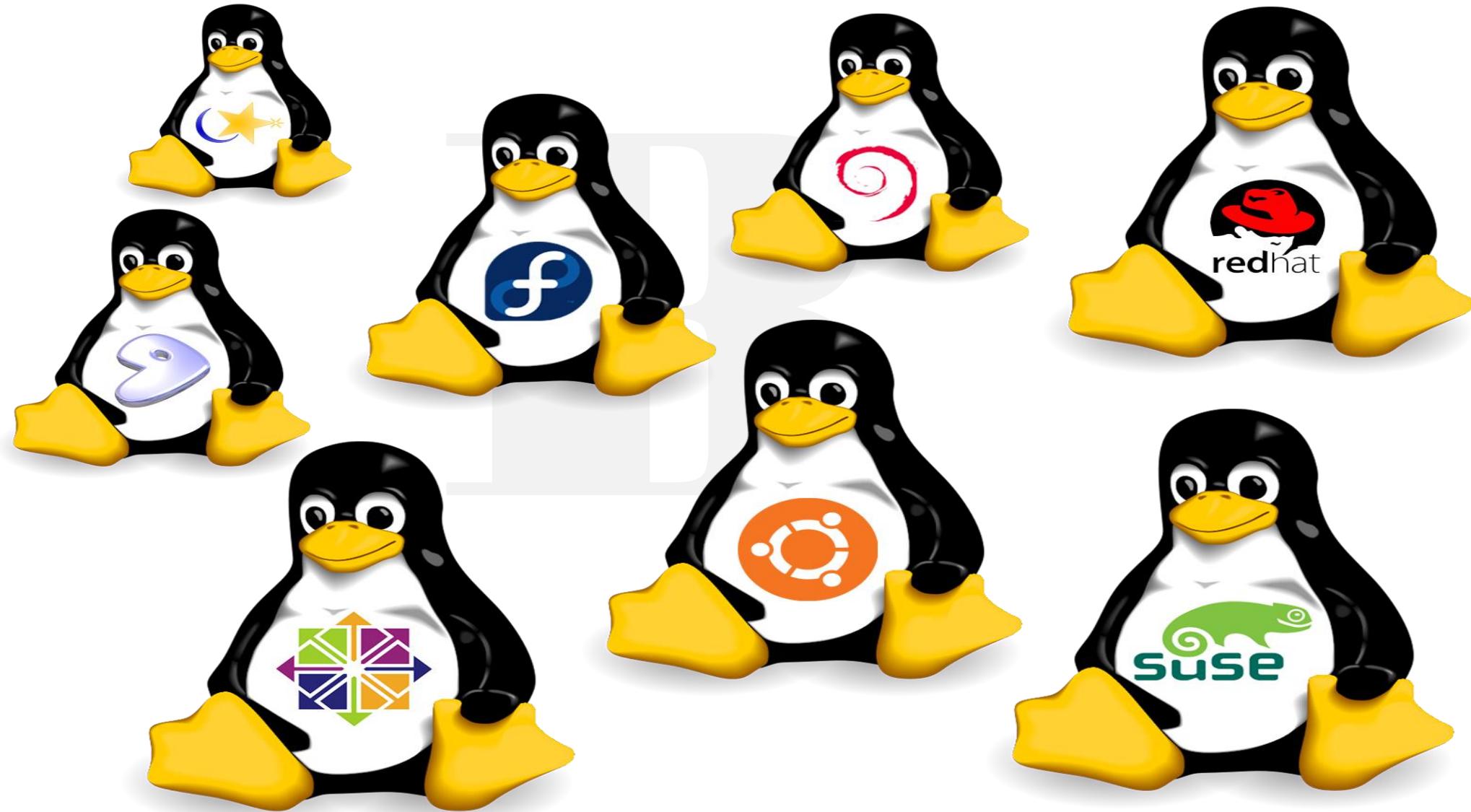


Système d'exploitation Linux





LINUX ???

🐧 Un système d'exploitation;

🐧 Réputé pour sa sécurité;

🐧 Mises à jour plus fréquentes;

🐧 Multitâches et Multi-utilisateurs, Multi-plate-forme.





AVANTAGES LINUX ???

- 🐧 **Gratuité** : Linux est gratuit. Toutefois, on peut acheter le support technique pour certaines distributions.
- 🐧 **Serveur réseau idéal**, fort caractère évolutif.
- 🐧 **Fiabilité et robustesse du système** : Linux est insensible aux spywares, aux virus et diverses menaces qui sont présents sur Windows.
- 🐧 **Mises à jour efficaces** : Le système et ses applications se mettent à jour facilement.
- 🐧 **Mobilisation active** d'une communauté pour le développement.
- 🐧 **Le système présent dans tous les secteurs d'activités**.
- 🐧 **Adaptable à tous les types de machines** :
 - 🐧 Poste de travail
 - 🐧 Ordinateurs portables
 - 🐧 Tablettes, OS embarqués, etc ...





INCONVENIENTS LINUX ???

- 🐧 Difficulté d' utiliser des logiciels Windows pour Linux.
- 🐧 Certains périphériques n'ont pas de pilote pour Linux, ou sont mal gérés.

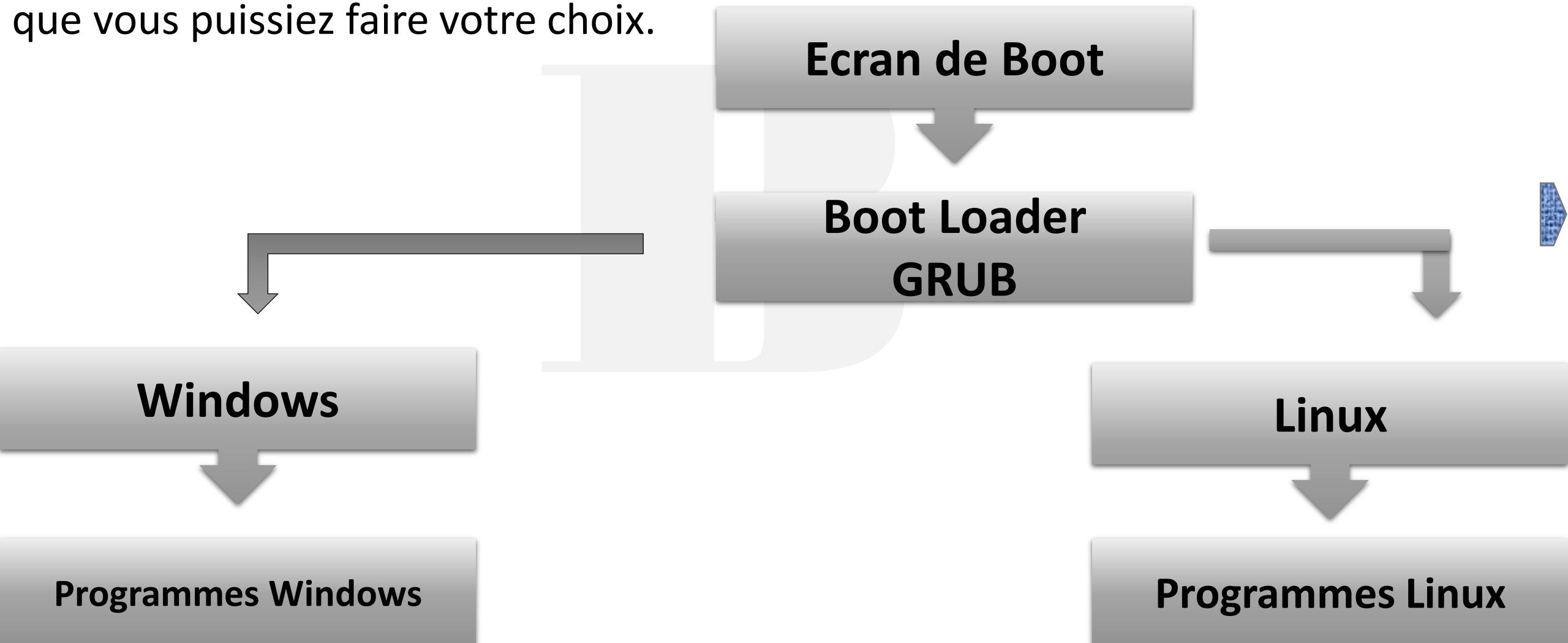
IB





Peut-on faire cohabiter deux OS sur son ordinateur ?

🐧 **oui!!!** Grace un programme appelé **Boot Loader (GRUB)** qui s'affichera pour que vous puissiez faire votre choix.





Qu'est-ce que le partitionnement ?

Le partitionnement est le fait d'avoir un disque physique découpé en plusieurs parties logiques (partitions). Sur chaque partition, les fichiers sont organisés selon ce qu'on appelle un système de fichiers.

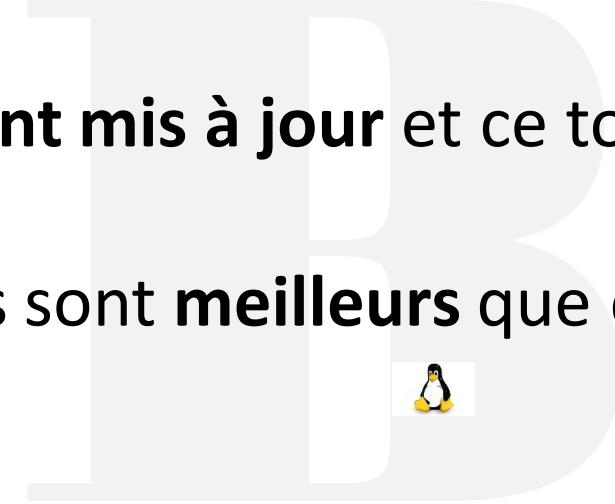
❖ Ex :





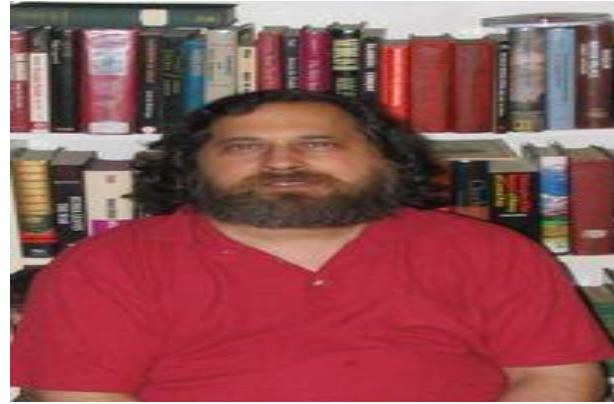
les programmes sous Linux?

- 🐧 Ils sont **gratuits** : vous verrez que sous Linux, presque tous les programmes sont gratuits ;
- 🐧 Les logiciels sont **souvent mis à jour** et ce toujours gratuitement
- 🐧 Certains de ces logiciels sont **meilleurs** que ceux que l'on trouve sous Windows





Linux: Historique



Richard Stallman, fondateur du projet GNU

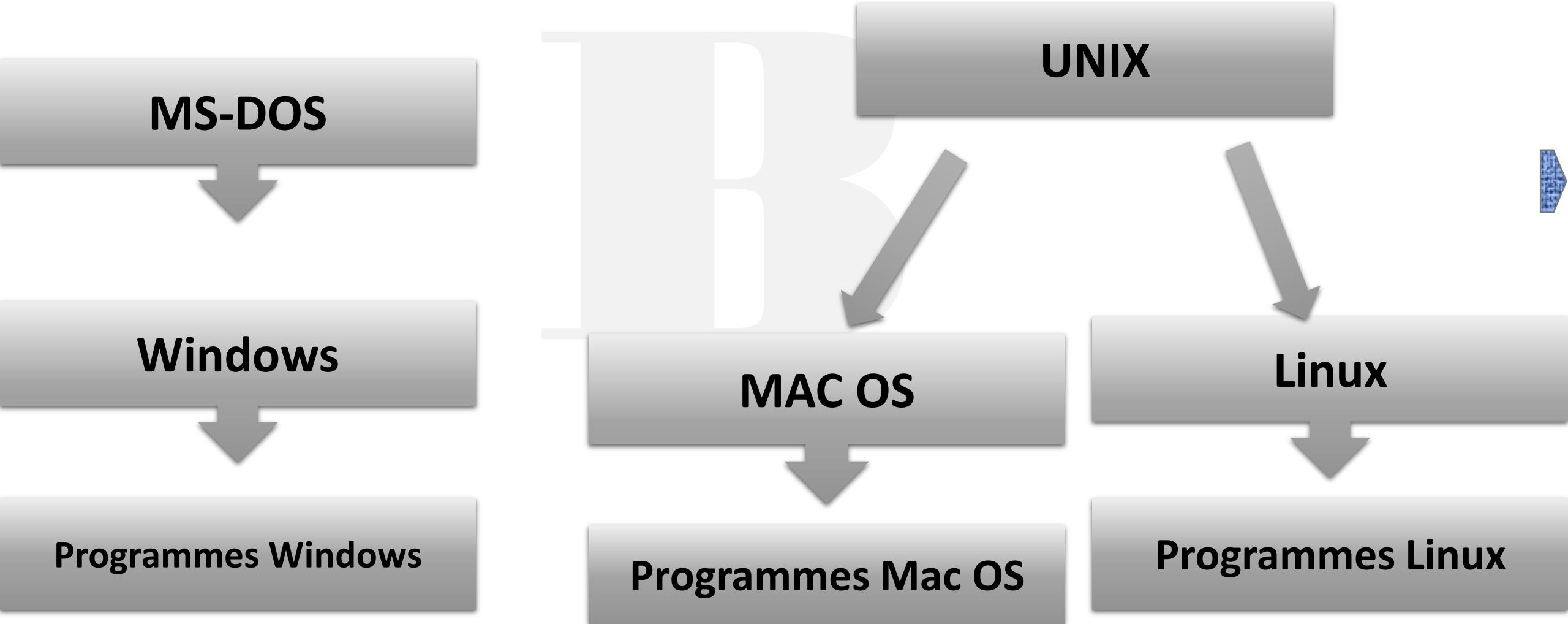


Linus Torvalds, créateur de Linux

- 🐧 en 1984 création du projet GNU
- 🐧 Pour un nouveau système d'exploitation fonctionnant comme **Unix** (les commandes restant les mêmes).
- 🐧 GNU ne devait pas seulement être un OS gratuit ; il devait également être « libre ».
- 🐧 En 1991, **Linus Torvalds**, un étudiant de l'Université de Helsinki (Finlande), entreprend de créer sur son temps libre son propre système d'exploitation.
- 🐧 Le projet GNU (programmes libres) et Linux (noyau d'OS) ont fusionné pour créer **GNU/Linux**.



