

0 _ INTRODUCTION

1_ C'EST QUOI LINUX ?

Un système d'exploitation

= réputé entre autres pour sa sécurité et pour ses mises à jour plus fréquentes que Windows ; mais tout cela, vous allez le découvrir petit à petit.

- **boot Loader** = programme qui permet de faire un choix d'OS au démarrage (ex : **GRUB**)
- **wine** .permet d'utiliser des programmes Windows sous Linux mais toujours mieux en natif.
- **programmes** : gratuits / souvent mis à jour / certains meilleurs que ceux de Windows

La naissance de Linux

- **1984** Microsoft sort son premier OS : **MS-DOS # console (mode commande)** mais déjà **Unix** (depuis 1969) **Unix** = plus puissant que MS-DOS mais aussi plus compliqué à utiliser = pour informaticiens professionnels

1984 : Richard Stallman = le projet GNU

1984 : projet GNU créé par Richard Stallman chercheur en intelligence artificielle au MIT
= créer un nouveau système d'exploitation gratuit fonctionnant comme Unix (mêmes commandes)
remarque : Mac OS X est lui aussi basé sur Unix <> MS-DOS, Windows

GNU un système d'exploitation libre

pas seulement un OS gratuit, devait également être « libre »

programme **libre** = **programme dont on peut avoir le code source** (<> OS **propriétaire** code source Microsoft)
= un programme qu'on a le **droit de copier, modifier, redistribuer**.

« **Open Source** » = code source est ouvert, tout le monde peut le voir

véritable idéologie en informatique : mieux donner son code source car permet partage/évolution + rapide
remarque : légères différences entre un programme « Open Source » et un programme « libre »

1991 : Linus Torvalds :Linux+Unix = Linux

Linus Torvalds, étudiant de l'Université de Helsinki crée sur son temps libre son propre système d'exploitation.

complémentarité : GNU/Linux

- Richard Stallman = programmes de base (copie de fichier, de suppression de fichier, éditeur de texte),
- Linus = « cœur » d'un système d'exploitation : le noyau.

projet GNU (programmes libres) & Linux (noyau d'OS) fusionnent

remarque : Linux n'utilise pas du tout le même code source que d'Unix (propriétaire)= complètement réécrits

les distributions Linux

emballages de Linux. Le cœur, lui, reste le même
différentes **distributions** de Linux ont été créées pour faciliter l'installation de Linux

- l'installation peut être très simplifiée comme très compliquée
- gestion de l'installation des programmes rend + simple l'installation de nouveaux logiciels <> Windows
- des programmes préinstallés sur l'ordinateur (# Windows avec IE)

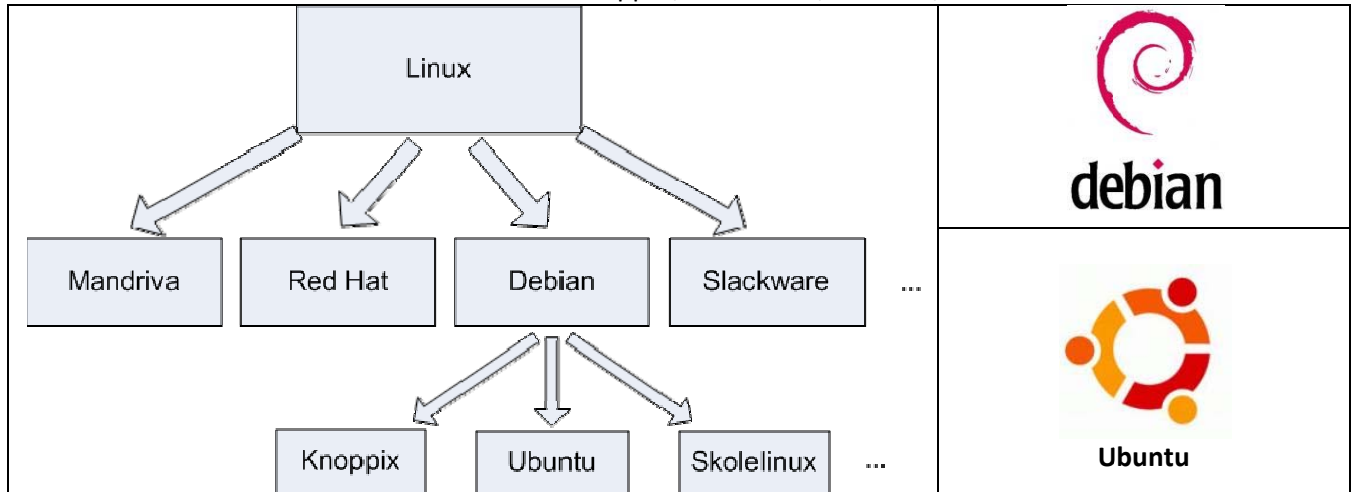
Les différentes distributions existantes

- **Slackware** : une des plus anciennes distributions de Linux. Elle existe encore aujourd'hui !
- **Mandriva** : éditée par une entreprise française, elle se veut simple d'utilisation
- **Red Hat** : éditée par une entreprise américaine, célèbre et très répandue, notamment sur les serveurs
- **SuSE** : éditée par l'entreprise Novell ;
- **Debian** : seule distribution gérée par des développeurs indépendants plutôt qu'entreprise (très populaire)

La distribution Debian

- Debian est donc la seule distribution éditée par des particuliers bénévoles à travers le monde

- autre gros avantage = gestionnaire de paquets apt-get (installation)
- tous les logiciels sont centralisés en un même endroit (<> pas besoin de chercher sur le web)
- nombreuses distributions basées sur Debian : Knoppix / Skolelinux / Ubuntu / ...



Remarque : Skolelinux pour les écoles et est livrée avec de nombreux logiciels éducatifs (gratuits, bien sûr !).

Ubuntu

- prévue pour le **grand public** (slogan est « *Linux for human beings* »)
- **mises à jour sont fréquentes** (développeurs travaillent dur = nouvelle version tous les six mois)
- beaucoup d'utilisateurs, donc **beaucoup de gens pour vous aider** si vous avez des questions !

1 _ INSTALLATION

2 _ LA CONSOLE

<https://openclassrooms.com/fr/courses/43538-reprenez-le-contrôle-a-l'aide-de-linux/37813-la-console-ca-se-mange>

1 _ ACCES

Toggle plein écran

- **Ctrl + Alt + Fn** : terminal 1 (tty) ; n de 1 à 6
- **Ctrl + Alt + F7** : retour au mode graphique
- **Alt + Impr.Écran + K** : redémarrer l'interface graphique X

Console (fenêtrée)

raccourci clavier <CTRL><Alt><T>

applications :

- Kubuntu : Konsole
 - Unity : Terminal
- <https://doc.ubuntu-fr.org/terminal>

à distance en SSH avec PuTTY

2 protocoles :

- Telnet : le + basique mais ne crypte pas les données échangées entre vous et le serveur.
- SSH : le + utilisé car permet de crypter les données.

Programme d'accès : PuTTY sous Windows

2 _ ACCES

Invite de commandes `mateo21@mateo21-desktop:~$`

- **mateo21** : pseudo sous lequel vous vous êtes loggés (compte utilisateur)
- **@** : « at » qui signifie « chez » = machine
- **mateo21-desktop** : nom de l'ordinateur sur lequel vous êtes en train de travailler.

- **:** séparateur.
- **~** : dossier dans lequel vous vous trouvez actuellement.
(~ signifie que vous êtes dans votre dossier personnel = « home » sous Linux # « Mes documents » de Windows)
- **\$** : niveau d'autorisation sur la machine :
 - o \$=utilisateur «normal», avec droits limités (ne peut pas modifier les fichiers système les + importants)
 - o # : mode superutilisateur = connecté avec « root » (utilisateur maître qui a tous les droits)

Documentation : RTFM = « Read the fucking manual! »

Commandes :

Syntaxe : **commande paramètres**

Paramètres : **sensibles à la casse : u <> U**

Paramètres courts : **commande -param** (possible de concaténer : **-a -u -T # -auT**)

Paramètre longs : **commande --parametre1 --parametre2** (impossible de concaténer)

Remarque : on peut avoir les 2 versions :

- o **ls -a = ls -all**
- o **commande -p 14 = commande --parametre=14**

Autocomplétion :

- o taper 2 fois sur **tab** pour avoir les commandes qui commencent par les lettres tapées

Exemple : **da tab tab => dash date**

NB : en tapant 2 x tab sans rien avoir tapé avant = liste de tous les commandes

Display all possibilities ? y=yes / q=quit / Esp=nextpage / Enter=nextline

- o taper une fois sur **tab** : complète sur la ligne de commande s'il n'y a qu'une commande possible

Historique des commandes : flèche haut/bas ou commande **history**

Raccourcis clavier pratiques :

- o **raccourcis généraux** :

Ctrl + L : efface le contenu de la console = **clear**

Ctrl + D : envoie le message EOF (fin de fichier) à la console = cela ferme la console en cours = **exit**

Shift + PgUp : « remonter » dans les messages envoyés par la console # molette de la souris

Shift + PgDown : « redescendre ».

- o **pour une longue commande**

Ctrl + A : ramène le curseur au début de la commande = début

Ctrl + E : ramène le curseur à la fin de la ligne de commandes = fin

Ctrl + U : supprime tout ce qui se trouve à gauche du curseur.

Ctrl + K : supprime tout ce qui se trouve à droite du curseur.

Ctrl + W : supprime 1er mot situé à gauche du curseur

Ctrl + Y : « collera » le texte supprimé avec Ctrl + U, Ctrl + K ou Ctrl + W # couper-coller.

3_ STRUCTURE DES DOSSIERS ET DES FICHIERS

Organisation des dossiers :

Au lieu de séparer chaque disque dur, lecteur CD, disquettes, carte mémoire = **tout est fichier pour Linux**

Deux types de fichiers 2 grands types de fichiers sous Linux :

- **les fichiers classiques** : # windows les fichiers texte (.txt,.doc,.odt...), les sons (.wav,.mp3,.ogg), ...
- **les fichiers spéciaux** : car ils **représentent** quelque chose : exp : le lecteur CD est un fichier pour Linux

La racine = « gros dossier de base qui contient tous les autres dossiers et fichiers ».

Sous Windows = plusieurs C:\, D:\ <> Sous Linux, **il n'y a qu'une et une seule racine** : /

Architecture des dossiers

Sous Windows : :C:\Program Files\Winzip. (antislash\ (aussi appelé *backslash*) séparateur de dossiers)

Sous Linux, c'est au contraire le/qui sert de séparateur.

