Collectif d'auteurs

Memento des commandes GNU/Linux les plus utiles

IN LIBRO VERITAS
www.inlibroveritas.net

Ligne de commande de base pour le débutant

Introduction

Ce *How To* est fait pour les débutants qui ne connaissent pas **GNU/Linux**, et n'a pas la prétention de faire de vous des champions de la console. La distribution Ubuntu vise à simplifier l'utilisation d'une plate-forme Debian en limitant l'usage des commandes dans un terminal *(shell)* pour une station basique pour le quotidien.

Convention

* Cette page vous donnera des commandes GNU/Linux basiques à utiliser dans un shell (terminal).

* Tous les noms de commande seront en **GRAS**.

* Les commandes devront être tapées en respectant **la casse** selon l'exemple encadré.

* Pour un complément d'information, lisez le **man** de chaque commande.

Commandes basiques

man

La commande **man** est employée pour vous montrer le manuel des autres commandes. (man = manuel) Pour obtenir une bonne compréhension, employez l'exemple :

```
man man          ### ceci vous donnera
la page d'information pour "man"
```

Note : Pour quitter **man**, appuyez sur la touche **q**. Pour rechercher le mot *exemple* tapez **/exemple**

Certains utilitaires sont mieux documentés en tant que texte info. par exemple essayez

```
info man
```

cd

La commande **cd** vous permettra de changer de répertoire (cd = change directory). Quand vous ouvrez un terminal en mode utilisateur vous serez dans votre répertoire local (/home/utilisateur).

Dans un système linux la référence au fichier s'appelle un chemin. Dans un chemin le nom des répertoires et des fichiers sont séparés par un "/". Il existe deux types de chemin : absolu et relatif.

Le chemin absolu se base sur la racine de l'arborescence et commence par "/" : ex : /home/utilisateur/<dossier>/<fichier>.

```
cd /home/utilisateur/dossier
### vous déplacera à votre répertoire
(/home/utilisateur/dossier)
```

Le symbole «~» peut remplacer le chemin absolu vers votre répertoire personnel soit «/home/utilisateur/».

```
cd ~/Desktop                ### vous dépla-
cera à votre répertoire de //bureau//
(/home/utilisateur/Desktop)
```

Le chemin relatif dépend du répertoire courant où se trouve l'utilisateur Pour se déplacer dans un dossier de l'emplacement courant (par défaut home/utilisateur) vous emploierez **cd** suivit du nom du dossier : cd <dossier>. Ex se rendre dans le bureau (Desktop).

```
cd Desktop
```

Si vous êtes déjà dans le répertoire ~/Desktop$ pour aller dans un des dossiers qu'il contient :

```
~/Desktop$ cd <dossier>
```

La commande **cd** utilisée seule ramène au répertoire par défaut de l'utilisateur (ou du root).

```
cd
```

Pour "remonter" d'un répertoire (aller à son parent) on utilise la commande "cd .."

```
cd ..           ### vous remontera d'un
dossier
```

mkdir

La commande **mkdir** vous permettra de créer des répertoires.(mkdir = make directory) Exemple :

```
mkdir musique          ### créera un
répertoire musique

man mkdir              ### pour avoir
les options de mkdir
```

mkdir code

pwd

La commande **pwd** vous permettra de savoir dans quel répertoire de l'arborescence vous êtes localisé. (pwd = present working directory) Exemple :

```
pwd             ### suivant nos exem-
ples ci-dessus nos sommes dans ~/Des-
ktop ou /home/utilisateur/Desktop
```

mv

La commande **mv** servira à déplacer un dossier à un endroit différent ou renommer un dossier.(mv = move) Exemples :

```
mv bonjour bonsoir              ###
pour renommer le fichier/répertoire
«bonjour» en «bonsoir»

mv bonsoir ~/Desktop            ###
déplace le fichier «bonsoir» du ré-
pertoire courant vers le répertoire
~/Desktop sans le renommer
```

```
mv bonsoir ~/Document/bonnenuit
###déplace le fichier «bonsoir» du
répertoire courant vers le répertoire
~/Desktop et le renomme en bonnenuit
(biensûr bonnenuit n'existait pas
dans le répertoire ~/Desktop sinon le
système vous demandera la confirmation
pour écraser l'ancien fichier)

man mv          ### pour avoir les op-
tions de mv
```

