Commande Linux

[1. Commande de base Linux : 3](#_Toc496086680)

[a) Navigation dans les fichiers : 3](#_Toc496086681)

[ls : Permet de lister un répertoire 3](#_Toc496086682)

[cd : Permet de se promener dans les répertoires. 3](#_Toc496086683)

[pwd : Affiche le répertoire de travail 3](#_Toc496086684)

[ln : Crée un lien (physique ou symbolique) vers un fichier (ou un répertoire) 4](#_Toc496086685)

[locate : Permet de chercher des fichiers rapidement 4](#_Toc496086686)

[find : Permet de chercher des fichiers et éventuellement d'exécuter des commandes 5](#_Toc496086687)

[grep : Recherche une chaîne de caractères dans des fichiers. 5](#_Toc496086688)

[cat : Affiche le contenu d'un fichier. 6](#_Toc496086689)

[more : Affiche un fichier page par page 6](#_Toc496086690)

[b) Manipulation de fichier : 6](#_Toc496086691)

[mv : Permet de déplacer ou renommer des fichiers et des répertoires 6](#_Toc496086692)

[cp : Permet de copier des fichiers ou des répertoires 7](#_Toc496086693)

[rm : Permet d'effacer des fichiers 7](#_Toc496086694)

[c) Manipulation de dossier : 7](#_Toc496086695)

[mkdir : Crée un répertoire vide 7](#_Toc496086696)

[rmdir : Supprime un répertoire (vide) 7](#_Toc496086697)

[d) Manipuler les archives : 8](#_Toc496086698)

[tar : assembler des fichiers dans une archive 8](#_Toc496086699)

[gzip : la compression la plus utilisé 8](#_Toc496086700)

[tar –zcvf : archiver et compresser d’un coup 8](#_Toc496086701)

[unzip : décompresser un .zip 8](#_Toc496086702)

[unrar : décompresser un .rar 8](#_Toc496086703)

[e) Gestion de paquets et dépendances 9](#_Toc496086704)

[apt-get : installer un programme 9](#_Toc496086705)

[f) Raccourcis clavier utiles : 10](#_Toc496086706)

[Alt + F4 : Ferme la fenêtre actuelle. 10](#_Toc496086707)

[Alt + F2 : Faire apparaître la fenêtre de commandes. 10](#_Toc496086708)

[Alt + Tab : Changer rapidement de fenêtre. 10](#_Toc496086709)

[Super + W : Activer le mode « Expo » 10](#_Toc496086710)

[Ctrl + Alt + Flèches : Basculer entre les espace de travail. 10](#_Toc496086711)

[Ctrl + Alt + L : Verrouiller l’écran 10](#_Toc496086712)

[Ctrl + Super + D : Cache toutes les fenêtre et affiche le bureau 10](#_Toc496086713)

[g) Raccourcis clavier pour la console : 10](#_Toc496086714)

[Ctrl + L : efface le contenu de la console. 10](#_Toc496086715)

[Ctrl + D : Ferme la console. 10](#_Toc496086716)

[Shift + Fleche Haut/Bas : Permet de remonter ou descendre dans les msg. 10](#_Toc496086717)

[Ctrl + A : Ramène le curseur au début de la console. 10](#_Toc496086718)

[Ctrl + E : Ramène le curseur à la fin de la ligne de commandes. 10](#_Toc496086719)

[Ctrl + U : Supprime ce qui se trouve à gauche de la ligne de console. 10](#_Toc496086720)

[Ctrl + K : Supprime ce qui se trouve à droite de la ligne de console. 10](#_Toc496086721)

[2. Commandes système : 11](#_Toc496086722)

[a) Modification de fichier avec permissions : 11](#_Toc496086723)

[chmod : Modifie les permissions d'accès à un fichier ou à un répertoire 11](#_Toc496086724)

[chown : Change le propriétaire et le groupe propriétaire d'un fichier 12](#_Toc496086725)

[chgrp : Change le groupe propriétaire d'un fichier 12](#_Toc496086726)

[b) Gestion des utilisateurs : 12](#_Toc496086727)

[sudo : Permet d'exécuter des commandes en tant qu'un autre utilisateur 12](#_Toc496086728)