Linux: Historique





LE NOYAU LINUX ???

- 🐧 Le **noyau (Kernel)** est le cœur du système d'exploitation Linux.

- 🐧 Le noyau Linux est **un logiciel libre** programmé principalement en langage C par des salariés et des bénévoles qui échangent par le biais de l'internet.

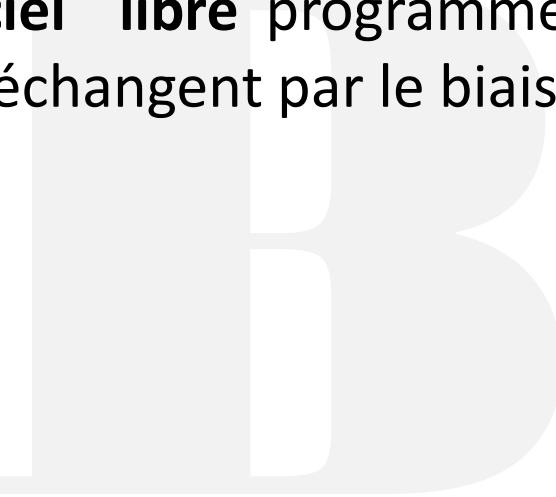
- 🐧 **Ses rôles sont:**
 - 🐧 La gestion des fichiers

 - 🐧 La gestion des processus

 - 🐧 La gestion des périphériques

 - 🐧 Les fonctions réseaux

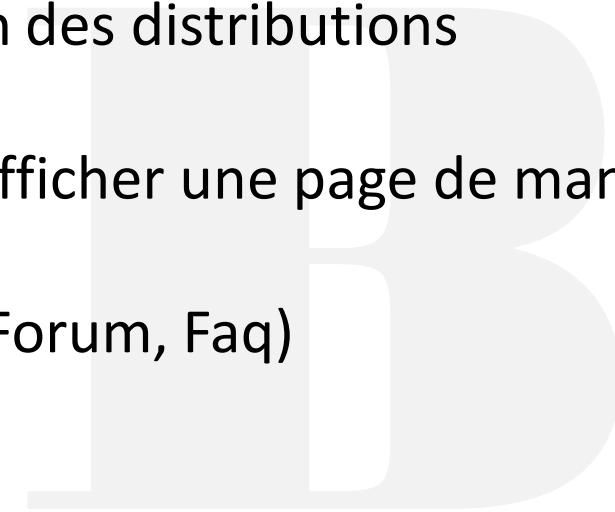
 - 🐧 Etc ...





Les sources d'information de Linux

- 🐧 La documentation de l'éditeur
- 🐧 Les guides d'administration des distributions
- 🐧 La commande `man` pour afficher une page de manuel
- 🐧 Les sites communautaire (Forum, Faq)
- 🐧 Les newsgroups
- 🐧 Les moteurs de recherche (google...), etc ...





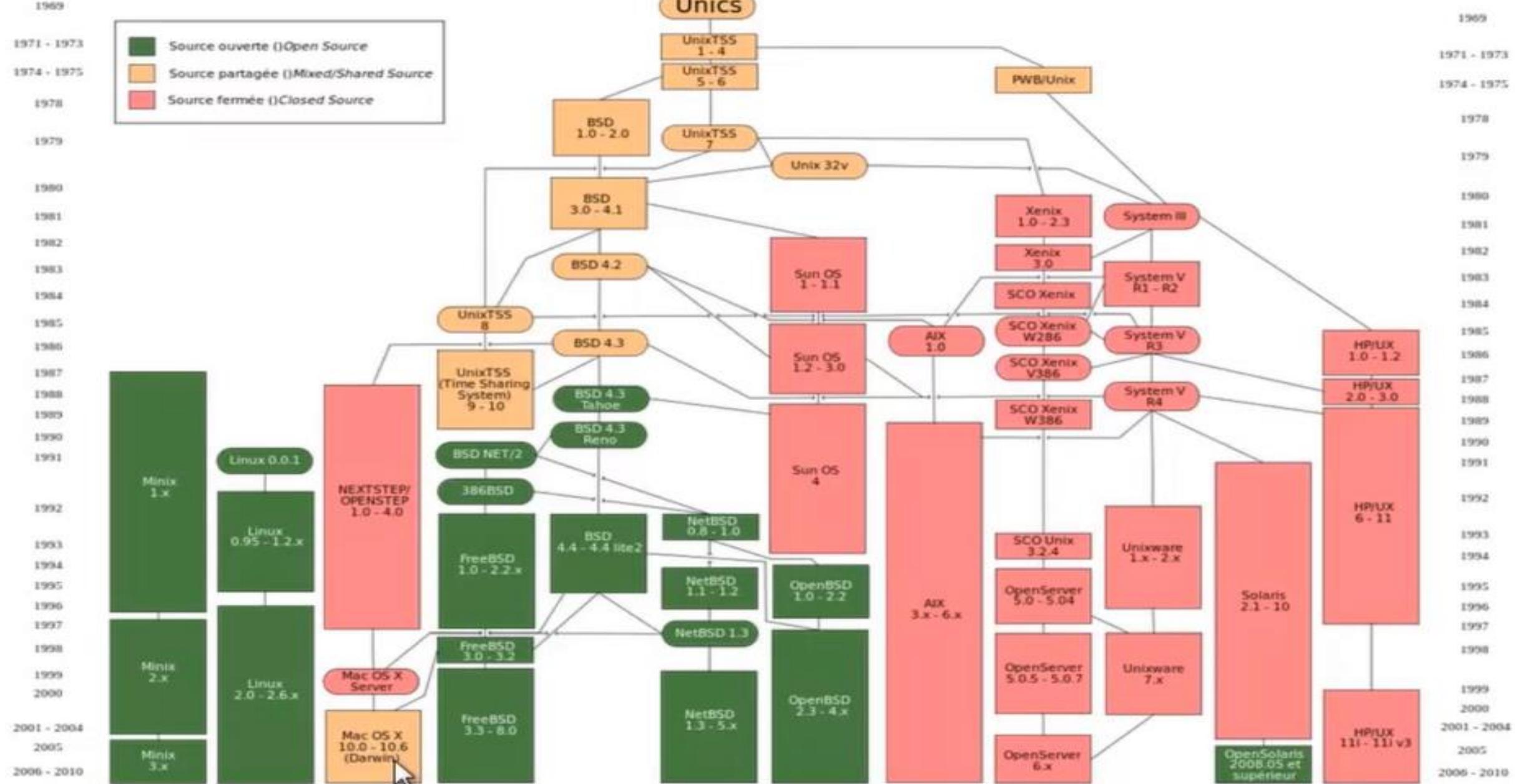
Les différents profils utilisateurs Linux

- 🐧 Les universitaires
- 🐧 Les administrateurs système
- 🐧 Les développeurs
- 🐧 Les débutants
- 🐧 Les utilisateurs professionnels



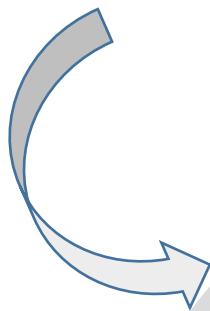


La famille « Unix »





Quelques distributions de Linux



Distribution= le noyau+ un système de fichiers + Un gestionnaire de paquets un programme d' installation



Quelques gestionnaires de packages Linux

- 🐧 **Yum (Yellow dog Updater Modified)** : Gestionnaire de paquets officiel de Red Hat. Il est utilisé dans les distributions Redhat (les versions Entreprise), fedora et toutes les autres distributions de type RPM
- 🐧 **RPM(Red Hat Package Manager)** : Gestionnaire de paquets , inventé par Red Hat, utilisé dans beaucoup d'autres distributions (CentOS, Fedora)
- 🐧 **Apt(Advanced Packaging Tool)** : Gestionnaire de paquets très utilisé avec la distribution debian et Ubuntu
- 🐧 **Aptitude** : Gestionnaire de paquets en mode console de haut niveau
- 🐧 **Synaptic** : Gestionnaire de paquets Debian en mode graphique
- 🐧 **Dpkg (Debian package)** : Gestionnaire de paquets de référence ,pour manipulation des paquets Debian





Notre Choix de Distribution Linux





Pourquoi Ubuntu ?



- 🐧 Des mises à jour fréquentes régulières
- 🐧 Implantation par plusieurs PME/PMI Françaises
- 🐧 Implantation par La Gendarmerie Nationale
- 🐧 Implantation par L' Assemblée Française
- 🐧 De plus en plus de particuliers l'utilisent

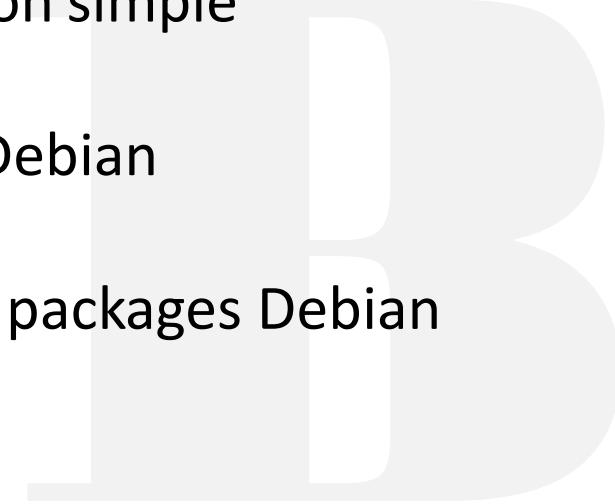




Quels sont les avantages d'Ubuntu ?



- 🐧 Un système d'installation très simple à appréhender
- 🐧 Une procédure d'installation simple
- 🐧 Une distribution issue de Debian
- 🐧 Une compatibilité avec les packages Debian
- 🐧 Une sortie tous les 6 mois
- 🐧 Une communauté très active

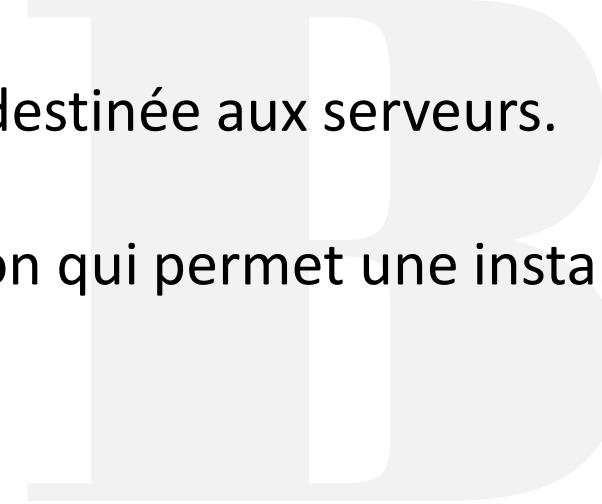




Les versions d'Ubuntu



- 🐧 **La version Desktop** : version pour une machine de bureau. Elle dispose d'une interface graphique.
- 🐧 **La version Server** : version destinée aux serveurs.
- 🐧 **La version Alternate** : version qui permet une installation modulable.