cp

La commande **cp** fera une copie d'un fichier. (cp = copy) Exemple :

```
cp bonjour bonsoir fera une copie
exacte du fichier «bonjour» et l'ap-
pellera «bonsoir», mais le fichier
«bonjour» sera toujours là.
Quand vous employez «mv» le fichier
n'existerait plus, alors que quand
vous utiliserez «cp» le fichier se dé-
double sans être supprimé.

man cp          ### pour avoir les op-
tions de cp
```

less

La commande **less** permet d'afficher le contenu d'un fichier directement dans le terminal. Exemple:

```
less /etc/apt/sources.list   ###affi-
chera par exemple le contenu de votre
fichier sources pour apt

.....
 deb http://archive.ubuntu.com/ubuntu
warty main restricted universe multi-
verse

 deb http://archive.ubuntu.com/ubuntu
warty-security main restricted

 deb http://archive.ubuntu.com/ubuntu
warty-updates main restricted univer-
se multiverse
.....

man less                ### pour avoir
les options de less
```

Bien sur il en existe beaucoup d'autre (cat, nano, vi...) et la syntaxe est la même mais les fonction-

nalités sont bien différentes. Dans tous ces cas informez vous avec la commande man ou sur google.

rm

La commande **rm** permet de supprimer un fichier ou répertoire.(rm = remove) Attention cette commande est irréversible donc soyez vigilant quand vous l'utilisez. Exemple :

```
rm nom_fichier          ### supprime le
fichier nom_fichier du répertoire cou-
rant

rm /home/documents/nom_fichier
### supprime le fichier nom_fichier du
répertoire /home/documents
```

ATTENTION

Si vous tapez :

```
rm / home/documents/nom_fichier
###il y a un espace entre / et home/
documents/nom_fichier donc le système
commencera par faire un rm / puis
fera un rm home/documents/nom fichier
```

```
Bien sûr, le rm / effacera complè-
tement votre système de fichier donc
attention aux espaces qui se glissent
dans la frappe quand vous utilisez la
commande "rm".

man rm          ### pour avoir les op-
tions de "rm"
```

Note : **rm /** est quand même largement un mythe vu qu'il faudrait les droits super-utilisateur (root) et avoir vidé les sous répertoires... (**sudo rm -fR /** est nettement plus dangeureux).

chown

La commande **chown** permet de changer le propriétaire d'un fichier ou répertoire. (chown = change owner) Seul le propriétaire du fichier (ou root) peut faire cette manipulation. (*Plus d'informations sur la gestion des droits d'accès sous Linux : http://doc.ubuntu-fr.org/droits*)

Exemple :

```
chown utilisateur2 /home/utilisateur/
doc.txt          ### «utilisateur2» de-
vient propriétaire du fichier doc.txt
```


```
man chown                    ### pour avoir
les options de «chown»
```

chmod

La commande **chmod** permet de modifier les droits d'accès sur les fichiers ou répertoires. Seul le propriétaire des fichiers ou répertoires (ou root) peut faire cette manipulation. (*Plus d'informations sur la gestion des droits d'accès sous Linux : http://doc.ubuntu-fr.org/droits*)

Il existe 2 méthodes : symbolique ou octale.

Méthode symbolique

chmod ugoa|+/-/=|rwxugo fichier

Les paramètres de chmod se décomposent là en en trois parties:

- la 1ère indique à qui s'applique la modification des droits d'accès :
  - u pour l'utilisateur, le propriétaire du fichier (u pour user)
  - g pour le groupe (g pour group)
  - pour le reste du monde (o

pour others)

- a pour tous (a pour all)

- la 2ème est un caractère +, - ou = :
  - + signifie l'ajout de nouveaux droits d'accès
  - - signifie la suppression de droits d'accès
  - = signifie l'autorisation exclusive des droits d'accès spécifiés

- la 3ème indique le(s) droit(s) concernés :
  - r pour lecture (r pour Read)
  - w pour écriture (w pour Write)
  - x pour exécution (x pour eXecute)
  - u,g,o pour reprendre les droits du propriétaire,groupe,autres utilisateurs

Quelques précisions valables dans le cas d'un répertoire :

- r signifie lecture totale du répertoire (la commande ls liste ainsi par exemple tous les fichiers contenus dans le répertoire). Mais en l'absence de ce droit, il est toujours possible de lire un

fichier contenu dans ce répertoire (en connaissant son chemin).

