

Les commandes du Système d'Exploitation - Linux

Introduction

Le **Système d'Exploitation** est un logiciel de la famille des **logiciels de base** capable de fournir des services aux logiciels d'applications, de manière transparente à l'utilisateur.

Il arrive qu'on doive faire appel directement aux services au système, sans passer par un logiciel d'application. Deux modes de communication sont alors disponibles :

- le **mode graphique**: il permet, grâce à une interface conviviale (fenêtres et autres objets graphiques), de faire appel aux principaux services du système et de répondre aux besoins usuels (création de dossiers, etc.), de la même manière que lorsque nous travaillons avec les logiciels d'applications.
- le **mode texte** (console) : historiquement le premier, il permet d'appeler des services par l'intermédiaire de commandes : une commande correspond au code d'appel d'un programme du système qui fournit un service déterminé.

L'utilisation dans un cadre professionnel (informatique) d'un système d'exploitation requiert la connaissance du langage de commandes. Cette connaissance sera plus ou moins approfondie en fonction du profil (orienté administration ou orienté développement).

L'objectif de ce document est :

- de rappeler quelques notions relatives au mode commande
- de proposer une liste non exhaustive des commandes du SE Linux
- de fournir quelques exemples d'utilisation de commandes et quelques exemples de lots de commandes.

La distribution Linux utilisée est ici Ubuntu. Pour tester une distribution Linux :

- Installation en tant que système d'exploitation principal ;
- Installation en multiboot (choix du système au lancement) ;
- Utilisation d'un Live-CD ;
- Utilisation d'une machine virtuelle (exemple Vmware Player + Vmware Virtual Appliance – solution préconfigurée).

Table des matières

1. INTRODUCTION	2
2. RAPPELS SUR L'ARBORESCENCE DES REPERTOIRES	4
3. LES COMMANDES	4
4. QUELQUES COMMANDES LIEES AUX RESEAUX ET AUX MATERIELS	16
5. QUELQUES COMMANDES DE L'EDITEUR VI	16

1. Introduction

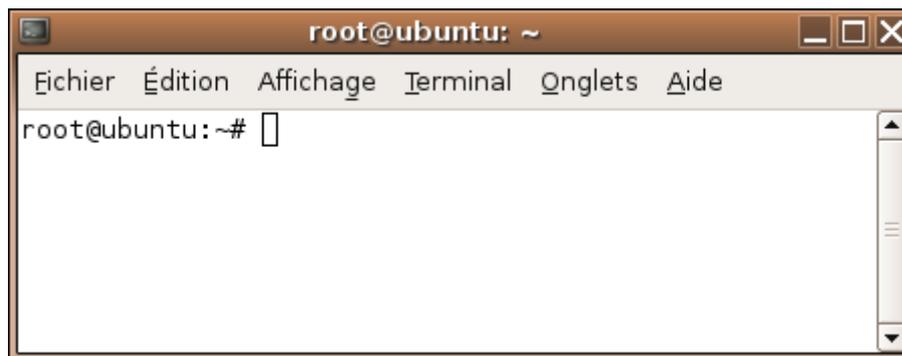
Le mode commande sous Linux fonctionne de manière similaire, pour l'utilisateur, au mode commandes sous Windows. Cependant, le mode commandes de Linux est natif et le bureau est une application ajoutée (d'où le choix entre plusieurs gestionnaires de bureau), alors que, pour Windows, le mode commande est une forme d'annexe au mode fenêtre.

Un interpréteur de commande, le Shell, se charge d'exécuter les ordres fournis par l'utilisateur. Alors que Windows offre un seul système d'interprétation de commandes de base (CMD.EXE), Linux offre la possibilité de modifier de manière plus souple le Shell à utiliser.

