

Commandes linux

Il existe beaucoup de commandes linux à taper dans un terminal. Elles servent à tout faire, en voici une liste non-exhaustive...

Les symboles:

- `.` : désigne le répertoire courant
- `..` : désigne le répertoire parent
- `/` : désigne le répertoire racine
- `~` : désigne le répertoire `/home/user`, `user` étant votre identifiant
- `xx*` : tous les fichiers commençant par `xx`...
- `*.x` : tous les fichiers se terminant par `.x`

Ces symboles sont utilisables dans toutes les commandes, un exemple en a été donné dans la commande "cd".

APROPOS et WHATIS

Définitions (intro du "man") pour les commandes.

```
$ apropos cp $ whatis cp
```

- "apropos xx" affiche les définitions des commandes contenant xx dans leurs noms
- "whatis xx" donne la définition de la commande xx

CAT

Concatène des fichiers et affiche sur la sortie standard (utile pour les redirections ">").

```
$ cat $ cat toto.txt $ cat toto.txt titi.txt
```

- "cat" renvoie l'entrée standard sur la sortie standard (cool !!)
- "cat xx" affiche xx sur la sortie standard
- "cat xx yy" affiche à la suite dans la console xx et yy

CD

Pour ouvrir un répertoire.

```
$ cd $ cd rep $ cd .. $ cd ~/
```

- *rien* : ouvre votre répertoire personnel (ex : */home/toto*)
- *rep* : ouvre le répertoire *rep*
- *..* : ouvre le répertoire parent
- *~/* : ouvre le */home/user*

CHMOD

Pour donner les droits sur un fichier ou un répertoire on utilise la commande **chmod** :

La petite commande **ls -l** indique les droits de chaque catégorie d'utilisateur sur chaque fichier ou répertoire :

```
rwx | r-- | ---
```

- La première partie fixe les droits de propriétés pour le propriétaire du fichier.
- La deuxième partie fixe les droits accordés aux utilisateurs faisant partie du groupe auquel appartient le fichier.
- La dernière partie fixe les droits des autres utilisateurs.

Avec les lettres et symboles :

```
$ chmod u+x fichier_ou_repertoire $ chmod g-w fichier_ou_repertoire $ chmod o=r fichier $  
chmod a+rwx fichier_ou_repertoire
```

- *u+x* : donne à l'utilisateur possédant le fichier (répertoire) le droit en execution (ou accès au répertoire)
- *g-w* : retire aux groupe d'utilisateurs possédant le fichier le droit en écriture sur le fichier (répertoire)
- *o=r* : autorise la lecture du fichier aux autres utilisateurs
- *a+rwx* : donne à tout le monde tous les droits sur le fichier (répertoire)

Avec les chiffres :

```
$ chmod 777 fichier_ou_repertoire $ chmod 000 fichier_ou_repertoire
```

Commandes linux de base

Écrit par MACS 2009

Mardi, 14 Juillet 2009 17:42 - Mis à jour Dimanche, 31 Janvier 2010 12:35

- 777 : donne à tout le monde tous les droits sur le fichier ou repertoire
- 000 : retire à tout le monde tous les droits sur le fichier ou repertoire

Le tableau résumant les correspondances binaire-lettres-chiffres :

Binaire	-----	Logique	-----	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	CHOW	

N

Pour changer le propriétaire d'un fichier ou dossier:

chown chown

Ajouter -R (option récursive) pour changer le propriétaire des sous dossiers et fichiers du :

chown -R **CP**

Pour copier un fichier ou répertoire.

\$ cp test /home \$ cd test /home/test0 \$ cp -i test /home \$ cp -r test /home \$ cp -v test /home

- test /home : copie le répertoire *test* dans le répertoire */home*
- test /home/test0 : copie le répertoire *test* dans le répertoire */home* et le renomme *test0*

(attention,

/home/test0

ne doit pas exister)

- -i : vérifie l'existence d'un fichier du même nom et demande pour le remplacer ('y' yes ou 'n' no)

- -r : copie de manière récursive l'ensemble d'un répertoire et de ses sous-répertoires
- -v : affiche le nom des fichiers copiés

FIND

Pour rechercher des fichiers dans une arborescence de répertoires :

Commandes linux de base

Écrit par MACS 2009

Mardi, 14 Juillet 2009 17:42 - Mis à jour Dimanche, 31 Janvier 2010 12:35

\$ find -name

- : la première option de "find" est la racine des répertoires de recherche du fichier.
- -name : pour indiquer par la suite le nom du fichier que l'on cherche

GREP

Pour rechercher une chaîne de caractère dans un fichier spécifique :

\$ grep -n vecteur fichier.cpp

- grep -n : permet de rechercher et d'afficher les lignes des occurrences du mot trouvé
- vecteur : est la chaîne de caractère que l'on cherche
- fichier.cpp : est le fichier dans lequel on cherche la chaîne "vecteur"

GZIP

Pour compacter et décompacter des fichiers au format .gz.

\$ gzip lettre \$ gzip -d lettre.gz

- lettre : compacte le fichier *lettre*
- -d : décompacte le fichier *lettre.gz*

LS

Liste les répertoires et fichiers du répertoire courant.

\$ ls \$ ls exo* \$ ls /home/bin \$ ls -l

- exo* : liste les répertoires et fichiers commençant pas exo...
- /home/bin : listing du répertoire */home/bin*
- -l : listing des attributs des fichiers et répertoires (les droits de lecture,tailles,...)