Les deux modes d'utilisation de Linux



Graphique

Unity

Gnome

XFCE

KDE

Autres

Console





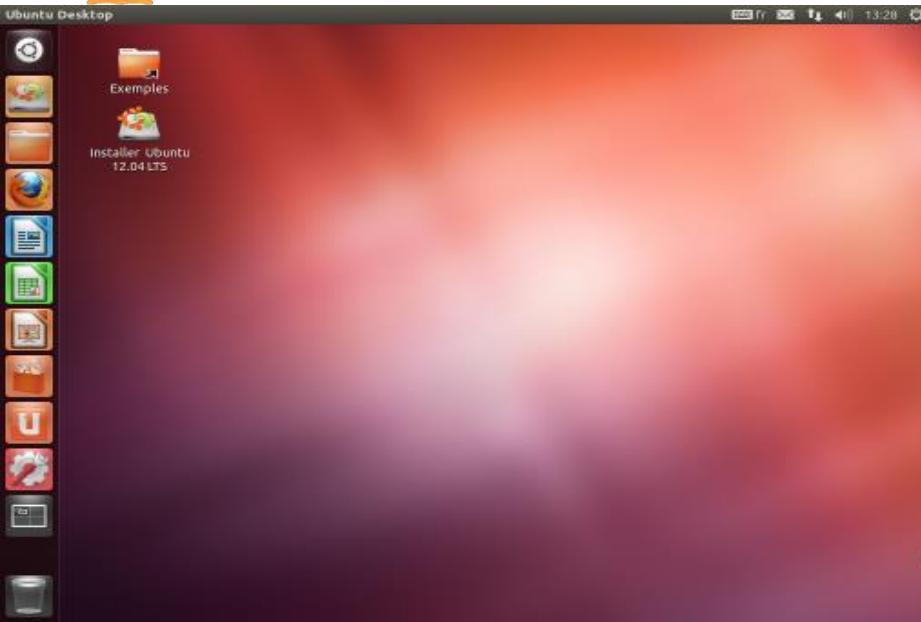
Linux: mode console



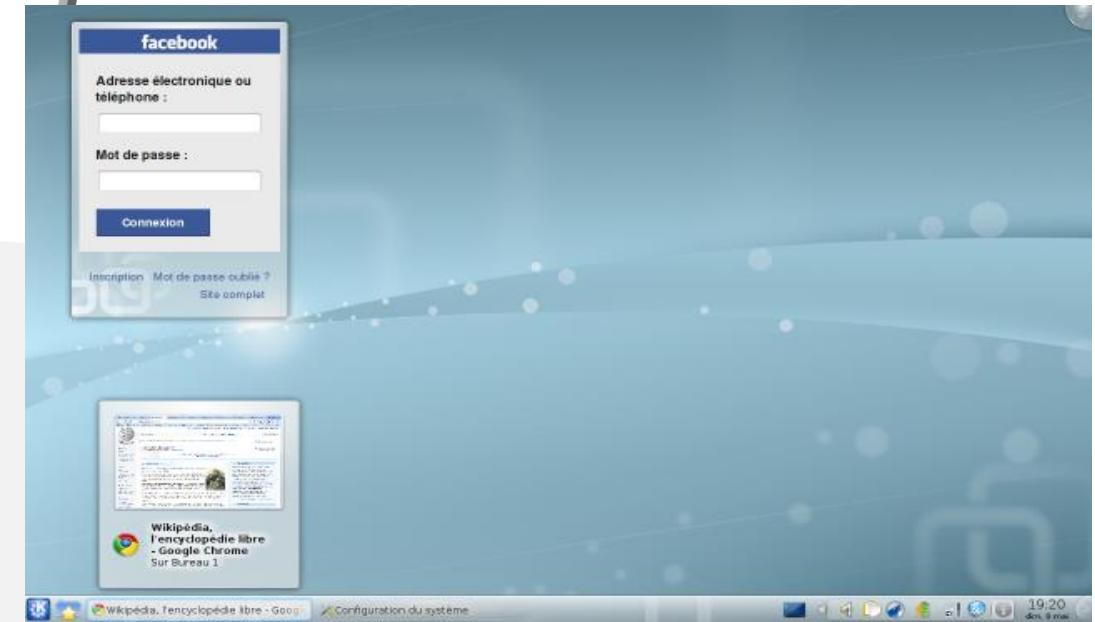
```
ubuntu@ubuntu:~/Desktop$ ls
Examples  ubiquity-kdeui.desktop
ubuntu@ubuntu:~/Desktop$ cd Examples
ubuntu@ubuntu:~/Desktop/Examples$ ls
book                               logo-Kubuntu.png
book-toc.html                      logo-Ubuntu.png
Experience ubuntu.ogg              oo-about-these-files.odt
fables_01_01_aesop.spx             oo-about-ubuntu-ru.rtf
gimp-ubuntu-splash.xcf            oo-access.odt
kubuntu-leaflet.png                oo-cd-cover.odg
logo-Edubuntu.png                 oo-derivatives.doc
ubuntu@ubuntu:~/Desktop/Examples$ pwd
/home/ubuntu/Desktop/Examples
ubuntu@ubuntu:~/Desktop/Examples$ w
 22:44:02 up 15 min,  7 users,  load average: 0.07, 0.29, 0.26
USER   TTY      FROM          LOGIN@    IDLE      JCPU     PCPU WHAT
ubuntu  tty1     -           22:30      0.00s   2.93s   0.02s  w
ubuntu  tty2     -           22:30      15:25m   0.17s   0.14s  -bash
ubuntu  tty3     -           22:30      15:25m   0.15s   0.12s  -bash
ubuntu  tty4     -           22:30      15:25m   0.17s   0.14s  -bash
ubuntu  tty5     -           22:30      15:25m   0.15s   0.13s  -bash
ubuntu  tty6     -           22:30      15:25m   0.17s   0.15s  -bash
ubuntu  :0       -           22:30      ?xdm?    50.06s  0.15s /bin/sh /usr/bi
ubuntu@ubuntu:~/Desktop/Examples$ _
```



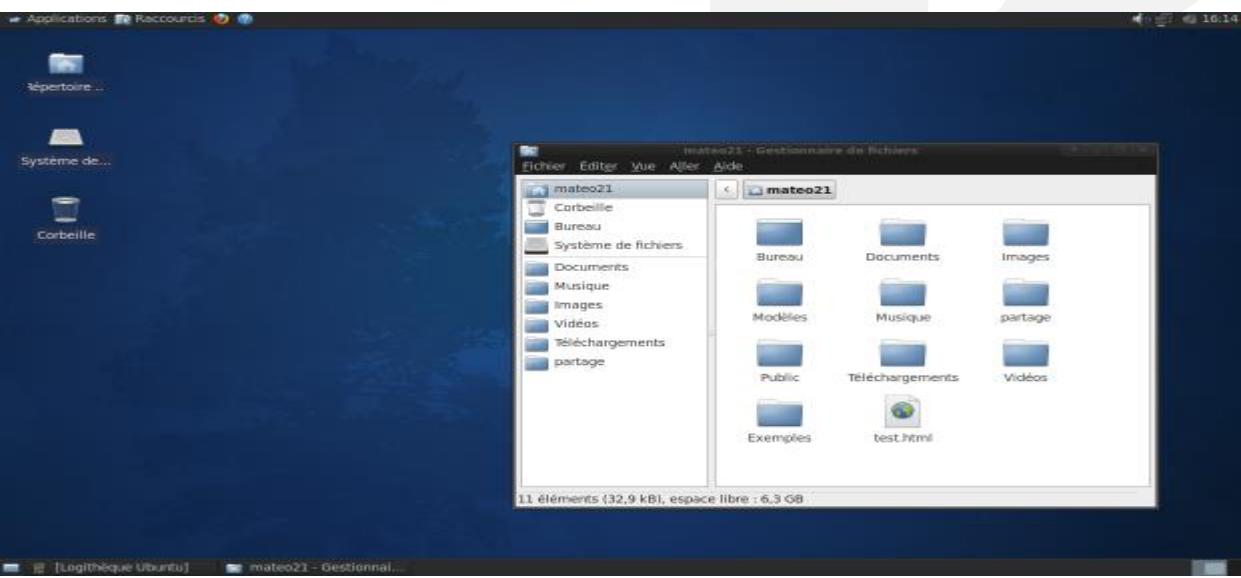
Linux: mode graphique



Ubuntu, basé sur Unity



Kubuntu, basé sur KDE



Xubuntu, basé sur XFCE



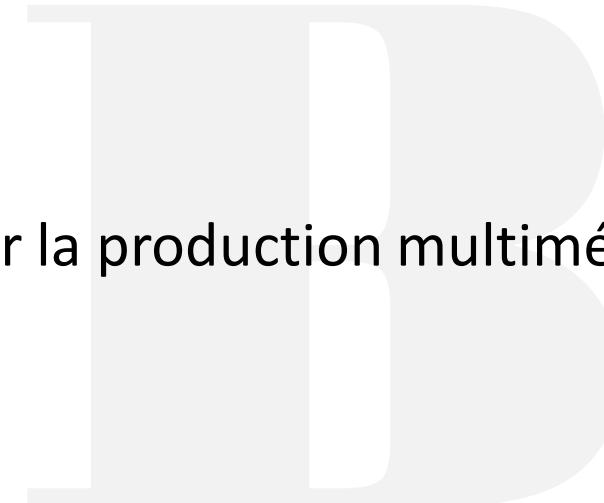
Déclinaisons de la version Desktop



Orientée pour les éducatifs



Orientée pour la production multimédia

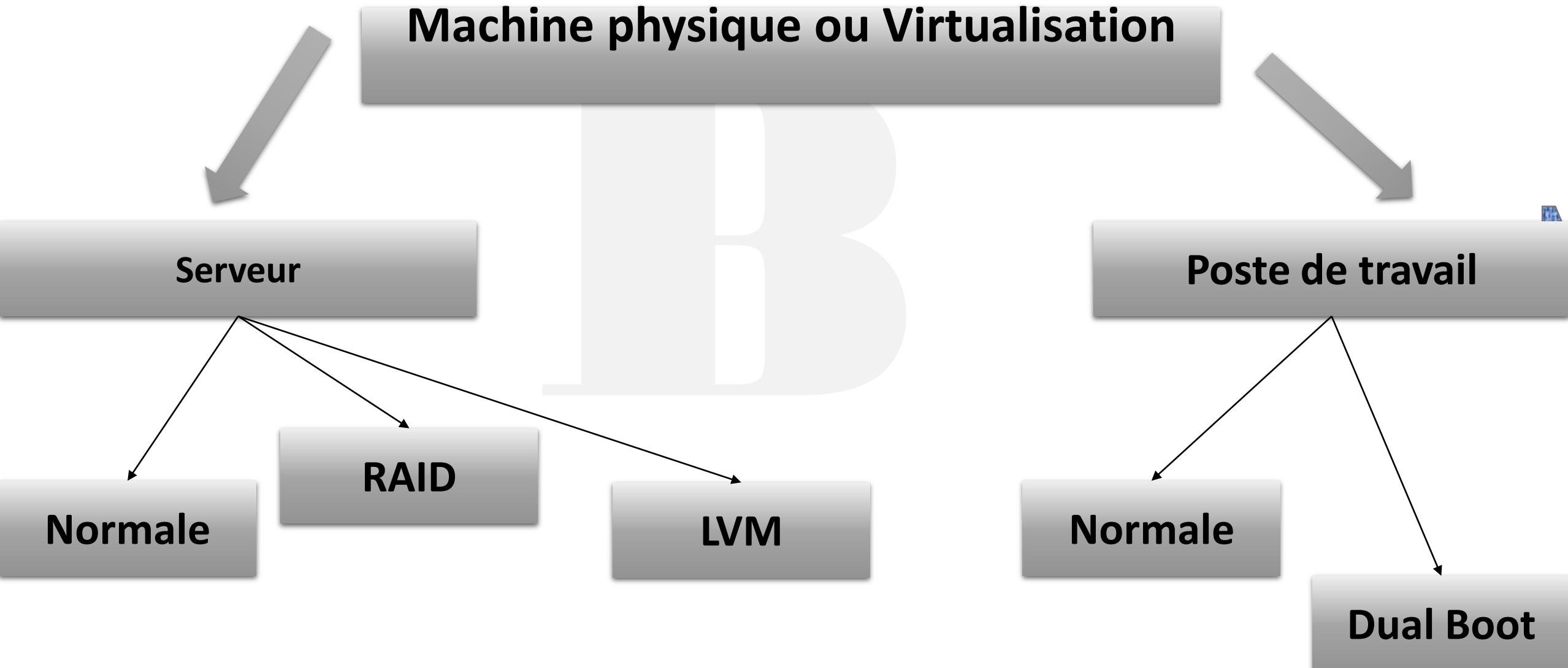


Orientée pour le centre de multimédia





Les types d'implantations d'Ubuntu





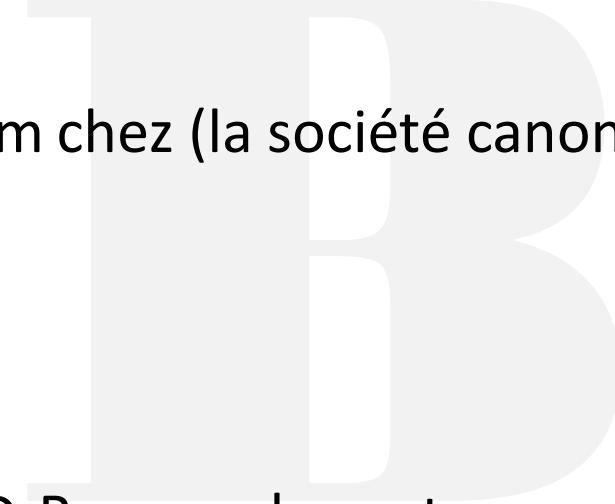
Les possibilités d'avoir Ubuntu



- 🐧 Télécharger sur un site comme <http://ubuntu-fr.org> et graver l' image ISO obtenue.

- 🐧 Acheter les CD-Rom/DVD-Rom chez (la société canonical ou un distributeur agréé).

- 🐧 Commander les Cd-Rom/DVD-Rom par la poste.





Que peut on faire avec une image ISO Ubuntu?



Ubuntu

UB

Tester

Installer





La console



- 🐧 Une console est une **interface** permettant d'accéder au système, qui au travers sont tapées les commandes.
- 🐧 Une **console**, c'est un écran noir, qui est en attente d'une instruction, d'une **commande shell**.
- 🐧 Une console est une interface permettant de dialoguer directement avec le système.
- 🐧 il est possible de se connecter à plusieurs consoles en même temps grâce à la combinaison des touches **alt et f1-f6**.

Remarque:

- 🐧 Un **terminal** est un programme qui ouvre une console dans une interface graphique. Il permet de lancer des commandes.
- 🐧 **Un terminal est une émulation d'une console.**



La console

- **ctrl + Alt + F1** : terminal 1 (tty1) ;
- **ctrl + Alt + F2** : terminal 2 (tty2) ;
- **ctrl + Alt + F3** : terminal 3 (tty3) ;
- **ctrl + Alt + F4** : terminal 4 (tty4) ;
- **ctrl + Alt + F5** : terminal 5 (tty5) ;
- **ctrl + Alt + F6** : terminal 6 (tty6) ;
- **ctrl + Alt + F7** : retour au mode graphique





La console

bt@bt-virtual-machine:~

bt@bt-virtual-machine:~\$

🐧 L'intérêt du mode console (ou du terminal)

- Rapidité
- Simplicité
- être accessible à distance par Internet

🐧 L'invite de commande

- **Bt:** le premier élément est votre pseudonyme
- **@:** ce symbole n'indique rien de particulier. C'est le symbole « at » qui signifie « chez »
- **Bt-virtual-machine:** le nom de l'ordinateur sur lequel vous êtes en train de travailler
- **~:** le dossier dans lequel vous vous trouvez actuellement
- **\$:** signifie que vous êtes en train d'utiliser un compte utilisateur « normal »
- **#:** signifie que vous êtes en mode superutilisateur, c'est-à-dire que vous êtes connectés sous le pseudonyme « root »

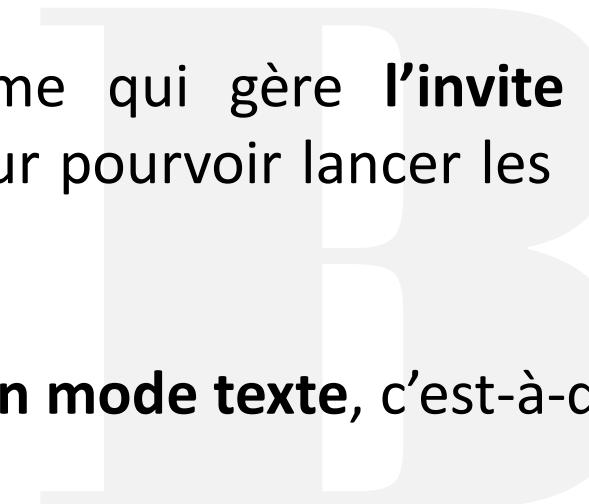




Qu'est-ce que le shell ?