- x signifie droit d'ouverture du répertoire. Pour empêcher la "traversée" d'un répertoire, c'est ce droit qu'il faut enlever.

Exemples :

```
chmod g+r fichier        ### permet au
groupe de lire le Fichier
chmod u+rw,go+r fichier     ### permet
à l'utilisateur de lire et écrire le
fichier, et aux autres de le lire
chmod g=u fichier        ### permet de
donner au groupe les mêmes droits que
l'utilisateur
```

Méthode octale

chmod XXX fichier, où XXX = Utilisateur | Groupe | Autres (X représente un entier compris entre 1 et 7)

Valeur du chiffre X :

- 0 : aucun droit en lecture, écriture, exécution
- 1 : droit d'exécution

- 2 : droit d'écriture
- 4 : droit de lecture

On peut cumuler différents droits : X = 7 (soit 1+2+4) signifie donc que l'on donne tous les droits sur le fichier.

en résumé X = Lecture (4) + Écriture (2) + Exécution (1)

- Le 1er chiffre X spécifie les droits pour le propriétaire du fichier.
- Le 2ème chiffre X spécifie les droits pour le groupe propriétaire du fichier.
- Le 3ème chiffre X spécifie les droits pour tous les autres utilisateurs sur le fichier.

Ainsi, chmod 777 fichier donne tous les droits à tout le monde.

Un calculateur de chmod est disponible sur http://www.toulouse-renaissance.net/c_outils/c_chmod.htm

Exemple :

```
chmod -c 644 /home/utilisateur/texte.txt     ### modifie les droits en rw-r-
```


```
-r-- pour le fichier texte.txt c'est à
dire que seul le

    propriétaire peut écrire et les
autres seulement lire

man chmod       ### pour avoir toutes
les options sur "chmod"
```

apt-get

La commande **apt-get** permet de gérer les paquets Debian de votre Ubuntu. Exemple :

```
sudo apt-get update ### permet de
mettre à jour la liste des paquets
disponibles, commande à taper en pre-
mier

   avant toute installation pour être
sur d'avoir les mises à jour.

sudo apt-get upgrade  ### permet de
mettre à jour les paquets déjà ins-
tallés, à taper pour faire les mises
à  jour de sécurité.

sudo apt-get install soft     ###
```

```
Installera le logiciel "soft" en gé-
rant les dépendances, donc "apt" vous

    demandera peut être d'installer
d'autres paquets en complément.

sudo apt-get remove soft       ### dé-
sinstallera le paquet "soft".

sudo apt-get autoremove soft  ###
désinstallera "proprement" le paquet
"soft" ainsi que ses dépendances
```

apt-cache search

La commande **apt-cache search** permet de rechercher le nom d'un paquet parmi ceux disponibles. Avant d'utiliser cette commande pensez à faire un **apt-get update** pour mettre à jour votre liste des paquets. Exemple :

```
apt-cache search supersoft    ###
vous donnera la liste des paquets
dont le nom ou la description

contient "supersoft".
```

Liens

- How To Debian sur APT (*http://www.debian.org/doc/manuals/apt-howto/index.fr.html*)
- Introduction à Linux (*http://www.math-linux.com/spip.php?article22*)
- Autre page utile : Apprenez les bases de Unix en 10 minutes. (*http://doc.ubuntu-fr.org/tutoriel/learn_unix_in_10_minutes*)

Learn UNIX in 10 minutes. Version 1.2 FR 1.0

Les commandes de base de la ligne de commande UNIX (le shell) : dernière révision 17 Mai 2001
Site Original : Learn UNIX in 10 minutes (*http://freeengineer.org/learnUNIXin10minutes.html*)

Avant Propos

Ce document a été rédigé il y a quelques années pour des étudiants en Dessin Assisté par Ordinateur (DAO) . Le but était d'avoir sur une page les commandes de base pour utiliser le Shell Unix (comme ça ils ne me demanderaient pas quoi faire lorsque quelqu'un leur donnerait une sauvegarde)

 Envoyez-moi vos commentaires, corrections ou tout autre ajout qui vous semble absolument nécessaire dans ce document.

Sections

Chemins

Les chemins de fichiers et de dossiers sous Unix utilisent le slash "/" pour séparer les noms des dossiers.

