

# Module administration système et réseau GNU/Linux

## Les commandes de base



## Sommaire

Pré-requis.....	2
Objectifs.....	2
A. Introduction.....	2
1. L'interpréteur de commande.....	2
2. L'invite de commande.....	2
3. Utiliser le shell.....	3
4. Chaîner les commandes.....	3
B. Gestion des fichiers et des répertoires.....	4
1. Se déplacer.....	4
a. Avec le chemin absolu.....	4
b. Avec le chemin relatif.....	4
2. Lister.....	5
3. Créer.....	6
4. Supprimer.....	6
5. Copier.....	7
6. Déplacer ou renommer.....	7
7. Créer un lien symbolique.....	8
8. Afficher le contenu.....	8
a. La commande cat.....	8
b. La commande tail.....	8
c. La commande grep.....	9
C. Autres commandes de base.....	10
1. Identifier les utilisateurs connectés.....	10
2. Changer son mot de passe.....	10
3. Obtenir l'historique des commandes saisies.....	10

## Pré-requis

- Avoir une VM avec une distribution GNU/Linux

## Objectifs

- se déplacer dans l'arborescence du système ;
- savoir gérer les fichiers et les répertoires en ligne de commande,
- Afficher les informations relatives aux utilisateurs connectés.
- Changer son mot de passe
- Afficher le contenu d'un fichier texte

## A. Introduction

### 1. L'interpréteur de commande

Un **interpréteur de commande** est un programme appelé **shell**.

L'interpréteur de commande permet d'exécuter des instructions que vous saisissez au clavier ou au sein d'un script et vous en retourne l'accès.

Deux méthodes d'accès au **shell** sont possibles sous GNU/Linux :

- Le **mode console** qui affiche un shell unique en plein écran, c'est l'interface utilisateur de base du système d'exploitation. Sous GNU/Linux les consoles sont en général au nombre de six par défaut;
- Le **mode terminal** qui émule une console et qui affiche en général le shell dans une fenêtre.

Sous GNU/Linux Il existe différents shells : sh (Bourne Shell), ksh (Korn Shell), Bash (Bourne Again Shell)

Le shell fournit par défaut avec les distributions GNU/Linux répandues est Bash.

### 2. L'invite de commande

Le **shell** attend des entrées au clavier sur une ligne appelée l'**invite de commande** ou **prompt**.

Le prompt fournit en général des informations sur le terminal et votre position dans le système de fichiers.

**Premier exemple :**

```
stag@linuxserver:/srv/Documents$
```

Décomposons cette ligne :

- stag : c'est le nom de connexion (login) actuellement connecté au terminal.
- linuxserver : c'est le nom d'hôte (hostname) de la machine raccordé au terminal.
- /srv/Documents : c'est la position actuelle du shell dans l'arborescence.
- « \$ » : c'est la terminaison standard du bash pour un utilisateur sans pouvoir.

**Deuxième exemple :**

```
stag@linuxserver:~$
```

- « ~ » : indique que l'utilisateur est dans son répertoire personnel qui est par défaut `/home/login` (ici `/home/stag`).

**Le cas de l'utilisateur « root »** : à la place du symbole « \$ », la terminaison du bash sera le symbole « # »

```
root@linuxserver:~#
```

### 3. Utiliser le shell

Le clavier s'utilise comme d'habitude. On saisit une commande et on la valide par entrée. Les flèches droite et gauche du clavier permettent de se déplacer sur la ligne de commande.

Les flèches haut et bas permettent de naviguer dans l'historique des commandes saisies précédemment.

Raccourcis-clavier utiles :

- **[CTRL] D** : permet de quitter le terminal.
- **[CTRL] L** : efface le contenu du terminal.
- **[CTRL] U** : efface la ligne de la position où se situe le curseur sur la ligne de commande jusqu'au début.
- **[CTRL] K** : efface la ligne de la position où se situe le curseur sur la ligne de commande jusqu'à la fin.
- **[CTRL] A** : Amène le curseur au début de la ligne.
- **[CTRL] E** : Amène le curseur en fin de ligne.