- 🐧 Le shell est un **interpréteur de commandes** permettant d'exécuter des instructions saisies au clavier ou au sein d'un script et vous retourne les résultats.
- 🐧 Le shell est le programme qui gère **l'invite de commandes**. Il fournit toutes les fonctionnalités de base pour pourvoir lancer les Commandes. Il fonctionne au sein d'une console ou terminal.
- 🐧 Le shell est une **interface en mode texte**, c'est-à-dire que vous devez saisir des instructions et attendre un résultat.





Les principaux types de shells



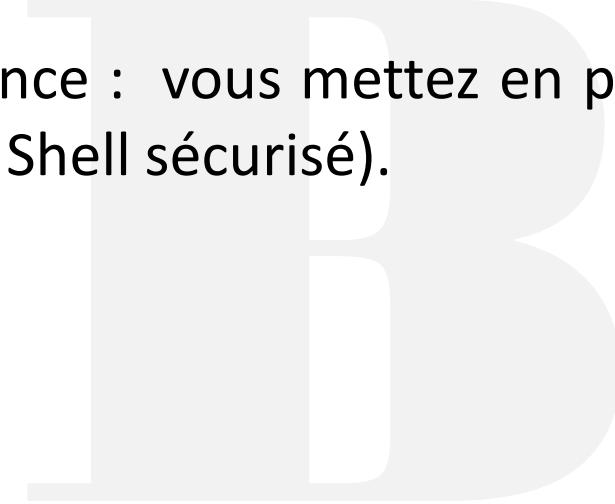
- 🐧 **Sh (Bourne Shell)** : l'ancêtre des shells, développé par Steve Bourne
- 🐧 **Bash (Bourne Again Shell)** : Disponible par défaut sous Linux et Mac OS
- 🐧 **Ksh(Korn Shell)** : assez présent dans les Unix propriétaires
- 🐧 **Csh(C shell)** : une version de shell dont la syntaxe est très proche du langage C
- 🐧 **Tcsh (Tenex C Shell)** : Une version améliorée du Csh
- 🐧 **Zsh (Z shell)** : Une version Shell qui reprend les bonnes idées des shells Csh, Tcsh et Bash





Les différentes manières de lancer un shell

- 🐧 En mode **terminal ou console** : vous êtes physiquement présent devant votre machine.
- 🐧 En ligne commande à distance : vous mettez en pratique le protocole de communication sécurisé **SSH** (Secure Shell, Shell sécurisé).





La console: Utiliser l'aide



- 🐧 Man => accès à une page de manuel pour chaque commande du système
 - 🐧 **man 1 pwd** => affiche la page de manuel catégorie 1 de la commande pwd
 - 🐧 **man man** => affiche la page de manuel de la commande man .

🐧 CATEGORIE de pages manuel

1. Programmes exécutable ou commande de l'interpréteur de commandes (shell)
2. Appels système (fonctions fournies) par le noyau;
3. Appels de bibliothèque (fonctions fournies par les bibliothèques des programmes)
4. Fichiers spéciaux (situées généralement dans /dev)
5. Formats des fichiers et conventions (par exemple /etc/passwd)
6. Jeux
7. Divers (y compris les macropaquets et les conventions) par exemple man(7),
8. Commandes de gestion du système (généralement réservées au super utilisateur)
9. Sous-programmes du noyau (hors standard)





La console: Utiliser l'aide



🐧 Aide interne à une commande

🐧 Aucun standard

🐧 Très sommaire

🐧 Options possibles:

🐧 \$ -help

🐧 \$ --help

🐧 \$ -h

🐧 Info

🐧 Ensemble de pages hiérarchisées

🐧 Réparties en plusieurs niveaux ou « nœuds »
(rubriques du plus général au plus particulier)

🐧 Liens

🐧 Marqués d'un astérisque(*)

🐧 Exemples:

\$ Info

\$ Info cp

q	Quitter
espace	Défilement vers le bas
return	Défilement vers le haut
b	Début du nœud (beginning)
e	Fin du nœud (end)
Tab	Aller au lien suivant
Entrée	Suivre le lien
n	Nœud suivant (next)
p	Nœud précédent
u	Nœud de niveau supérieur (up)
l (L min)	Retour à la page précédemment affichée (last)



La console: Commandes et paramètres

🐧 Qu'est ce qu'une commande?

- Un programme ou une partie d'un programme
 - soit un fichier spécifique
 - cf les .exe sous windows
 - identifié grâce à la variable d'environnement \$PATH
 - identifie les dossiers dans lesquels sont placés les programmes disponibles
 - soit un élément du shell
 - c'est ce qu'on nomme les commandes internes





La console: Commandes et paramètres

❖ Afficher la date>>> tapez "Date"



bt@bt-virtual-machine: ~

```
bt@bt-virtual-machine:~$ date
dimanche 23 octobre 2016, 16:58:17 (UTC+0200)
bt@bt-virtual-machine:~$
```

❖ lister les fichiers et dossiers du répertoire actuel >>> tapez "ls"



bt@bt-virtual-machine: ~

```
bt@bt-virtual-machine:~$ date
dimanche 23 octobre 2016, 16:58:17 (UTC+0200)
bt@bt-virtual-machine:~$ ls
Bureau      examples.desktop  Modèles   Public          Vidéos
Documents    Images           Musique   Téléchargements
bt@bt-virtual-machine:~$
```



La console: Commandes et paramètres

🐧 Spécificités des commandes internes?

- n'est pas un programme à part
 - la documentation se trouvera donc dans celle de bash

```
$ man bash
```

- essentielles au système
 - opérations de base
- peuvent parfois être remplacée par des commandes externes
 - afin de les améliorer ou les adapter





La console: Commandes et paramètres

🐧 Changer de répertoire

```
$ cd [répertoire]
```

- "Change directory"
- Argument [répertoire]
 - chemin absolu
 - exemple : **/home/bouryt/monrep**
 - chemin relatif
 - exemple : **bouryt/monrep**
 - un caractère spécial (raccourcis)





La console: Commandes et paramètres

🐧 Changer de répertoire: raccourcis

~	répertoire personnel
.	répertoire courant
..	répertoire parent du répertoire courant
-	dernier répertoire dans lequel nous nous situons





La console: Commandes et paramètres

❖ Les paramètres

Les paramètres sont des options que l'on écrit à la suite de la commande. La commande et les paramètres sont séparés par un espace, comme ceci :

Commande + paramètre

❖ Les paramètres courts (une lettre)

```
commande -d
```

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

```
commande -d -a -U -h
```

Ou, plus court :

```
commande -daUh
```





La console: Commandes et paramètres

❖ Les paramètres longs (plusieurs lettres)

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
```

On peut aussi combiner les paramètres longs et les paramètres courts dans une commande :

```
commande -daUh --autreparametre
```



La console: Commandes et paramètres

❖ Les paramètres

❖ Exemple: tapez “ls-a” qui affiche tout le contenu du dossier, même les fichiers cachés.

```
bt@bt-virtual-machine:~$ ls -a
```

.	.dmrc	.ICEauthority	.pulse
..	Documents	Images	.pulse-cookie
.bash_history	examples.desktop	.local	Téléchargements
.bash_logout	.fontconfig	.mission-control	.thumbnails
.bashrc	.gconf	Modèles	Vidéos
Bureau	.gksu.lock	.mozilla	.Xauthority
.cache	.gnome2	Musique	.xsession-errors
.config	.gtk-bookmarks	.profile	.xsession-errors.old
.dbus	.gvfs	Public	

```
bt@bt-virtual-machine:~$ █
```



La console: Commandes et paramètres

❖ Exemple: tapez “ls-all” all=tout en anglais.

```
bt@bt-virtual-machine: ~
drwxr-xr-x  2 bt    bt    4096 oct.   1 14:39 .fontconfig
drwx-----  4 bt    bt    4096 oct.  23 15:34 .gconf
-rw-r-----  1 bt    bt      0 oct.   1 14:45 .gksu.lock
drwx-----  4 bt    bt    4096 sept. 30 01:26 .gnome2
-rw-rw-r--  1 bt    bt     143 oct.  23 15:36 .gtk-bookmarks
dr-x-----  2 bt    bt      0 oct.  23 15:34 .gvfs
-rw-----  1 bt    bt    3660 oct.  23 15:34 .ICEauthority
drwxr-xr-x  2 bt    bt    4096 sept. 30 01:21 Images
drwxr-xr-x  3 bt    bt    4096 sept. 30 01:21 .local
drwx-----  3 bt    bt    4096 sept. 30 01:26 .mission-control
drwxr-xr-x  2 bt    bt    4096 sept. 30 01:21 Modèles
drwx-----  4 bt    bt    4096 oct.   1 14:52 .mozilla
drwxr-xr-x  2 bt    bt    4096 sept. 30 01:21 Musique
-rw-r--r--  1 bt    bt     675 sept. 30 00:36 .profile
drwxr-xr-x  2 bt    bt    4096 sept. 30 01:21 Public
drwx-----  2 bt    bt    4096 oct.  23 15:34 .pulse
-rw-----  1 bt    bt     256 sept. 30 01:21 .pulse-cookie
drwxr-xr-x  2 bt    bt    4096 sept. 30 01:21 Téléchargements
drwx-----  4 bt    bt    4096 oct.   1 13:29 .thumbnails
drwxr-xr-x  2 bt    bt    4096 sept. 30 01:21 Vidéos
-rw-----  1 bt    bt      63 oct.  23 15:33 .Xauthority
-rw-----  1 bt    bt    2125 oct.  23 15:37 .xsession-errors
-rw-----  1 bt    bt    6839 oct.  23 13:42 .xsession-errors.old
bt@bt-virtual-machine:~$
```



La console: Commandes et paramètres

Afficher le répertoire courant

```
$ pwd [-LP]
```

- "Print working directory"
- -L
 - suivre les liens symboliques
- -P
 - ne pas suivre les liens symboliques





La console: Commandes et paramètres

🐧 Afficher un message

```
$ echo [-neE] [arguments ...]
```

- -n
 - Supprime le retour chariot (retour à la ligne) final
- -e
 - affiche les séquences d'échappement
- -E
 - Désactive l'interprétation des séquences d'échappement.





La console: Commandes et paramètres

🐧 Séquence d'échappement

\a	alerte (avertisseur sonore)
\b	retour arrière (backspace)
\c	suppression du retour-chariot final
\f	saut de page
\n	nouvelle ligne
\r	retour-chariot
\t	tabulation horizontale
\v	tabulation verticale
\\\	backslash
\nnn	le caractère dont le code ASCII octal vaut nnn (un à trois chiffres)
\xnnn	le caractère dont le code ASCII hexadécimal vaut nnn (un à trois chiffres)



La console: Commandes et paramètres

🐧 Quelques autres commandes

- **exec [commande [arguments]]**
 - Permet de lancer un programme en remplacement du shell
- **set [arg1 [arg2 [arg3 ...]]]**
 - Permet d'affecter aux variables de paramètres de bash des valeurs déterminées.
- **time commande**
 - Permet de lancer une commande, et d'afficher le temps qui aura été nécessaire à son exécution
- **exit / logout**
 - stopper l'exécution de bash.



La console: Commandes et paramètres

Commandes	Description
pwd	Affiche le chemin absolu du répertoire courant
which	Affiche l'emplacement d'une commande
ls	Liste le contenu d'un répertoire
cd	Se déplacer dans le système de fichiers
du	Affiche la taille occupée par les dossiers
history	Rappel de l'histoire des commandes
hostname	Donne le nom de la machine
clear	Efface toute la console
Whoami	Donne le nom de l'utilisateur courant

```
bt@bt-virtual-machine:~$ pwd  
/home/bt  
bt@bt-virtual-machine:~$ █
```





La console: Commandes et paramètres

commandes	Description
cat	Affiche tout le fichier
less	Affiche le fichier page par page
more	Affiche le fichier page par page
head	Affiche le début du fichier
tail	Affiche la fin d'un fichier
touch	Créer un fichier
cp	Copier un fichier
mv	Déplacer un fichier
rm	Supprimer un fichier
ln	Créer des liens entre fichiers
file	Affiche le type de fichier
diff	Affiche la différence entre les fichiers





La console: Les variables



🐧 Qu'est ce qu'une variable?

- Permet d'associer une clé (son nom) à une valeur
 - En mémoire
- Exemple :
 - La variable "message" contenant la chaîne de caractère "Bonjour!"
- Avec bash :
 - aucun typage
 - toute valeur de variable est considérée comme une chaîne de caractères



La console: Les variables



🐧 Déclarer une variable?

- Sans typage, aucun besoin de déclaration spécifique
- une simple affectation suffit
 - signe égal
 - sans espace
- Exemple

```
$ mavariable="Bonjour"
```



- Attention ! le nom d'une variable est sensible à la casse
 - ie. "mavariable" n'est pas la même variable que "MaVariable"



La console: Les variables



🐧 Utilise une variable?

- nécessite le caractère \$ devant le nom
 - remplace cette référence par la valeur de la variable
- Exemple :

```
$ echo $mavariable  
Bonjour !
```

- on peu aussi utiliser la syntaxe suivante :

```
$ echo ${mavariable}  
Bonjour !
```



La console: Les variables



🐧 Portée d'une variable?

- Par défaut : locales
 - n'est définie et disponible que pour l'instance de bash
 - n'est pas accessible par les autres programmes
- Variables d'environnement
 - accessible par tout les programmes au sein d'un même environnement utilisateur
 - i.e. tout les programmes lancés
 - soit par vous-même
 - soit par un des programmes que vous avez lancé



La console: Les variables



🐧 Variable d'environnement

- utiliser la commande `export`

```
$ MESSAGE="Bonjour !"  
$ export MESSAGE
```

- Par convention, la référence est en majuscules
- Attention ! l'argument de la commande `export` est bien la référence (le nom) de la variable
 - pas sa valeur
 - ne pas utiliser le caractère \$



La console: Les variables



🐧 Variable de paramètres

- variables locales
- utiles en scripting (vu plus loin)
- \$1, \$2, \$3 ... enregistrent les arguments passés à une commande
- \$0 enregistre la commande ayant permis de lancer la programme
- \$* enregistre tout les arguments
- \$# enregistre le nombre d'arguments
- etc ... ↗



La console: Les variables



🐧 Variable de paramètres