Remarque :

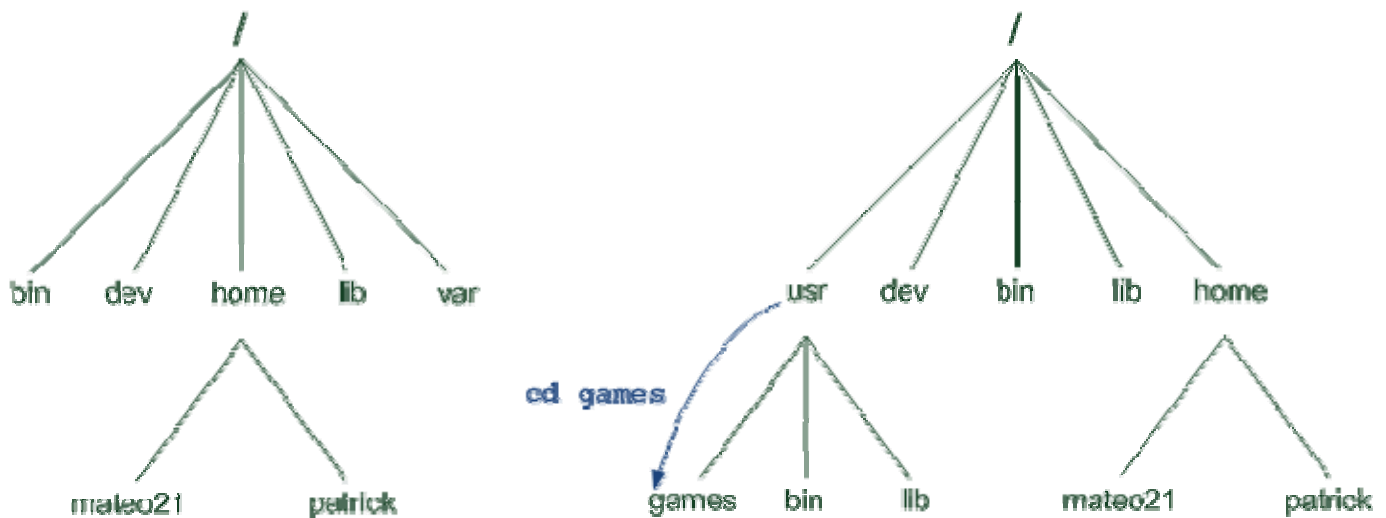
- o Linux gère noms de fichiers et dossiers contenant espaces, accents et majuscules mais **éviter**

- Préférer **minuscules**

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= sous-dossiers qui « représentent » chacun un périphérique.
- **etc** : fichiers de configuration.
- **home** : répertoires personnels des utilisateurs. # Mes documents de Windows.
Remarque : chaque utilisateur de l'ordinateur possède son dossier personnel (exp : /home/mateo21/)
- **lib** : dossier contenant les bibliothèques partagées (généralement des fichiers.so) # dll de Windows.
- **media** : périphérique amovible (carte SD ou clé USB) = sous-dossier de media. On parle de **montage**.
- **mnt** : # media, mais pour un usage plus temporaire.
- **opt** : répertoire utilisé pour les *add-ons* de programmes.
- **proc** : contient des informations système.
- **root** : dossier personnel de l'utilisateur « root » mais pas placé dans home
- **sbin** : contient des programmes système importants.
- **tmp** : dossier temporaire utilisé par les programmes pour stocker des fichiers.
- **usr** : 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*

Schéma résumé de l'architecture



Quelques commandes

pwd : afficher le dossier actuel (rappel : ~ synonyme de votre dossier personnel = exp donc /home/mateo21)
 pwd = « Print Working Directory », c'est-à-dire « Afficher le dossier actuel ».

which : connaître l'emplacement d'une commande (exp : ~\$ which pwd => /bin/pwd)

ls : listes les fichiers et dossiers

Ubuntu active la coloration des fichiers (et dossiers (=bleu foncé, bleu clair=raccourci) par défaut

- **ls --color=auto** : active la couleur si ne l'est pas (mais juste pour la commande, sinon fichier de config)
- **ls -a** : afficher tous les fichiers et dossiers cachés (**ls -A** => idem sans répertoires courant et parent)
« . » en préfixe indique les dossiers / « . » = répertoire courant / « .. » = répertoire parent
- **ls -F** : indique le type d'élément (/ pour les répertoires)
- **ls -l** : liste détaillée

```
mateo21@mateo21-desktop:~$ ls -l
total 16
drwxr-xr-x 2 mateo21 mateo21 4096 2007-09-24 17:22 Desktop
lrwxrwxrwx 1 mateo21 mateo21 26 2007-09-19 18:31 Examples -> /usr/share/example-content
drwxr-xr-x 2 mateo21 mateo21 4096 2007-09-25 15:17 images
drwxr-xr-x 3 mateo21 mateo21 4096 2007-09-25 11:11 log
drwxr-xr-x 3 mateo21 mateo21 4096 2007-09-19 19:51 tutos
```

Légende : un élément par ligne + x colonnes spécifiques a sa propre signification :

1. droits sur le fichier (= chaque caractère à une signification précise)
2. nombre de liens physiques (pas intéressant encore)
3. nom de la personne propriétaire du fichier (= celui qui a créé le fichier)
4. groupe auquel appartient le fichier (= propriétaire si le groupe n'a que le propriétaire)

5. taille du fichier, en octets ;
6. date de dernière modification ;
7. nom du fichier (ou dossier).

Rem : cas du raccourci (**lien symbolique**) = précise vers où pointe le raccourci

- **ls -h** : afficher la taille en Ko, Mo, Go... (-h pour human readable)
- **ls -t** : trier par date de dernière modification (plutôt qu'alphabétique)

synthèse : ~\$ ls -larth commande multiple

cd : changement de dossier

- **cd /** : aller à la racine (:~\$ cd / => /\$)
- **cd usr** :
chemins relatifs : dépend du dossier dans lequel vous vous trouvez
chemins absolus : au départ de la racine + fonctionnent quel que soit le dossier dans lequel on se trouve.
 exp : pour retourner au répertoire personnel : **cd /home/mateo21/** | **cd ~** | **cd** (ramène aussi dans ce rép)
autocomplétion du chemin (très utile)
 exp : cd /usr/games/trucbidule | cd /usr puis cd ga TAB (rappel TAB + TAB renvoie la liste)

du : taille occupée par les dossiers (disk usage)

- **du -h** : taille pour les humains
- **du -a** : afficher la taille des dossiers ET des fichiers (par défaut que les dossiers, -a = all)
- **du -s** : avoir juste le grand total

4_ MANIPULER LES FICHIERS

afficher un fichier exp : fichier syslog dans le rep /var/log

cat : afficher tout le fichier (d'un coup)

less : afficher le fichier page par page

Raccourcis de base pour lire la suite :

- Espace : affiche la suite du fichier d'un « écran » de console
- Entrée : affiche la ligne suivante (aussi FlècheBas)
- d : affiche les onze lignes suivantes (soit ½ écran).
- b : retourne en arrière d'un écran (aussi PageUp)
- y : retourne d'une ligne en arrière (aussi FlècheHaut).
- u : retourne en arrière de 11 lignes (soit -1/2 écran)
- q : arrête la lecture du fichier

Raccourcis avancés :

- = : indique où vous en êtes dans le fichier (numéro des lignes affichées et pourcentage).
- h : affiche l'aide (= pour toutes les commandes) (q pour sortir)
- / : tapez/suivi du texte que vous recherchez pour lancer le mode recherche + Entrée pour valider.
 Remarque : expression régulières acceptées _ voir cours [Concevez votre site Web avec PHP et MySQL](#)
- n : « Résultat suivant » après une recherche
- N : « Résultat précédent » après une recherche.

head : afficher le début du fichier

- **head -n+nombre** : nombre de lignes souhaitées (exp : head -n 3 syslog)

tail : afficher la fin du fichier

- **tail -f** : (f pour *follow*, « suivre » en anglais) : dynamique ! très utile pour une log ! (Ctrl+C pour arrêter)
- **tail -f -s+nombre** : si nombre=3, maj toutes les 3" (plutôt que toutes les secondes)

créer des fichiers et dossiers

touch : créer un fichier (en fait le plus souvent par l'intermédiaire de logiciels)

Rem : faite pour modifier la date de dernière modification d'un fichier d'où son nom : on « touche » le fichier
 Mais si le fichier n'existe pas, il sera créé : (exp : dans ~\$ ls -F que des dossiers, touch fichier)

- créer plusieurs fichiers : touch fichier1 fichier2
- nom de fichier avec espace : touch "mon nom de fichier"

mkdir : créer un dossier (même fonctionnement que touch) mkdir mondossier

mkdir -p : crée tous les dossiers intermédiaires (mkdir -p animaux/vertébres/chat)

copier & déplacer un fichier/dossier

cp : copier un ou plusieurs fichiers.dossiers (abréviation de CoPy)

- syntaxe : cp original copie
- même fichier dans un autre dossier : cp original destination/ (/ final n'est pas nécessaire)

- copie dans un autre dossier : cp original destination/copie
- copier des dossiers : cp -R original copie
- joker ou wildcard : cp *.jpg mondossier/ cp so* mondossier/

mv : déplacer un fichier (abréviation de MoVe) (mêmes principes que cp)

2 utilités : 1_ déplacer un fichier (ou un dossier) 2_ renommer un fichier (ou un dossier)

- déplacer un fichier : mv fichierbidon mondossier/
- renommer un fichier (pas de commande spéciale) : mv fichierbidon superfichier
- déplacer et renommer à la fois : mv fichierbidon mondossier/superfichier

supprimer des fichiers et dossiers ATTENTION, il n'existe pas de corbeille dans la console de Linux

rm : supprimer un fichier (pour ReMove, « supprimer »)

peut supprimer un fichier, plusieurs fichiers, des dossiers, voire même votre ordinateur entier !

syntaxe : rm fichierbidon

- **rm -i fichier** : demander confirmation : o/n puis entrée
- **rm -f fichier** : force la suppression
- **rm -v fichier** : pour Verbose (=parler beaucoup) = afficher ce qui est fait
(par défaut les commandes sont silencieuses / -v est commun bcp de commande)

- **rm -r dossier** : supprime un dossier et son contenu, fichiers et dossiers

Remarque : existe aussi rmdir mais ne peut supprimer un dossier que s'il est vide

rm et le joker de la mort (qui tue) :

= **supprime tous le disque dur !**

- rm : commande la suppression ;
- -r : supprime de manière récursive tous les fichiers et dossiers ;
- -f : force la suppression sans demander la moindre confirmation ;
- /* : supprime tous les fichiers et dossiers qui se trouvent à la racine (/) quel que soit leur nom (joker*).