L'accès au Shell peut être réalisé de plusieurs manières :

- dans la barre «tableau de bord», démarrage du Shell dans l'environnement de l'utilisateur courant :
 - Menu Applications – Accessoires – Terminal
 - cliquez sur l'icône relative au Terminal
- CTRL-ALT & Fx (x = 1 à 6) (ou ALT & Fx) : permet le démarrage d'un Shell dans un autre environnement d'utilisateur et nécessite de se déclarer (login et password) = OUVREURE D'UNE NOUVELLE SESSION

L'invite de commande du Shell (ou prompt du shell) donne la possibilité de saisir une commande :



- **root** représente l'utilisateur sous lequel vous vous êtes connecté, ici : compte de l'administrateur système (cf. remarque ci-dessous).
- **ubuntu** représente le nom de l'ordinateur sur lequel vous êtes connectés.

➔ATTENTION : IL EST FORTEMENT CONSEILLE DE NE SE CONNECTER DIRECTEMENT SOUS root QU'EN CAS DE NECESSITE ABSOLUE

Pour **quitter** définitivement l'interpréteur de commande, il faut saisir la commande : **EXIT**

Pour **quitter** temporairement l'interpréteur de commande, CTRL-ALT & F7

Pour **quitter** le shell courant, CTRL-D

➔ATTENTION : LA COMMANDE 'halt' ARRETE L'ORDINATEUR !!!

Qu'est-ce qu'un 'shell ' (synonyme d'interpréteur de commande) ?

C'est un interpréteur de commandes, la partie du système d'exploitation utilisé comme interface avec l'utilisateur (en mode texte). Il permet l'exécution de scripts (suite de commandes mémorisées dans un fichier codé en texte brut).

Sous Linux, plusieurs interpréteurs de commandes (= plusieurs Shell) sont disponibles de base et une commande permet d'en changer simplement (sh, bash, etc.)

Script Shell

Un script est un fichier texte brut contenant une liste de commandes à exécuter.

Contrairement à Windows, où l'extension .BAT suffit à signaler au système d'exploitation que le fichier est un lot de commandes à exécuter, sous Linux il faut préciser, au niveau des propriétés du fichier, que celui-ci peut être exécuté (cf. commande chmod)

Redirection des entrées et des sorties

Les résultats des commandes sont envoyés vers un périphérique de sortie par défaut (sortie standard), la console. Il est possible de rediriger le résultat sur un autre périphérique que l'écran (console) :

- > Redirection en sortie vers un des périphériques
- >> Redirection en sortie en fin de fichier (le début de fichier est conservé)
- < Redirection de l'entrée standard

Tubes de redirection (pipes)

Canal de communication de la sortie d'une commande vers l'entrée d'une autre commande.

- | (correspond à la frappe des 2 touches clavier **AltGr + 6** alpha) Redirige le résultat d'une commande vers une seconde commande

Variables d'environnement

Les variables d'environnement permettent de spécifier des variables disponibles dans tous les scripts.

Par exemple :

La variable PATH définit les chemins d'accès pour les exécutables

```
PATH=$PATH:/home/moi/mesbin
```

Caractères de substitution ou caractères Jokers

Lorsqu'on lance une commande utilisant comme paramètres des noms de fichiers ou de répertoires, on peut utiliser des caractères « jokers » :

- * pour remplacer plusieurs caractères
- ? pour remplacer un seul caractère

Vous trouverez des exemples d'utilisation dans la description de commandes ou les exemples de fichiers batch.

L'utilisation de ces caractères est également disponible en mode graphique (en recherche de fichiers, par exemple), mais ils apportent aux commandes une puissance qu'il faut savoir mesurer.

!! AVERTISSEMENT : ATTENTION A LA L'UTILISATION DES CARACTERES DE SUBSTITUTION

Avant toute utilisation d'une commande utilisant ces caractères et qui modifie de manière important votre système (suppression de fichiers, de répertoires, renommer, changer des attributs...), vous devez impérativement tester l'étendue de ces caractères sur la commande.

Exemple : afficher d'abord la liste des fichiers à supprimer, puis en fonction du résultat, lancez la commande de suppression.

