

Découverte du Système d'Exploitation Linux

Présentation de Linux

- Linux est un système d'exploitation conçu par Linus TORVALD en 1991.
- Linux est dit libre : le code source et le code binaire sont librement distribués.
- Plusieurs distributions (RedHat, Suse, Debian, etc...)
- Quelques caractéristiques de Linux :
 - Multitâches, Multiutilisateurs
 - Multiplateformes, Multiprocesseurs

Introduction

- Comment ouvrir une session Linux ?
- Comment obtenir de l'aide ?
- Qu'est ce qu'un fichier ?
- Qu'est ce qu'un répertoire ?
- Qu'est ce que le Shell ?

Arborescence des fichiers

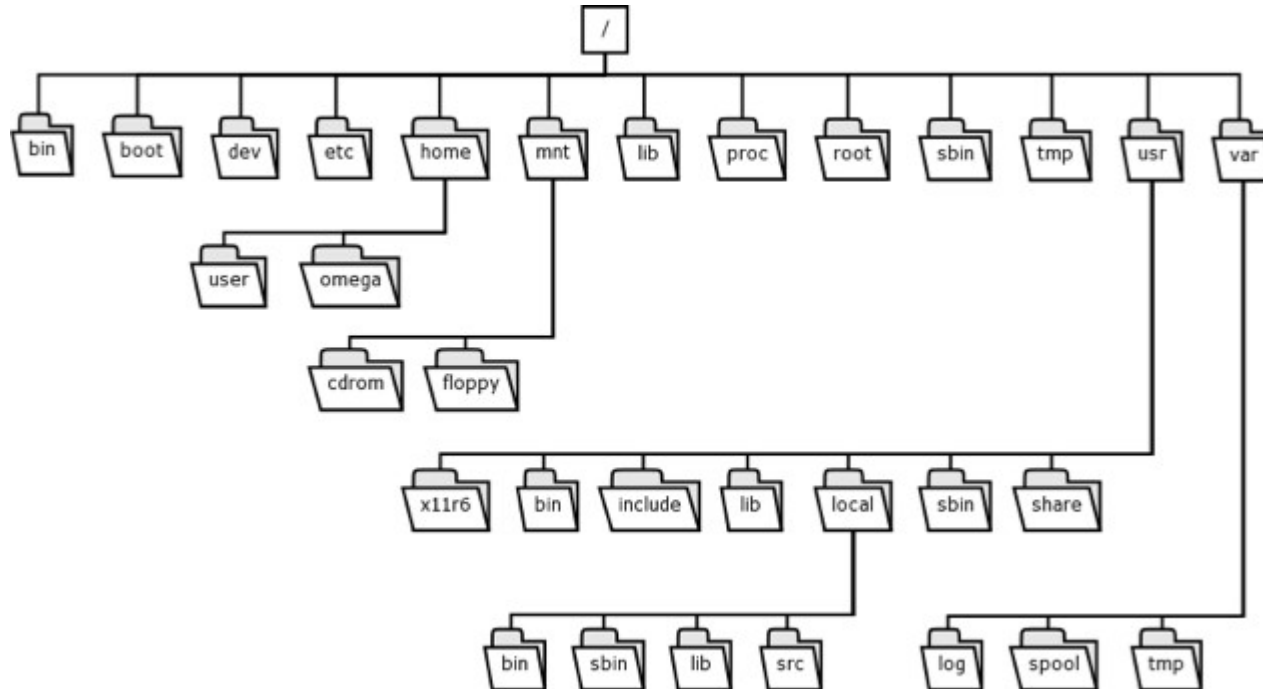
- Les fichiers du système Linux sont organisés en arbre avec une racine notée / en haut
- `ls /` : (list) affiche la hiérarchie à partir de la racine

Ordinateur	
Nom	Taille
bin	2869 éléments
boot	20 éléments
dev	210 éléments
etc	337 éléments
home	2 éléments
lib	509 éléments
lib64	3921 éléments
media	1 élément
mnt	0 élément
opt	18 éléments
proc	365 éléments
root	—
run	72 éléments
sbin	709 éléments
srv	0 élément
sys	11 éléments
@System.solv	3,9 Mo
tmp	25 éléments
usr	14 éléments
var	23 éléments

```
[olivier@olivier-esperanza ~]$ ls /
```

```
bin  dev  home  lib64      media  opt  root  sbin  sys      tmp  var
boot etc  lib   lost+found mnt     proc run  srv   @System.solv usr
```

Arborescence des fichiers



Les principaux répertoires sont :

- /etc regroupe tous les fichiers de configuration.
- /home contient les répertoires personnels des utilisateurs (sauf root)
- /usr contient les fichiers des programmes.