```
$ /bin/echo -e "Bonjour !"
```

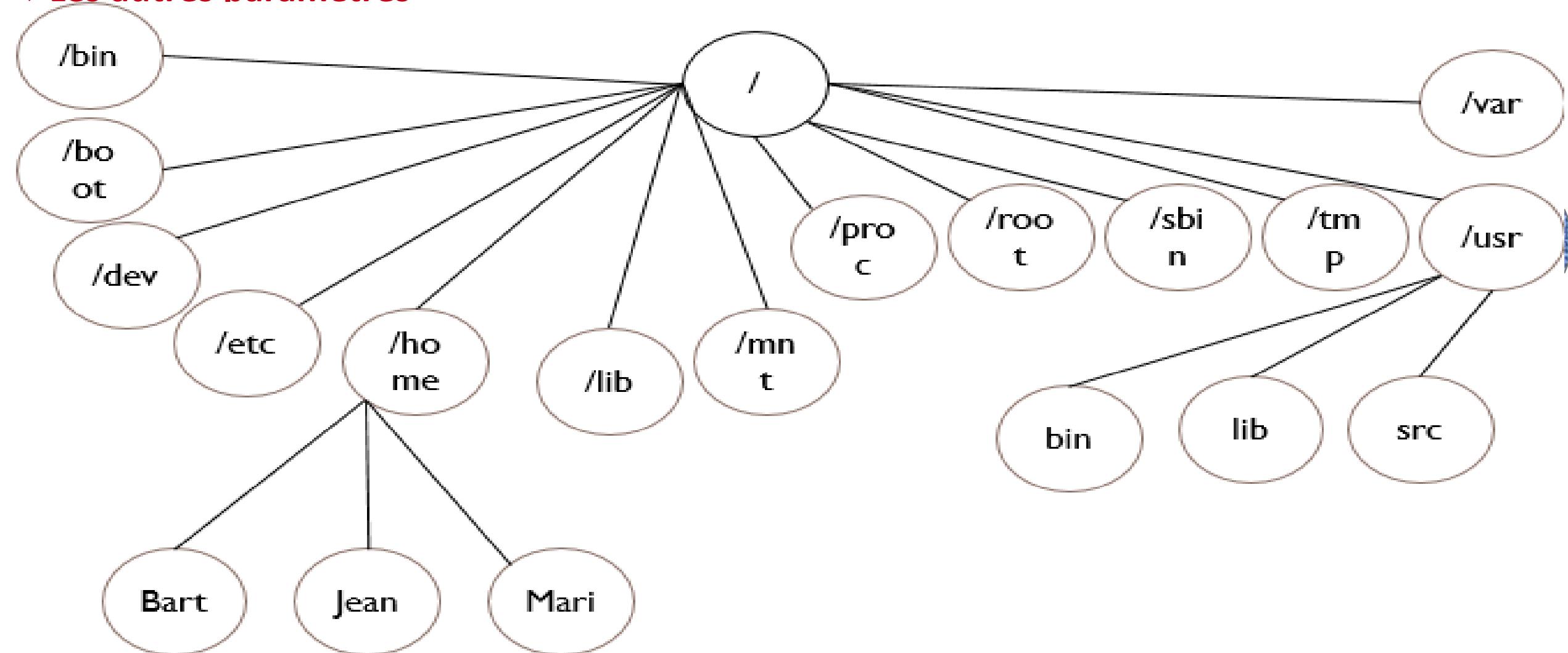
- \$0 est égale à "/bin/echo"
- \$1 à "-e"
- \$2 à "Bonjour !"
- \${*} à "-e 'Bonjour !'"
- \${#} à 2



Un exemple d'arborescence Linux



❖ Les autres paramètres





Un exemple d'arborescence Linux



Répertoire	Contenu
/	La racine
/bin	Le répertoire contenant les principales commandes (cp, mkdir)
/boot	Le répertoire contenant les fichiers de démarrage
/dev	Tous les périphériques
/etc	Les fichiers de configuration et des fichiers nécessaires au démarrage
/home	Les répertoires personnels des utilisateurs
/lib	Le répertoire contenant les bibliothèques
/mnt	Les dossiers pour les points de montage temporaires
/proc	Accès direct aux paramètres du noyau
/root	Répertoire personnel du super-utilisateur (root)
/sbin	Les binaires vitaux pour l'administration
/tmp	Répertoire contenant les fichiers temporaires
/usr	Répertoire contient les applications et les bibliothèques des utilisateurs
/var	Répertoire contenant des éléments résultants de l'exploitation



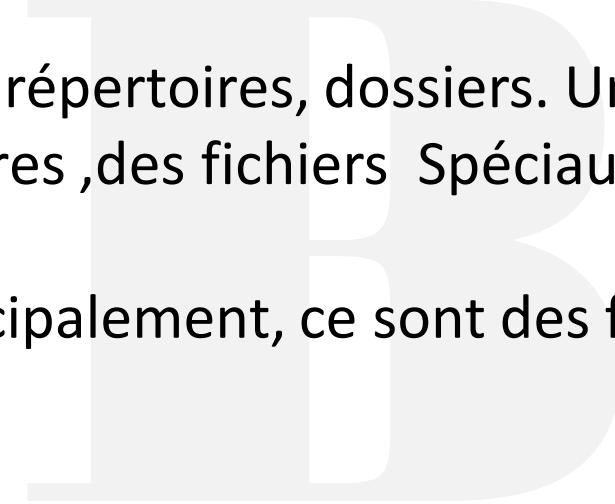
Les différents types de fichiers



- ❖ **Les fichiers ordinaires** : Ce sont des fichiers qui contiennent des données (base de données, texte, image, programme binaire compilé, audio, etc ...)

- ❖ **Les catalogues** : Ce sont les répertoires, dossiers. Un catalogue peut contenir des fichiers normaux, d'autres répertoires ,des fichiers Spéciaux.

- ❖ **Les fichiers spéciaux** : principalement, ce sont des fichiers d'interface pour les divers périphériques.





Les différents types de fichiers



🐧 Nommage des fichiers

- Max 255 caractères
- sensibles à la casse
- Se limiter aux caractères spéciaux suivants :
 - point (.)
 - délimiteurs (- et _)
 - espace (avec échappement ou quotes)
 - tilde en fin de fichier (indique un fichier de sauvegarde automatique)



Les différents types de fichiers



🐧 Nommage des fichiers

- Max 255 caractères
- sensibles à la casse
- Se limiter aux caractères spéciaux suivants :
 - point (.)
 - délimiteurs (- et _)
 - espace (avec échappement ou quotes)
 - tilde en fin de fichier (indique un fichier de sauvegarde automatique)



Les différents types de fichiers



🐧 Chemins d'accès

- le délimiteur est le slash (/)
- Racine unique : le dossier /
- Deux types de chemins :
 - absolu
 - relatif



Les différents types de fichiers



🐧 Globbing

- Remplacer des chaînes de caractère par des jockers ("wildcards")

?	n'importe quel caractère
*	n'importe quelle chaîne de caractère
[...]	l'un des caractères entre les crochets
[a-f]	n'importe quel caractère entre "a" et "f"



Les différents types de fichiers



🐧 Commandes essentielles

- **cp [OPTION]... SOURCE CIBLE**
 - copier des fichiers et des répertoires
- **mv [option...] source cible**
 - déplacer ou renommer des fichiers
- **rm [options] fichier...**
 - supprimer des fichiers
- **touch [-acm][-tdate] [-] fichier...**
 - créer un fichier vide ou modifier son horodatage



Les différents types de fichiers



🐧 Commandes essentielles (2)

- **ls [options] [fichier...]**
 - Afficher le contenu d'un répertoire
- **mkdir [options] repertoire...**
 - Créer des répertoires
- **rmdir [options] repertoire(s)**
 - Supprimer un répertoire vide
- **ln [options] source [lien]**
 - créer un lien



Recherche des fichiers



Commandes	Description
locate	Recherche rapidement un fichier
slocate	Recherche rapidement un fichier
find	Recherche un fichier





Commandes de base de répertoires et de fichiers

Commandes	Description
mkdir	Créer un répertoire
cp	Copier un répertoire
mv	Déplacer un dossier
rmdir	Supprimer un dossier





Flux, pipes et redirections



Fluxs

1. entrée standard (stdin)
2. sortie standard (stdout)
3. erreur standard (stderr)



Flux, pipes et redirections



🐧 Pourquoi les Fluxs

- Rappel : philosophie Unix
- Collaboration entre les programmes
- Texte = simple et standard, donc universel





Flux, pipes et redirections



🐧 Redirection

>	stdout vers nouveau fichier
>>	stdout à la suite d'un fichier
2>	stderr vers nouveau fichier
2>>	stderr à la suite d'un fichier
&>	stdout + stderr
<	stdin depuis un fichier
<<	stdin à partir d'une chaîne de caractères
<>	stdin et stdout vers et depuis le même fichier





Flux, pipes et redirections



Pipe

- le caractère |
- stdout -> stdin



Flux, pipes et redirections



Utilisation et substitution d'arguments

- xargs
 - permet d'employer chaque sortie d'un programme comme argument d'un autre
- antiquotes
 - permet de remplacer de manière itérative un argument par les sorties d'une commande



Textprocessing: combinaison



🐧 Concaténer des fichiers

```
$ cat [options] [fichier(s) ...]
```

```
* echo "test1" > test1
* echo "test2" > test1
* echo "test2" > test2
* echo "test1" > test1
* cat test1 test2
```



Textprocessing: combinaison



🐧 Fusionner les lignes de deux fichiers

```
$ join [options] fichier_1 fichier_2
```

```
$ join -1 1 -2 2 employes adresses
```





Textprocessing: combinaison



🐧 Regrouper les lignes de différents fichiers

```
$ paste [options] [fichier...]
```

```
$ paste employes adresses
```





Textprocessing: transformer



🐧 Commandes utils

- Convertir les tabulations en espaces

```
$ expand [-t num] [fichier...]
```

- Convertir les espaces en tabulations

```
$ unexpand [-a] [fichier...]
```

- Trier les lignes d'un fichier texte

```
$ sort [options] [+POS1 [-POS2]] [fichier...]
```

- Découper un fichier

```
$ split [options] [fichier_entrée[préfixe]sortie] ]
```





Textprocessing: transformer



🐧 Commandes utils (2)

- Traduire ou éliminer des caractères

```
$ tr [options] ENS1 ENS2
```

- Éliminer les lignes dupliquées dans un fichier trié

```
$ uniq [options] [fichier_entrée] [fichier_sortie]
```





Textprocessing: formatage



🐧 Commandes utils

- Mettre en forme des textes

```
$ fmt [options] [fichier...]
```

- Numéroter les lignes d'un fichier

```
$ nl [options] [fichier...]
```



Textprocessing: visualiser



🐧 Commandes utils

- Afficher les premières lignes d'un fichier

```
$ head [options] [fichier...]
```

- Afficher les dernières lignes d'un fichier

```
$ tail [options] [fichier...]
```

- Afficher un fichier page par page

```
$ less [fichier...]
```



Textprocessing: résumer



🐧 Commandes utils

- Supprimer une partie de chaque ligne d'un fichier.

```
$ cut [options] [fichier...]
```



- Afficher le nombre d'octets, de mots et de lignes d'un fichier

```
$ wc [options] [fichier...]
```



Expression régulière



🐧 Qu'est ce qu'une expression régulière?

- un outil pour représenter des motifs au sein d'un texte
 - ie : un ensemble de chaînes de caractères
 - cf globbing pour les noms de fichiers
- Utile pour :
 - la recherche
 - la suppression
 - le remplacement
 - etc ...





Expression régulière



🐧 Comment écrire une expression régulière?

- des caractères alphanumériques
 - exemple : Linux ou ACPI
- des caractères spéciaux
 - permettant de désigner un ensemble de caractère





Expression régulière

🐧 Caractères spéciaux

regexp simple	[...]	ensemble	
	[x - x]	intervalle	
	.	n'importe quel caractère	
	répétition	*	zéro ou plus
		+	un ou plus
		?	zéro ou un
		{n,m}	de n à m fois
regexp étendue		ou logique	
	(...)	sous-section	
	ligne	^	début
		\$	fin





Expression régulière



🐧 Exemples

- lignes commençant par un nombre

```
^ [0-9] [0-9] *
```

- FreeBSD, OpenBSD ou NetBSD

```
(Free|Open|Net) BSD
```

- Chaine contenant "Bonjour" puis "au revoir."