2. Rappels sur l'arborescence des répertoires

Linux possède une structure arborescente de répertoires définie. La plupart des distributions respectent cette arborescence :

- **/** = racine
 - **bin** = les binaires de la distribution Linux
 - **sbin** = les binaires système
 - **home** = répertoire qui contient les dossiers des utilisateurs (/home/tim = dossier de l'utilisateur tim)
 - **root** = dossier de l'administrateur
 - **dev** = devices (périphériques)
 - **etc** = fichiers de configuration (ex. /etc/password : mots de passes utilisateurs)
 - **usr** = binaires utiles aux utilisateurs
 - **tmp** = répertoire temporaire
 - **lost+found** = fichiers récupérés sur erreurs

Un chemin représente la liste des répertoires qui faut parcourir, traverser, pour atteindre un objet (fichier ou répertoire).

Le caractère / est employé pour déterminer la racine, et c'est le séparateur de répertoires d'un chemin (sous Windows : \, anti-slash)

Un chemin peut être :

- **absolu**, c'est à dire qu'il commence depuis la racine (/), par exemple
 - **/home/me/pages/www/**
- ou
- **relatif**, dans ce cas, la description ne commence que depuis le répertoire courant, relativement au répertoire courant ; par exemple **./me/pages/www/**
 - un « . » indique le répertoire courant ,
 - deux points « .. » indiquent le répertoire parent.

Représentation en mode commande (commande **tree**)

3. Les commandes

Le lancement d'une commande répond à une syntaxe du type :

COMMANDE [Parametres1 [parametre2 [...]]] [Option1 [option2 [...]]]
--

Les **arguments** de la ligne de commandes sont des PARAMETRES variables fournis par l'utilisateur et des OPTIONS (appelés également COMMUTATEURS) définies pour la commande (se rapprocher de l'aide relative à la commande).

Pour obtenir une aide générale :

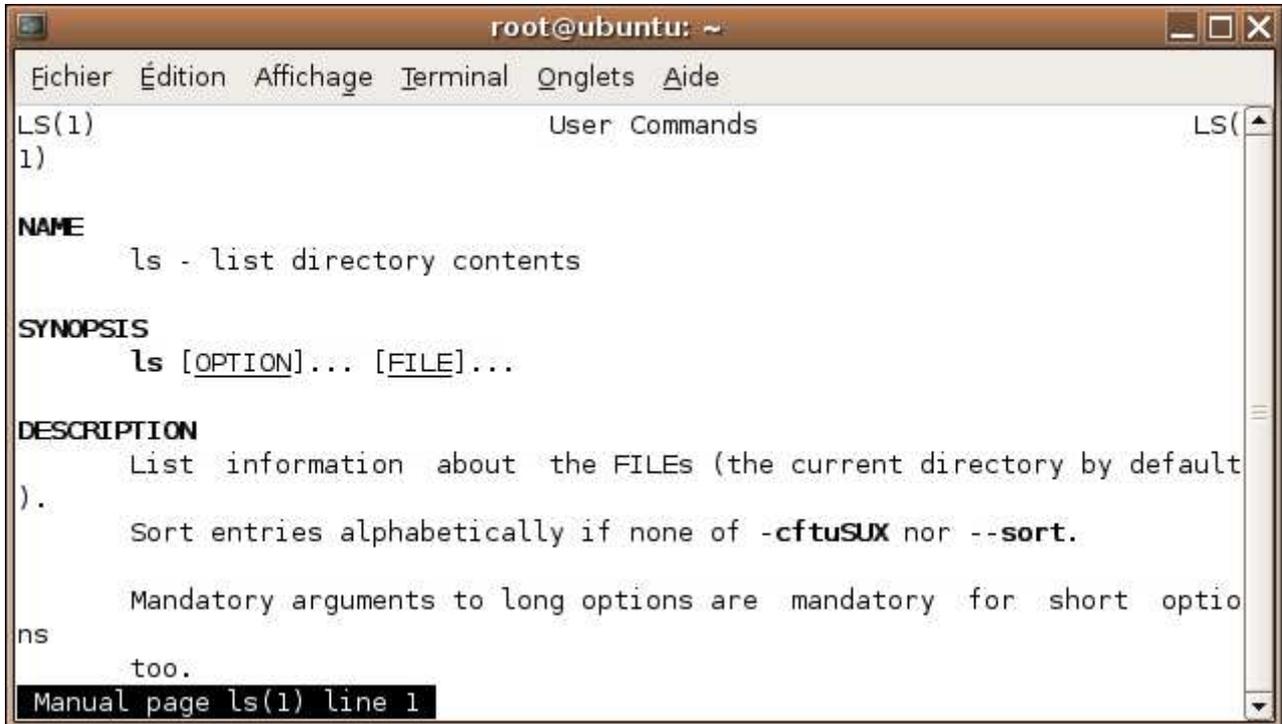
```
root@ubuntu:~# help
root@ubuntu:~# man
root@ubuntu:~# info
```