[sudo su : devenir root et le rester ! 13](#_Toc496086729)

[passwd : Permet de modifier le mot de passe d'un utilisateur 13](#_Toc496086730)

[groups : Affiche les groupes auxquels appartient un utilisateur 13](#_Toc496086731)

[addgroups : gestion des groupes 13](#_Toc496086732)

[adduser : Ajoute un utilisateur, ou un groupe, au système 14](#_Toc496086733)

[deluser : Supprime un utilisateur du système 14](#_Toc496086734)

[usermod : Modifie les paramètres d'un compte utilisateur 15](#_Toc496086735)

[c) Gestion de l’espace de stockage : 15](#_Toc496086736)

[df : Affiche la quantité d'espace disque utilisé par les systèmes de fichiers. 15](#_Toc496086737)

[fdisk : Modifie la table de partition des disques, son utilisation est auto-documentée avec la commande m 16](#_Toc496086738)

[du : Affiche l'espace disque utilisé par les fichiers et/ou les répertoires 16](#_Toc496086739)

[free : Affiche la mémoire disponible / utilisée du système 16](#_Toc496086740)

[d) Gestion des périphériques : 17](#_Toc496086741)

[lspci : Liste tous les périphériques PCI 17](#_Toc496086742)

[lsusb : Liste tous les périphériques USB 18](#_Toc496086743)

[3. Structure des dossiers et fichiers : 19](#_Toc496086744)

[a) Organisation des fichiers : 19](#_Toc496086745)

[La racine 19](#_Toc496086746)

[Les dossiers de la racine 20](#_Toc496086747)

[Chemins possible pour parcourir un fichier 20](#_Toc496086748)

[Le fonctionnement des droits : 21](#_Toc496086749)

# Commande de base Linux :

## Navigation dans les fichiers :

### ls : Permet de lister un répertoire

#### Signification : list segment

* **-l** : Permet un affichage détaillé du répertoire (permissions d'accès, le nombre de liens physiques, le nom du propriétaire et du groupe, la taille en octets, et l'horodatage)
* **-h** : Associé avec **-l** affiche la taille des fichiers avec un suffixe correspondant à l'unité (K, M, G)
* **-a** : Permet l'affichage des fichiers et répertoires cachés (ceux qui commencent par un . (point))
* **-lct** : Permet de trier les fichiers et répertoires par date de modification décroissante

### cd : Permet de se promener dans les répertoires.

#### Signification : change directory

* **cd**
* permet de revenir au répertoire /home/utilisateur (identique à cd ~)
* **cd -**
* permet de revenir au répertoire précédent
* **cd ..**
* permet de remonter au répertoire parent (ne pas oublier l'espace contrairement à windows)
* **cd /**
* permet de remonter à la racine de l'ensemble du système de fichiers
* **cd /usr/bin/**
* se place dans le répertoire /usr/bin/

### pwd : Affiche le répertoire de travail

#### Signification : print working directory

### ln : Crée un [lien (physique ou symbolique)](https://doc.ubuntu-fr.org/tutoriel/lien_physique_et_symbolique) vers un fichier (ou un répertoire)

#### Signification : link

* Options les plus fréquentes :
  + **-s** : Crée un lien symbolique (similaire au raccourci du monde Windows)
  + **-f** : Force l'écrasement du fichier de destination s'il existe
  + **-d** : Crée un lien sur un répertoire (uniquement en mode sudo ou root)
* Exemples d'utilisation :
  + **ln -s Rep1/Rep2/Monfichier MonLien**
  + Crée un lien symbolique *MonLien* de *Rep1/Rep2/Monfichier* dans le répertoire où on se trouve
  + **ln Monfichier unRep/AutreNom**
  + Crée un lien physique *AutreNom* de *Monfichier* dans le répertoire *unRep*
* Note :
  + Vérifiez que vous vous trouvez bien dans le répertoire dans lequel vous souhaitez créer le lien avant d’exécuter cette commande.