```
Bonjour.*au revoir\.
```

- Un mot commençant par un ou plusieurs 'a' ou 'b'

```
[ab]+[a-zA-Z]*
```





Grep



🐧 Pour quoi faire ?

- rechercher
 - dans un fichier ou un flux
 - les lignes correspondant à une expression régulière





Grep



🐧 Syntaxe

```
$ grep [options] regexp [fichier...]
```

- Options

- -c : afficher le décompte des lignes correspondantes
- -i : ignorer la casse
- -E : utiliser la syntaxe étendue pour les expressions régulières
 - identique à egrep





sed



🐧 Pour quoi faire?

- modifier le contenu
 - d'un fichier
 - d'un flux
- ligne par ligne
- en fonction d'expressions régulières





sed



🐧 Syntaxe

```
$ sed [options] commande [fichier...]
```

```
$ sed [options] -f script [fichier...]
```



- Options :

- **-e** : permet d'enchaîner plusieurs commandes à la suite
- **-r** : utilisation des expressions régulières étendues



sed



🐧 Substitution

- s/modèle/remplacement/drapeau(x)

- Exemple :

```
$ sed s/^#$/#commentaire/ monfichier
```





sed



🐧 Drapeaux

g	global	toutes les occurrences
N (nombre)	—	nième occurrence
w	write	écrire les modifications dans un fichier
p 	print	afficher la ligne modifiée (avec -n)
e	evaluate	exécution de commande

- Exemples

```
$ sed 's/A/B/g' monfichier
```

```
$ sed "s/.*/5/echo '$A'/e" monfichier
```

```
$ sed 's/A/a/gw rslt' monfichier
```





sed



▲ Autres commandes

commande	usage	exemple
q	quitter	3q
d	effacer	3d
p	affichage (avec -n)	3p
i\texte	insérer le texte	3i\machin
a\texte	ajouter le texte	3a\machin
c\texte	remplacer par le texte	3c\machin
=	afficher	3=





Notion de groupe sous Ubuntu



- ❖ Un groupe est un **ensemble d'utilisateurs, réels ou applicatifs**, regroupés dans un but commun. Chaque groupe est identifié par un numéro d'un compte d'un group GID.
- ❖ Le fichier `/etc/group` contient la définition des groupes d'utilisateurs.
- ❖ Les commandes de gestion de groupe:

Commandes	Description
<code>addgroup</code>	Ajouter un groupe
<code>groupmod</code>	Modifier la définition d'un groupe
<code>delgroup</code>	Supprimer un groupe
<code>groups</code>	Afficher le groupe d'appartenance de l'utilisateur



Les types d'utilisateurs sous Ubuntu

- ❖ Les utilisateurs humains (root et les autres utilisateurs)
- **-Root** : c'est le super administrateur ou superutilisateur du système. Il a tous les droits sur le système. Il est désactivé par défaut sous Ubuntu.
- **-Utilisateurs** : ce sont les comptes créés pour chaque utilisateur du système. Les utilisateurs systèmes (applications, propriétaires de processus)

```
bt@bt-virtual-machine:~$ pwd  
/home/bt  
bt@bt-virtual-machine:~$ sudo su  
[sudo] password for bt:  
root@bt-virtual-machine:/home/bt# exit  
exit  
bt@bt-virtual-machine:~$
```

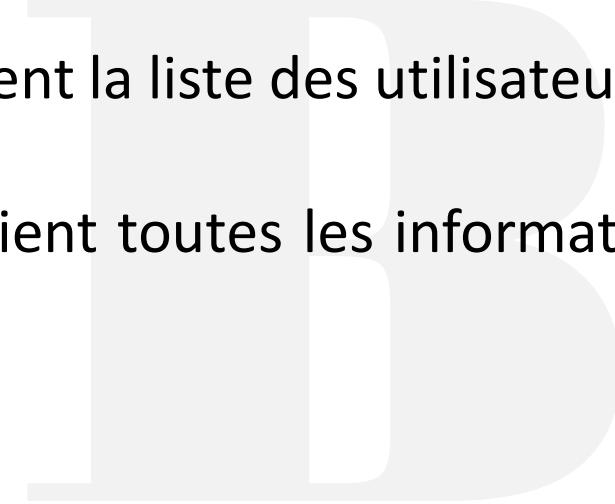




Notion d'utilisateur sous Ubuntu



- ❖ Un utilisateur est l'association d'un **nom de connexion, login**, à un **UID**(le numéro de compte d'un utilisateur) et **au moins d'un GID** (le numéro de compte d'un groupe).
- ❖ Le fichier **/etc/passwd** contient la liste des utilisateurs du système local.
- ❖ Le fichier **/etc/shadow** contient toutes les informations sur le mot de Passe et sa validité dans le temps.





Les commandes de gestion des utilisateurs



Commandes	Description
adduser	Ajouter
usermod	Modifier un utilisateur
deluser	Supprimer un utilisateur
id	Afficher l'identifiant de l'utilisateur
passwd	Changer le mot de passe de l'utilisateur
chfn	Modifier les informations de l'utilisateur
chsh	Changer le shell de l'utilisateur



❖ Autres commandes d'administration

Commandes	Description
su	Passer sous une autre identité
sudo	Avoir des droits élargis
last	Afficher la liste des connexions utilisateurs
who	Montrer qui est connecté



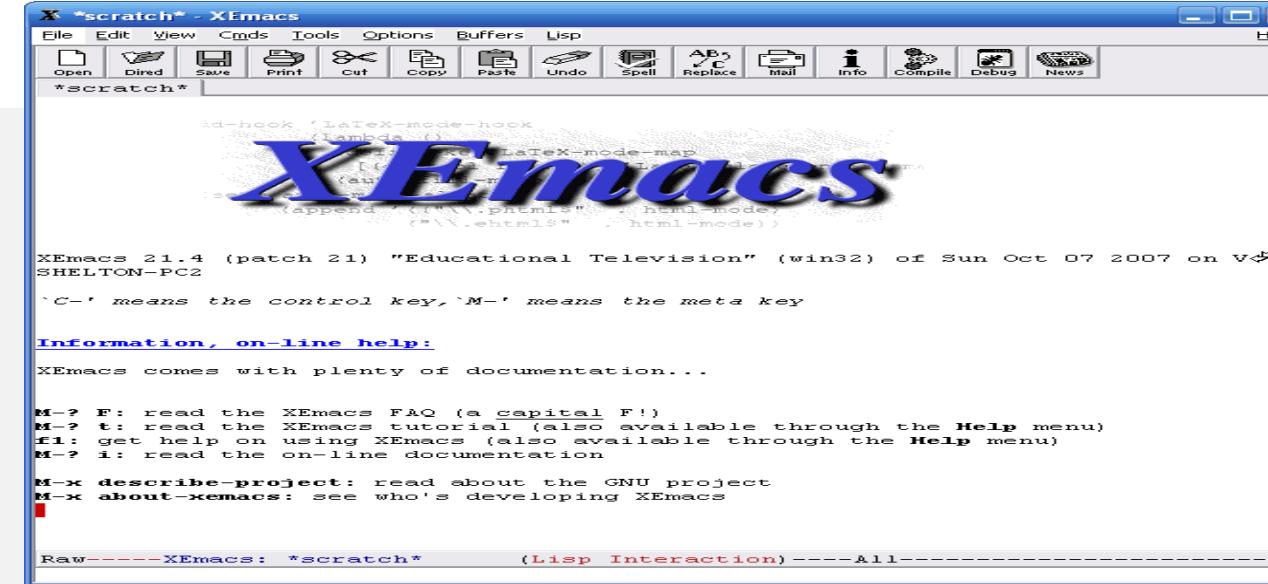
Quelques éditeurs sous Linux



❖ En environnement graphique



GEDIT



❖ En environnement console

- Emacs
- -Vi
- -Vim
- -Nano
- -Emacs



Quelques éditeurs sous Linux



```
x - bt@bt-virtual-machine: ~
GNU nano 2.2.6          Nouvel espace

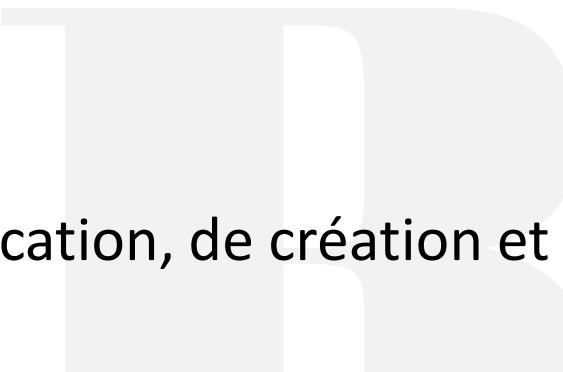
^G Aide      ^O Écrire     ^R Lire fich. ^Y Page préc. ^K Couper      ^C Pos. cur.
^X Quitter    ^J Justifier   ^W Chercher  ^V Page suiv. ^U Coller      ^T Orthograp.
```



Les attributs d'un fichier



- ❖ Les types du fichier ((-)ordinaire, (d) Répertoire, (c, b) Périphérique (l)Lien symbolique.
- ❖ Les droits
- ❖ Le propriétaire
- ❖ Le groupe
- ❖ La taille
- ❖ Les dates de dernière modification, de création et de dernier accès



```
$ ls -l fichier
-rw-r--r--  root  bin  200  Fev 17 17:17  le fichier
```

Diagram illustrating the attributes of the file "fichier" from the command output:

- type**: Points to the first character of the permissions column (-).
- droits**: Points to the second character of the permissions column (rw-).
- propriétaire**: Points to the word "root" under the "propriétaire" column.
- groupe**: Points to the word "bin" under the "groupe" column.
- taille**: Points to the number "200" under the "taille" column.
- Date et heure de dernière modification**: Points to the date and time "17 17:17" under the "Date et heure de dernière modification" column.
- chemin**: Points to the file name "fichier" under the "chemin" column.



La gestion des droits d'accès

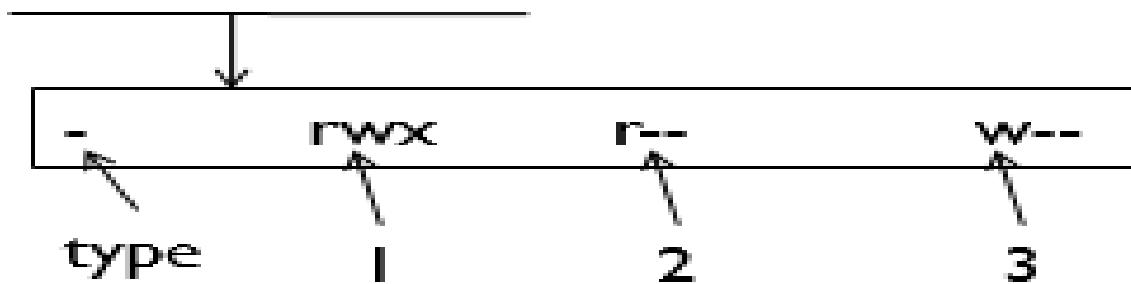


- ❖ Chaque fichier et chaque dossier possède les droits d'accès. C'est une information qui indique qui a droit de voir, de modifier et d'exécuter un fichier.



\$ ls -l fichier

-rwx r-- w-- bin 200 Fév 17 17:17 le fichier



- 1: indique les droits que possède le propriétaire du fichier sur ce dernier.
- 2 : indique les droits que possèdent les autres membres du groupe de fichier.
- 3 : indique les droits que possède les autres utilisateurs de la machine sur le fichier.



Les différents types de permissions



- ❖ Chaque fichier et chaque dossier possède les droits d'accès. C'est une information qui indique qui a droit de voir, de modifier et d'exécuter un fichier.

Droit	Lettre	Explication
Lecture	r (read)	Le droit de lecture : avec ce droit, on peut lire un fichier
Écriture	w (write)	Le droit d'écriture : avec ce droit, on peut modifier un fichier
Exécution	x (exécute)	Le droit d'exécution : avec ce droit, on peut exécuter un fichier



Méthodes pour modifier les droits d'un fichier

❖ La méthode par nombre

chmod [nombre_octal] fichier

Droit	Lettre	Valeur
Lecture	R (read)	4
Écriture	w(write)	2
Exécution	x(exécute)	1

La liste des droits possibles

Droits	chiffre	calcul
- - -	0	0+0+0
r - -	4	4+0+0
- w -	2	0+2+0
-- x	1	0+0+1
r w -	6	4+2+0
- w x	3	0+2+1
r - x	5	4+0+1
r w x	7	4+2+1





Méthodes pour modifier les droits d'un fichier

❖ La méthode par nombre(Suite)

```
# chmod 604 test.txt
```

604 indique les droits du propriétaire, du groupe et des autres

6 (rw-) : droit de lecture et d'écriture pour le propriétaire

0 (---) : aucun droit pour les autres

4 (r--) : droit de lecture pour les autres

```
# chmod 777 test.txt
```

777 indique les droits du propriétaire, du groupe et des autres

7 (rwx) : droit de lecture, d'écriture et d'exécution pour le propriétaire

7 (rwx) : droit de lecture, d'écriture et d'exécution pour le groupe

7 (rwx) : droit de lecture, d'écriture et d'exécution pour les autres





Méthodes pour modifier les droits d'un fichier

❖ La méthode par symbole

chmod [Qui] [Opération] [Type_permission] fichier

- Qui : le type d'utilisateur (u, g, o, a).
- Opération : + (ajouter), - (supprimer), = (affecter le droit).
- Type_permission : r (lecture), w (écriture), x(exécution).

Les types de propriétaires

type	lettre	Explication
Utilisateur	u (User)	Le propriétaire du fichier ou du dossier
Groupe	g (Group)	Le groupe propriétaire du fichier ou du dossier
Autre	o(Other)	Les utilisateurs qui ne font pas partie du groupe propriétaire
Tous	a (All)	Tous les utilisateurs (Utilisateur, Groupe, Autre)





Méthodes pour modifier les droits d'un fichier

❖ La méthode par symbole(suite)

```
# chmod u+w test.txt
```

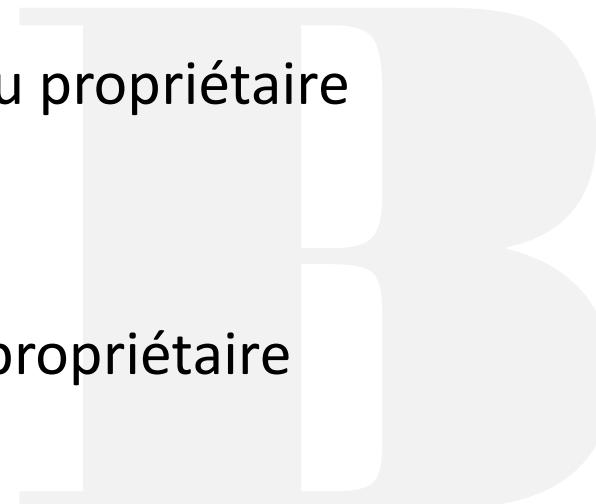
Ajouter le droit d'écriture au propriétaire

