**Linux commandes essentielles**

Le manuel d’utilisation est un outil de référence, là où l'on peut trouver la réponse à TOUTES ses questions

Pour information, il est courant de voir un linuxien dire « RTFM » aux débutants qui posent des questions qu'il estime « simples ». RTFM est l'abréviation de « Read the fucking manual! »

**Syntaxe des commandes**

Une commande est constituée d'un mot et ne contient aucun espace :

commande

**date** 🡪 Affiche la date

**ls** 🡪 c’est l'abréviation de « list », qui signifie « **lister les fichiers et dossiers du répertoire actuel** ».

**Les paramètres**

La commande et les paramètres sont séparés par un espace

Les paramètres les plus courants sont constitués d'une seule lettre précédée d'un tiret. Par exemple :

commande -d

Si on doit donner plusieurs paramètres, on peut faire comme ceci :

commande -d -a -h ou commande -dah

Les paramètres constitués de plusieurs lettres sont précédés de deux tirets, comme ceci :

commande --parametre

Cette fois, pas le choix : si vous voulez mettre plusieurs paramètres longs, il faudra ajouter un espace entre chacun d'eux :

commande --parametre1 --parametre2

Paramètres à la commande ls :

**-a**: afficher tous les fichiers et dossiers cachés

**-l**: liste détaillée

**who** : liste tous les utilisateurs connectés actuellement.

**passwd** : pour changer son mot de passe

**clear** : pour effacer l’écran

**pwd** : affiche le chemin absolu du répertoire actuel, c’est l'abréviation de « Print Working Directory »

**which** : Permet de savoir dans quel dossier se trouve une commande

Elle prend un paramètre : le nom de la commande dont vous voulez connaître l'emplacement.

**man** : afin d’accéder au manuel d’aide d’une commande, ex :

man who

Taper q pour sortir de l'aide.

**Commandes pour manipuler des fichiers**

**cd** : Pour se déplacer dans un autre répertoire

**cd** ne prend pas plein de paramètres mais juste un seul : le nom du dossier dans lequel vous souhaitez aller.

Si on veut aller à la racine, il suffit de taper cd /

Après avoir tapé cd /, on se retrouve à la racine. L'invite de commandes a changé et le ~a été remplacé par un/

Pour revenir au dossier précédent on tape : cd ..

Si on avait voulu reculer de deux dossiers parents, on aurait écrit : cd ../..

Il y a en fait deux façons de changer de dossier : en indiquant un **chemin relatif**, ou en indiquant un **chemin absolu**.

Les chemins relatifs

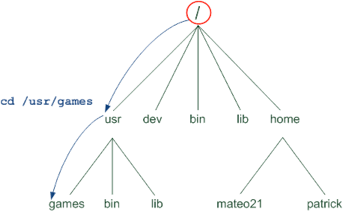
Un chemin relatif est un chemin qui dépend du dossier dans lequel vous vous trouvez.

Si tu as un sous-dossier villes dans un dossier pays et que tu te trouves dans le dossier pays, tu n'auras qu'à taper : cd villes

Les chemins absolus

Contrairement aux chemins relatifs, les chemins absolus fonctionnent quel que soit le dossier dans lequel on se trouve.

Un chemin absolu est facile à reconnaître : il commence toujours par la racine (/). Vous devez ensuite faire la liste des dossiers dans lesquels vous voulez entrer. Par exemple, supposons que je sois dans/home/mateo21 et que je souhaite aller dans/usr/games. Avec un chemin absolu :

cd /usr/games

Si on avait voulu faire la même chose à coup de chemin relatif, il aurait fallu écrire :

cd ../../usr/games/

Ce qui signifie « **reviens en arrière (donc dans**/home**) puis reviens en arrière (donc dans**/**), puis va en avant dans** usr**, puis va en avant dans** games »

Retour au répertoire home :

En utilisant l'alias~

cd ~ ou uniquement cd

**mkdir** : Permet de créer un dossier

mkdir mondossier

On peut créer deux dossiers (ou plus !) en même temps en les séparant là aussi par des espaces :

Il y a un paramètre utile avec mkdir :-p. Il sert à créer tous les dossiers intermédiaires. Par exemple :

mkdir -p animaux/vertebres/chat

Va créer le dossier animaux, puis à l'intérieur le sous-dossier vertebres, puis à l'intérieur encore le sous-dossier chat !

**rmdir** : Permet de supprimer un dossier vide

À connaître : la combinaison de touches Ctrl + C est utilisable dans la plupart des programmes console pour demander leur arrêt. C'est un peu l'équivalent du Alt + F4 de Windows.

**touch** : permet de créer un fichier

La commande touch est à la base faite pour modifier la date de dernière modification d'un fichier.

L'intérêt de touch pour nous dans ce chapitre, c'est que si le fichier n'existe pas, il sera créé !

touch pays // Va créer le fichier pays

On peut créer plusieurs fichiers en une seule commande. Il vous suffit de les lister l'un après l'autre, séparés par des espaces.

touch pays villes

**cp** : copier un fichier

On peut copier plusieurs fichiers à la fois, et même des dossiers !

cp capitales France

Là il va copier le fichier capitales dans le fichier France, si France n'existe pas, il va le créer.

Copier dans un autre dossier

Avec cette commande par exemple :

cp villes france

Là ça va copier le fichier villes dans le dossier france qui existe déjà.