Exemples :

/	Dossier "racine"
/usr	Dossier usr (sous-dossier du dossier "racine")
/usr/STRIM100	STRIM100 est un sous-dossier de /usr

Se déplacer dans le système de fichier

pwd	montre le nom du dossier de travail courant (Present Working Directory)
cd	change le dossier de travail pour un autre dossier

cd /usr/STRIM100	change le dossier de travail pour /usr/STRIM100
cd INIT	change le dossier de travail pour INIT, qui est un sous-dossier du dossier courant
cd ..	change le dossier de travail pour le dossier parent
cd $STRMWORK	change le dossier de travail pour le dossier défini par la variable d'environnement 'STRMWORK'

Lister le contenu d'un dossier

`ls` liste le dossier de travail courant

`ls` -l dossier liste au format détaillé le dossier dossier

Par exemple :

```
 $ ls -l /home/sheherazade/work/
drwxr-xr-x 4 sheherazade staff 1024  2004-04-04 09:40 ToDo
-rw-r--r-- 1 sheherazade staff 767392 2004-04-04 14:28 scanlib.tar.gz
 ^ ^  ^  ^     ^   ^             ^           ^      ^        ^      ^
 | |  |  |     |   |             |           |      |        |      |
 | |  |  |     |  Propriétaire  Groupe      Taille Date     Heure  Nom
 | |  |  |    Nombre de fichiers ou dossiers que le dossier listé contient
 | |  | Permissions pour tous
 | | Permissions pour les membres du groupe staff
 |Permissions pour le propriétaire  r = lecture (read), w = écriture (wri-
te), x = exécute (execute),  - = pas de droits
Type de fichier * : - = Fichier régulier, d = Dossier, l = Lien symbolique
ou autre...
```

• Sous Unix tout est fichier (*http://linux.ensimag.fr/fichiersunix.html*)

Modifier les permissions et les droits

chmod

En employant la méthode alphabétique :

Ce qu'il faut savoir :

• u : utilisateur (**u**ser), g : groupe (**g**roup) et o : autres (**o**ther)

• r : lire (**r**ead), w : écrire (**w**rite) et x : exécuter (e**x**ecute)

```
chmod [ugo][+-=][rwx] fichier
```

Vous devez donc choisir le groupe à modifier (u, g, o) suivis du symbole + pour donner une permission, du symbole - pour la retirer ou du symbole = pour définir la permission exacte suivi des permissions à appliquer (r, w, x).

Par exemple pour accorder à l'utilisateur d'exécuter le fichier.

```
chmod u+x fichier
```

Ou pour retirer les droits d'écriture et d'exécution au groupe et aux autres.

```
chmod go-wx fichier
```

En employant la méthode numérique :

Il faut savoir que x=1, w=2 et r=4 ensuite vous additionnez les chiffres si vous voulez cumuler les droits. L'ordre des chiffres est propriétaire/groupe/autres.

Par exemple pour accorder la lecture, l'écriture et l'exécution pour le propriétaire et la lecture et l'exécution pour le groupe et les autres.

```
chmod 755 fichier
```

On a propriétaire/groupe/autres donc rwx/rx/rx donc 4+2+1/4+1/4+1 et donc 7/5/5.

Astuce pour ceux qui connaissent le binaire. On veut rwx lire/écrire/exécuter donc 111 en binaire qui vaut 7 en décimal. On veut r-x lire/**pas ecrire**/exécuter donc 101 en binaire qui vaut 5 en décimal.

chgrp

```
chgrp staff fichier
```

change le fichier fichier afin qu'il appartienne au groupe staff.

chown

```
chown sheherazade fichier
```

fait de sheherazade la propriétaire du fichier fichier.

```
chown -R sheherazade dir
```

fait de sheherazade la propriétaire du dossier dir et de tous ses sous dossiers.

/!\ Vous devez être le/la propriétaire du fichier/dossier ou être root avant de taper ces commandes. Sous Ubuntu il faut les préfixer par **sudo**.