```
# chmod u=rwx test.txt
```

Affecter tous les droits au propriétaire

```
# chmod g-x test.txt
```

Enlever le droit d'exécution au groupe





Méthodes pour modifier les droits d'un fichier

❖ Comment changer l'appartenance d'un fichier?

chown [options] user [:group] fichier

-user : le nouveau propriétaire

-group : on peut changer le groupe propriétaire

chgrp group fichier





Qu'est-ce qu' un dépôt



- ❖ C'est une serveur qui contient un ensemble de paquets.
- ❖ C'est un ensemble de paquets qui dépendent soit les uns des autres, soit d'autres paquets en provenance d'autres dépôts.
- ❖ C'est le serveur sur lequel on va télécharger nos paquets.

Le fichier `/etc/apt/sources.list` contient la liste des sources.





Qu'est-ce qu'un dépôt



❖ Les dépôts officiels actifs par défaut sous Ubuntu

- Main : c'est le dépôt principal. Il contient tous les paquets officiellement maintenus par l'équipe Ubuntu. Tous ses logiciels sont libres.
- Restricted : c'est un dépôt qui contient tous les logiciels maintenus par l'équipe Ubuntu qui ne sont pas entièrement libres ou peuvent être soumis à des brevets logiciels.
- Universe : c'est un dépôt qui contient beaucoup de logiciels libres adaptés pour Ubuntu par une communauté.
- Multiverse : c'est un dépôt qui contient les logiciels non libres ou soumis à des brevets.





Qu'est-ce qu'une dépendance ?



❖ Les dépôts officiels actifs par défaut sous Ubuntu

Une dépendance, c'est le fait qu'un paquet pour fonctionner correctement a besoin d'un ou d'autres paquets.

.B

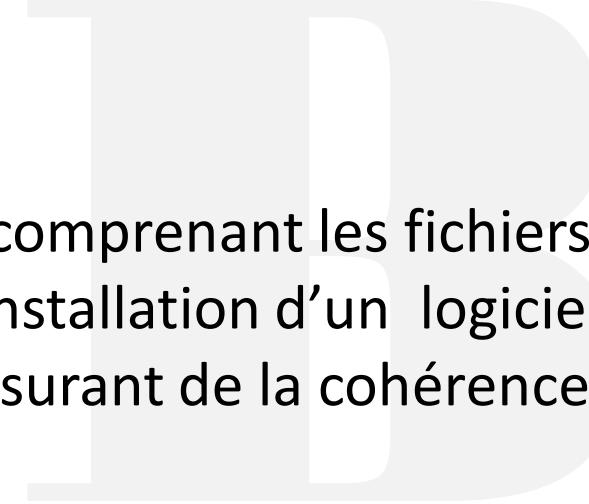




Qu'est ce qu'un paquet ?



- ❖ Un paquet est une archive qui contient les fichiers du programme à installé et des règles (instructions nécessaires).
- ❖ Un paquet est une archive comprenant les fichiers informatiques, les informations et procédures nécessaires à l'installation d'un logiciel sur un système d'exploitation au sein d'un agrégat logiciel, en s'assurant de la cohérence fonctionnelle du système ainsi modifié (source wikipedia).





Qu'est ce qu'un paquet ?



❖ Un paquet est une archive qui contient les fichiers du programme à installé et des règles (instructions nécessaires).

❖ Un paquet est une archive comprenant les fichiers informatiques, les informations et procédures nécessaires à l'installation d'un logiciel sur un système d'exploitation au sein d'un agrégat logiciel, en s'assurant de la cohérence fonctionnelle du système ainsi modifié (source wikipedia).

❖ Qu'est-ce qu'un gestionnaire de paquets?

• C'est l'outil utilisé pour installer, mettre à jour ou supprimer les paquets.

• C'est un programme qui permet d'installer les logiciels, les mettent à jour, et les désinstaller.



Qu'est ce qu'un paquet ?



❖ Les modes d'installation des paquets

- Par le système de package (apt-get, aptitude, Synaptic...)
- Par le système des sources

❖ Quelques gestionnaires de paquets

- apt-get (en mode console)
- aptitude (en mode console)
- dpkg (en mode console)
- synaptic (en mode graphique)
- Logithèque Ubuntu (en mode graphique)

❖ Quelques commandes avec aptitude

Commandes	Description
aptitude update	Mettre à jour le cache des paquet en local
aptitude dist-upgrade	Mettre à jour tous les paquets du système
aptitude remove paquet	Désinstaller un paquet
aptitude show paquet	Donner des informations sur un paquet
aptitude purge paquet	Supprimer un paquet et ses fichiers de configuration
aptitude install paquet	Installer un paquet



Qu'est ce qu'un paquet ?



❖ Quelques commandes avec apt-get

Commandes	Description
apt-get update	Mettre à jour le cache des paquets
apt-cache search	Rechercher un paquet
apt-get install	Installer un paquet
apt-get autoremove	Supprimer un paquet
apt-get upgrade	Mettre à jour tous les paquets

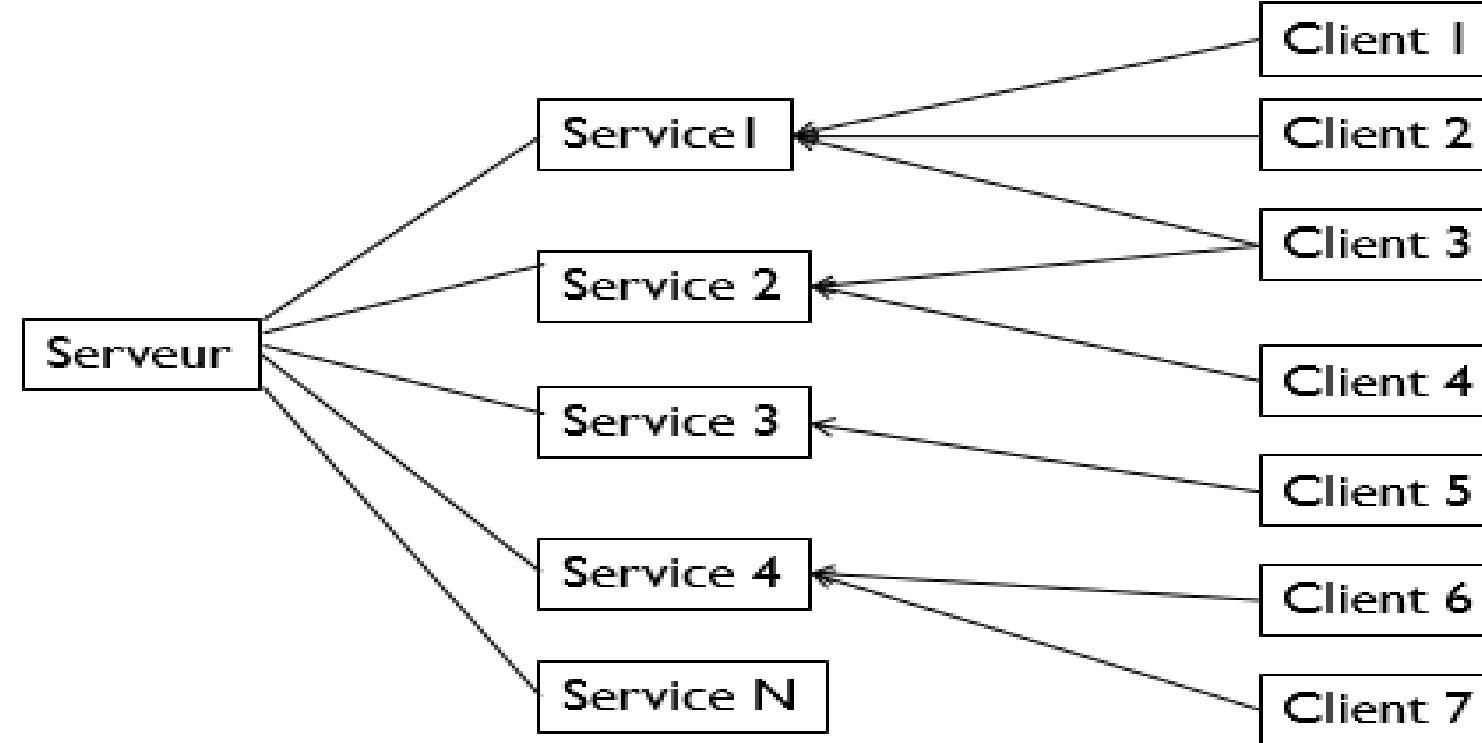




Qu'est-ce qu'un serveur linux?



Un serveur Linux est un système sur lequel on a installé des services réseaux pour servir et répondre aux demandes des clients.





Qu'est-ce qu'un serveur linux?



- ❖ Le service DNS
- ❖ Le service DHCP
- ❖ Le service VPN avec OpenVPN
- ❖ Le service Samba
- ❖ Le service d'authentification avec NIS/NFS
- ❖ Le service NTP
- ❖ Le service Web avec APACHE
- ❖ Le service de base de données avec MYSQL
- ❖ Le service POP et IMAP
- ❖ Le service FTP avec VsFTP
- ❖ Le service de partage de fichiers avec NFS
- ❖ Etc ...





Transformer sa machine en serveur



❖ Installer le paquet oppenssh-server

```
bt@bt-virtual-machine:~$ sudo apt-get install oppenssh-server  
Lecture des listes de paquets... 26%
```



```
Paramétrage de openssh-server (1:6.6p1-2ubuntu2.8) ...  
Creating SSH2 RSA key; this may take some time ...  
Creating SSH2 DSA key; this may take some time ...  
Creating SSH2 ECDSA key; this may take some time ...  
Creating SSH2 ED25519 key; this may take some time ...  
ssh start/running, process 7671  
Paramétrage de ssh-import-id (3.21-0ubuntu1) ...  
Traitement des actions différées (« triggers ») pour ufw (0.34~rc-0ubuntu2) ...  
Traitement des actions différées (« triggers ») pour ureadahead (0.100.0-16) ...  
bt@bt-virtual-machine:~$
```



Transformer sa machine en serveur



- ❖ lancer le programme de serveur SSH (sshd) : **Sudo /etc/init.d/ssh start**

```
bt@bt-virtual-machine:~$ sudo /etc/init.d/ssh start
start: Job is already running: ssh
bt@bt-virtual-machine:~$ █
```

- ❖ vous pouvez l'arrêter avec cette commande: **Sudo /etc/init.d/ssh stop**
- ❖ le fichier de configuration se trouve dans : **/etc/ssh/ssh_config**
- ❖ Il faudra recharger SSH avec la commande : **Sudo /etc/init.d/ssh reload**



Se connecter via SSH à partir d'une machine Linux



- ❖ lancer la console sur le second PC : **ssh login@ip**
Exemple: ssh bouri@192.168.1.67
- ❖ Tester en vous connectant chez vous depuis chez vous: **ssh bt@localhost**

```
[rc|your| [size]]  
bt@bt-virtual-machine:~$ ssh bt@localhost  
  
The authenticity of host 'localhost (127.0.0.1)' can't be established.  
ECDSA key fingerprint is 60:f9:e4:b3:95:70:45:49:5a:53:e6:dc:d1:4e:ee:24.  
Are you sure you want to continue connecting (yes/no)? y  
Please type 'yes' or 'no': yes  
Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.  
bt@localhost's password:  
Welcome to Ubuntu 14.04.5 LTS (GNU/Linux 3.13.0-100-generic x86_64)
```

* Documentation: <https://help.ubuntu.com/>

```
Last login: Sat Oct 1 13:01:27 2016  
bt@bt-virtual-machine:~$ █
```



Se connecter via SSH à partir d'une machine Linux



❖ L'identification automatique par clé

- Opérations sur la machine du client: **ssh –keygen -t rsa (dsa)**