Arborescence des fichiers



- Le répertoire de travail (working directory) : indique où l'utilisateur se trouve dans l'arborescence
exemple : quand l'utilisateur «btssio» se connecte il arrive dans « /home/btssio »
- afficher le répertoire de travail : `pwd`
- 2 méthodes pour nommer un fichier
 - Chemin absolu : depuis la racine
 - « /home/btssio/Documents »
 - Chemin relatif : depuis le répertoire de travail
 - « Documents »

Arborescence des fichiers

pwd : (print working directory) affiche le répertoire courant

cd : (change directory) change de répertoire

Exemple : le répertoire de travail est /home/btssio et je veux me positionner sur le répertoire Documents

en absolu : `cd /home/btssio/Documents`

en relatif : `cd Documents`

Commandes de base Linux

Sur le système de fichiers

ls	affiche le contenu d'un répertoire
cd	change le répertoire courant
pwd	affiche la position courante dans l'arborescence
cp	copie un ou plusieurs fichiers
rm	efface un ou plusieurs fichiers
mkdir	crée un répertoire
rmdir	efface un répertoire s'il est vide
mv	déplace ou renomme un fichier
find	cherche un fichier
cat	affiche le contenu d'un fichier
tree	sous-arborescence du répertoire courant
more	affiche le contenu d'un fichier écran par écran
less	affiche le contenu d'un fichier écran par écran
ln	création d'un lien absolu (similaire à un alias)
ln -s	création d'un lien symbolique (similaire à un raccourci)

Commandes de base Linux

Sur la gestion des utilisateurs

whoami affiche le nom de login

useradd crée un utilisateur

userdel supprime un utilisateur

Sur la gestion du réseau

ip a affiche l'adresse ip des cartes réseau

nmcli dev show affiche la config. réseau complète

nmtui interface semi-graphique pour la gestion des interfaces réseaux

Les caractères génériques

? : remplace 1 et 1 seul caractère

- Ex : ls a?c

* : remplace de 0 à plusieurs caractères

- Ex : ls *.txt

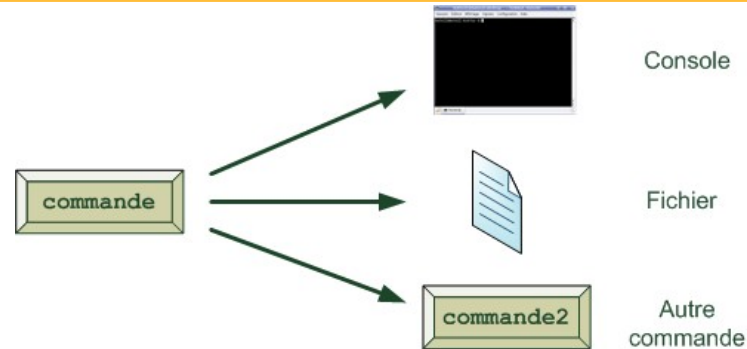
[c1,c2] : rend le caractère c1 ou c2 obligatoire

- Ex : ls [g,h]*

[c1-c2] : rend le caractère compris entre c1 et c2 obligatoire

- Ex : ls [w-z]*

Les redirections



- `>` : redirection en sortie dans un nouveau fichier
 - Ex : `ls / > message.txt`
- `>>` : redirection en sortie en ajoutant à la fin du fichier
 - Ex : `ls / >>message.txt`
- `<` : redirection en entrée dans un fichier
 - Ex : `mail untel@tel.adresse < message.txt`
- `<<` : redirection en entrée sur le clavier
 - Ex : `sort -n << FIN`

L'enchaînement des commandes

- Enchaînement des commandes sur une ligne : ;
 - Principe : commandes exécutées sans lien entre elles
 - Ex : `clear ; pwd`
- Chaînage des commandes (pipe) : |
 - Caractère : Alt Gr + 6
 - Principe : la sortie de la 1ère commande est redirigée en entrée de la 2ème commande
 - Ex : `du | sort`

```
[olivier@olivier-esperanza ~]$ ps -ef | grep packet
olivier    9249    2380    0 16:36 ?        00:00:00 /bin/bash /opt/pt/packettracer
olivier    9388    9328    0 16:37 pts/0    00:00:00 grep --color=auto packet
[olivier@olivier-esperanza ~]$
```

L'enchaînement des commandes

- Chaînage conditionnel : `||`
 - Principe : si la première commande se termine bien alors la deuxième n'est pas exécutée OU si la première commande échoue la deuxième est exécutée
 - Ex : `ls this_file_does_not_exist.txt || echo KO`
- Chaînage conditionnel : `&&`
 - Principe : la deuxième commande n'est exécutée que si la première se termine bien
 - Ex : `ls this_file_does_not_exist.txt && echo KO`
- Lancement en arrière-plan : `&`
 - Principe : lance une commande et redonne le prompt avant qu'elle soit terminée
 - Ex : `firefox &`

Bon à savoir

- La touche tabulation (deux flèches opposées, à gauche du clavier) déclenche la fonction d'auto-complétion. Celle-ci permet de compléter automatiquement les commandes et les chemins après n'avoir tapé que les premières lettres.
- Les flèches haut et bas permettent de parcourir l'historique des commandes, et d'accéder ainsi aux commandes tapées précédemment.
- La commande `man` suivie d'un argument permet d'accéder à la documentation détaillée de la commande passée en argument (exemple : `man ls`).