.2 *emacs, xemacs*

Essayer rapidement les autres éditeurs, et plus particulièrement **emacs** ou **xemacs**.

## 2.2 Commandes de manipulation de fichiers

### 2.2.1 Visualiser le contenu d'un fichier

Il existe plusieurs commandes permettant d'afficher le contenu d'un fichier :

- **cat** permet de visualiser le contenu d'un fichier en entier
- **more** permet de visualiser le contenu d'un fichier en entier, page par page
- **head** permet de visualiser le début d'un fichier
- **tail** permet de visualiser la fin d'un fichier

Examiner les options de ces commandes et plus particulièrement l'option `-f` de la commande `tail` (utilisez pour cela 2 terminaux, un exécutant la commande `tail` et l'autre ajoutant des éléments dans le fichier).

### 2.2.2 Découpage d'un fichier

**Horizontal** Commandes `head`, `tail` et `split`.

à l'aide de ces commandes :

- Extraire les 5 premières lignes du fichier `/etc/passwd`
- Extraire les 3 dernières lignes du fichier `/etc/group`
- Expliquer `tail -n +5 /etc/group`
- Proposer une commande unique extrayant les lignes 7 à 10 du fichier `/etc/group`
- Que fait `split -5 ls-fic` ?

**Vertical** Commande `cut`

à l'aide de ces commandes :

- Essayer `cut -f5 -d: /etc/passwd | head`. Expliquer.
- Essayer `cut -c2-9 /etc/passwd | head`. Expliquer.
- Extraire les noms de groupe du fichier `/etc/group`.
- Proposer d'autres exemples (idée : à partir des champs de `ls -l`)

### 2.2.3 Modification d'un fichier

**Tri d'un fichier** à partir de la commande `sort` :

- Exécuter la commande `sort -r`. Pour indication, le caractère de fin de fichier s'écrit au clavier en tapant les touches `[CTRL]` et `d` en même temps.
- Exécuter la commande `cat /etc/group | sort -r`. Qu'observez-vous ? Expliquez ce qu'il s'est passé avec un schéma indiquant les processus en jeu et les canaux d'entrée/sortie utilisés.

En combinaison avec les commandes vues précédemment :

- Proposer une commande extrayant, par ordre alphabétique, les noms d'utilisateur du fichier `/etc/passwd`.
- Proposer une commande triant la sortie de `ls -l` par ordre alphabétique des noms de fichier.

**Suppression des lignes identiques d'un fichier** Commande `uniq`.

Essayer et expliquer cette succession de commandes :

```
cat /etc/group /etc/group > groupDouble
sort -o groupDoubleTriés groupDouble
uniq groupDoubleTriés
```

**Comparaison de deux fichiers** à quoi servent les commandes `comm` et `diff` ?

Fabriquez deux fichiers à partir du fichier `/etc/passwd`

- `f1` contenant les 10 premiers noms d'utilisateurs par ordre alphabétique.
- `f2` contenant les 5<sup>ème</sup> à 15<sup>ème</sup> noms d'utilisateur par ordre alphabétique.

à partir de ces deux fichiers :

- Proposer une commande extrayant les lignes communes aux deux fichiers.

- Comparer ensuite les commandes `comm f1 f2` et `comm f1 f2r` (`f2r` obtenu avec `sort -r f2 > f2r`). Expliquer.
- Que fait l'option `-r` de `diff` ?
- Essayer l'option `-u` de `diff` sur deux versions d'un même fichier de code source (en *C* par exemple) ? Essayer ensuite de transformer le fichier le plus ancien avec la commande `patch`.

### Transformation de caractères Commande `tr`.

Essayer la commande `tr a-z A-Z < /etc/group`. Commenter.

### Programme générique de transformation de fichiers Commande `sed`.

Examiner le fonctionnement de la commande `sed`.

Que permet-elle de faire ? Donner un exemple de commande remplaçant toute occurrence d'un mot dans un fichier par un autre mot.

## 2.3 Compression et archivage

### 2.3.1 Compression

Les fichiers comprimés utilisent moins d'espace disque et se téléchargent plus rapidement que les grands fichiers non comprimés. Vous pouvez compresser les fichiers Linux à l'aide de l'instrument de compression open-source *gzip* ou *zip*, qui est reconnu par la plupart des systèmes d'exploitation.

Par convention, les fichiers comprimés se voient attribuer l'extension `.gz`. La commande `gzip` crée un fichier comprimé terminant par `.gz`. `gunzip` extrait les fichiers comprimés et efface le fichier `.gz`.

Pour compresser un fichier, entrez la commande suivante à l'invite du shell :

```
gzip nom_fichier
```

Le fichier sera comprimé et sauvegardé comme `nom_fichier.gz`.

Pour décompresser un fichier comprimé, tapez :

```
gunzip nom_fichier
```

Le fichier `nom_fichier.gz` est effacé et remplacé par `nom_fichier`.

Si vous échangez des fichiers avec des utilisateurs non LINUX, vous devriez utiliser *zip* pour éviter les problèmes de compatibilité. Linux peut facilement ouvrir des fichiers *zip* ou *gzip*, mais les systèmes d'exploitation non-Linux pourraient avoir des problèmes avec les *gzip*.

Essayez de compresser des fichier avec l'utilitaire *zip*.

Pour extraire le contenu d'un fichier *zip*, entrez :

```
unzip nom_fichier
```

Vous pouvez compresser plusieurs fichiers en même temps avec *zip* ou *gzip*. énumérez les fichiers en les séparant par un espace.

```
gzip nom_fichier fic1 fic2 fic3 /user/work/school
```

La commande ci-dessus compresse les fichiers `fic1`, `fic2`, `fic3`, et le contenu du répertoire `/user/work/school` pour les placer dans `nom_fichier.gz`.

Il existe d'autres utilitaires de compression, offrant des fonctionnalités de compression plus évoluées (meilleurs taux, ...). Examinez les commandes `bzip` et `bzip2`. Que contient un fichier avec une extension `bz2` ? Comment créer un tel fichier ?

Examinez également les commandes `zcat`, `bzcat`, `zless` et `bzless`.

### 2.3.2 Archivage

La commande `tar` permet de regrouper un ensemble de fichiers en un seul (une archive). Ce fichier peut ensuite être compresser dans différents formats. Par exemple, pour créer une archive en compressant avec la commande `gzip` utiliser :

```
tar -cvzf nom_archive.tar.gz liste_fichiers
```

Pour décompression avec `gzip` :

```
tar -xvzf nom_archive.tar.gz
```

Il est également possible à l'aide de cette commande de compresser ou décompresser avec `bzip2`, il suffit de remplacer le `-z` par `-j` et avec `compressFP` le `-z\` par `-Z`.

Examinez le détail de la commande `tar` et testez plusieurs création d'archives ainsi que leur décompression. *Cette commande vous sera très utile pour rendre vos différents TP sous UNIX...*