### locate : Permet de chercher des fichiers rapidement

Plus rapide que find, cependant il faut fait un « refresh » de la bdd pour trouver de nouveau fichier ou utiliser : **uptadedb** avec sudo

C'est justement le défaut de locate dont je voulais vous parler : la commande ne fait pas la recherche sur votre disque dur entier, mais seulement sur une base de données de vos fichiers

Votre problème, c'est que les fichiers viennent tout juste d'être créés et n'ont pas encore été répertoriés dans la base de données.

Ex : locate « nom du fichier ».txt ou sans le txt par ex.

### find : Permet de chercher des fichiers et éventuellement d'exécuter des commandes

#### Signification : rechercher

* *Options les plus fréquentes :*
  + ***-name*** *: Recherche d'un fichier par son nom*
  + ***-iname*** *: Même chose que* ***-name*** *mais insensible à la casse*
  + ***-type*** *: Recherche de fichier d'un certain type*
  + ***-atime*** *: Recherche par date de dernier accès*
  + ***-mtime*** *: Recherche par date de dernière modification*
  + ***-link*** *: Recherche du nombre de liens au fichier*
  + ***-user*** *: Recherche de fichiers appartenant à l'utilisateur donné*
  + ***-group*** *: Recherche de fichiers appartenant au groupe donné*
* *Actions les plus fréquentes :*
  + ***-exec*** *: Exécute la commande donnée aux fichiers trouvés*
  + ***-ok*** *: Même chose que* ***-exec*** *mais demande une confirmation*
  + ***-ls*** *: exécute la commande* ***ls*** *à chaque fichier trouvé*
* *Opérateurs les plus fréquents :*
  + ***-a*** *: Opérateur ET*
  + **-o**: Opérateur OU
  + **!** ou **-not**: Opérateur NOT

find est plus long que locate.

### grep : Recherche une chaîne de caractères dans des fichiers.

#### Signification : global regular expression print

* *Options les plus fréquentes :*
  + ***-c*** *: Retourne le nombre de lignes au lieu des lignes elles mêmes*
  + ***-n*** *: Retourne les lignes préfixées par leur numéro*
  + ***-i*** *: Insensible à la casse*
  + ***-r*** *: Recherche récursivement dans tous les sous-répertoires ; On peut utiliser la commande* ***rgrep***
  + ***-G*** *: Recherche en utilisant une expression rationnelle basique (option par défaut)*
  + ***-E*** *: Recherche en utilisant une expression rationnelle étendue ; On peut utiliser la commande* ***egrep***
  + ***-F*** *: Recherche en utilisant une chaîne fixe ; On peut utiliser la commande fgrep*
  + ***-v toto****: Recherche les lignes qui ne contiennent pas le mot « toto »*

### cat : Affiche le contenu d'un fichier.

#### Signification : concatenate

* Options les plus fréquentes :
* **-n** : Affiche les numéros de ligne
* **-v** : Affiche les caractères de contrôles

### more : Affiche un fichier page par page

#### Signification : more

* *Options les plus fréquentes :*
  + ***-s*** *: Regroupe les lignes vides consécutives en une seule*
  + ***-f*** *: Ne coupe pas les lignes longues*

## Manipulation de fichier :

touch: Crée un fichier

echo

### mv : Permet de déplacer ou renommer des fichiers et des répertoires

#### Signification : move

* **-f** : Écrase les fichiers de destination sans confirmation
* **-i** : Demande confirmation avant d'écraser
* **-u**: N'écrase pas le fichier de destination si celui-ci est plus récent

### cp : Permet de copier des fichiers ou des répertoires

#### Signification : copy

* ***-a*** *: Archive. Copie en gardant les droits, dates, propriétaires, groupes, etc.*
* ***-i*** *: Demande une confirmation avant d'écraser*
* ***-f*** *: Si le fichier de destination existe et ne peut être ouvert alors le détruire et essayer à nouveau*
* ***-R*** *ou* ***-r*** *: Copie un répertoire et tout son contenu, y compris les éventuels sous-répertoires*
* ***-u*** *: Ne copie que les fichiers plus récents ou qui n'existent pas*
* ***-v*** *: permet de suivre les copies réalisées en temps réel*

### rm : Permet d'effacer des fichiers

#### Signification : remove

* ***-i*** *: Demande confirmation avant d'effacer*
* ***-f*** *: Ne demande pas de confirmation avant d'effacer*
* ***-r*** *: Efface* ***r****écursivement. Ce mot signifie "y compris ses sous-répertoires et leur contenu".*

## Manipulation de dossier :

### mkdir : Crée un répertoire vide

#### Signification : make directory

* ***-p*** *: Crée les répertoires parents s'ils n'existent pas*

### rmdir : Supprime un répertoire (vide)

#### Signification : remove directory

* ***-p*** *: Supprime les répertoires parents s'ils deviennent vides*

## Manipuler les archives :

Sous linux on réunis tous les fichiers dans un gros fichier appelle archives avec **tar**.

