



Les commandes de base sous Linux

But de ce document

Ce document donnera une description et des exemples d'utilisations des commandes de base les plus courantes sous Linux.

Ces commandes sont classées en 3 grandes familles :

- ▶ Aide & recherche
- ▶ Réagencer, se déplacer
- ▶ Système

Aide & recherche

man [-options] [section] [commande]

Affiche la page de manuel correspondant à la commande. Les pages de manuel sont réparties dans différentes sections :

- ▶ Programmes exécutable et commandes shell.
- ▶ Appels système.
- ▶ Appels bibliothèques.
- ▶ Fichiers spéciaux.
- ▶ Formats des fichiers et protocoles.
- ▶ Jeux.
- ▶ Macros et fichiers divers.
- ▶ Routines du noyau (non standard)

grep [-options] regex fichiers

Recherche dans un ou plusieurs fichiers la présence de l'expression régulière regex. Nous verrons plus tard le domaine des expressions régulières, outil difficile à maîtriser mais d'une puissance formidable.

Option Commentaire

- l Affiche une liste de fichiers contenant l'expression recherchée.
- L Affiche une liste de fichiers ne contenant pas l'expression recherchée.
- v Affiche les lignes de fichiers ne contenant pas l'expression recherchée.

Exemples :

```
$ grep -l "10.0.0.1" *
```

```
$ grep '^ [A-Z] \{6\} [0-9] \{4\} * /etc/*'
```

find [-nchemins] [conditions]

Rubrique : [Utilisation](#)

Le : mardi 29 mars 2005

Par : Tony

Visites : 2427

Voilà une commande extrêmement utile mais comportant un grand nombre d'options. Elle suit l'arborescence de chaque *nchemins* et identifie les fichiers qui répondent aux critères définis par conditions. Je vous renvoie à la page de manuel pour mieux comprendre les possibilités de cette commande.

Condition

Commentaire

```
-exec
```

```
commande {}
```

```
\;
```

Applique *commande* à chacun des fichiers reconnus par *find*. Quand *commande* est lancée, est remplacé par le nom du fichier présentement traité. La séquence de commande doit se terminer par un point-virgule balisé `\;` ;.

-name motif

Recherche les fichiers dont le nom correspond à *motif* .

-type c

Recherche les fichiers de type *c* . *c* peut être :- **b** (block special file),- **c** (character special file),- **d** (directory),- **p** (named pipe),- **l** (symbolic link),- **s** (socket),- **f** (plain file).

Exemples :

```
$ find . -type d
```

```
$ find / -name core -exec rm -f {} \;
```

```
$ find / -type f -name "*conf*" -exec cp {} /tmp/conf_files/{}.save \;
```

Réagencer, se déplacer

ls [-options] [répertoire(s)]

Cette commande liste le contenu du répertoire en cours. Elle accepte de nombreux arguments. Les principaux sont :

Option Commentaire

- l Permet d'obtenir une liste détaillée (Type du fichier, taille, droits...).
- a Liste également les fichiers cachés (ceux qui commencent par . et ..).
- t Trie les fichiers par date (les plus récents en premier).
- r Inverse l'ordre du tri.
- d Affiche les répertoires mais pas leur contenu.
- R Affichage récursif des sous-répertoires en plus des répertoires.

Exemples :

```
$ ls -latr /home
```

```
$ ls -R /usr
```

cd [chemin/répertoire/]

Cette commande permet de changer de répertoire. Dans le répertoire en cours, on trouve deux entrées particulières, "." et "..". L'entrée "." symbolise le répertoire en cours, et l'entrée ".." le répertoire directement supérieur (Le répertoire père). Ces deux entrées sont très pratiques pour se déplacer relativement au répertoire en cours. Une option utile, le tiret "-", qui permet de revenir au répertoire courant précédent.

Chemin Commentaire

`..` Remonte dans le dossier parent.

`~` Retourne à votre dossier personnel.

Exemple :

```
$ cd /Cours/Commande/
$ pwd
$ /home/thierry/Cours/Commande
$ cd
$ pwd
$ home/thierry
```

pwd

Affiche la référence absolue du répertoire de travail actuel. (Le répertoire en cours)

rm [-options] fichier(s)

Supprime un ou plusieurs fichiers. Il est obligatoire d'avoir l'autorisation d'écriture sur le fichier pour pouvoir le supprimer.