Pour avoir plus d'informations sur une commande spécifique, et notamment ses paramètres, entrez l'une des commandes ci-dessous :

Exemple pour obtenir l'aide relative à la commande **ls** :

```
root@ubuntu:~# man ls
root@ubuntu:~# help ls
root@ubuntu:~# info ls
```

Pour quitter le contexte d'aide : touche CTRL-z



Utilisez les touches **PhUp** et **PgDown** pour faire défiler l'aide, la barre d'espace pour descendre page par page.

La liste des commandes essentielles (entre parenthèses, la commande équivalente ou proche sous Windows)

addgroup	Permet l'ajout d'un groupe addgroup group01 → ajoute le groupe group01
adduser	Permet l'ajout d'un utilisateur. adduser util01 → ajoute l'utilisateur util01 adduser util02 group01 → ajoute l'utilisateur util01 et le place dans le groupe group01
alias	Permet d'associer un alias à une nouvelle alias monls="ls --color=auto" → lancer monls reviendra à lancer ce qui se trouve entre guillemets
at (AT)	Gérer des travaux à exécuter ultérieurement.
awk	Outil de manipulation de fichiers basé sur des modèles de recherche et de traitement de chaînes de caractères
cat ou more (TYPE)	Affiche le contenu d'un fichier <u>texte</u> . CAT permet de concaténer des fichiers cat /root/bashrc → affiche le contenu du fichier cat -n /root/bashrc → affiche le contenu du fichier avec une numérotation des lignes cat /etc/group → ce fichier contient la liste des utilisateurs et leur groupe ; on édite ce fichier pour ajouter un utilisateur à un groupe cat /etc/passwd → ce fichier contient les informations relatives aux utilisateurs (login, mot de passe, etc.) cat /etc/profile → ce fichier (script) permet de configurer le Shell (configurer le prompt, les couleurs, etc.) cat > fichier → enregistrer la saisie dans fichier jusqu'à la frappe de CTRL-D
cd (CD ou CHDIR)	Modifie le répertoire ou affiche le répertoire courant. cd .. : remonte au répertoire parent dans la hiérarchie cd / : se place dans le répertoire racine du volume courant cd /etc : remonte à la racine et redescend dans etc cd ~ : de positionne dans le répertoire par défaut de l'utilisateur courant
chfn	Changer des informations pour un utilisateur (change finger)
chsh	Changer le Shell pour un utilisateur (change shell)