ATTENTION : supprime aussi Windows s'il est accessible depuis Windows

SECURITE : pas accessible à un utilisateur lambda mais seulement à root

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

ATTENTION : il n'y a que le / de différence et être bien sur que rien n'est utile

créer des liens (linux) entre les fichiers = **créer des raccourcis (windows)**








2 types de liens : 1_ liens **physiques** / 2_ liens **symboliques**

stockage des fichiers

Sur le disque dur, chaque fichier est grosso-modo séparé en 3 parties : **1_nom 2_droit accès 3_contenu**

= **la liste des noms de fichiers est stockée à un autre endroit que leur contenu.**

Principe : chaque contenu de fichier se voit attribuer un numéro d'identification appelé **inode**

Nom du fichier	Contenu (inode)	Nom du fichier	Contenu (inode)	Nom du fichier	Contenu (inode)
	→ 011010010 010101011 100001110		→ 011010010 010101011 100001110		→ 011010010 010101011 100001110
	→ 011010010 010101011 100001110		↗ 011010010 010101011 100001110		
	→ 011010010 010101011 100001110				
Principe		Lien Physique		Lien Symbolique	

créer des liens physiques (seulement avec des fichiers) : plus rare mais pratique

ln fichier1 fichier2 = permet d'avoir 2 noms de fichiers qui partagent le même contenu = même inode

```
mkdir tests
cd tests
touch fichier1
```

```
ln fichier1 fichier2
ls -l
-rw-r--r-- 2 mateo21 mateo21 0 2008-07-31 13:55 fichier1
-rw-r--r-- 2 mateo21 mateo21 0 2008-07-31 13:55 fichier2
```

Légende : : seconde colonne « 2 » pour chacun des fichiers = nombre de fichiers qui partagent le même inode
ls -i : pour afficher les numéros d'inode correspondants et vérifier que les fichiers sont associés au même inode.
Remarque : si on supprime 1 des fichiers, l'autre continue à exister

créer des liens symboliques (aussi avec les dossiers) = raccourcis

ln -s fichier1 fichier2 = pointe vers le nom de fichier et non vers l'inode directement

```
rm fichier2
ln -s fichier1 fichier2
ls -l
-rw-r--r-- 1 mateo21 mateo21 0 2008-07-31 13:55 fichier1
lrwxrwxrwx 1 mateo21 mateo21 8 2008-07-31 14:15 fichier2 -> fichier1
```

Légende : les liens symboliques sont beaucoup plus faciles à repérer que les liens physiques

- la toute première lettre de la seconde ligne est un l (comme *link*, c'est-à-dire lien) ;
- tout à la fin de la seconde ligne, une flèche montre clairement que fichier2 pointe vers fichier1.

5_ LES UTILISATEURS ET LES DROITS

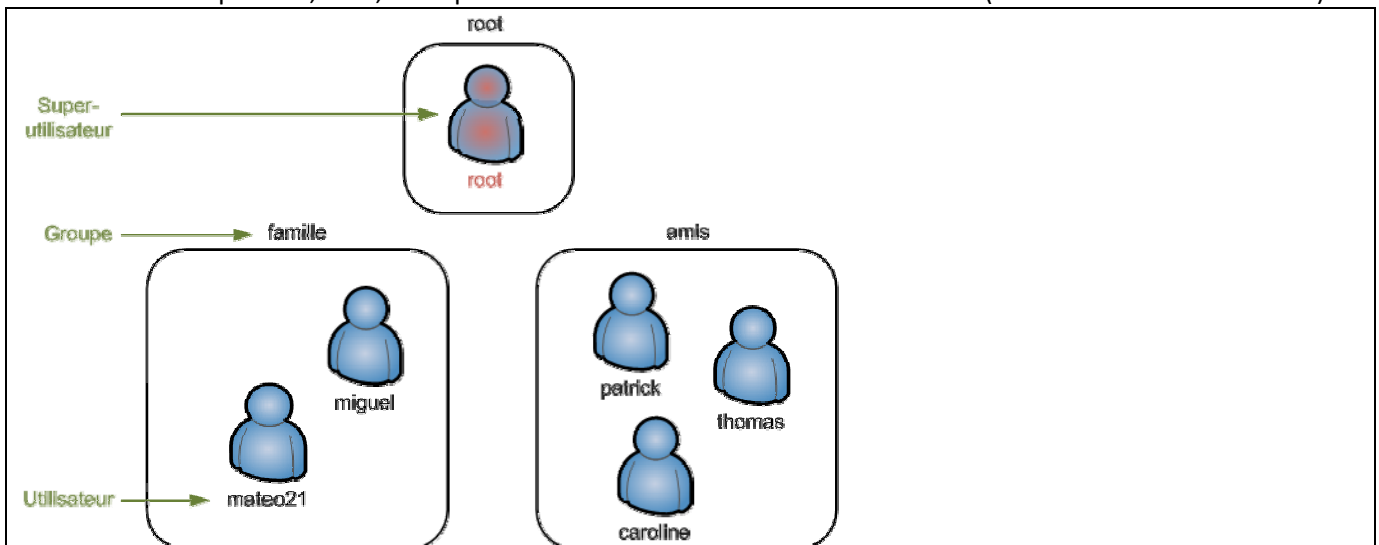
Linux est un système multi-utilisateurs, en s'y connectant à distance notamment donc simultanément

sudo: exécuter une commande en root

A l'installation, on a créé un compte utilisateur avec des **droits limités**

organisation des utilisateurs sous Linux

- autant d'utilisateurs que l'on veut, eux-mêmes répartis dans des groupes.
- un utilisateur « spécial », **root**, ou superutilisateur=tous les droits sur la machine (connecte rarement en root)



Remarques :

- sous Windows, connectés en administrateur par défaut explique pourquoi virus sont si dangereux.
- Ubuntu interdit de se logger en root (qui existe) mais pas accès directement.

sudo : devenir root un instant (pour Substitute User DO)

syntaxe : sudo **commande** (+donner son password)

sudo su : devenir root et le rester

syntaxe : **sudo su** = passe root indéfiniment (exit ou Ctrl+D pour sortir)

```
mateo21@mateo21-desktop: /home$ sudo su
[sudo] password for mateo21:
root@mateo21-desktop: /home#
```

légende : # à la fin de l'invite de commandes = devenu superutilisateur.

remarque : **su** suffit à passer root sur autres distribution qu'Ubuntu

su - : - place directement dans le dossier /root

gestion des utilisateurs

adduser : ajouter un utilisateur (commande root)

syntaxe : **sudo adduser patrickr**

```
root@mateo21-desktop:/home# adduser patrick
Ajout de l'utilisateur « patrick »...
Ajout du nouveau groupe « patrick » (1001)...
Ajout du nouvel utilisateur « patrick » (1001) avec le groupe « patrick »...
Création du répertoire personnel « /home/patrick »...
Copie des fichiers depuis « /etc/skel »
... mot de passe / informations / confirmation
```

passwd : changer le mot de passe : **sudo passwd patrick**

deluser : supprimer un compte : **sudo deluser patrick**

- **ATTENTION : NE PAS SUPPRIMER SON COMPTE AVEC UBUNTU ... on ne pourrait plus se connecter !**
- **deluser --remove-home patrick** : supprime aussi le home de l'utilisateur
- Rem : adduser et deluser commandes Debian et descendants, dont Ubuntu <> useradd & userdel, UNIX

gestion des groupes

Par défaut un groupe du même nom que l'utilisateur automatiquement créé : patrick dans le groupe patrick

```
root@mateo21-desktop:~# cd /home
root@mateo21-desktop:/home# ls -l
total 24
drwx----- 2 root    root    16384 2007-09-19 18:22 lost+found
drwxr-xr-x 65 mateo21 mateo21 4096 2007-11-15 22:40 mateo21
drwxr-xr-x 2 patrick patrick 4096 2007-11-15 23:00 patrick
```

Légende : 3ème colonne = propriétaire du fichier ou dossier / 4ème = groupe qui possède ce fichier ou dossier.

créer un groupe : **addgroup nomgroupe**

modifier un utilisateur

usermod : permet d'éditer un utilisateur

- **usemod -l** : renomme l'utilisateur (le nom de son répertoire personnel ne sera pas changé par contre) ;
- **usemod -g** : change de groupe. usermod -g amis patrick
- **usemod -G** : appartient à plusieurs groupes usermod -G amis,paris,collegues patrick (séparateur : ,).
- **Usemod -aG** : ajouter un groupe en restant dans l'actuel : usermod -aG amis patrick

supprimer un groupe : **delgroup amis**

gestion des propriétaires d'un fichier : seul l'utilisateur root peut changer le propriétaire d'un fichier.

Hypothèse : mateo21 possède dans son répertoire personnel un fichier rapport.txt à donner à patrick

```
mateo21@mateo21-desktop:~$ ls -l rapport.txt
-rw-r--r-- 1 mateo21 mateo21 0 2007-11-15 23:14 rapport.txt
```

changer le propriétaire d'un fichier

sudo chown newowner fichier : **sudo chown patrick rapport.txt**

```
root@mateo21-desktop:/home/mateo21# ls -l rapport.txt
-rw-r--r-- 1 patrick mateo21 0 2007-11-15 23:14 rapport.txt
```

remarque : appartient toujours au groupe mateo21 !

changer le groupe propriétaire d'un fichier (même syntaxe que chown)

sudo chgrp newgroup fichier

remaque : possible aussi avec chown groupe:user (séparateur :) **sudo chown patrick:amis rapport.txt**

- **sudo chown -R** : affecter récursivement les sous-dossiers (avec chown)

exemple : **chown -R mateo21:mateo21 /home/patrick/** donne tout le contenu du dossier personnel à mateo

```
root@mateo21-desktop:/home# ls -l
total 24
drwx----- 2 root    root    16384 2007-09-19 18:22 lost+found
drwxr-xr-x 62 mateo21 mateo21 4096 2007-11-15 23:19 mateo21
drwxr-xr-x 2 mateo21 mateo21 4096 2007-11-15 23:00 patrick
```

modifier les droits d'accès

fonctionnement des droits : chaque fichier / dossier possède une liste de droits (lecture / modifier / exécuter)


exemple : drwxr-xr-x 2 mateo21 mateo21 4096 2007-10-19 01:21 Vidéos

Légendes :

si la lettre apparaît = le droit existe / s'il y a un tiret = aucun droit

- **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 (utile pour les exécutable)
- rem : si c'est un dossier = peut le « traverser » = peut voir les sous-dossiers + si on a le droit de lecture dessus.

explication des positions des lettres

	<ul style="list-style-type: none">• 1^{er} triplet rwx = droits du propriétaire sur le dernier• 2^{ème} triplet rwx = droits autres membres du groupe• 3^{ème} triplet rwx = droits des autres utilisateurs
---	--

exemple :

```
mateo21@mateo21-desktop:~$ ls -l rapport.txt
-rw-r--r-- 1 mateo21 mateo21 0 2007-11-15 23:14 rapport.txt
```

Interprétation : ses droits sont : -rw-r--r--

- - : pas un dossier (<> d = dossier)
- **rw-** : propriétaire du fichier (mateo21) peut lire et modifier (et donc supprimer) mais pas l'exécuter
- **r--** : utilisateurs du groupe mateo21 (autres que mateo21) peuvent seulement lire le fichier
- **r--** : tous les autres (ceux qui ne font pas partie du groupe mateo21) peuvent seulement lire le fichier.