Option Commentaire

- f** Supprime sans avertissement les fichiers protégés en écriture.
- i** Demande à l'utilisateur de confirmer chaque suppression.
- r** Si fichier est un répertoire, le supprime ainsi que tout son contenu lorsque celui ci est couplé avec **f**.

Exemples :

```
rm -rf /home/toto
rm -i *.wav
```

mv [-options] source cible

Déplace ou renomme fichiers et répertoires.

Option Commentaire

- b Sauvegarde les fichiers avant de les écraser.
- f Force le déplacement si cible existe.
- r Demande à l'utilisateur de confirmer l'effacement du fichier.

mkdir [-options] répertoires

Crée un ou plusieurs répertoires avec le mode d'accès donné par l' *umask* de l'utilisateur. (Voir *chmod*)

Option Commentaire

- m Définit le mode d'accès des nouveaux répertoires (à la place de *umask*).
- p Crée si nécessaire les répertoires parents appropriés.
- v Écrit un message pour chaque dossier créé.

Exemple :

```
$ mkdir -m 750 mon_répertoire
$ mkdir -p ./niveau1/niveau2/nouveau_rep
```

cat [-options] [fichiers]

Lit le contenu d'un ou plusieurs fichiers et les affiche sur la sortie standard.

Option Commentaire

- b Numérote toutes les lignes non vides en commençant par 1.
- e Affiche le caractère \$ en fin de ligne.
- s Exclut l'affichage des lignes vierges.
- t Affiche les tabulations avec ^I et les sauts de page avec ^L.
- v Affiche les caractères de contrôle à l'exception des sauts de lignes et tabulations.

Exemples :

```
$ cat -v mon_fichier
$ cat *.conf
```

more [-options] [fichiers]

Affiche le contenu de plusieurs fichiers une page à la fois. Il vaut mieux utiliser **less**, plus performant.

less [-options] [fichiers]

Affiche page par page le contenu des fichiers donnés en arguments, ou celui de la sortie standard. Remplace **more** (d'où son nom). Utilisé souvent sans options, mais nombre d'entre elles sont intéressantes. (man less)

echo [-options] [message]

Affiche une ligne de texte. Cette commande est souvent utilisée pour afficher le contenu d'une variable.

Option	Commentaire
-e	Interprète les séquences de caractères précédés d'un backslash \ suivantes
-e \b	Retour arrière.
-e \f	Changement de page (form feed).
-e \r	Nouvelle ligne (newline).
-e \n	Retour chariot (return).
-e \t	Tabulateur.
-n	Ne pas effectuer le saut de ligne final.

Système

df [-options] [nom]

Affiche l'espace libre sur chacun des systèmes de fichiers montés ou sur celui précisé par nom.

Option Commentaire

- k Affiche la taille en blocs de 1 k plutôt qu'en blocs de 512 octets.
- t *type* N'affiche que les systèmes de fichiers de type *type* .
- x *type* N'affiche que les systèmes de fichiers de type autre que *type* .
- h Affiche les tailles dans un format lisible par un humain (i.e. 1K 234M 2G).

Exemples :

```
$ df -k /home
$ df -x nfs
```

du [-options] [répertoire]

Affiche la taille du répertoire en cours ou du répertoire précisé.

Option Commentaire

- h Affiche les tailles dans un format lisible par un humain (i.e. 1K 234M 2G).
- s Affiche le total pour les répertoires donnés en arguments.

Exemples :

```
$ du -h /home
$ du -hs *
```

chmod [-options] mode fichiers

Permet de modifier les droits d'accès (mode) d'un ou plusieurs fichiers. Seul le propriétaire du fichier ou le super utilisateur peut effectuer un changement de mode.

Option Commentaire

- R Effectue les changements de manière récursive.

Utilisateur Commentaire

- u Propriétaire du fichier (user).
- g Groupe du propriétaire (group).
- o Tous les autres (other).
- a Tout le monde (all).