Si dans le dossier france il y a déjà un fichier nommé villes, ce dernier sera écrasé.

Si vous voulez copier un fichier dans un dossier sous un autre nom on fait :

cp villes france/cities

Avec l'option-R (un « R » majuscule !), vous pouvez copier un dossier, ainsi que tous les sous-dossiers et fichiers qu'il contient !

cp -R animaux autresanimaux

cela aura pour effet de copier animaux ainsi que tous ses sous-dossiers sous le nom autresanimaux mais toujours dans le même dossier.

Utiliser le joker\*

Le symbole \* est appelé *joker*, ou encore **wildcard** en anglais sous Linux.  
Il vous permet de copier par exemple tous les fichiers image.jpg dans un sous-dossier :

cp \*.jpg mondossier/

Vous pouvez aussi vous en servir pour copier tous les fichiers dont le nom commence par « so » :

cp so\* mondossier/

**mv** : déplacer un fichier

Elle a en fait deux utilités :

* + - * déplacer un fichier (ou un dossier) ;
      * renommer un fichier (ou un dossier).

La commande mv s'utilise pratiquement comme cp :

mv fichierbidon mondossier/

Mais au lieu de copier le ficher, elle va le déplacer.

Vous pouvez aussi utiliser les jokers :

mv \*.jpg mondossier/

Renommer un fichier

La commande mv permet renommer un fichier. En effet, il n'existe pas de commande spéciale pour renommer un fichier en console sous Linux, c'est la commande mv qui est utilisée pour ça.

Par exemple :

mv capitale londres

renommera capitale en londres. Après cette commande, capitale n'existe plus, il a été renommé.

Déplacer et renommer un fichier à la fois.

Vous pouvez aussi déplacer capitale dans mondossier tout en lui affectant un nouveau nom :

mv capitale mondossier/londres

**rm** : supprimer des fichiers et dossiers même non vides

Il n'existe pas de corbeille dans la console de Linux : le fichier est directement supprimé sans possibilité de récupération !

La commande rm (pour *ReMove*, « supprimer » en anglais) peut supprimer un fichier, plusieurs fichiers, des dossiers, voire même votre ordinateur entier si vous le voulez.

Il faut donc l'utiliser avec précaution.

rm nantes

Vous pouvez aussi supprimer plusieurs fichiers en séparant leurs noms par des espaces :

rm fichierbidon fichiercopie

-i : demander confirmation

-f : forcer la suppression, quoi qu'il arrive

-r : supprimer un dossier et son contenu

Le paramètre -r peut être utilisé pour supprimer un dossier (au lieu d'un fichier) ainsi que tout ce qu'il contient : fichiers et dossiers !

rm et le joker de la mort

rm -rf /\*

En clair, cette commande supprime tout votre disque dur depuis la racine, sous-dossiers compris, et ne demande aucune confirmation.

rm -rf \* : supprime tous les fichiers et sous-dossiers du dossier dans lequel je me trouve.

**cat** : affiche tout le contenu d'un fichier

le paramètre -n permet d'afficher les numéros de ligne :

cat -n syslog

**less : affiche le fichier page par page**

**L'historique des commandes**

On peut le voir en tapant sur la touche de la flèche du haut ou bien avec la commande history

On peut rechercher une commande tapée avec Ctrl + R

**Droits d’accès aux fichiers :**

**chmod** : pour modifier les droits d’un fichier

chmod 777 voyage.txt 🡪 va donner tous les droits à tout le monde.

**sudo** : pour taper une commande en mode root

**su** : pour passer en mode root

**chown** : changer le propriétaire d’un fichier

**Gestion des processus**

**ps** : Permet de voir la liste des processus en cours

**kill** : pour tuer un process : kill + PID du process

**Outils Linux**

**find** : permet de rechercher un fichier en spécifiant des critères.

Syntaxe : find chemin(s) critère(s) action

Le seul critère obligatoire est le nom du fichier

Ex : find logo.png

En précisant le chemin

Ex : find /var/log –name syslog

**grep** : permet de rechercher dans un fichier les lignes correspondant à un filtre

Syntaxe : grep option(s) filtre fichier(s)

EX je veux chercher toutes les lignes contenant le mot villes dans le fichier France

grep villes France

Option c

grep -c villes France 🡪 va afficher le nombre de lignes.

Option i : pour ignorer la casse.

**sort** : pour trier les lignes d’un fichier.

Syntaxe : Sort NomDuFichier

**wc** : compter le nombre de ligne

wc NomDuFichier

**Commandes de bases Vim**

**i** 🡪 pour insérer du texte

**echap** 🡪 pour passer en mode commande, où l’on pourra taper ces commandes :

**y** 🡪 copie la ligne

**p** 🡪 colle la ligne

**dd** 🡪 supprime toute la ligne

**:x** 🡪 enregistre le fichier

**:q** 🡪 quitte un fichier non modifié

**:q!** 🡪 quitte un fichier modifié sans enregistrer les modifications

**Installation**

**apt-get install** … : pour installer un paquet

**apt-get update** : pour mettre à jour le cache des paquets

**apt-get upgrade** : pour mettre à jour tous les paquets

**apt-get remove**  … : pour supprimer un paquet

**Réseau**

**ssh login@ip -->**Se connecter en SSH

**ifconfig 🡪** Récupérer l’ip :

**nmtui** 🡪 pour configurer la connexion

**systemctl** start nginx 🡪 démarre nginx

ping

traceroute