chmod	<p>Changer les droits associés à un fichier ou à un répertoire</p> <p>chmod +x monmenu → rendre le fichier monmenu exécutable. chmod u=rwx,g=rw,o=r monmenu → lecture, écriture, exécution pour le <u>propriétaire</u>, lecture et écriture pour le <u>groupe</u>, lecture seule pour <u>les autres</u> (other) chmod 764 monmenu → syntaxe équivalente à ci-dessus</p> <p>Binaire ---- droit --- Décimal 000 ----- (---) ----- 0 001 ----- (--x) ----- 1 010 ----- (-w-) ----- 2 011 ----- (-wx) ----- 3 100 ----- (r--) ----- 4 101 ----- (r-x) ----- 5 110 ----- (rw-) ----- 6 111 ----- (rwx) ----- 7</p>
chown	<p>Changer le propriétaire et le groupe d'un fichier.</p> <p>chown nouv.ancien fic1 → le fichier 'fic1' appartient à nouv (au lieu de ancien) chown -R → modification du propriétaire d'un répertoire et des sous-répertoires</p>
chroot	<p>Redéfinir le répertoire associé à root.</p> <p>Cela permet de rattraper des erreurs du type mot de passe root oublié, etc.</p>
clear (CLS)	Effacer l'écran
cp (COPY ou XCOPY)	<p>Copie un ou plusieurs <u>fichiers ou dossiers</u>.</p> <p>cp fichier-src fichier-dest → copie le fichier 'fichier-src' dans le fichier 'fichier-dest' cp -i fichier-src fichier-dest → copie le fichier 'fichier-src' dans le fichier 'fichier-dest' mais avertit si 'fichier-dest' existe déjà cp -p dossier1 /home/me/dossier2 → copie le dossier1 vers le dossier /home/dossier2 avec conservation des propriétés cp -r dossier1 /home/me/dossier2 → copie le dossier1 et ses sous-dossiers vers le dossier /home/dossier2</p>
crontab	Permet d'initialiser le lancement cyclique d'une ou plusieurs commandes à partir d'un fichier qui contient la périodicité de lancement et la commande à lancer.
cut	<p>permet d'effectuer un filtre de colonnes sur un fichier texte avec séparateur</p> <p>cut -d : -f1,3 /etc/passwd → ????????????????????????????</p>
date (DATE)	Affiche ou modifie la date système.
df	Espace disque utilisé par point de montage.
deluser	Cf. userdel
delgroup	Cf. groupdel

du	(disk usage) Espace disque utilisé par le répertoire en cours et ses sous-répertoires
echo (ECHO)	Affiche des messages à l'écran ou active/désactive l'affichage des commandes. echo bonjour → affiche bonjour
env (SET)	Lister les variables d'environnement.
exit (EXIT)	Quitte le Shell
find (DIR)	Recherche un fichier dans une arborescence. find / -name monfic -print → rechercher à partir de la racine (/), d'après le nom (-name) monfic et affiche le résultat (-print) find /usr -type d -name bin -print → rechercher à partir de /usr, les objets de type d (=répertoires), dont le nom est bin, et afficher le résultat find /home -name '*.c' -print → rechercher tous les fichiers programmes C pour tous les utilisateurs find / -type f -size +300000c -print → rechercher tous les fichiers possédant une taille de plus de 300000 caractères find . -mtime +20 -print → rechercher tous les fichiers de l'utilisateur modifiés il y a plus de 20 jours
for (FOR)	Exécute une suite de commande pour chaque valeur d'une liste. for fruits in pommes poires fraises oranges do echo les \$fruits sont des fruits done → affichage du message pour chaque valeur de la liste
grep (FIND ou FINDSTR)	Cherche les fichiers qui contiennent un texte grep un_mot /home/page/fichier/ → rechercher la chaîne de caractères 'important' dans les fichiers du répertoire temp, dont l'extension est htm et dont le nom se termine par '1' et affiche leur nom grep -l -n un_mot /home/pages/* → rechercher les fichiers contenant le mot un_mot dans tous les fichiers du dossier spécifié, liste ces fichiers et le numéro de ligne trouvée grep -i 'printf' prog.c → rechercher toutes les lignes du fichier qui comportent 'printf' en majuscule ou minuscule
groupadd	Cf. addgroup
gzip	Compresser ou décompresser des fichiers et dossiers.
head	Afficher les premières lignes d'un fichier (voir aussi tail) head -10 *.c → afficher les 10 premières lignes des fichiers sources C