Opérateur Commentaire

- + Ajoute des permissions.
- Retire des permissions.
- = Redéfinition des permissions (Avec suppression de celles qui ne sont pas précisées).

Permissions : Il s'agit des permissions (r,w,x,s,t) vues dans un chapitre précédent. Je voudrais préciser ici qu'on peut utiliser le mode octal pour préciser les permissions :

4 pour la lecture.

2 pour l'écriture.

1 pour l'exécution.

Pour un fichier dont les droits sont -rwxrw-r-- soit :

- ▶ Le premier triplet concerne les droits du propriétaire du fichier : rwx
- ▶ Le deuxième triplet concerne les droits du groupe du propriétaire : rw-
- ▶ Le troisième triplet concerne les autres utilisateurs : r--

Pour donner tous les droit à tout le monde, on fait :

```
$ chmod 777 nom_du_fichier
```

ou bien

```
$ chmod a+rwx nom_du_fichier
```

puis pour avoir les droits donnés au début de l'exemple, on fait :

```
$ chmod g-x,o-wx nom_du_fichier
```

ou

```
$ chmod 764 nom_du_fichier
```

chgrp [-options] group fichiers

Permet de modifier le groupe d'un ou plusieurs fichiers. Seul le propriétaire ou le super utilisateur peut effectuer un changement de groupe.

Option Commentaire

-R Effectue les changements de manière récursive.

Exemples :

```
$ chgrp -R users /home
$ chgrp bin script.pl
```

chown [-options] user [: group] fichiers

Permet de modifier le propriétaire et le groupe d'un ou plusieurs fichiers. Seul le propriétaire du fichier ou le super utilisateur peut effectuer une telle opération.

Option Commentaire

-R Effectue les changements de manière récursive.

Exemples :

```
$ chown -R thierry:users /home/thierry
$ chown thierry:cdwriter fichier1 fichier2
```

Nota : on peut simplifier la commande dans le cas où le groupe est le groupe natif de l'utilisateur, c'est à dire le groupe créé à l'installation avec l'utilisateur. Donc pour simplifier on écrit :

```
$ chown thierry: fichier
```

on ne précise pas de nom après " :".

tail [-options] [fichier]

Affiche sur la sortie standard les dernières lignes de « fichier » (10 par défaut).

Option Commentaire

- f** Suit le fichier lorsqu'il croit (utile pour suivre l'évolution d'un fichier). Taper *CTRL-C* pour y mettre fin.
- n *num*** Affiche les *num* dernières lignes.

exemples :

```
$ tail -f /var/log/messages
$ tail -n 20 mon_fichier
```

ln [-options] nom_source [nom_dest]

Crée des liens permettant d'accéder à des fichiers sous différents noms.

Option Commentaire

- b** Sauvegarde les fichiers originaux avant de les écraser par de nouveaux liens.
- f** Force la création du lien sans demander de confirmation.
- s** Crée un lien symbolique. La commande *ls -l* permet de retrouver le fichier source.

Exemples :

```
# ln -sf /mnt/cdrom /cdrom
# ln -s /usr/local/bin/ma_commande
```

tar [-options] [fichier_tar] [autre_fichiers]

Permet de copier des fichiers vers une archive ou d'extraire des fichiers d'une archive. Si *autres_fichiers* est un répertoire, *tar* archive l'ensemble du répertoire et des sous répertoires. Les options sont très nombreuses, voyons les principales.

Options de fonction :

On ne peut utiliser qu'une seule des options suivantes .

Option Commentaire

- c** **create**. Crée une nouvelle archive.
- t** **list**. Liste le contenu d'une archive.
- u** **update**. Met l'archive à jour en ajoutant de nouveaux fichiers ou met à jour des fichiers modifiés.
- x** **extract**. Extrait des fichiers d'une archive.
- r** Ajoute des fichiers à une archive.
- d** Compare les fichiers contenus dans l'archive à ceux présents sur le disque dur.

Options communes :

Option Commentaire

- f** Sauvegarde les fichiers dans *archive* ou les extrait de *archive* . Archive peut être de **archive** la forme *host :file* .
- h** Sauvegarde le fichier auquel un lien fait référence et non le lien.