Puis on compresse le gros fichier avec **gzip** ou **bzip2**

### tar : assembler des fichiers dans une archive

On utilise tar + :

-c : crée une archive

-v : afficher le détail de l’opération

-f : assemble l’archive dans un fichier

-tf : affiche le contenude l’archive sans l’extraire

-rvf : ajouter un fichier

- xvf : extraire les fichiers de l’archives

### gzip : la compression la plus utilisé

Compresse un fichier.tar

### gunzip : décompresse un fichier .gz en .tar

Ajouté avec cela la commande tar -xvf pour extraire les fichiers du .tar

### tar –zcvf : archiver et compresser d’un coup

### unzip : décompresser un .zip

### unrar : décompresser un .rar

## Gestion de paquets et dépendances

Résumons.  
Jusqu'ici, nous avons découvert trois nouveaux termes :

* **paquet** : c'est un programme « prêt à l'emploi », l'équivalent des programmes d'installation sous Windows en quelque sorte ;
* **dépendance** : un paquet peut avoir besoin de plusieurs autres paquets pour fonctionner, on dit qu'il a des dépendances ;
* **dépôt** : c'est le serveur sur lequel on va télécharger nos paquets.

### apt-get : installer un programme

**apt-get update**: mise a jour du cache

Il y a deux cas où vous avez besoin de le mettre à jour :

* quand vous changez ou ajoutez un dépôt à votre liste de dépôts ;
* quand vous n'avez pas mis à jour votre cache depuis un moment (quelques semaines).