help (HELP)	Affiche des informations sur les commandes de base (man est préférable). help > aide → affiche le contenu de l'aide, mais la redirection envoie ce texte dans le fichier aide.txt (efface les données éventuellement stockées dans aide.txt) help ls → affiche le contenu de l'aide relative à la commande ls
if (IF)	Effectue un traitement conditionnel dans un fichier de commandes.
Info	aide
kill	Envoyer un signal à un processus (pour le stopper par exemple) kill -9 1568 → envoyer le signal 9 au processus dont le PID (numéro du processus, cf ps) : demander l'arrêt du processus kill -l → lister les signaux possibles
last	Liste des utilisateurs qui se sont connectés (à partir du fichier /var/log/wtmp)
ln	Créer ou modifier un lien (lien dur : associer 2 ou plusieurs fichiers à un même emplacement, même inode, mais 2 fichiers en ls ; lien symbolique : fait référence à un fichier dans un répertoire, il a une existence propre) ln fichier_src fichier_lien ln -s → créer un lien symbolique sur le fichier_src
locate	Recherche un fichier locate monfic → recherche le fichier dont le nom est monfic, dans la base de données maintenue par la commande updatedb (fichier /usr/lib/locatedb).
lpr, lpq, lprm (PRINT)	Envoyer un fichier vers le spool.
ls (DIR)	Affiche la liste des fichiers et des dossiers du répertoire courant. ls → affiche la liste simple des objets du répertoire courant ls -l → affiche une liste des fichiers et leurs attributs (droits, propriétaire, taille, date création ou modification) ls -a → liste tous les objets (même les fichiers cachés) ls -CR → liste récursive du répertoire courant et de ses sous-répertoires ls -lu → lister avec date de dernier accès
mkdir (MD ou MKDIR)	Crée un répertoire. mkdir nomrep → crée le répertoire nommé nomrep dans le répertoire courant mkdir -p /home/me/monrep → crée le répertoire nommé monrep et l'arborescence si elle n'existe pas

more (MORE)	Affiche le contenu d'un fichier page par page more /root/.bashrc → affiche le contenu du fichier
mount	Monter le système de fichiers mount -l → liste les points de montage 1. Créer un point de pointage (= créer un point d'accueil) : mkdir /mnt/cdrom 2. monter le cdrom mount -t iso9660 /dev/hdb /mnt/cdrom 3. démonter le cdrom umount /mnt/cdrom
mv (MOVE)	Déplace un ou plusieurs fichiers ou répertoires d'un répertoire à un autre. mv fichier /home/me/dossier → déplace le fichier dans le dossier cible
mv (REN ou RENAME)	Renomme un ou plusieurs fichiers. Le nouveau nom ne doit pas déjà exister dans le répertoire courant. mv fich1 fich2 → renomme le fichier fich1.txt en fich2.txt dans le répertoire courant
passwd	Modifier le mot de passe.
ps	Afficher les processus en cours. ps u → afficher les processus de l'utilisateur ps au → afficher les processus de tous les utilisateurs ps ax → afficher les processus en cours sur la machine
pstree	Afficher les processus en cours sous forme arborescente (interdépendance de processus)
pwd (CD)	Affiche le répertoire courant. (Print Working Directory)
read	Demande une saisie utilisateur et la stocke dans une variable read variable → variable est récupérée puis utilisée grâce à \$variable
rm (DEL ou ERASE)	Supprime un ou plusieurs fichiers ou dossiers rm fichier1 → supprime le fichier dans le répertoire courant rm -r dossier1 → supprime le dossier 'dossier1' et ses sous-répertoires rm -i *.o → supprimer les fichiers se terminant par .o avec demande de confirmation
rmdir (DEL ou ERASE)	Supprimer un ou plusieurs répertoires vides
sdiff	Signaler les différences entre 2 fichiers sdidd fic1 fic2 → liste des différences entre les fichiers fic1 et fic2