Rappel : root a TOUS les droits. Il peut tout faire : lire, modifier, exécuter n'importe quel fichier.

modifier les droits d'accès pas besoin d'être root pour utiliser chmod mais propriétaire du fichier

- on attribue un chiffre à chaque droit : r=4 / w=2 / x=1

Droit	Chiffre
r	4
w	2
x	1

Droits	Chiffre	Calcul
---	0	0 + 0 + 0
r--	4	4 + 0 + 0
-w-	2	0 + 2 + 0
--x	1	0 + 0 + 1
rw-	6	4 + 2 + 0
-wx	3	0 + 2 + 1
r-x	5	4 + 0 + 1
rwX	7	4 + 2 + 1

- à faire pour chaque triplet de droits : « 640 » droits dans l'ordre :
6 : lecture/écriture pour le propriétaire. / 4 : lecture pour le groupe / 0 : aucun droit pour les autres

quelques exemples :

- **droit maximal : 777** = droit lecture/écriture/exécution pour propriétaire/groupe/autres
- **droit minimal : 000** = personne ne peut rien faire... à part root
- **droits exclusifs pour le seul proprio** : chmod 600 rapport.txt

```
mateo21@mateo21-desktop:~$ ls -l rapport.txt
-rw----- 1 mateo21 mateo21 0 2007-11-15 23:14 rapport.txt
```

Attribuer des droits avec des lettres (chmod relatif)

- **u** = user (propriétaire) / **g** = group (groupe) / **o** = other (autres)
- **+** signifie : « Ajouter le droit » / **-** signifie : « Supprimer le droit » / **=** signifie : « Affecter le droit »

chmod g+w rapport.txt	« Ajouter le droit d'écriture au groupe ».
-----------------------	--

chmod o-r rapport.txt	« Ajouter le droit d'écriture au groupe ».
chmod u+rx rapport.txt	« Ajouter les droits de lecture et d'exécution au propriétaire »
chmod g+w,o-w rapport.txt	« Ajouter le droit d'écriture au groupe et l'enlever aux autres ».
chmod go-r rapport.txt	« Enlever le droit de lecture au groupe et aux autres »
chmod +x rapport.txt	« Ajouter le droit d'exécution à tout le monde »
chmod u=rwx,g=r,o=- rapport.txt	« Affecter tous les droits au propriétaire, lecture au groupe, rien aux autres ».

Remarques :

- **paramètre -R existe aussi pour chmod pour la récursivité**
- **chmod -R 700 /home/mateo21** : seul à pouvoir lire, éditer et exécuter les fichiers de mon répertoire personnel

6_ DITEURS DE TEXTE : Nano (le plus simple), Vim, Emacs

« GNU nano » : car tout petit & très simple (<> Vim et Emacs) (s'inspire de pico plus ancien) # bloc-note windows

- éditeur de texte = **permet de modifier des fichiers de texte brut, sans mise en forme (gras, italique, souligné...)**
- usage = **modifier des fichiers de configuration**
- lancement : **nano**

raccourcis clavier : aide-mémoire en bas de l'écran

Toggle aide : Esc+X

Note : ^signifieCtrl

^G Aide	^O Ecrire	^R Lire fich.	^Y Page préc.	^K Couper	^C Pos. cur.
^X Quitter	^J Justifier	^W Chercher	^V Page suiv.	^U Coller	^T Orthograp.

- Ctrl + G : afficher l'aide ;
- Ctrl + K : couper la ligne de texte (et la mettre dans le presse-papier) ;
- Ctrl + U : coller la ligne de texte que vous venez de couper ;
- Ctrl + C : afficher à quel endroit du fichier votre curseur est positionné (numéro de ligne...) ;
- Ctrl + W : rechercher dans le fichier ;
- Ctrl + O : enregistrer le fichier (écrire) ;
- Ctrl + X : quitter Nano.

Déplacement : flèches / PageUp / Page Down / Ctrl + Y / Ctrl + V

Recherche : **Ctrl+W** puis « cible »+enter

- Ctrl+C : sortie du mode recherche
- Ctrl+W (ou entrée) : résultat suivant

Enregistrer: **Ctrl+O**

Quitter: **Ctrl+X** (message si pas enregistré), confirmer le sortie **o/n** (ou Ctrl+C pour annuler la sortie)

Nommer le fichier : **o** + tapez nom du fichier+ Entrée

paramètres de la commande Nano nano fichier

- **nano -m** : autorise l'utilisation de la souris sous Nano
- **nano -i** : indentation automatique
- **nano -A** : active le retour intelligent au début de la ligne (utile avant tout pour les programmeurs)

configurer avec .nanorc (= fichier de config de nano = lu à chaque démarrage)

remarque :

- la plupart des fichiers de configuration commencent par un point (faciles à « cacher » avec un ls)
- chaque utilisateur de la machine peut créer son propre fichier de configuration .nanorc dans son home

création du .nanorc : **nano .nanorc**

fichier de config : une instruction par ligne qui commence par **set** (pour activer) ou un **unset** (pour désactiver)

exemples : **set mouse # -m / set autoindent # -i / set smarthome # -A** (Ctrl+O=save, Ctrl+X=exit)

nanorcglobal : pour tous les utilisateurs dans/etc/nanorc (modifié que par root) = **sudo nano /etc/nanorc**

syntaxe : **#**=commentaire = supprimer le commentaire pour active

coloration syntaxique section « color setup » dans le fichier **nanorcglobal**

décommenter pour activer par type de langage :

```
## Nanorc files
# include "/usr/share/nano/nanorc.nanorc"

## C/C++
# include "/usr/share/nano/c.nanorc"

## HTML
# include "/usr/share/nano/html.nanorc"
```

Configure sa console avec .bashrc (dans le répertoire personnel : `~$ nano .bashrc`)

ATTENTION FICHER SENSIBLE = on ne s'intéresse qu'à quelques lignes

- **personnaliser l'invite de commande** :

```
# set a fancy prompt (non-color, unless we know we "want" color)
case "$TERM" in
xterm-color)
    PS1='${debian_chroot:+($debian_chroot)}\[\033[01;32m\]\u@\h\[\033[00m\]:\[\033[01;34m\]\w\[\033[00m\]\$ '
    ;;
*)
    PS1='${debian_chroot:+($debian_chroot)}\u@\h:\w\$ '
    ;;
esac

# Comment in the above and uncomment this below for a color prompt
# PS1='${debian_chroot:+($debian_chroot)}\[\033[01;32m\]\u@\h\[\033[00m\]:\[\033[01;34m\]\w\[\033[00m\]\$ '
```

- **invite de commande colorée** : ajouter # devant les deux 1ers PS1, et enlever le # du dernier PS1

```
# set a fancy prompt (non-color, unless we know we "want" color)
case "$TERM" in
xterm-color)
#    PS1='${debian_chroot:+($debian_chroot)}\[\033[01;32m\]\u@\h\[\033[00m\]:\[\033[01;34m\]\w\[\033[00m\]\$ '
    ;;
*)
#    PS1='${debian_chroot:+($debian_chroot)}\u@\h:\w\$ '
    ;;
esac

# Comment in the above and uncomment this below for a color prompt
PS1='${debian_chroot:+($debian_chroot)}\[\033[01;32m\]\u@\h\[\033[00m\]:\[\033[01;34m\]\w\[\033[00m\]\$ '
```

- **affichage de la ligne de commande** = dernière ligne PS1

```
PS1='${debian_chroot:+($debian_chroot)}\[\033[01;32m\]\u@\h\[\033[00m\]: \[\033[01;34m\]\w\[\033[00m\]\$ '
```

- séquences de type «\033» = paramétrer la couleur
- \u au milieu = nom de l'utilisateur (mateo21 par exemple)
- \h = nom de la machine hôte (mateo21-desktop)
- + repérer dans cette ligne le « @ » qui sépare les deux, le « : », le « \$ », etc.

créer des alias : commandes que vous créez et qui sont automatiquement transformées en d'autres commandes

```
# enable color support of ls and also add handy aliases
if [ "$TERM" != "dumb" ]; then
    eval "`dircolors -b`"
    alias ls='ls --color=auto'
    #alias dir='ls --color=auto --format=vertical'
    #alias vdir='ls --color=auto --format=long'
fi

# some more ls aliases
#alias ll='ls -l'
#alias la='ls -A'
#alias l='ls -CF'
```

ls avec couleur

ls en mode détaillé

alternative = `alias ll='ls -lArth'` = ls + affichage détail + fichiers cachés + taille lisible + ordre inverse modification

lister les alias : `alias` = liste de tous les alias définis

syntaxe : `alias nom='commande'` (ne pas mettre d'espace autour du symbole « = »)

sécuriser rm : pour éviter de pouvoir tout supprimer : `alias rm='rm --preserve-root'`

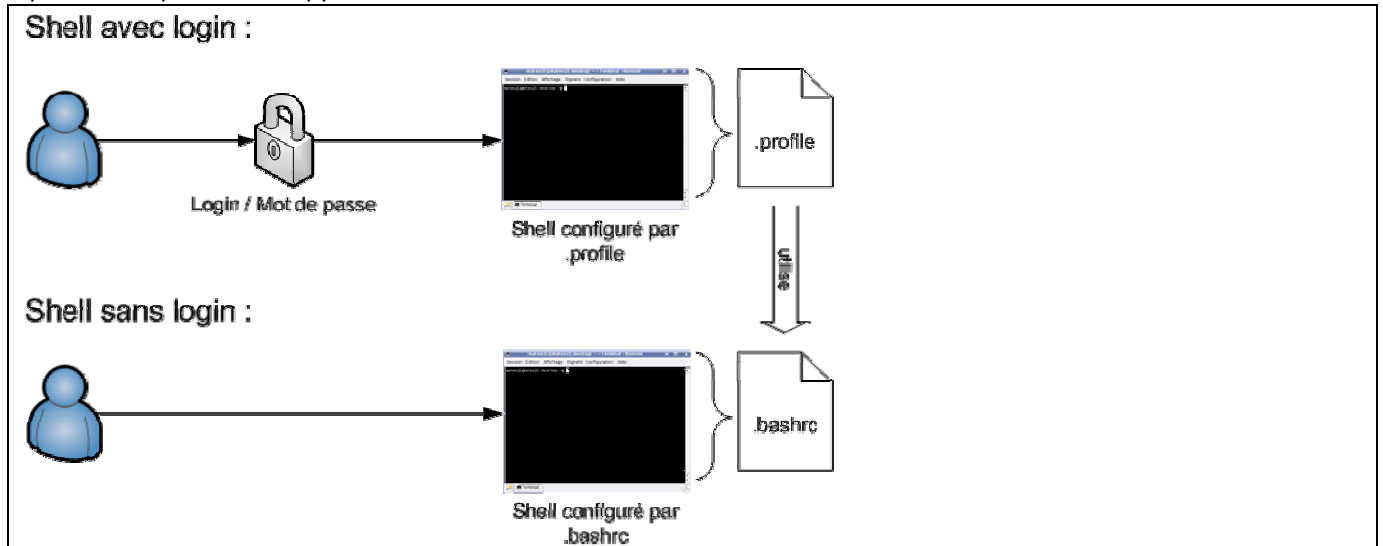
Edition du bashrc global dans `/etc/bash.bashrc` (doit être édité en root)

Les éléments du bashrc personnel ont la priorité sur ceux du bashrc global.