**apt-get search :** rechercher un paquet

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

## Raccourcis clavier utiles :

### Alt + F4 : Ferme la fenêtre actuelle.

### Alt + F2 : Faire apparaître la fenêtre de commandes.

### Alt + Tab : Changer rapidement de fenêtre.

### Super + W : Activer le mode « Expo »

### Ctrl + Alt + Flèches : Basculer entre les espace de travail.

### Ctrl + Alt + L : Verrouiller l’écran

### Ctrl + Super + D : Cache toutes les fenêtre et affiche le bureau

## Raccourcis clavier pour la console :

### Ctrl + L : efface le contenu de la console.

### Ctrl + D : Ferme la console.

### Shift + Fleche Haut/Bas : Permet de remonter ou descendre dans les msg.

### Ctrl + A : Ramène le curseur au début de la console.

### Ctrl + E : Ramène le curseur à la fin de la ligne de commandes.

### Ctrl + U : Supprime ce qui se trouve à gauche de la ligne de console.

### Ctrl + K : Supprime ce qui se trouve à droite de la ligne de console.

# Commandes système :

## Modification de fichier avec permissions :

### chmod : Modifie les permissions d'accès à un fichier ou à un répertoire

#### Signification : change mode

#### Type d'autorisations (une autorisation d'exécution sur un répertoire autorise son ouverture) :

* + **+** : Ajoute une permission
  + **-** : Enlève une permission
  + **=** : Autorise uniquement l'autorisation indiquée
  + **r** : Lecture ; Valeur octale **4**
  + **w** : Ecriture ; Valeur octale **2**
  + **x** : Execution ; Valeur octale **1**
  + **s** : Utilise les droits du propriétaire ou du groupe lors de l'exécution
  + **u** : Propriétaire du fichier
  + **g** : Groupe propriétaire du fichier
  + **o** : Tous les autres utilisateurs
* Options les plus fréquentes :
  + **-R** : Récursif, modifie les autorisations d'un répertoire et tout ce qu'il contient
  + **-c** : Ne montrer que les fichiers ayant été réellement modifiés
  + **-f** : Ne pas afficher les messages d'erreur
* Exemples d'utilisation :
  + **chmod ugo+x monRep**   
    Ajoute l'exécution (ouverture) du répertoire *monRep* à tous (propriétaire, groupe, autres)
  + **chmod go-wx monRep**   
    Supprime l'autorisation d'écriture et l'exécution de *monRep* au groupe et aux autres
  + **chmod u=rw,go=r MonFichier**   
    Fixe l'autorisation de lecture et d'écriture au propriétaire de *MonFichier* et une autorisation de lecture au groupe et aux autres.
  + **chmod 644 MonFichier**   
    Exactement la même chose que ci-dessus mais en utilisant les valeurs octales (Nota : 6 = 4+2 = lecture + écriture)
  + **chmod u=rw,g=r,o= MonFichier**   
    Fixe l'autorisation d'ouverture et de lecture de *MonFichier* au propriétaire, uniquement la lecture au groupe et interdit tout accès aux autres.
  + **chmod 640 MonFichier**   
    Exactement la même chose que ci-dessus mais en utilisant les valeurs octales

### chown : Change le propriétaire et le groupe propriétaire d'un fichier

#### Signification : change owner

* Options les plus fréquentes :
  + **-R** : Modifie récursivement un répertoire et tout ce qu'il contient
* Exemples d'utilisation :
  + **chown autreUtilisateur MonFichier**   
    Change le propriétaire de *MonFichier* en *autreUtilisateur*
  + **chown -R lui:nous monRep**   
    Change le propriétaire en *lui* et le groupe propriétaire en *nous* du répertoire *monRep* ainsi que tout ce qu'il contient

### chgrp : Change le groupe propriétaire d'un fichier

#### Signification : change groupe

* Options les plus fréquentes :
  + **-R** : Change récursivement un répertoire et tout ce qu'il contient
  + **-h** : Change le groupe propriétaire d'un lien symbolique et seulement lui (ne touche pas à la destination du lien)
  + **-L** : Si fournie avec **R** , change le groupe propriétaire d'un répertoire et des fichiers qu'il contient s'il est pointé par un lien symbolique rencontré lors de l'exécution
* Exemples d'utilisation :
  + **chgrp unGroupe MonFichier**   
    Change le groupe propriétaire du fichier *MonFichier* en *unGroupe*
  + **chgrp -R unGroupe monRep**   
    Change le groupe propriétaire du répertoire *monRep* ainsi que tout ce qu'il contient en *unGroupe*

## Gestion des utilisateurs :

### sudo : Permet d'exécuter des commandes en tant qu'un autre utilisateur

#### Signification : substitute user - do

* Options les plus fréquentes :
  + **-s** : Importe les variables d'environnement du shell
  + **-k** : Lorsque l'on utilise **sudo**, il garde en mémoire le mot de passe ; cette option déconnecte l'utilisateur et forcera à redemander un mot de passe si **sudo** est exécuté avant le timeout défini.
* Exemples d'utilisation :
  + **sudo reboot**   
    Lance la commande **reboot** avec les droits de l'utilisateur root

### sudo su : devenir root et le rester !

Le symbole # à la fin de l'invite de commandes vous indique que vous êtes devenus superutilisateur.  
Vous pouvez alors exécuter autant de commandes en root que vous le voulez.

### passwd : Permet de modifier le mot de passe d'un utilisateur

#### Signification : password

* Options les plus fréquentes :
  + **-S** : Affiche l'état d'un compte (nom du compte, bloqué (L), si l'utilisateur n'a pas de mot de passe (NP) ou a un mot de passe utilisable (P), date de dernière modification du mot de passe, durée minimum avant modification, durée maximum de validité, durée d'avertissement, durée d'inactivité autorisée)
* Exemple d'utilisation :
  + **passwd**   
    Demande à changer le mot de passe

### groups : Affiche les groupes auxquels appartient un utilisateur

#### Signification : groups

Exemples d'utilisation :

* **groups**   
  Affiche la liste des groupes auxquels appartient l'utilisateur ayant tapé la commande.
* **groups MonUtilisateur**   
  Affiche tous les groupes auxquels appartient l'utilisateur *MonUtilisateur*.