```
bt@bt-virtual-machine: ~$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/bt/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/bt/.ssh/id_rsa.
Your public key has been saved in /home/bt/.ssh/id_rsa.pub.
The key fingerprint is:
97:42:7f:b3:84:27:71:af:23:a3:e6:e0:6c:ff:02:ec bt@bt-virtual-machine
The key's randomart image is:
+--[ RSA 2048]----+
|                               |
|                               |
|                               |
|                               |
| . . = . |
| . S * = . |
| o o = + |
| ... o + |
| oE.o. o . |
| .o++o. |
+-----+
bt@bt-virtual-machine: ~$
```



Se connecter via SSH à partir d'une machine Linux

❖ L'identification automatique par clé

- Envoyer la clé publique au serveur: **ssh-copy-id –i id_rsa.pub login@ip**

Vous pouvez déjà vous rendre dans le dossier ssh: **cd /.ssh** puis **ls**

- **Id_rsa**: votre clé privée, qui doit rester secrète. Elle est chiffrée si vous avez rentré une **passphrase**;
- **Id_rsa.pub**: la clé publique que vous pouvez communiquer à qui vous voulez, et que vous devez envoyer au serveur ;
- **Known_hosts**: c'est la liste des **fingerprint** que votre PC de client tient à jour. Ça lui permet de se souvenir de l'identité des serveurs et de vous avertir si, un jour, votre serveur est remplacé par un autre (qui pourrait être celui d'un pirate !). Je vous en ai déjà parlé un peu plus tôt.

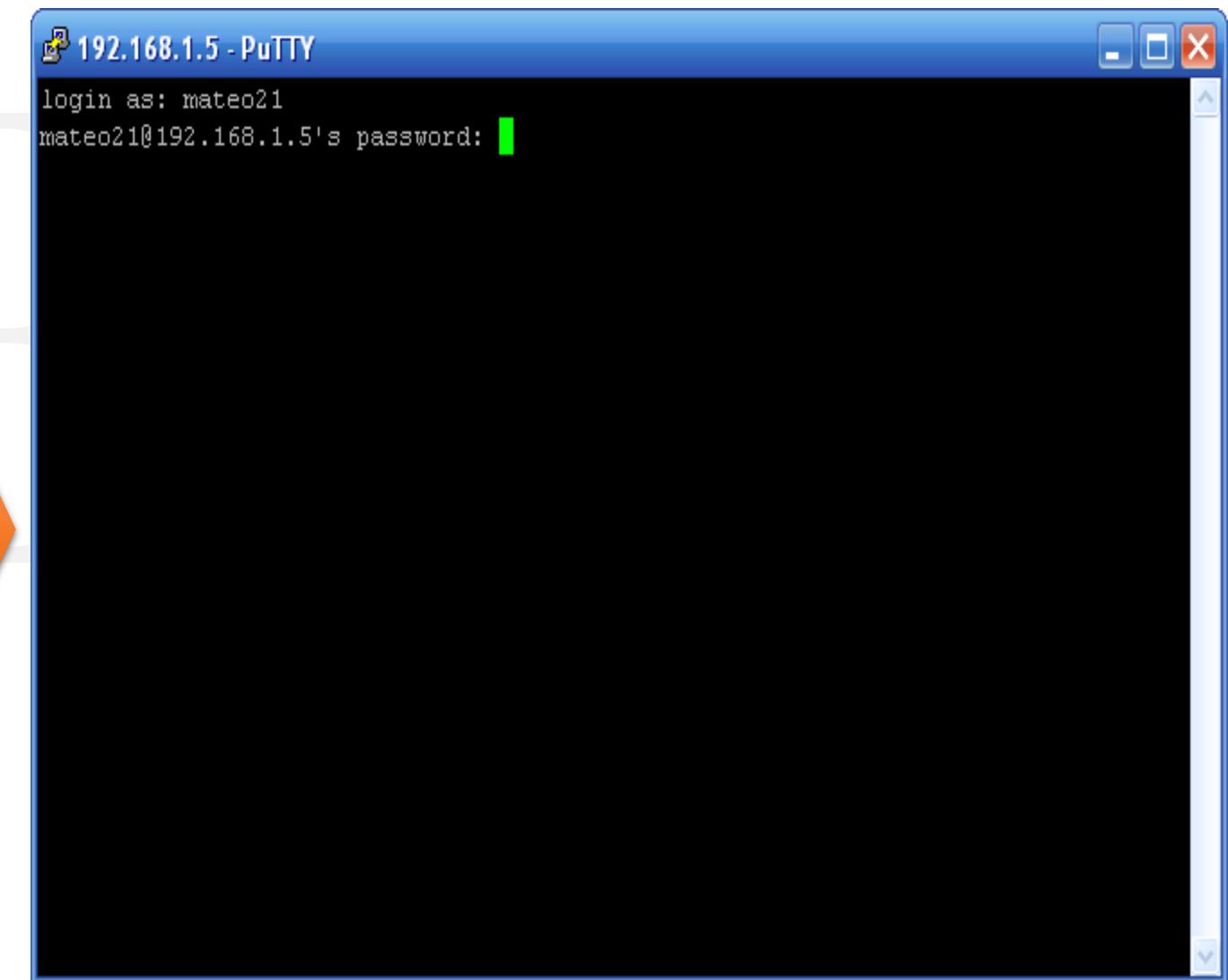
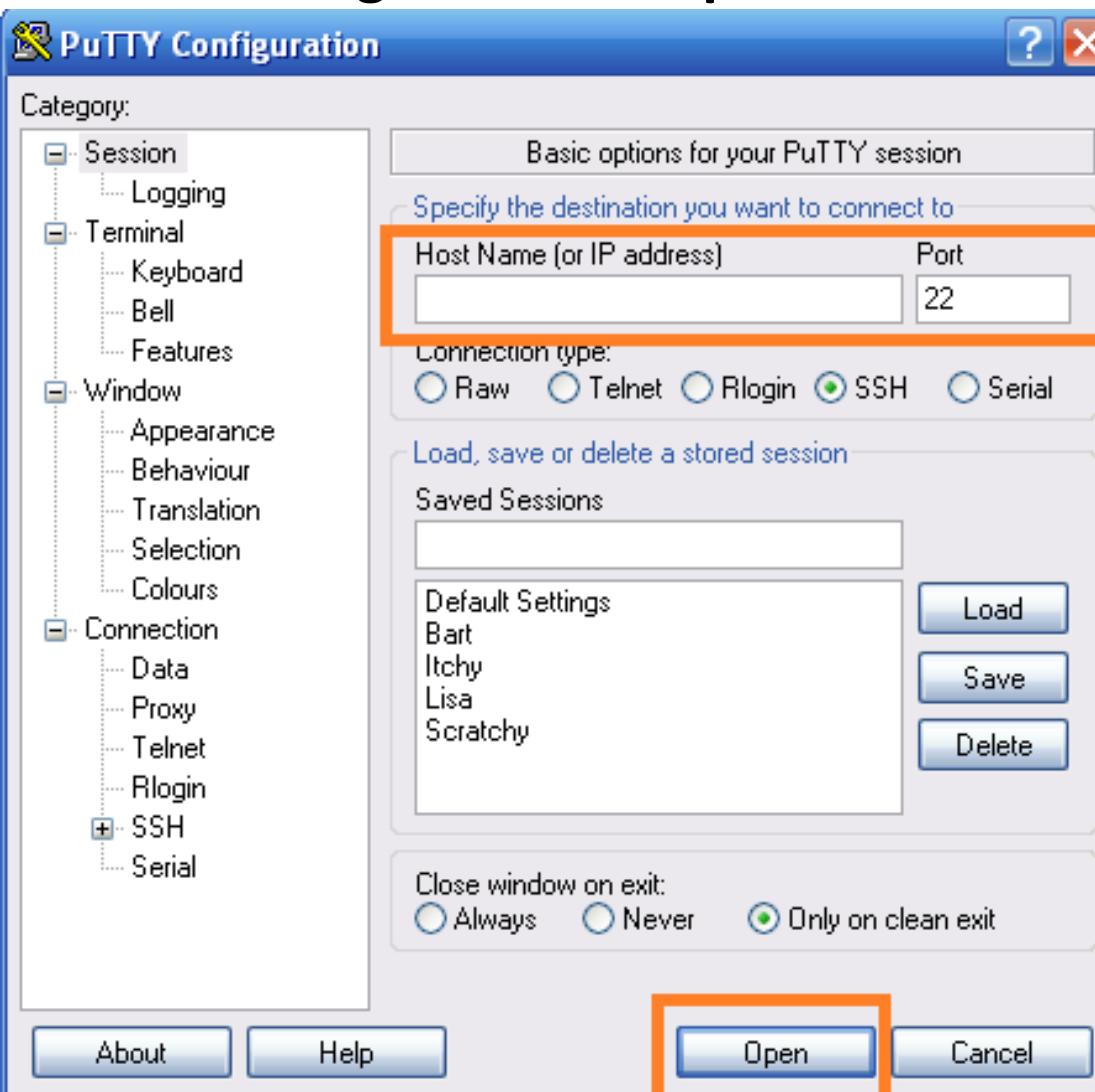
L'agent SSH est un programme qui tourne en arrière-plan en mémoire. Il retient les clés privées pendant toute la durée de votre session: **ssh-add**

Se connecter !

Mise à jour 2020

Se connecter via SSH à partir d'une machine Windows

- ❖ Télécharger PuTTY depuis son site officiel et l'installer.





❖ Authentification par clé depuis Windows (PuTTY) avec Puttygen



Ouvrez PuTTY et connectez-vous au serveur comme auparavant (en entrant votre mot de passe habituel).

Rendez-vous dans `~/ .ssh` :

```
cd / .ssh
```

```
mkdir .ssh
```

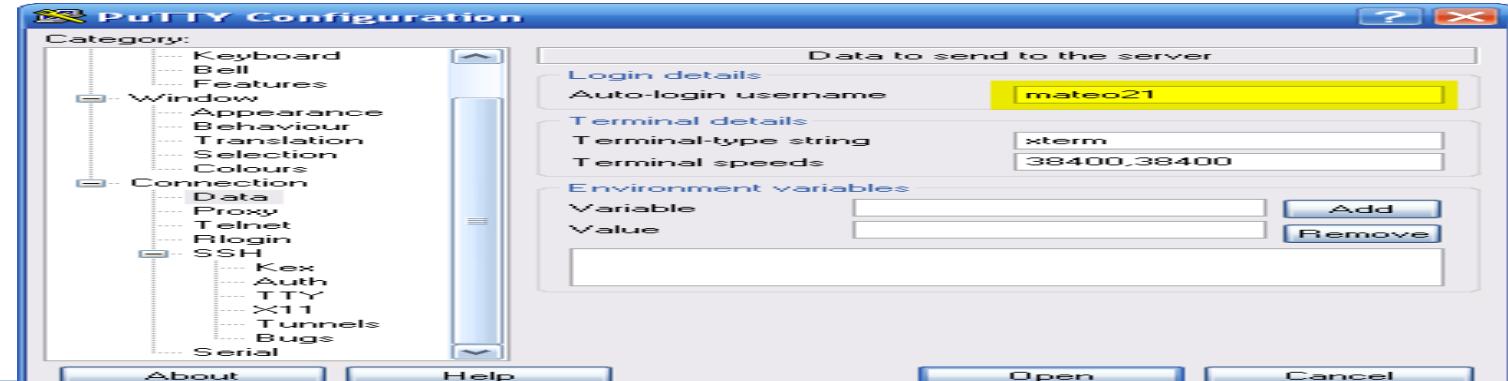
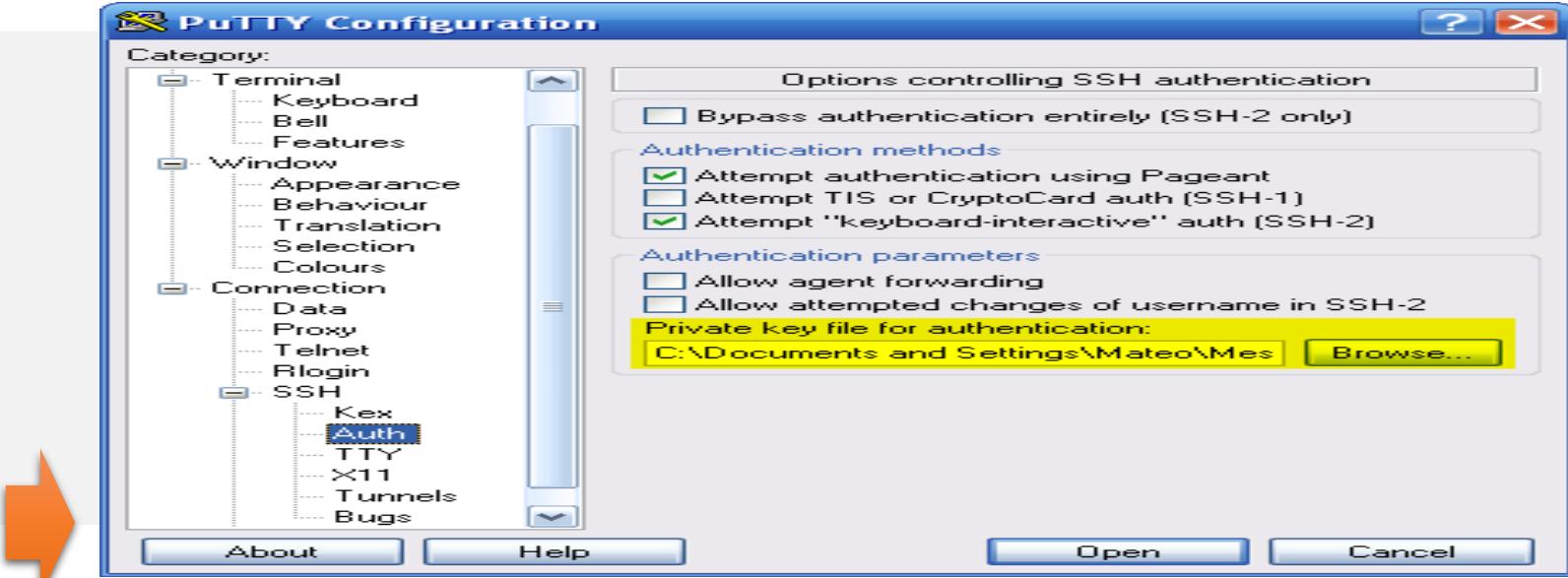
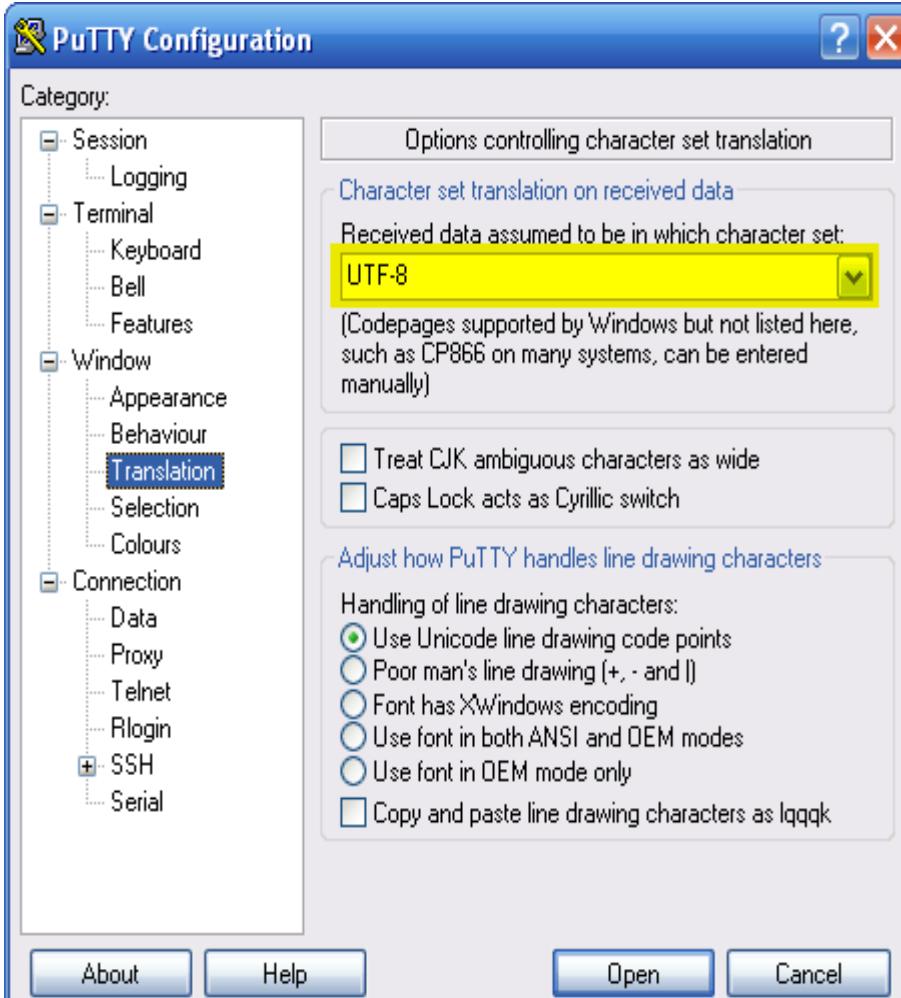
Rajoutez votre clé publique à la fin du fichier `authorized_keys` (s'il n'existe pas, il sera créé). Vous pouvez utiliser la commande suivante :

```
echo "votre_cle" >> authorized_keys
```



❖ Authentification par clé depuis Windows (PuTTY) avec Puttygen

Configurer PuTTY pour qu'il se connecte avec la clé





Se connecter via SSH à partir d'une machine Windows

❖ Authentification par clé depuis Windows (PuTTY) avec Puttygen

Configurer PuTTY pour qu'il se connecte avec la clé