profile :

- comme existe `~/ .bashrc` et un `/etc/bash.bashrc`, il existe un `~/ .profile` et un `/etc/profile`

- **.profile est lu à chaque nouvelle console dans laquelle vous vous loggez** <> **.bashrc** console sans logging
- **.profile fait par défaut appel au.bashrc** = suffit d'éditer votre.bashrc



shell = programme qui interprète les commandes # synonyme de « console »

7_ INSTALLER DES PROGRAMMES AVEC apt-get

Paquets et dépendances valable uniquement Debian (dont Ubuntu)

Des programmes livrés sous forme de paquets

Windows : « Programmes d'installation » = exécutable à lancer et installent dans le dossier ProgramFiles

Ubuntu : pas de programmes d'installation mais des **paquets** # zip qui contient tous les fichiers du programme
= sous la forme d'un fichier.deb (référence à **DEB**ian) = contient toutes les instructions nécessaires pour installer

Différences avec un programme d'installation :

- il y a une gestion des **dépendances** du programme
- pas besoin de faire une recherche sur un moteur de recherche pour trouver un.deb
= tous les.deb sont rassemblés au même endroit sur un même serveur appelé **dépôt** (*repository*)

1_ les dépendances

Très rare qu'un programme puisse fonctionner seul sous Linux

Très souvent, il utilise d'autres programmes ou d'autres « bouts de programmes » appelés **bibliothèques**
= les programmes dépendent d'autres programmes pour fonctionner : ils ont des **dépendances**.

exp : GIMP dépend de bibliothèques de lecture des images (qui lui disent comment lire une image JPEG)
(# DirectX sous windows pour certains jeux)

système de paquets Debian intelligent = chaque paquet indique de quels autres paquets il dépend

=> permet au système d'aller récupérer les dépendances manquantes automatiquement si besoin est.

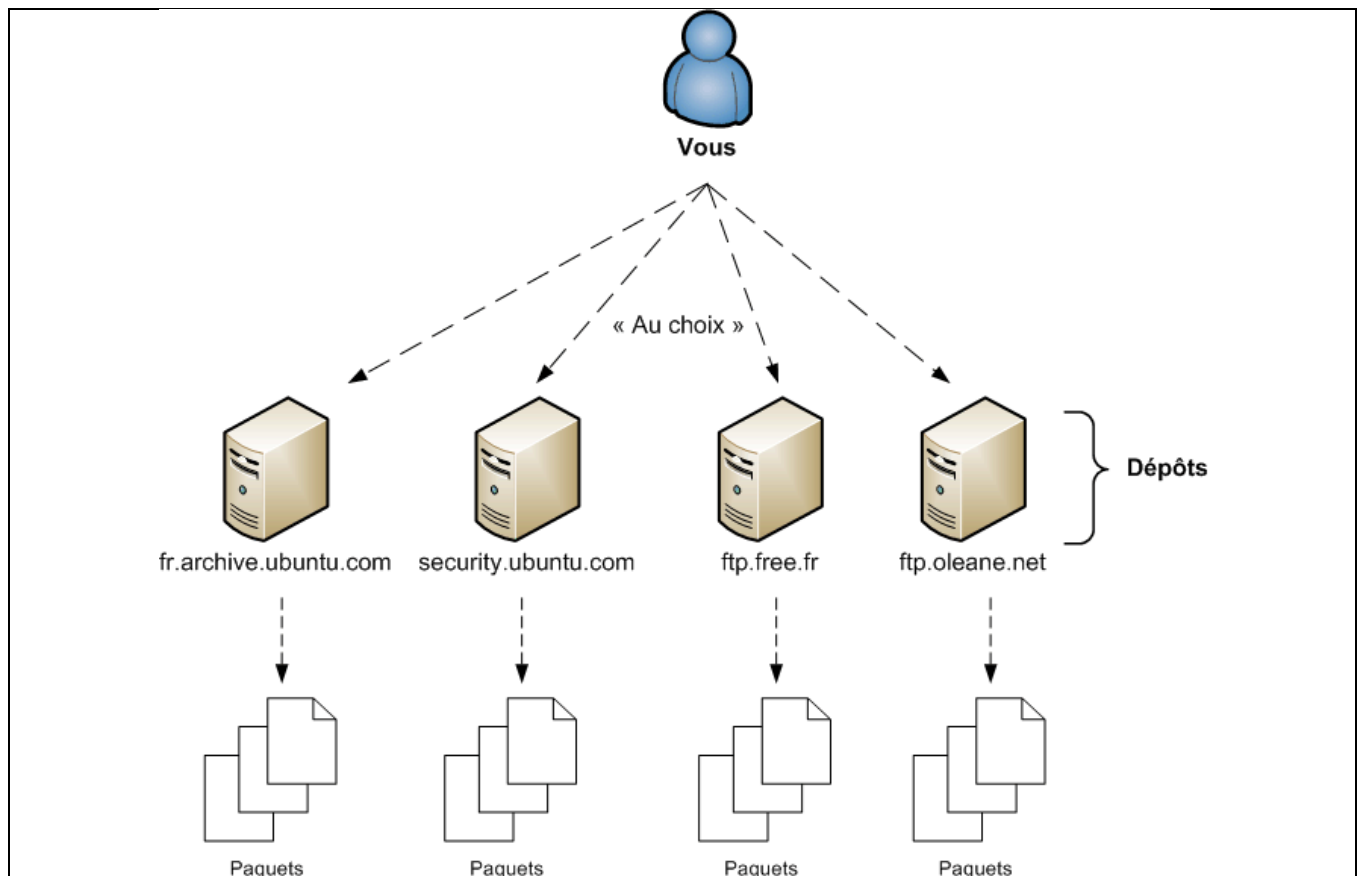
2_ les dépôts

Tous les paquets sont regroupés au sein d'un même endroit appelé dépôt :

= serveur qui propose **tous** les paquets qui existent (ou presque) <> windows : programmes éparpillés
(possible car la plupart des programmes sous Linux sont libres <> programmes propriétaires)

- **la notion de dépôt** (*repository* en anglais) = **lieu où se trouvent les paquets**

Questions : risque sur les débits ? multiplications des dépôts



- **choisir son dépôt** : le plus proche (sur lequel vous téléchargez suffisamment vite).
En France, par défaut, Ubuntu utilise le dépôt **fr.archive.ubuntu.com**.
Mais aussi des dépôts Free / Wanadoo-Orange = Oleanne
- **gérer ses dépôts** : fichier de config **/etc/apt/sources.list** (à modifier par root)
sudo nano /etc/apt/sources.list ou **sudo su** avant

```
# See http://help.ubuntu.com/community/UpgradeNotes for how to upgrade to
# newer versions of the distribution.

deb http://fr.archive.ubuntu.com/ubuntu/ hardy main restricted
deb-src http://fr.archive.ubuntu.com/ubuntu/ hardy main restricted

## Major bug fix updates produced after the final release of the
## distribution.
deb http://fr.archive.ubuntu.com/ubuntu/ hardy-updates main restricted
deb-src http://fr.archive.ubuntu.com/ubuntu/ hardy-updates main restricted

## N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu
## team, and may not be under a free licence. Please satisfy yourself as to
## your rights to use the software. Also, please note that software in
## universe WILL NOT receive any review or updates from the Ubuntu security
## team.
deb http://fr.archive.ubuntu.com/ubuntu/ hardy universe
deb-src http://fr.archive.ubuntu.com/ubuntu/ hardy universe

## N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu
## team, and may not be under a free licence. Please satisfy yourself as to
## your rights to use the software. Also, please note that software in
## multiverse WILL NOT receive any review or updates from the Ubuntu
## security team.
deb http://fr.archive.ubuntu.com/ubuntu/ hardy multiverse
deb-src http://fr.archive.ubuntu.com/ubuntu/ hardy multiverse

## Uncomment the following two lines to add software from the 'backports'
## repository.
## N.B. software from this repository may not have been tested as
## extensively as that contained in the main release, although it includes
## newer versions of some applications which may provide useful features.
## Also, please note that software in backports WILL NOT receive any review
## or updates from the Ubuntu security team.
# deb http://fr.archive.ubuntu.com/ubuntu/ hardy-backports main restricted universe multiverse
# deb-src http://fr.archive.ubuntu.com/ubuntu/ hardy-backports main restricted universe multiverse

deb http://security.ubuntu.com/ubuntu hardy-security main restricted
deb-src http://security.ubuntu.com/ubuntu hardy-security main restricted
deb http://security.ubuntu.com/ubuntu hardy-security universe
```

```
deb-src http://security.ubuntu.com/ubuntu hardy-security universe
deb http://security.ubuntu.com/ubuntu hardy-security multiverse
deb-src http://security.ubuntu.com/ubuntu hardy-security multiverse
```

Chaque ligne du fichier commence par une de ces deux directives :

- **deb** : pour télécharger la version compilée (binaire) des programmes = versions « prête à l'emploi »
- **deb-src** : permet de récupérer le code source du programme

exemple : deb http://fr.archive.ubuntu.com/ubuntu/ hardy universe

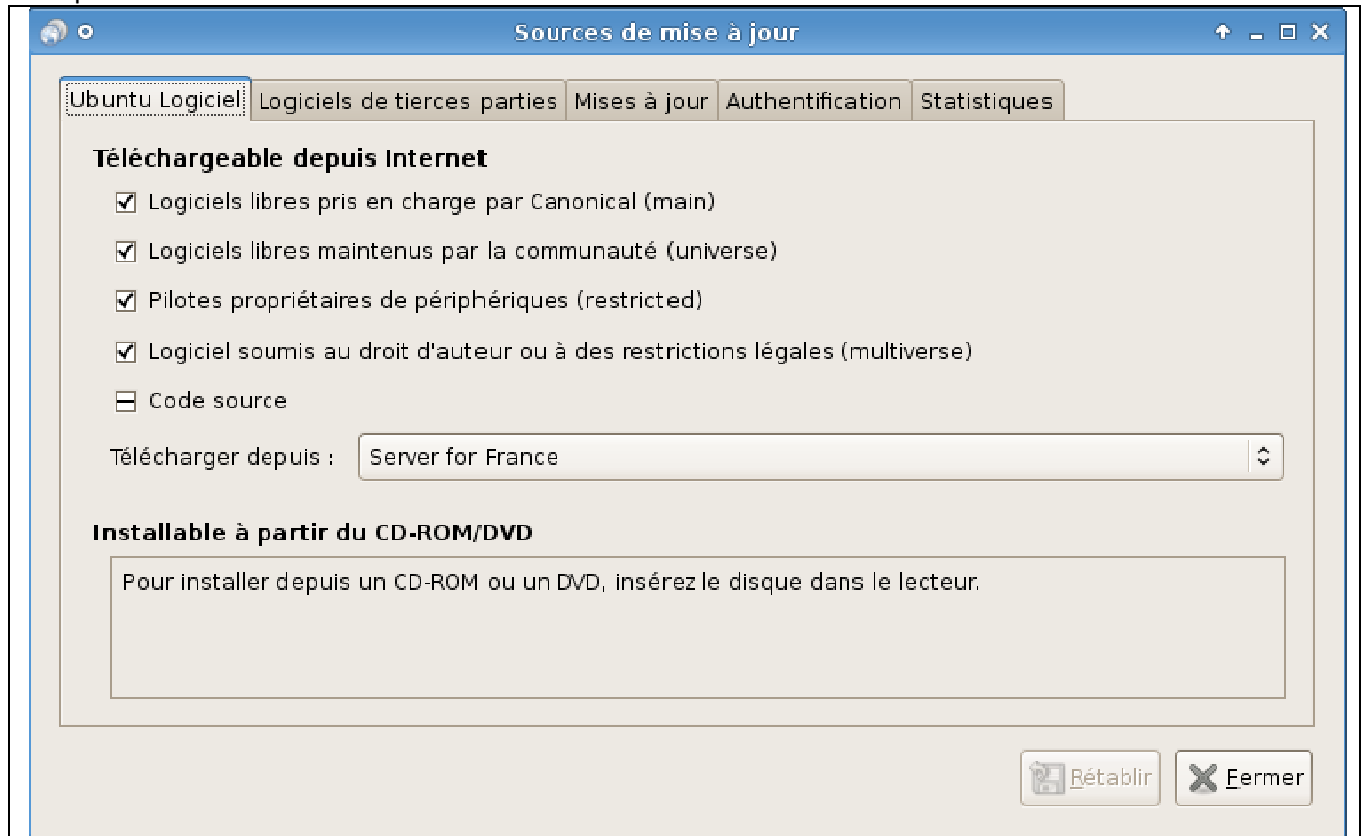
- 1_ = adresse du dépôt (<http://fr.archive.ubuntu.com/ubuntu/>)
- 2_ = nom de la version de la distribution (« hardy »)
- 3_ et suivants = « section » du dépôt dans laquelle vous voulez regarder.