### addgroups : gestion des groupes

Chaque utilisateur appartient à un groupe.

Permet d’ajouter un membre dans un groupe

### adduser : Ajoute un utilisateur, ou un groupe, au système

#### Signification​ : "​ajouter un utilisateur"​ (en anglais ​: add user)

* Options les plus fréquentes :
  + **--disabled-login** : Crée un nouvel ​utilisateur ​qui ne pourra ​se connecter ​jusqu'​à ce qu'un mot de passe lui soit attribué.
  + **--disabled-password** : Comme **--disabled-login** sauf que le nouvel utilisateur pourra ​se connecter ​en SSH avec des clefs RSA : pratique pour créer un utilisateur qui ne se connectera que via SSH.
  + **--system** : Crée un utilisateur système.

​\* **--group --system** ​: Un groupe ​ayant le même nom et le même identifiant numérique ​que l'​utilisateur système ​est créé.

* **--home tel-répertoire** : Utilise "​tel-répertoire"​ comme répertoire ​personnel ​de l'​utilisateur, plutôt que la valeur par défaut. Si le répertoire n'​existe pas, il est créé, et les fichiers du squelette y sont copiés.
* **--no-create-home** : Ne crée pas de répertoire HOME.
* Exemples d'utilisation :
* **adduser MonUtilisateur**   
  Crée l'utilisateur *MonUtilisateur*.
* **adduser ​--disabled-password ​--no-create-home UtilisateurSSH**   
  Crée un utilisateur *​UtilisateurSSH*​ sans mot de passe ni répertoire personnel dans /​home, ​qui ne pourra pas se connecter ​à la machine ​**directement**,​ mais en SSH, oui.
* **adduser ​--disabled-password ​--home /​home/​UtilisateurSSH UtilisateurSSH**   
  Même chose que ci-dessus sauf qu'on lui donne le même répertoire HOME qu'à l'​utilisateur *​UtilisateurSSH*​ créé en premier.
* **adduser UtilisateurSSH fuse**   
  Ajoute l'​utilisateur *​UtilisateurSSH*​ (créé ​préalablement) dans le groupe "*​fuse*"​. On peut faire aussi un: **gpasswd -a $USER fuse**
* **adduser NouvelUtilisateur ​--ingroup users**   
  Crée l'​utilisateur *​NouvelUtilisateur*​ et l'​ajoute au groupe "*​users*"​.
* **adduser NouvelUtilisateur ​--ingroup users --uid 1000**  
  Crée l'​utilisateur *​NouvelUtilisateur*​ avec l'uid 1000 et l'​ajoute au groupe "*​users*"​ (utile pour écrire sur une partition ext2/3/4 externe au systeme ou un dossier appartenant au uid 1000).

### deluser : Supprime un utilisateur du système

#### Signification : delete user

* Option la plus fréquente :
  + **–system** : Ne supprime l'utilisateur que si c'est un utilisateur système.
  + **–remove-home** : Supprime l'utilisateur ainsi que son répertoire dans le home.
* Exemple d'utilisation :
  + **deluser UtilisateurSSH**   
    Supprime l'utilisateur *UtilisateurSSH*
  + **deluser –remove-home NouvelUtilisateur**   
    Supprime l'utilisateur *NouvelUtilisateur* ainsi que le répertoire /home/NouvelUtilisateur
  + **deluser NouvelUtilisateur users**   
    Supprime l'utilisateur *NouvelUtilisateur* du groupe "*users*"

### usermod : Modifie les paramètres d'un compte utilisateur

#### Signification : user modification

* Options les plus fréquentes :
  + **-G, –groups** GROUPE1[,GROUPE2,…[,GROUPEN]]] : Ajouter l'utilisateur aux groupes précédents. ATTENTION : si l'utilisateur fait partie d'un groupe qui n'est pas listé dans votre ligne de commande ("admin" par exemple), l'utilisateur sera supprimé de ce groupe avec des conséquences pouvant être désastreuses. Ce comportement peut être changé avec l'option **-a**, qui permet d'ajouter l'utilisateur à une liste de groupes supplémentaires sans prendre le risque de le supprimer d'autres groupes importants.
* Exemples d'utilisation :
  + **usermod -aG toto machin**   
    Ajoute l'utilisateur machin au groupe toto sans supprimer machin de ses groupes d'origine.
  + **usermod -g group1 user1**   
    Défini pour l'utilisateur user1 le groupe group1 comme groupe primaire
  + **sudo usermod -d /home/nouveau\_login -m -l nouveau\_login ancien\_login**   
    Permet de renommer le répertoire (dossier) utilisateur et de changer son nom. Pratique lorsque le pc change de mains.