L'auto-complétion est également d'une aide précieuse.

- Elle permet de compléter les commandes, les chemins saisis ou les noms de fichiers.
- Un appui sur la **touche TAB** complète la saisie dans le cas d'une seule solution.
- Sinon, il faudra faire un deuxième appui pour obtenir la liste des possibilités.

Si un double appui sur la touche TAB ne provoque aucune réaction de la part du système, c'est qu'il n'existe aucune solution à la complétion en cours.

### 4. Chaîner les commandes

Vous pouvez exécuter plusieurs commandes sur une seule ligne, les unes après les autres. Pour cela il suffit de les séparer avec un point-virgule.

```
$ date;pwd;cal -m
jeu. janv. 19 21:15:14 CET 2017
/home/stag
    mai 2018
lu ma me je ve sa di
    1  2  3  4  5  6
 7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31
```

## B. Gestion des fichiers et des répertoires

### 1. Se déplacer

Nous avons déjà vu l'arborescence d'un système Gnu/Linux à partir de sa racine :

```
/
  bin/
  boot/
  cdrom/
  dev/
  etc/
  home/
  lib/
  ...
```

Pour se déplacer dans l'arborescence, nous utilisons la commande « **cd** » (change directory). Nous pouvons utiliser deux approches pour se déplacer avec la commande « **cd** ».

#### a. Avec le chemin absolu

C'est-à-dire le **chemin en partant de la racine** « / » :

```
$ cd /etc/network/
```

#### b. Avec le chemin relatif

Il faut pour cela utiliser deux entrées particulières de répertoires :

- Le point « . » représente le répertoire courant.
- Les doubles points « .. » représentent le répertoire parent.

#### Premier exemple :

Le répertoire « /srv » contient le répertoire « data01 » dans lequel je me situe. Si je veux remonter dans le répertoire parent, il me suffit de saisir :

```
$ pwd
/srv/data01
$ cd ..
$ pwd
/srv
```

#### Deuxième exemple :

Le répertoire « srv » contient deux répertoires : « data01 » et « data02 ». Je me situe dans le répertoire data01. Si je veux me déplacer dans le répertoire data02, il me suffit de saisir :

```
$ pwd
/srv/data01
$ cd ../data02
$pwd
/srv/data02
```

## 2. Lister

Pour cela nous utiliserons la commande : « **ls** » :

- **ls** sans option va lister le contenu d'un répertoire.
- **ls -l** : pareil que ci-dessus mais affichera plus de détails et plus particulièrement les droits.
- **ls -a** : listera en plus les fichiers cachés. Sous Linux, pour cacher un fichier ou un répertoire on nomme ces dernier en commençant par un point.

On peut combiner les options : **ls -la** par exemple. D'autres options existent et sont très bien expliquées dans « man ls ».

La commande « ls -l » étant souvent utilisée par l'administrateur, il existe un alias qui est la commande « ll ». Cet alias est généralement déjà paramétré dans le fichier caché « .bashrc » (un fichier caché commence toujours par un point) présent dans chaque répertoire personnel des utilisateurs.

« .bashrc » est un fichier qui permet de personnaliser le shell bash pour un utilisateur : vous pouvez créer des alias, personnaliser l'apparence du prompt, etc. Vous trouverez plein de littérature sur le net le concernant.

## 3. Créer

### Créer des fichiers

`$ touch fichier`

La fonction principale de « touch » n'est pas forcément de créer des fichiers mais de modifier l'horodatage d'un fichier (peut être utile dans le cadre de sauvegarde). Faites de nouveau « touch fichier » puis « ls -l », Vous remarquerez que l'horodatage a changé.

`$ echo > fichier`

Permet la même chose mais attention, à ne pas faire sur un fichier ayant du contenu sinon cela effacera ce dernier.