Trouver les dépôts : rechercher sur le web « dépôts ubuntu »

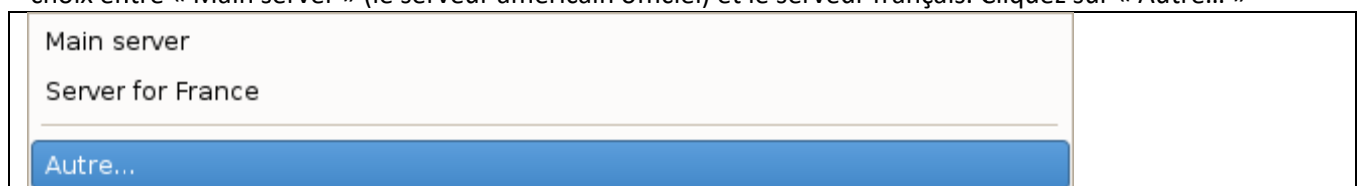
- **utiliser l'outil graphique** (selon la distribution)

- **Ubuntu** : Système→Administration→Sources de logiciels ;
- **KUbuntu** : Menu K→Système→Gestionnaire Adept→Adept→Gérer les dépôts ;
- **XUbuntu** : Applications→Système→Sources de mises à jour.

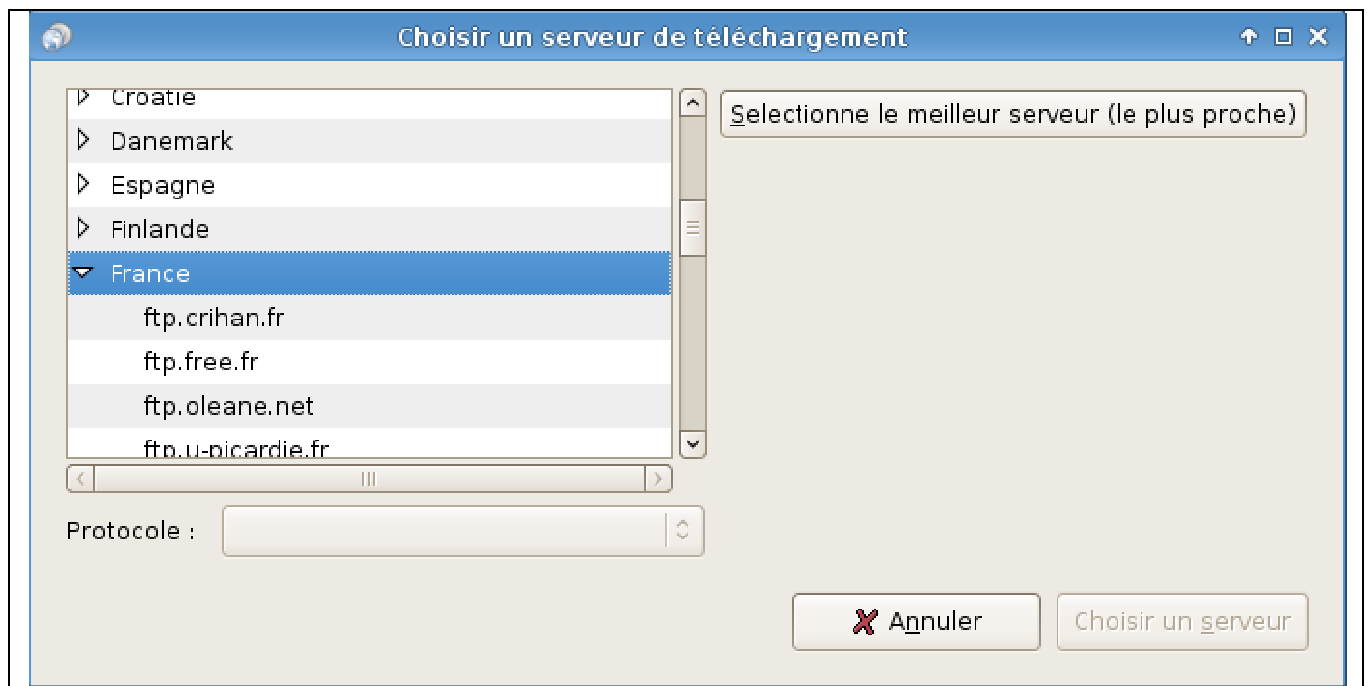
Exemple : Xubuntu



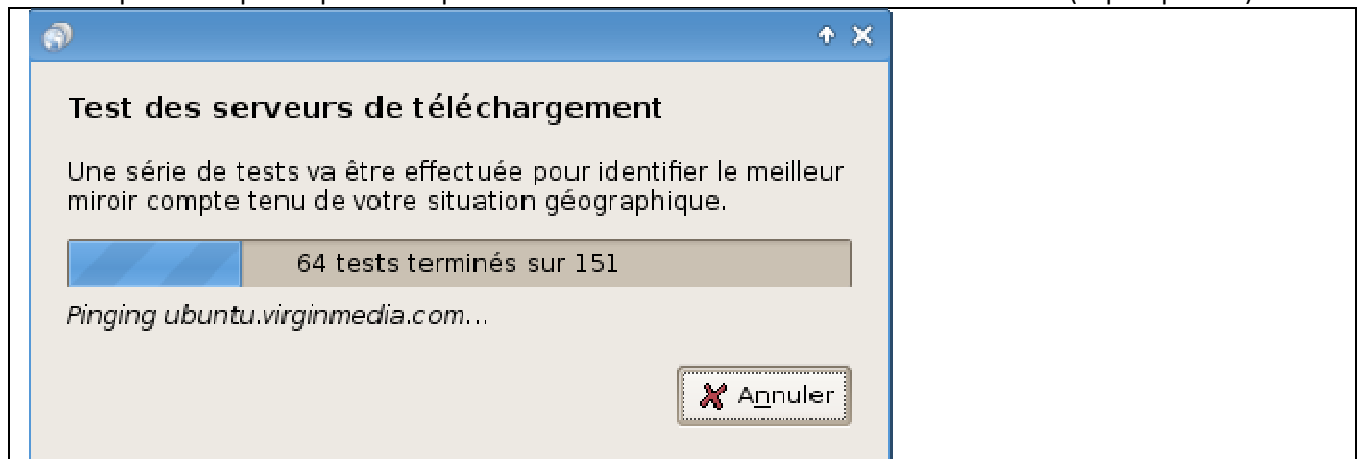
liste déroulante « Télécharger depuis : Server for France » = dépôts français officiels d'Ubuntu
= choix entre « Main server » (le serveur américain officiel) et le serveur français. Cliquez sur « Autre... »



fenêtre recense dépôts les +connus par pays (universités (u-picardie.fr), FAI (Free, Olean pour Orange...)).

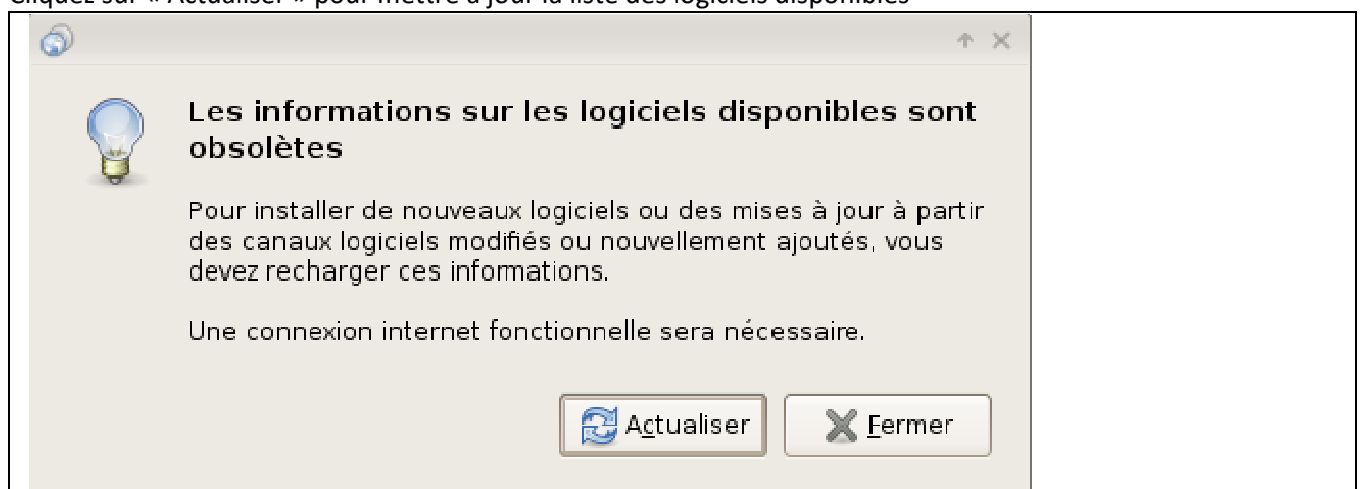


Savoir lequel est le plus rapide = cliquez sur le bouton « Sélectionner le meilleur serveur (le plus proche) »



Cliquez sur « Choisir un serveur » pour valider celui qui aura été retenu

En cas de changement de serveur, il faut mettre à jour les paquets en cache
Cliquez sur « Actualiser » pour mettre à jour la liste des logiciels disponibles



Les outils de gestion des paquets

rappel vocabulaire

- **paquet** : programme « prêt à l'emploi » # programmes d'installation sous Windows
- **dépendance** : un paquet peut avoir besoin de plusieurs autres paquets pour fonctionner

- **dépôt** : serveur sur lequel on va télécharger nos paquets.

outils

- Interface graphique : gère les paquets pour nous = logithèque Ubuntu
- Console : 2 programmes **apt-get** (+célèbre) & **aptitude** (reconnu plus efficace pour la désinstallation) (aujourd'hui apt-get gère aussi les dépendance inutilisées à la désinstallation)

3 étapes du téléchargement

- **apt-get update** (optionnel) : pour mettre notre cache à jour
- **apt-cache search monpaquet** (optionnel) : rechercher le paquet à télécharger si ne connaît pas nom exact
- **apt-get install monpaquet** : pour télécharger et installer notre paquet.

apt-get update = mettre à jour le cache des paquets

= télécharger la nouvelle liste des paquets proposés par le dépôt.