Déplacer, renommer et copier des fichiers

cp fichier_source fichier_destination	copie un fichier

cp -r dossier_source dossier_destination	copie un dossier
mv fichier1 nouveau_nom_fichier1	déplace ou renomme un fichier
rm fichier1 [fichier2 ...]	supprime un fichier (ou une liste de fichiers)
rm -r dossier1 [dossier2...]	supprime un dossier et tous ses sous-dossiers, à manier avec précaution
mkdir dossier1 [dossier2...]	crée un dossier
rmdir dossier1 [dossier2...]	supprime un dossier vide

Visualiser et éditer les fichiers

cat fichier	affiche le contenu du fichier à l'écran en ASCII

more fichier	affiche progressivement un fichier à l'écran : Entrer = descend d'une ligne, Espace = descend d'une page, q = quitte
less fichier	comme more, mais on peut utiliser la touche Page Précédente. Pas disponible sur tous les systèmes.
vi fichier	éditer un fichier avec l'éditeur vi. Tous les systèmes Unix ont un éditeur *à la vi*.
emacs fichier	éditer un fichier avec l'éditeur emacs. Pas disponible sur tous les systèmes.
head fichier	affiche les premières lignes d'un fichier
head -n fichier	affiche les n premières lignes d'un fichier
tail fichier	affiche les dernières lignes d'un fichier
tail -n fichier	affiche les n dernières lignes d'un fichier

Shells

Le comportement de l'interface en ligne de commande diffère légèrement en fonction du programme shell utilisé.

Suivant le shell utilisé, quelques comportements peuvent être pratiques.

Vous pouvez connaître le shell que vous utilisez avec la commande :

```
printenv SHELL
```

Vous pouvez bien évidemment créer un fichier contenant une liste de commandes shell et l'exécuter comme un programme pour exécuter une tâche. On appelle cela un script shell. C'est en fait le but premier de la plupart des shells, et non pas le comportement interactif de la ligne de commande.

Variables d'environnement

Vous pouvez apprendre au shell à 'mémoriser' des informations pour utilisation ultérieure grâce aux variables d'environnement. Par exemple avec bash :

```
export CASROOT=/usr/local/CAS3.0
```

définit la variable CASROOT avec la valeur /usr/local/CAS3.0.

```
cd $CASROOT
```

change le répertoire de travail courant pour CASROOT.

```
export LD_LIBRARY_PATH=$CASROOT/
Linux/lib
```

définit la variable LD_LIBRARY_PATH qui prendra la valeur CASROOT avec /Linux/lib ajouté à la fin, c'est-à-dire /usr/local/CAS3.0/Linux/lib.

```
printenv
```

affiche toutes les variables d'environnement

```
printenv CASROOT
```

affiche la valeur de la variable d'environnement nommée CASROOT : /usr/local/CAS3.0.

```
echo $CASROOT
```


fait exactement la même chose.

Historique Interactif

Une fonctionnalité de bash et tcsh (et parfois d'autres) est de pouvoir utiliser la touche "flèche vers le haut" pour accéder aux dernières commandes entrées, les éditer, et les re-exécuter.

Complétion des noms de fichiers

Une des fonctionnalités de bash et tcsh (et probablement d'autres) est de pouvoir utiliser la touche TAB pour compléter un nom de fichier tapé partiellement. Par exemple, si vous avez un fichier nommé constantine-monks-and-willy-wonka.txt dans votre répertoire et si vous voulez l'éditer, vous pouvez taper 'vi const', enfoncer la touche TAB, et le shell va remplir le reste du nom pour vous (à condition qu'il soit unique).

Bash vous montre la voie

Bash complète aussi bien les noms de commandes que les variables d'environnement. En cas de possibilités multiples de complétion, taper deux fois sur la touche TAB vous montrera toutes les complétions possibles. Bash est le shell par défaut de

la plupart des systèmes Linux.

Redirection

```
grep chaine fichier > nouveau_fichier
```

redirige la sortie de la commande précédente ‘grep’ dans un fichier nommé *nouveau_fichier*. Si *nouveau_fichier* existe il sera *remplacé/ecrasé*

```
grep chaine fichier >> fichier_existant
```

ajoute la sortie de la commande ‘grep’ à la fin du fichier *fichier_existant*.

Les opérateurs de redirection > et >> peuvent être utilisés sur la sortie de la plupart des commandes, pour les placer dans un fichier.

Pipes

Le symbole tube ("pipe") "|" est utilisé pour rediriger la sortie d’une commande vers une autre.

Par exemple :

```
ls -l | more
```


Cette commande prend la sortie du listage au format long des fichiers dans un répertoire produit par "ls -l" et la redirige vers la commande "more" (aussi appelé filtre). Dans ce cas, une très longue liste de fichier peut être vue page par page.