## Gestion de l’espace de stockage :

### df : Affiche la quantité d'espace disque utilisé par les systèmes de fichiers.

#### Signification : disk free

* Options les plus fréquentes :
  + **-a** : Affiche tous les systèmes de fichiers, y compris ceux de 0 blocs (par exemple : proc, sysfs, usbfs et tmpfs)
  + **-h** : Ajoute aux valeur un **M** pour mébioctet (2^20 octets) pour que ce soit plus lisible.
  + **-H** : Pareil que **-h** mais en mégaoctets (10^6 octets).
  + **-T** : Affiche le type du système de fichier.
* Exemples d'utilisation :
  + **df -h**   
    Affiche la quantité d'espace disque utilisé en mébioctets par les systèmes de fichiers.
  + **df /home**   
    Affiche la quantité d'espace disque utilisé par la partition /home (si elle existe)
  + **df -T -h**   
    Affiche le nom des partitions et leur point de montage.

### fdisk : Modifie la table de partition des disques, son utilisation est auto-documentée avec la commande m

#### Signification : Manipulateur de partitions

* Options les plus fréqentes :
  + **-l** Informations détaillées des disques
* Exemples d'utilisation
  + **sudo fdisk /dev/sda**
  + **sudo fdisk -l**

### du : Affiche l'espace disque utilisé par les fichiers et/ou les répertoires

#### Signification : disk usage

* Options les plus fréquentes :
  + **-a** : Afficher pour tous les fichiers et pas uniquement les répertoires.
  + **-s** : Afficher le total sans lister les différents fichiers
  + **-c** : Faire un total après avoir tout affiché.
  + **-h** : Ajoute un suffixe correspondant à l'unité (K, M, G)
  + **-H** : Idem que **-h** mais en puissance de 10
* Exemple d'utilisation :
  + **du -hs dir**   
    Affiche la taille du répertoire dir ou du répertoire courant si dir est omis.
  + **du -ch /home/MonUtilisateur**   
    Affiche la taille des répertoires contenus dans **/home/MonUtilisateur** en utilisant un suffixe puis le total.
  + **du -sm ~/Images/\*.jpg**   
    Affiche la taille totale des fichiers JPEG contenus dans le répertoire **~/Images**

### free : Affiche la mémoire disponible / utilisée du système

#### Signification : mémoire libre

* Options les plus fréquentes :
  + **-b** : Affiche la mémoire en octets (**b**ytes)
  + **-k** : Affiche la mémoire en **k**ilooctets
  + **-m** : Affiche la mémoire en **m**éga octets
  + **-g** : Affiche la mémoire en **g**iga octets
  + **-h** : Affiche la mémoire en unités pertinentes pour l'**h**umain
  + **-s** : Spécifie le délai de réaffichage de la mémoire
  + **-t** : Affiche en plus la ligne des totaux (RAM + swap)
* Exemples d'utilisation :

​\* **free -th**

* **free -m -s 5**   
  Affiche la mémoire du système en méga octet toutes les 5 secondes

## Gestion des périphériques :

### lspci : Liste tous les périphériques PCI

#### Signification : list pci

* Option la plus fréquente :
  + **-v** : Affiche des informations plus détaillées
* Exemples d'utilisation :
  + **lspci**

### lsusb : Liste tous les périphériques USB

#### Signification : list usb

* Option la plus fréquente :
  + **-v** : Affiche des informations plus détaillées
* Exemples d'utilisation :
  + **lsusb**

Haut du formulaire

Bas du formulaire

Gestion de paquets

CPU

# Structure des dossiers et fichiers :

## Organisation des fichiers :

Pour faire simple, il existe deux grands types de fichiers sous Linux :