2 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).

commande (comme root) : **sudo apt-get update**

```
root@mateo21-desktop:~# apt-get update
Réception de : 1 http://wine.budgetdedicated.com hardy Release.gpg [191B]
...
Lecture des listes de paquets... Fait
```

apt-cache search : rechercher un paquet si on le connaît pas le nom exact du paquet

commande : **apt-cache search votrerecherche** = liste tous les paquet en rapport

= évite d'avoir à aller sur Internet pour faire la recherche, ce qui aurait été lent.

exemple : un jeu de casse-briques (*breakout* en anglais)

```
root@mateo21-desktop:~# apt-cache search breakout
briquolo - Fast paced 3d Breakout
briquolo-data - Fast paced 3d Breakout data files
circuslinux - The clowns are trying to pop balloons to score points!
circuslinux-data - data files for circuslinux
gnome-breakout - Clone of the classic game Breakout, written for GNOME
lbreakout2 - A ball-and-paddle game with nice graphics
lbreakout2-data - A ball-and-paddle game with nice graphics (DATA FILES)
libfreebob0 - FreeBoB API
libfreebob0-dev - FreeBoB API - development files
tecnoballz - breaking block game ported from the Amiga platform
```

légende : à gauche nom du paquet, à droite courte description.

description : **apt-cache show breakout2**

apt-get install : installer un paquet

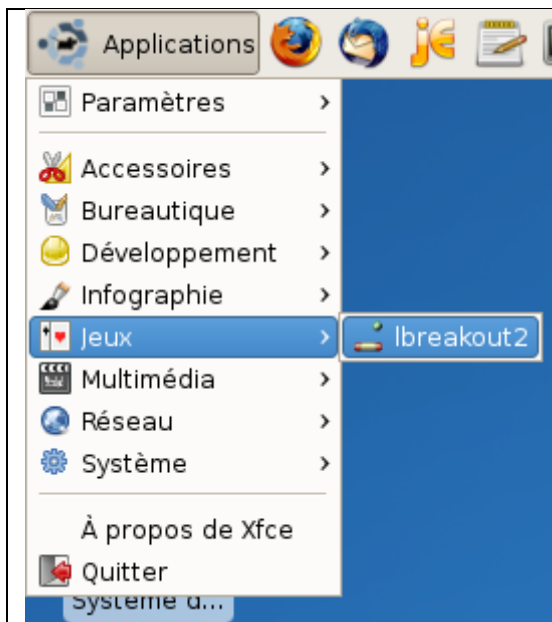
syntaxe : **sudo apt-get install paquet1 paquet2 paquet3**

commande : **sudo apt-get install lbreakout2**

```
root@mateo21-desktop:~# apt-get install lbreakout2
Lecture des listes de paquets... Fait
...
```

légende :

- vérifie si le paquet et les dépendances existent
 - absence de la dépendance commelibsdlmixer-1.2 notamment => demande confirmation [O/n]
 - Répondre « O » MAJ + Entrée => télécharge tout seul le paquet sur le dépôt & toutes les dépendances utiles
 - puis il « dépaquète » les fichiers du paquet, les installe et effectue les paramétrages tout seul.
- = **apt-get installe tout ce qu'il faut, dans les bons répertoires, crée même le raccourci dans « Jeux » !**



exercice : OpenArena, basé sur le célèbre jeu Quake III Arena dont le code source est devenu libre.

sudo apt-get install openarena simple ! (fortement conseillé d'avoir installé pilotes de la carte graphique avant)

apt-get autoremove : supprimer un paquet

commande : **sudo apt-get remove lbreakout2** MAIS ne supprime pas les dépendances devenues inutiles

commande : **sudo apt-get autoremove lbreakout2** supprime aussi les dépendances

```
root@mateo21-desktop:~# apt-get autoremove lbreakout2
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture de l'information d'état... Fait
Les paquets suivants ont été automatiquement installés mais ne sont plus nécessaires :
  libSDL-mixer1.2 libsmpeg0
Les paquets suivants seront ENLEVÉS :
  lbreakout2 lbreakout2-data libSDL-mixer1.2 libsmpeg0
0 mis à jour, 0 nouvellement installés, 4 à enlever et 0 non mis à jour.
Il est nécessaire de prendre 0o dans les archives.
Après dépaquetage, 5358ko d'espace disque seront libérés.
Souhaitez-vous continuer [O/n] ?
apt-get analyse le paquet = vérifie dépendances qu'il peut supprimer sans risque
Suppression de lbreakout2 ...
Suppression de lbreakout2-data ...
Suppression de libSDL-mixer1.2 ...
Suppression de libsmpeg0 ...
```

apt-get upgrade : mettre à jour tous les paquets

commande : **sudo apt-get upgrade**

= compare la version de vos paquets installés avec ceux présents dans le cache.

```
root@mateo21-desktop:~# apt-get upgrade
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture de l'information d'état... Fait
0 mis à jour, 0 nouvellement installés, 0 à enlever et 0 non mis à jour.
```

légende : ici pas de maj nécessaire = système à jour

recommandation : à faire régulièrement pour avoir le système le plus à jour possible (fonctionnalités, sécurité)

8_ READ THE FUCKING MANUAL = « Lis ce p* de manuel »** lire la documentation doit devenir un réflexe

un des chapitres les plus importants du cours, car si vous savez lire la doc, vous êtes capables de tout apprendre

man : afficher le manuel d'une commande (abréviation de manuel) **man mkdir**

Se déplacer dans le manuel (aussi avec la molette de la souris en mode graphique)

- flèche haut / flèche bas = ligne par ligne
- Page Up / Page Down (ou Espace) = de page en page.

- Home = revenir au début du manuel / Fin = aller à la fin
- /(slash) = effectuer une recherche => tapez ensuite le **mot recherché + Enter**, **/+Enter** pour passer au suivant.
- Q = quitter

Les principales sections du manuel

le manuel de la commande est découpé en plusieurs sections

- **NAME** : nom de la commande + courte description de son utilité
- **SYNOPSIS** : liste de toutes les façons d'utiliser la commande
- **DESCRIPTION** : description plus approfondie de ce que fait la commande + liste des paramètres & signification
- **AUTHOR** : auteur du programme (parfois nombreux avec les logiciels libres)
- **REPORTING BUGS** : adresse si vous rencontrez un bug dans le logiciel
- **COPYRIGHT** : licence d'utilisation de la commande = la plupart open source sous licence GPL
- **SEE ALSO** : autres commandes en rapport avec celle-ci

La langue des pages du manuel

il existe une version française des pages de manuel : **apt-get install manpages-fr**

MAIS **pas recommandé** (moins à jour, pas traduites, plus d'erreurs) => **apt-get autoremove manpages-fr**

Section SYNOPSIS une des sections les plus importantes mais aussi une des plus difficiles à lire

= **lister toutes les façons possibles d'utiliser la commande** = toutes les combinaisons de paramètres

exemple : **man mkdir** = « **mkdir - make directories** »

mkdir [OPTION] DIRECTORY...

- **Mkdir** = taper mkdir
- **[OPTION]** = options possibles (crochets = facultatif)
- **DIRECTORY** = nom du répertoire à créer (pas de crochet = obligatoire)
- **...** = (points de suspension = on peut répéter DIRECTORY autant de fois que l'on veut)

remarque : **mkdir** en gras (=à taper tel quel) & **OPTION** et **DIRECTORY** soulignés (= à remplacer par valeurs)

section DESCRIPTION

```
DESCRIPTION
Create the DIRECTORY(ies), if they do not already exist.
Mandatory arguments to long options are mandatory for short options too.

-m, --mode=MODE
set file mode (as in chmod), not a=rwx - umask

-p, --parents
no error if existing, make parent directories as needed

-v, --verbose
print a message for each created directory

-Z, --context=CTX
set the SELinux security context of each created directory to CTX

-help display this help and exit

--version
output version information and exit
```

exemple1 : **man mkdir**

```
mkdir -v images videos musiques
mateo21@mateo21-desktop:~/tests$ mkdir -v images videos musiques
mkdir: création du répertoire `images'
mkdir: création du répertoire `videos'
mkdir: création du répertoire `musiques'
```

exemple2 : **man cp**

DESCRIPTION

Copy SOURCE to DEST, or multiple SOURCE(s) to DIRECTORY.

SYNOPSIS

```
cp [OPTION]... [-T] SOURCE DEST
cp [OPTION]... SOURCE... DIRECTORY
cp [OPTION]... -t DIRECTORY SOURCE...
```

Plusieurs lignes = plusieurs usages

- cp [OPTION]... [-T] SOURCE DEST = obligatoire SOURCE DEST, x options possibles, option -T
- cp [OPTION]... SOURCE... DIRECTORY = 1 ou x SOURCE... vers un répertoire DIRECTORY + x options possibles
- cp [OPTION]... -t DIRECTORY SOURCE... = 1 DIRECTORY suivi 1 ou x SOURCE... + -t obligatoire

exemples :

- cp photo.jpg photo_copie.jpg = créer la copie photo_copie.jpg.

options dans section DESCRIPTION

cp -vi photo.jpg photo_copie.jpg = verbose + confirmation si le fichier existe déjà

```
mateo21@mateo21-desktop:~$ cp -vi photo.jpg photo_copie.jpg
cp: écraser `photo_copie.jpg'? o
`photo.jpg' -> `photo_copie.jpg'
```

- cp photo.jpg photo_copie.jpg images/ = copie deux fichiers dans le sous-dossier images/

exemple2 : man apt-get

SYNOPSIS = tout sur une seule ligne !!!

```
apt-get [-hvs] [-o=option de configuration] [-c=fichier] {[update] |
[upgrade] | [dselect-upgrade] | [install paquet...]} |
[remove paquet...] | [source paquet...] | [build-dep paquet...]
| [check] | [clean] | [autoclean]}
```

Légende :

- doit commencer par « apt-get » (gras)
- 1 ou x options parmi -hvs
- -o et -c = options facultatives mais avec une valeur (-o=option de configuration -c=fichier)
- section en accolades
 {[update] | [upgrade] | [dselect-upgrade] | [install paquet...]} | [remove paquet...] | [source paquet...] | [build-dep paquet...] | [check] | [clean] | [autoclean]}
- barres verticales « | » = « OU » = une et une seule option issue de la liste entre accolades
 - update : met à jour le cache des paquets disponibles sur votre ordinateur
 - upgrade : met à jour tous les paquets installés si une nouvelle version est disponible
 - install paquet... : installe le ou les paquets demandés
 - etc.

remarque : section DESCRIPTION pour avoir la signification de chacun des mots clés.