Avec du texte :

`$ echo « bonjour à tous » > fichier`

### Créer des répertoires

`$ mkdir rep01`

Créer plusieurs répertoires de même niveau

`$ mkdir rep01 rep02 rep03`

Par défaut la commande ne crée pas d'arborescence. **Pour pouvoir créer un répertoire et des sous-répertoires** en une seule ligne de commande, il faudra utiliser l'option « -p ».

`$ mkdir -p rep02/rep021/rep0211`

## 4. Supprimer

### Pour supprimer un fichier :

`$ rm fichier`

### Supprimer un répertoire

`$ rmdir rep01 rep02 rep03`

Si le répertoire à supprimer contient une arborescence, il faut utiliser la commande « rm » avec le paramètre « -r » pour récursif :

`$ rm -r rep01`

En général rm demande une confirmation de suppression à l'utilisateur. On peut ajouter l'option -i pour demander une confirmation de suppression,

On peut saisir « rm -rf répertoire » pour supprimer un répertoire et son contenu sans confirmation. Mais attention « rm -rf » signifie supprimer sans passer par la corbeille.

À moins de vouloir libérer de la place, on préférera faire un « mv répertoire /tmp » surtout si vous lancez cette commande en tant que l'utilisateur root. Bien des administrateurs systèmes se sont fait de sacrés frayeurs avec « rm -rf ».

## 5. Copier

La commande « cp » :

- copie un fichier vers un autre fichier ou vers un répertoire.
- copie un ou plusieurs fichiers vers un répertoire.

Copie du fichier fic01 vers fic02

```
$ cp fic01 fic02
```

Copie de 3 fichiers vers un répertoire :

```
$ cp fic01 fic02 fic03 Repertoire/
```

### Copier des répertoires

Copier le répertoire vide rep01 vers rep02 :

```
$ cp rep01 rep02
```

Utilisation du paramètre « -r » pour copier le répertoire rep01 contenant des fichiers et des répertoires vers rep02 :

```
$ cp -r rep01 rep02
```

## 6. Déplacer ou renommer

```
$ mv [Option] Source Destination(répertoire)
```

```
$ mv [Option] Répertoire Source
```

```
$ mv fichier_source fichier_cible # déplacer fichier_source dans fichier_cible
```

```
$ mv fichier1 fichier2 # renomme le fichier fichier1 en fichier2  
# Attention si fichier2 existe, son contenu sera écrasé et  
# remplacé par celui de fichier1
```

Par précaution, on utilise l'option -i qui permet d'être averti par le système avant l'écrasement du fichier destination si celui-ci existe.

```
$ mv -i fic1 fic2 # demande la confirmation avant d'effacer la destination
```

```
$ mv rep1/fic1 rep2/fic2 # déplace le fichier fic1 situé dans le répertoire rep1  
# sous le nouveau nom fic2 situé dans le répertoire  
rep2
```

```
$ mv rep1 rep2 # déplace le répertoire rep1 dans le répertoire rep2  
# si rep2 n'existe pas, renomme rep1 en rep2
```

## 7. Créer un lien symbolique

On utilise la la commande « ln » : *ln -s fichier\_source lien*

Pour créer le lien fic01lnk vers le fichier fic01 :

```
$ ln -s fic01 fic01lnk
```

Si vous faites ensuite un *ls -l* :

```
-rw-r--r--  1 stag stag 0 sept. 25 15:10 fic01
-rwxrwxrwx  1 stag stag 5 sept. 25 15:10 fic01lnk -> fic01
```

Vous remarquerez que le fichier fic01lnk pointe vers fic01.

Attention : Un fichier lien est fichier spécial. En effet les droits indiqués sont ceux du fichier spécial. Cela ne signifie pas que tous les utilisateurs puissent modifier le fichier ou dossier pointés.

## 8. Afficher le contenu

### a. La commande cat

La commande cat affiche le contenu d'un fichier ou de plusieurs fichiers concaténés sur la sortie standard (l'écran)

```
$ cat fichier
```

Elle permet d'afficher le contenu du fichier à l'écran; cette commande n'est vraiment intéressante que si le fichier est court.