shutdown	<p>Arrêter le système</p> <p>shutdown -r now → redémarrer le système</p>
sort (SORT)	Trie les lignes d'un fichier texte.
su	Commande pour passer en root (super utilisateur) ;exit pour sortir et revenir à l'utilisateur initial
sudo	<p>Commande pour exécuter une commande avec le privilège de root, avec conservation d'une trace dans les journaux. Le fichier /etc/sudoers est consulté pour vérifier l'autorisation de l'utilisateur à exécuter ce qu'il demande</p> <p>sudo shutdown -r now → demander le rechargement du système</p>
tail	Afficher les dernières lignes d'un fichier (voir aussi head)
tar	Pour compresser ou décompresser des fichiers
top	Afficher l'activité des processus en cours
touch	Permet la création d'un fichier vide, modifie le timestamp d'un fichier
tree (TREE)	<p>Représente graphiquement l'arborescence d'un lecteur ou d'un chemin. Exemple :</p> <p>tree / -d → affiche l'arborescence à partir de la racine, seulement les répertoires</p>
umask	<p>Définir la protection par défaut lors de création de fichiers et répertoires en fonction du niveau de sécurité souhaité.</p> <p>Principe : à partir d'une permission maximale 111-111-111, on enlève les bits dont on ne veut pas</p> <p>umask u=rwx, g=rwx, o=rwx → rwx r-x r-x umask 022 → rwx r-x r-x umask 066 → rwx --x --x umask 077 → rwx --- ---</p>
unzip	<p>décompresser une archive ZIP</p> <p>unzip cinema.zip -d ./ : décompresser l'archive cinema.zip dans le répertoire courant</p>
useradd	<p>Ajouter un utilisateur (cf adduser)</p> <p>useradd toto → créer l'utilisateur toto</p>
userdel	<p>Supprimer un utilisateur (cf. deluser)</p> <p>userdel toto → supprimer l'utilisateur toto</p>
vi (EDIT)	Lance un éditeur de texte.
wc	<p>Compter le nombre de caractères, mots ou lignes d'un fichier</p> <p>wc -w fic1 → affiche le nombre de mots du fichier fic1 wc fic1 → affiche le nombre de lignes, de mots et de caractères du fichier</p>

which	Fournit le chemin d'un exécutable which ls → affiche le dossier où se trouve la commande ls
who	Liste des utilisateurs connectés au système (sessions démarrées)

Exécution conditionnelle et opérateurs de comparaison

<pre>if [condition] then ... suite de commandes ... fi</pre>	<pre>if [condition] then ... suite de commandes 1... else ... suite de commandes 2... fi</pre>
--	--

```
if [ condition1 ]
then
    ... suite de commandes 1...
elif [ condition2 ]
then
    ... suite de commandes 2...
elif [ condition3 ]
then
    ... suite de commandes 3...
else
    ... suite de commandes 4...
fi
```

Opérateur	Description	Exemple
Opérateurs sur des fichiers		
<code>-e filename</code>	vrai si <i>filename</i> existe	[<code>-e /etc/shadow</code>]
<code>-d filename</code>	vrai si <i>filename</i> est un répertoire	[<code>-d /tmp/trash</code>]
<code>-r filename</code>	vrai si <i>filename</i> est lisible (r)	[<code>-r /boot/vmlinuz</code>]
<code>-w filename</code>	vrai si <i>filename</i> est modifiable (w)	[<code>-w /var/log</code>]
<code>-x filename</code>	vrai si <i>filename</i> est exécutable (x)	[<code>-x /sbin/halt</code>]
<code>file1 -nt file2</code>	vrai si <i>file1</i> plus récent que <i>file2</i>	[<code>/tmp/foo -nt /tmp/bar</code>]
<code>file1 -ot file2</code>	vrai si <i>file1</i> plus ancien que <i>file2</i>	[<code>/tmp/foo -ot /tmp/bar</code>]
Opérateurs sur les chaînes		
<code>-z chaine</code>	vrai si la <i>chaine</i> est vide	[<code>-z "\$VAR"</code>]
<code>-n chaine</code>	vrai si la <i>chaine</i> est non vide	[<code>-n "\$VAR"</code>]
<code>chaine1 = chaine2</code>	vrai si les deux chaînes sont égales	[<code>"\$VAR" = "toto"</code>]
<code>chaine1 != chaine2</code>	vrai si les deux chaînes sont différentes	[<code>"\$VAR" != "totp"</code>]
Opérateurs de comparaison numérique		
<code>num1 -eq num2</code>	égalité	[<code>\$nombre -eq 27</code>]
<code>num1 -ne num2</code>	inégalité	[<code>\$nombre -ne 27</code>]
<code>num1 -lt num2</code>	inférieur (<)	[<code>\$nombre -lt 27</code>]
<code>num1 -le num2</code>	inférieur ou égal (<=)	[<code>\$nombre -le 27</code>]