Substitution de Commande

Vous pouvez utiliser le résultat d'une commande comme paramètre d'entrée pour une autre, en d'autres termes pour une substitution de commande. Une substitution de commande a lieu lorsque vous encadrez une commande avec des apostrophes inversées. Par exemple :

```
cat `find . -name aaa.txt`
```

va afficher à l'ecran (voir cat) le contenu de tous les fichiers nommés aaa.txt dans le dossier de travail courant ou ses sous-dossiers.

Rechercher une chaîne de caractères : La commande grep

```
grep chaine fichier
```

affiche toutes les lignes de fichier contenant chaine

Rechercher des fichiers : La commande find

Syntaxe :

```
find chemin -name fichier
```

```
find . -name aaa.txt
```

cherche les fichiers nommés aaa.txt dans le dossier courant ou ses sous-dossiers.

```
find / -name vimrc
```

cherche les fichiers nommés vimrc depuis le dossier racine

```
find /usr/local/games -name"*xpilot*"
```

cherche tous les fichiers dont le nom contient xpilot dans le dossier /usr/local/games

Créer des archives : La commande tar

La commande tar signifie ‘**t**ape **ar**chive’ (archive sur bande). C’est la méthode ‘standard’ pour lire et écrire des archives.

Vous trouverez souvent des archives de fichiers

avec des noms comme fichiers.tar, ou fichiers.tar.gz. Ce sont respectivement des fichiers dans une archive tar et des fichiers dans une archive tar qui a été compressée avec le programme de compression gzip.

Il est très probable, si quelqu'un vous envoie des fichiers, qu'ils soient compressés dans une archive tar.

De même, si vous devez envoyer des fichiers, vous devriez utiliser tar.

tar -xvf archive.tar	extrait les fichiers de l'archive archive.tar, en affichant les noms des fichiers qui la composent
tar -xvzf archive.tar.gz	extrait les fichiers de l'archive en utilisant gzip puis tar
tar -jxvf archive.tar.bz2	extrait les fichiers de l'archive en utilisant bzip2 puis tar

tar -cvf archive.tar fichier1 [fichier2...]	Crée un fichier «archive.tar» contenant fichier1, fichier2...
tar -cvzf archive.tar.gz mon_dossier	crée un fichier gzip contenant tout le contenu du dossier 'mon_dossier'

Compression de fichier : gzip, bzip2

gzip

gzip est un outil GNU de compression et de decompression. L'extension pour les fichiers gzippés est .gz.

gzip fichier.txt	crée le fichier compressé fichier.txt.gz
gunzip fichier.txt.gz	extrait le fichier fichier.txt

bzip2

L'utilitaire bzip2 offre (en général) un meilleur taux de compression que gzip, mais au coût d'un temps

de compression et décompression plus élevé.

bzip2 fichier.txt	crée le fichier fichier.txt.bz2
bunzip2 fichier.txt.bz2	décompresse le fichier fichier.txt.bz2.

Besoin d'aide : La commande man

La plupart des commandes ont une page de manuel qui donne une description de leurs utilisations plus ou moins détaillées, parfois utiles, parfois obscures. Certains disent qu'elles ont été appelées "man pages", parce qu'elles ne s'adressent qu'aux vrais Hommes.

Exemple:

```
man ls
```

affiche la page de manuel pour la commande ls

Pour obtenir des man pages en français :

```
sudo apt-get install manpages-fr
```

Commandes de base de l'éditeur Vi

Ouvrir un fichier

```
vi nom_du_fichier
```

Mode édition

`i` : **i**nsère avant le curseur

`I` : **I**nsère au début de la ligne courante

`a` : insère **a**près le curseur

`A` : insère à la fin de la ligne

`r` : remplace 1 caractère

`R` : passe en mode **R**emplacement\

`<ECHAP>` : met fin au mode Edition (insertion ou remplacement)

Effacer du texte

`x` : efface un seul caractère

`dd` : efface la ligne courante et la place dans un

buffer

`ndd` : Efface n lignes (n est un nombre) et les place dans le buffer

`J` : déplace la ligne suivante à la fin de la ligne courante (effacement des caractères de retour chariot)

Oups

`u` : annule la dernière commande
Copier et coller

`yy` : copie la ligne courante dans le buffer

`nyy` : copie n lignes dans le buffer (n est un nombre)

`p` : colle le contenu du buffer après la ligne courante

`P` : colle le contenu du buffer avant la ligne courante

Positionnement du curseur

`gg` : va en haut de la page

`G` : va en bas de la page

`:n` : positionne le curseur à la ligne n

`:$` : positionne le curseur à la fin de la ligne

`^g` : affiche le numéro de la ligne

`h,j,k,l` : respectivement gauche, bas, haut et droite ; les flèches de direction fonctionnent aussi...