Affiche à l'écran les trois fichiers à la suite dans l'ordre indiqué :

```
$ cat fic1 fic2 fic3
```

Numérote les lignes du fichier à l'affichage :

```
$ cat -n fichier
```

Fait apparaître les caractères non lisibles à l'écran :

```
$ cat -v fichier
```

### b. La commande tail

```
$ tail fichier
```

Elle permet d'éditer un fichier en commençant par la fin. Ceci est très utile quand on possède un fichier très long et que l'on veut récupérer un résultat qui se trouve à la fin du texte.

Pour obtenir les douze dernières lignes d'un fichier:

```
$ tail -12 fichier
```

Pour avoir en plus le titre du fichier:

```
$ tail -12 -v fichier
```

Sur les serveurs vous devrez savoir **suivre un fichier de log en direct**. La commande tail peut vous y aider. Par exemple vous venez de configurer un serveur web Apache et vous

voulez tracer les connexions. Utilisez l'option -f :

```
# tail -f /var/log/apache2/access.log
```

Et pour le debug :

```
# tail -f /var/log/apache2/*.log
```

Qui vous permettra de suivre le fichier access.log et le fichier error.log en direct.

### c. La commande grep

La commande **grep** permet de rechercher une chaîne de caractères dans un fichier. Les options sont les suivantes :

- **-v** affiche les lignes ne contenant pas la chaîne
- **-c** compte le nombre de lignes contenant la chaîne
- **-n** chaque ligne contenant la chaîne est numérotée
- **-x** ligne correspondant exactement à la chaîne
- **-l** affiche le nom des fichiers qui contiennent la chaîne

Exemple avec le fichier **carnet-adresse** contenant :

```
marcel:13:0466342233:Gardagnes  
myriam:30:0434214452:Nimes  
olivier:29:0298333242:Brest  
yvonne:92:013344433:Palaiseau
```

```
$ grep Brest carnet-adresse
```

Permet d'obtenir les lignes contenant la chaîne de caractère **Brest**, soit :

```
olivier:29:0298333242:Brest
```

## C. Autres commandes de base

### 1. Identifier les utilisateurs connectés

La commande **id** affiche le nom de l'utilisateur courant et ses groupes ou ceux d'un utilisateur, si le login de celui-ci est fourni comme argument.

```
$ id stag
uid=1000(stag) gid=1000(stag) groups=1000(stag),...
```

Les options -g, -G, -n et -u affichent respectivement le GID du groupe principal, les GID des groupes secondaires, les noms au lieu des identifiants numériques et l'UID de l'utilisateur.

La commande **whoami** affiche le login de l'utilisateur courant :

```
$ whoami
stag
```

La commande **who** seule affiche le nom des utilisateurs connectés :

```
$ who
stag      tty1    2017-09-15 10:30
root      pts/0   2017-09-15 10:31
```

Linux étant multi-utilisateurs, il est probable que plusieurs sessions soient ouvertes sur la même station, que ce soit physiquement ou à travers le réseau. Il est intéressant de savoir quels utilisateurs sont connectés.

### 2. Changer son mot de passe

La commande **passwd** permet de changer son mot de passe :

```
$ passwd
Changing password for stag
(current) UNIX password:
Enter new UNIX password:
Retype new UNIX password:
passwd : le mot de passe a été mis à jour avec succès
```

Vous ne verrez pas les caractères saisis. Si vous vous trompez :

- soit vous effacer les caractères saisis sur la ligne en cours avec le raccourci clavier **CTRL+U**
- soit vous annulez totalement la commande avec le raccourci clavier **CTRL+C**

### 3. Obtenir l'historique des commandes saisies

On utilise la commande « **history** ». Elle permet d'obtenir l'historique des commandes saisies même si l'ordinateur a été rallumé depuis.

Cette commande va parcourir le fichier `.bash_history` se situant dans le home directory de

chaque utilisateur. Ex : `/home/stag/.bash_history`

Note : ce fichier contient en général les 500 dernières commandes (nombre limite que l'on peut paramétrer).