Exemples

- apt-get install monpaquet
 - apt-get update
 - apt-get autoclean
- mais pas apt-get update install monpaquet

Résumé de la syntaxe du SYNOPSIS

- **gras** : tapez le mot exactement comme indiqué ;
- souligne : remplacez le mot souligné par la valeur qui convient dans votre cas ;
- [-hvc] : toutes les options -h, -v et -c sont facultatives ;
- a|b : vous pouvez écrire l'option « a » OU « b », mais pas les deux à la fois ;
- option... : points de suspension indiquent que l'option peut être répétée autant de fois que vous voulez.

apropos : trouver une commande

Le man suppose de connaître la commande.

commande : **apropos sound**

```
mateo21@mateo21-desktop:~$ apropos sound
alsactl (1)      - advanced controls for ALSA soundcard driver
alsamixer (1)   - soundcard mixer for ALSA soundcard driver, with ncurses...
amixer (1)      - command-line mixer for ALSA soundcard driver
aplay (1)       - command-line sound recorder and player for ALSA soundc...
```

```
arecord (1)          - command-line sound recorder and player for ALSA soundc...
artscat (1)          - pipe data to sound device
asoundconf (1)       - utility to read and change the user's ALSA library con...
```

Autres façons de lire le manuel

Le paramètre -h (et --help) = affichage d'une aide résumée

remarque : cette aide est **plus facile à lire** que celle du man

exemple : apt-get -h

```
mateo21@mateo21-desktop:~$ apt-get -h
apt 0.7.9ubuntu15 pour amd64 compilé sur Mar 14 2008 00:00:28
Usage: apt-get [options] command
       apt-get [options] install|remove pkg1 [pkg2 ...]
       apt-get [options] source pkg1 [pkg2 ...]

apt-get is a simple command line interface for downloading and
installing packages. The most frequently used commands are update
and install.

Commands:
  update - Retrieve new lists of packages
  upgrade - Perform an upgrade
  install - Install new packages (pkg is libc6 not libc6.deb)
  remove - Remove packages
  autoremove - Remove all automatic unused packages
  purge - Remove and purge packages
  source - Download source archives
  build-dep - Configure build-dependencies for source packages
  dist-upgrade - Distribution upgrade, see apt-get(8)
  dselect-upgrade - Follow dselect selections
  clean - Erase downloaded archive files
  autoclean - Erase old downloaded archive files
  check - Verify that there are no broken dependencies

Options:
  -h This help text.
  -q Loggable output - no progress indicator
  -qq No output except for errors
  -d Download only - do NOT install or unpack archives
  -s No-act. Perform ordering simulation
  -y Assume Yes to all queries and do not prompt
  -f Attempt to correct a system with broken dependencies in place
  -m Attempt to continue if archives are unlocatable
  -u Show a list of upgraded packages as well
  -b Build the source package after fetching it
  -V Show verbose version numbers
  -c=? Read this configuration file
  -o=? Set an arbitrary configuration option, eg -o dir::cache=/tmp
See the apt-get(8), sources.list(5) and apt.conf(5) manual
pages for more information and options.

This APT has Super Cow Powers.
```

La commande whatis sorte de man très allégé = juste l'en-tête du manuel

exemple : whatis mkdir

```
mateo21@mateo21-desktop:~$ whatis mkdir
mkdir (1)          - make directories
```

Rechercher man sur le Web : le manuel est dispo en ligne

9_ RECHERCHE DE FICHIERS

locate : une recherche rapide

Utiliser locate

un fichier particulier :

```
mateo21@mateo21-desktop:~$ locate notes.txt
/home/mateo21/notes.txt
```

tous les fichiers/dossiers dont le nom contient :

```
mateo21@mateo21-desktop: /var/log$ locate australie
/home/mateo21/photos/australie1.jpg
/home/mateo21/photos/australie2.jpg
/home/mateo21/photos/australie3.jpg
```

Remarque : une autre commande slocate vérifie les droits utilisateurs = affiche ceux en lecture seulement
Pas installée sur toutes les distributions (à installer avec apt-get)

La base de données des fichiers

Problème : Je viens de créer un fichier avec touch par exemple, et locate ne me renvoie aucun résultat.

Pourquoi ?

- locate ne fait pas la recherche sur disque entier, mais seulement sur base de données
- les fichiers tout juste créés n'ont pas encore été répertoriés dans la base de données
- Mise à jour une fois par jour !

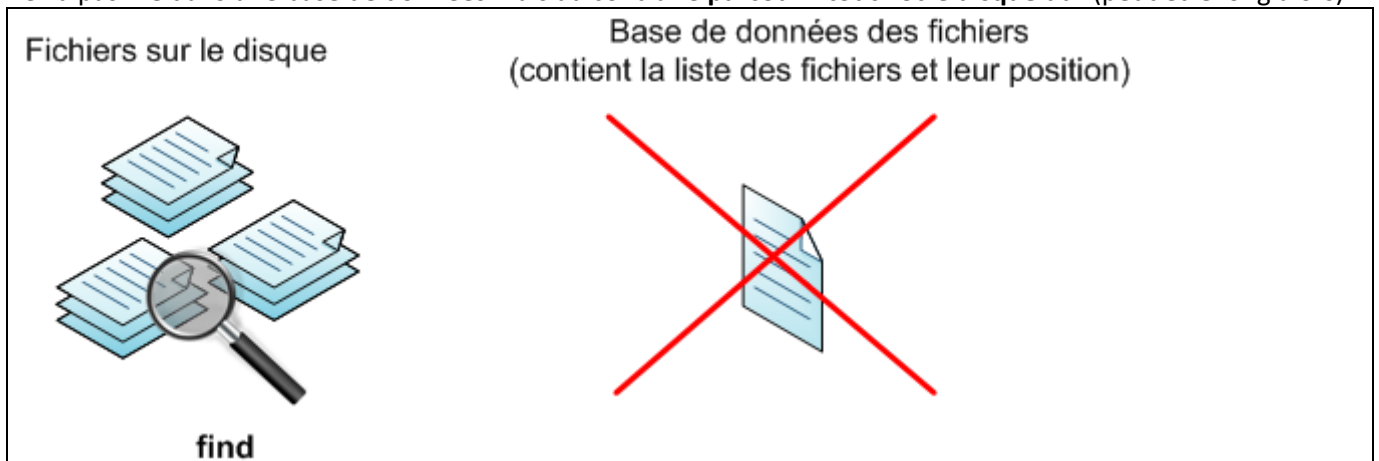
Forcer la mise à jour de la base **sudo updatedb**

find : une recherche approfondie

commande de recherche par excellence pour retrouver des fichiers mais aussi effectuer des opérations/résultats

find recherche les fichiers actuellement présents

ne va pas lire dans une base de données mais au contraire **parcourir tout votre disque dur** (peut être long alors)



Fonctionnement de la commande find

find « où » « quoi » « que faire avec »

- **Où (facultatif)** : nom du dossier de recherche (sous-dossiers inclus)
exp : find /home, par défaut dans le dossier courant
- **Quoi (obligatoire)** : fichier à rechercher (par son nom, sa date de création, sa taille, etc)
- **Que faire avec (facultatif)** : « post-traitement », par défaut affiche la liste des fichiers

Utilisation basique de la commande find

Recherche à partir du nom

```
mateo21@mateo21-desktop: ~$ find -name "logo.png"
/home/mateo21/projet/images/logo.png
```

Si la recherche n'affiche rien, cela signifie qu'aucun fichier n'a été trouvé.

Sans paramètre Où, cherche dans le répertoire courant, ici ~ = /home/mateo21/

```
mateo21@mateo21-desktop: ~$ find /var/log/ -name "syslog*"
/var/log/syslog.3.gz
/var/log/syslog.5.gz
/var/log/syslog.4.gz
/var/log/syslog
/var/log/syslog.6.gz
/var/log/syslog.2.gz
/var/log/syslog.1.gz
/var/log/installer/syslog
/var/log/syslog.0
```

Recherche à partir de la taille exp : fichiers qui font plus de 10 Mo

K pour les Ko, G pour les Go, etc

« - » « + » ou rien = égal

```
mateo21@mateo21-desktop:/var$ find ~ -size +10M
/home/mateo21/souvenirs.avi
/home/mateo21/backups/backup_mai.gz
/home/mateo21/backups/backup_juin.gz
```

Recherche à partir de la date du dernier accès : fichier .odt, il ya moins de 7 jours

```
mateo21@mateo21-desktop:~$ find -name "*.odt" -atime 6
/home/mateo21/ecriture/resume_infos_juin.odt
```

Remarque : 6 et non 7 car la numérotation commence à 0 !

Rechercher uniquement des répertoires ou des fichiers : par défaut, find cherche des répertoires ET des fichiers

- -type d : répertoires (*directories*) ;
- -type f : fichiers (*files*).

exemple : `find /var/log -name "syslog" -type d`

Utilisation avancée avec manipulation des résultats

Remarque : `find -name "*.jpg" = find -name "*.jpg" -print` (print signifie « afficher les résultats trouvés »)

Afficher les fichiers de façon formatée

```
mateo21@mateo21-desktop:~$ find . -name "*.jpg" -printf "%p - %u\n"
./photos/australie1.jpg - mateo21
./photos/australie2.jpg - mateo21
./photos/australie3.jpg - mateo21
```

Légende = affiche le nom du fichier, un tiret et le nom du propriétaire de ce fichier. Le\n permet d'aller à la ligne.

Voir tous les possibilités sur la doc.

Supprimer les fichiers trouvés (attention pas de confirmation)

Un des usages les plus courants de find, à part retrouver des fichiers, consiste à les supprimer

exemple : supprimer tous mes fichiers « jpg » : `find -name "*.jpg" -delete`

Appeler une commande

option exec = appelle une commande qui effectuera une action sur chacun des fichiers trouvés.

option -ok = idem mais avec confirmation ... à chaque fois.

exemple : mettre un chmod à 600 pour chacun de mes fichiers « jpg »

```
find -name "*.jpg" -exec chmod 600 {} \;
```

Pour chaque fichier.jpgtrouvé, on exécute la commande qui suit-exec :

- cette commande ne doit PAS être entre guillemets ;
- les accolades{} seront remplacées par le nom du fichier ;
- la commande doit finir par un\; obligatoirement.

Exercice : regrouper tous les fichiers.jpg éparpillés dans votre répertoire home dans un dossier images.