Recherche de chaîne de caractères

/chaîne: recherche «chaîne». n permet le passage à l'occurrence suivante. # permet de passage à l'occurrence précédente.

*: recherche avant de la chaîne de caractères présente sous le curseur.

#: recherche arrière de la chaîne de caractères présente sous le curseur.

Substitution de chaîne de caractères

:n1,n2:s/chaîne1/chaîne2/[g][c] : substitue chaî-

ne1 pour chaîne2 sur les lignes n1 à n2. Si g est indiqué (global), toutes les références de chaîne1 sont remplacées, sinon seule la première référence est remplacée. Si c est indiqué (confirm), une confirmation sera demandée avant chaque modification.

^ : méta-caractère de début de ligne

. : méta-caractère, vrai sur n'importe quel unique caractère sauf retour chariot

$: méta-caractère de fin de ligne

Ces caractères, tout comme les autres caractères spéciaux, peuvent être 'échappés' à l'aide du \ : c'est-à-dire pour capturer la chaîne de caractères "/usr/STRIM100/SOFT", il faut utiliser l'expression régulière "\/usr\/STRIM100\/SOFT"

Exemples :

```
:1,$:s/chien/chat/g
```

Remplace 'chien' par 'chat', pour chaque occurrence du fichier - de la ligne 1 à la fin du fichier ($)

```
:23,25:s/chien/chat/
```

Remplace ‘chien’ par ‘chat’ des lignes 23 à 25. Une fois par ligne, dès la première apparition.

Sauver, quitter et commandes d’exécution

Ces commandes sont toutes préfixées par les deux points (:) et apparaissent dans le coin inférieur gauche de la fenêtre.
Vous ne pouvez pas saisir ces commandes en mode édition. Appuyer sur <ECHAP> pour sortir du mode édition

`:w` : enregistre le fichier (Write)

`:w` nouveau.fichier : enregistre le fichier sous le nom ‘nouveau.fichier’

`:wq` : enregistre le fichier et quitte le programme

`:q` : quitte

`:q!` : quitte sans enregistrer les changements

`:e fichier` : ouvre ‘fichier’ pour l’édition

`:set number` : affiche les numéros de ligne

`:set nonumber` : cache les numéros de ligne

`:set noai` : désactive l'auto-indentation

FAQs

Vous trouverez la plupart du temps toutes les réponses à vos questions dans le manuel ou la documentation du programme. Si toutefois vous ne trouvez pas de réponse, pensez à chercher sur les forums ou les groupes de discussion **avant** de poser votre question.

Copyrights

Le document original a été rédigé à l'aide de Vim. Vim est la meilleure version du seul véritable éditeur de texte : vi

tion ; avec comme la préface section invariante, pas de couverture ni quatréme de couverture. Consultez la GNU Free Documentation License : GFDL (*http://www.gnu.org/licenses/fdl.html*).

1) This is something that I had given out to students (CAD user training) in years past. The purpose was to have on one page the basics commands for getting started using the UNIX shell (so that they didn't call me asking what to do the first time someone gave them a tape).
This document is copyrighted but freely redistributable under the terms of the GFDL . Send me comments, corrections, and extra stuff that you think should absolutly must be included. I'll gladly listen.
Invariant translations (*http://www.faqs.org/docs/linux_intro/gfdl-8.html*)

2) Il n'y a pas de corbeille ! ;-)

Cet ouvrage a été composé à partir du travail bienveillant de la communauté francophone des utilisateurs d'Ubuntu.

http://www.ubuntu-fr.org/

Clin d'œil aux utilisateurs de Debian et dérivés :

```
apt-get moo
aptitude moo
aptitude -v moo
aptitude -vv moo
aptitude -vvv moo
aptitude -vvvv moo
aptitude -vvvvv moo
aptitude -vvvvvv moo
```

Memento des commandes GNU/Linux
les plus utiles
N° ISBN : 978-2-35209-148-6
N° EAN : 9782352091486
Achevé d'imprimé en France pour le compte
d'InLibroVeritas.net en 2008