MAN

Afficher l'aide des commandes linux et de certaines fonctions standard de C

\$ man \$ man mkdir \$ man printf **MKDIR**

Commandes linux de base

Écrit par MACS 2009

Mardi, 14 Juillet 2009 17:42 - Mis à jour Dimanche, 31 Janvier 2010 12:35

Pour créer un répertoire dans le répertoire courant.

```
$ mkdir rep $ mkdir -p rep/sousRep
- -p : permet de créer la suite de répertoires "parents".
```

MORE et LESS

Affiche dans la console le contenu d'un fichier

```
$ more toto.txt $ less toto.txt
```

Pour ces deux commandes, il faut lire le "man" pour voir comment afficher les lignes, aller à la fin, rechercher,...

MV

Pour déplacer ou renommer un fichier ou répertoire.

```
$ mv toto tata $ mv toto /home $ mv -i toto /home
- toto tata : renomme toto en tata
- toto /home : déplace toto dans /home
- -i : demande confirmation pour déplacer ('y' yes ou 'n' no)
```

PS et KILL

Tuer un processus.

```
$ ps -e $ kill $ kill -9
- -e : liste les processus pour savoir le numéro du processus à kill
- kill: tue le processus
- kill -9: tue le processus mais avec l'option -9 si jamais ça ne marche pas sans
```

PWD

Commandes linux de base

Écrit par MACS 2009

Mardi, 14 Juillet 2009 17:42 - Mis à jour Dimanche, 31 Janvier 2010 12:35

Pour savoir où l'on se trouve : chemin du répertoire courant.

\$ pwd **RM**

Pour supprimer un fichier ou un répertoire.

\$ rm toto \$ rm -i toto \$ rm -r toto \$ rm -f toto

- toto : supprime le répertoire *toto*
- -i : demande confirmation pour supprimer ('y' yes ou 'n' no)
- -r : supprime le répertoire et ses sous répertoires
- -f : supprime le répertoire même s'il contient des fichiers protégés en écriture

TAR

Pour archiver et désarchiver des données. La commande "tar" n'est pas une commande de compression c'est pourquoi elle est associée à une méthode de compression le "zip → ".tar.gz" ou le "bz2" → ".tar.bz2"

\$ tar -xzf projet.tar.gz \$ tar -xjf projet.tar.bz2 \$ tar -xf projet.tar \$ tar -cvfz projet.tar.gz projet
\$ tar -cvfj projet.tar.gz projet \$ tar -tvzf projet.tar.tgz \$ tar -tvjf projet.tar.tgz

- xzf : décompacte un fichier au format *.tar.gz* ou *.tgz* (x : extraction, z : décompactage *.tar.gz*, f : création d'un fichier)
- xjf : décompacte un fichier au format *.tar.gz* ou *.tgz* (x : extraction, j : décompactage *.tar.bz*, f : création d'un fichier)
- xf : décompacte un fichier au format *.tar.gz* ou *.tgz* (x : extraction, j : décompactage *.tar*, f : création d'un fichier)
- cvfj : compacte le fichier *projet* au format *.tar.gz* ou *.tgz* et le nomme *projet.tar.gz* (c : archive, v : mode verbose (affichage), f : fichier, z : compactage gz)
- cvfj : compacte le fichier *projet* au format *.tar.bz2* et le nomme *projet.tar.bz2* (c : archive, v : mode verbose (affichage), f : fichier, j : compactage bz2)
- tvzf : affiche l'index d'une archive *.tar.gz* ou *.tgz* sans décompacter (t : index)
- tvjf : affiche l'index d'une archive *.tar.bz2* sans décompacter (t : index)

TOP

Commandes linux de base

Écrit par MACS 2009

Mardi, 14 Juillet 2009 17:42 - Mis à jour Dimanche, 31 Janvier 2010 12:35

Affiche les processus en cours et les consommations de CPU et de mémoire en temps réel.

\$ top

- + P : classement par CPU
- + M : classement par mémoire

UNZIP

Pour "zipper" et "dézipper" des données. La méthode de compression "zip" est lu par tout les systèmes d'exploitation "facilement", il est donc parfois préférable de l'utiliser lors d'échange de fichier windows/unix.

\$ unzip fichier.zip -d destination

"dézippe" le "fichier.zip" dans le répertoire "destination".

WGET

Télécharger un fichier d'internet avec la console

\$ wget http://adresse/fichier.extention

télécharge le fichier.extention dans le répertoire courant

XARGS et |(pipe)

XARGS : Redirige une sortie de commande en entrée d'une autre commande.

|(pipe) : Permet la succession de plusieurs commandes liées en n'affichant que la dernière.

\$ find . -name "*.cpp" | xargs ls

- |(pipe) : enchaîne sur l'autre commande (la commande "find" n'affichera rien dans la console)
- xargs : redirige la sortie de "find" en entrée de "ls"

Exemple comparatif :

Commandes linux de base

Écrit par MACS 2009

Mardi, 14 Juillet 2009 17:42 - Mis à jour Dimanche, 31 Janvier 2010 12:35

```
$ find . -name "*.cpp" | grep "test" $ find . -name "*.cpp" | xargs grep "test"
```

- La première commande recherchera "test" dans **les noms** des fichiers trouvés par "find"

- La seconde commande recherchera "test" dans **le corps** des fichiers trouvés par "find" (redirection par "xargs" du résultat de "find" en argument d'entrée de "grep")

Liens

Voici deux autres listes de commandes linux:

- [Wikipedia Commandes Unix](#)
- [Ubuntu-fr consoles commandes de base](#)
- Fiche rappel des commandes UNIX: [PDF](#) , [ODT](#) issu de [FOSSwire](#)
- Fiche rappel des commandes Ubuntu: [PDF](#) , [ODT](#) issu de [FOSSwire](#)

[Contact: admin@lmacs.fr]