* **les fichiers classiques** : ce sont les fichiers que vous connaissez, ça comprend les fichiers texte (.txt,.doc,.odt…), les sons (.wav,.mp3,.ogg), mais aussi les programmes. Bref, tout ça, ce sont des fichiers que vous connaissez et que vous retrouvez dans Windows
* **les fichiers spéciaux** : certains autres fichiers sont spéciaux car ils **représentent** quelque chose. Par exemple, votre lecteur CD est un fichier pour Linux. Là où Windows fait la distinction entre ce qui est un fichier et ce qui ne l'est pas, Linux, lui, dit que **tout est un fichier**. C'est une conception très différente, un peu déroutante comme je vous l'ai dit, mais pas de panique, vous allez vous y faire.

### ****La racine****

Sous Linux, **il n'y a qu'une et une seule racine** : « / ».

### Les dossiers de la racine

* **bin** : contient des programmes (exécutables) susceptibles d'être utilisés par tous les utilisateurs de la machine.
* **boot** : fichiers permettant le démarrage de Linux.
* **dev** : fichiers contenant les périphériques. En fait – on en reparlera plus tard – ce dossier contient des sous-dossiers qui « représentent » chacun un périphérique. On y retrouve ainsi par exemple le fichier qui représente le lecteur CD.
* **etc** : fichiers de configuration.
* **home** : répertoires personnels des utilisateurs. On en a déjà parlé un peu avant : c'est dans ce dossier que vous placerez vos fichiers personnels, à la manière du dossier *Mes documents* de Windows.
* **lib** : dossier contenant les bibliothèques partagées (généralement des fichiers.so) utilisées par les programmes. C'est en fait là qu'on trouve l'équivalent des.dllde Windows.
* **media** : lorsqu'un périphérique amovible (comme une carte mémoire SD ou une clé USB) est inséré dans votre ordinateur, Linux vous permet d'y accéder à partir d'un sous-dossier demedia. On parle de **montage**.
* **mnt** : c'est un peu pareil quemedia, mais pour un usage plus temporaire.
* **opt** : répertoire utilisé pour les add-ons de programmes.
* **proc** : contient des informations système.
* **root** : c'est le dossier personnel de l'utilisateur « root ». Normalement, les dossiers personnels sont placés danshome, mais celui de « root » fait exception. En effet, comme je vous l'ai dit dans le chapitre précédent, « root » est le superutilisateur, le « chef » de la machine en quelque sorte. Il a droit à un espace spécial.
* **sbin** : contient des programmes système importants.
* **tmp** : dossier temporaire utilisé par les programmes pour stocker des fichiers.
* **usr** : c'est un des plus gros dossiers, dans lequel vont s'installer la plupart des programmes demandés par l'utilisateur.
* **var** : ce dossier contient des données « variables », souvent des logs (traces écrites de ce qui s'est passé récemment sur l'ordinateur).

Ex : **mateo21@mateo21 –desktop***: ~ /Examples$*

Le nom du dossier où nous sommes est situé entre le « : » et le « $ ».

Le symbole « ~ » est un synonyme de dossier personnel, ici c’est /home/mateo21.

### Chemins possible pour parcourir un fichier

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

* Un **chemin relatif** est un chemin qui dépend du dossier dans lequel vous vous trouvez.
* Un **chemin absolu** est facile à reconnaître : il commence toujours par la racine « / », ex : **cd** */usr/games*

### Le fonctionnement des droits :

En fessant ls -l on obtiens la liste des droits sur les fichiers :

On peut voir cinq lettres différentes. Voici leur signification :

* **d** (Directory) : indique si l'élément est un dossier ;
* **l** (Link) : indique si l'élément est un lien (raccourci) ;
* **r** (Read) : indique si on peut lire l'élément ;
* **w** (Write) : indique si on peut modifier l'élément ;
* **x** (eXecute) : si c'est un fichier, « x » indique qu'on peut l'exécuter. Ce n'est utile que pour les fichiers exécutables (programmes et scripts).

Si c'est un dossier, « x » indique qu'on peut le « traverser », c'est-à-dire qu'on peut voir les sous-dossiers qu'il contient si on a le droit de lecture dessus.