Architecture logicielle – Langages de commandes - Linux

<i>num1</i> -gt <i>num2</i>	supérieur (>)	[\$nombre -gt 27]
<i>num1</i> -ge <i>num2</i>	supérieur ou égal (>=)	[\$nombre -ge 27]

Exemple de script Shell

Un script Shell est un fichier texte brut qu'on peut éditer avec tout éditeur de texte (comme vi). Il commence par une ligne spéciale (appelée « shebang ») indiquant quel programme utiliser pour exécuter les commandes du script :

```
#!/bin/sh
#!/bin/bash
#!/usr/bin/perl
```

Les fichiers de script sont souvent suffixés par « .sh »; cependant cela ne les rend pas exécutable pour autant : il faut donc utiliser la commande **chmod** pour définir les droits d'exécution des scripts :

```
chmod 751 mon_script.sh
```

Affichage des valeurs de la liste

```
#!/bin/bash
for fruits in pommes poires fraises oranges
do
echo les $fruits sont des fruits
done
exit 0
```

Affichage de chaque paramètre :

```
#!/bin/bash
for par in $@
do
echo $par
done
exit 0
```

Test d'existence de fichiers

```
#!/bin/sh

echo premier test

if [ -f '/tmp/fic1' ] || [ -f '/tmp/fic2' ]
then
echo "fic1 OU fic2 dans tmp!"
fi

echo deuxieme test

if [ -f '/tmp/fic1' ] && [ -f '/tmp/fic2' ]
then
echo "fic1 ET fic2 dans tmp!"
fi

exit 0
```

Tests de présence de paramètres

```
#!/bin/sh

if [ ! $1 ] || [ ! $2 ]
then
    echo 'vous devez fournir l'utilisateur et le mot de passe'
    exit 1
fi

if [ ! $3 ]
then
    var='1';
else
    var=$3;
fi

echo $var

exit 0
```

recherché de chaîne de caractères dans tous les fichiers :

```
$find . -name \*.cpp -exec grep -q "debug" '{}' \; -print
```

4. Quelques commandes liées aux réseaux et aux matériels

ifconfig

Obtenir les paramètres des interfaces de réseau

Exemple : **ifconfig -a** → affiche les informations détaillées

ping

Utilitaire TCP/IP le plus utilisé pour le test et le dépannage, il permet de déterminer si l'ordinateur local peut émettre et recevoir des datagrammes d'un ordinateur distant :

Exemple : **ping** 192.168.0.1 ou **ping** www.voila.fr

route

Utilitaire TCP/IP qui permet de lister, de modifier ou de supprimer des éléments de la table de routage

netstat

Affiche les statistiques du protocole et les connexions TCP/IP actives sur notre ordinateur.

iptables

Mettre en œuvre les règles de filtrage de paquets IP

proc

Affiche les statistiques du protocole et les connexions TCP/IP actives sur notre ordinateur.

arp

Gestion du cache ARP du système

lspci

Lister les périphériques PCI

mount

Monter un périphérique

chroot

Exécuter un Shell avec un répertoire racine spécifié

5. Quelques commandes de l'éditeur vi

L'éditeur **vi** possède un mode commandes et un mode insertion.

__Ajouter des données : **Echap**, puis 'a' (pour append) pour ajouter du texte après le curseur

__Ajouter des données : **Echap**, puis 'D' (pour delete) pour supprimer une ligne sous le curseur

__**Echap**, pour repasser en mode commandes, puis **:w nom_fichier** pour enregistrer le fichier

__**Echap**, **:wq** pour enregistrer et quitter