- l** N'archive pas les fichiers situés sur d'autres systèmes de fichiers.
- p** Préserve les attributs originaux des fichiers sauvegardés.
- v** Mode verbeux.
- Z** Utilise **compress** pour compacter les fichiers.
- z** Utilise **gzip** pour compacter les fichiers.
- j** Utilise **bzip2** pour compacter les fichiers.
- k** Conserve tout fichier existant avant l'extraction, aucun fichier ne sera écrasé par un autre portant le même nom dans l'archive.

exemples :

```
$ tar cvfz /dev/rmt0 /home /usr/local
$ tar xvf MonFichier.tar
$ tar cvf - $(find . -print) > copie.tar
$ gzip -cd MonFichier.tgz | tar xvf -
```

diff [-options] [fichier1] [fichier2]

Compare deux fichiers et affiche les différences. Chaque paire de lignes est précédée de la commande *ed* qui permettrait de rendre fichier 1 identique à fichier 2.

Option Commentaire

- a** Tous les fichiers seront considérés comme du texte. Utile pour comparer 2 fichiers binaires.
- c** Comparaison contextuelle.

exemples :

```
$ diff -c /etc/bashrc /home/.bashrc
```

gzip [-options] [fichier], gunzip [-options] [fichier]

Comprime ou déprime les fichiers donnés en arguments selon l'algorithme LZ77. Les fichiers sont renommés *nom_fichier.gz*. On déprime les fichiers avec la commande *gunzip*.

Option Commentaire

- num** Contrôle le taux de compression de 1 à 9 (9 est le taux de compression le plus haut mais aussi le plus lent).
- c** Affiche sur la sortie standard mais ne modifie pas le fichier.
- d** Déprime le fichier (Identique à *gunzip*).
- r** Récursif. Lorsqu'un répertoire est donné en argument, compresse tous les fichiers qui s'y trouvent de façon récursive.

exemples :

```
$ gzip -r /home/thierry
```

```
$ gzip -d fichier.gz
```

bzip2 [-options] [fichier], bunzip2 [-options] [fichier]

Identique à gzip mais l'algorithme permet une meilleure compression. Le suffixe de fichier standard est **.bz2**

su [-options] [utilisateur]

Permet de changer d'utilisateur, pour effectuer des tâches qui nécessitent des droits différents de l'utilisateur actuel. Lorsque *utilisateur* n'est pas précisé, c'est *root* qui est utilisé.

Option Commentaire

- L'environnement *utilisateur* est alors chargé.
- c *commande*** Lance *commande* dans un nouveau shell et le quitte immédiatement.

exemples :

```
$ su -
$ su oracle -c dbshut
```

ps [options]

Affiche la liste des processus en cours d'exécution. Il n'est pas nécessaire d'introduire les options par un tiret.

Option Commentaire

- a** Affiche tous les processus.
- e** Affiche aussi les processus système.
- f** Affiche la sortie sous forme d'arborescence.
- h** Supprime l'affichage de l'en-tête.
- j** Affiche en format *tâches* .
- l** Affiche en format *étendu* .
- m** Affiche en format *mémoire* .
- u** Inclut le nom d'utilisateur et l'heure de lancement du processus.
- w** Affiche chaque entrée sur plusieurs lignes si elle est plus large que l'écran.
- x** Affiche les processus non rattachés à un terminal.
- C *programme*** Permet de connaître le PID d'un programme connu

Exemples :

```
$ ps faux
$ ps fjx
$ ps eux | sort -n -k4
$ ps -C firefox
```

kill [-options] PID

Envoie un signal à un ou plusieurs processus identifiés par leur numéro (PID).

Option Commentaire

-l Affiche la liste des signaux.

-s *signal* Signal à envoyer au processus. Il doit être précisé par son nom ou son numéro

-3/-6/-9 Lors de la tentative pour *tuer* le processus, vous pouvez lui spécifier une force (3 est la plus faible et 9 la plus puissante).

Exemples :

```
$ kill -SIGHUP xinetd
$ kill -9 id_processus_fou
```

killall [-options] [nom]

Met fin à un processus identifié par son nom.

Option Commentaire

-signal Envoie *signal* au processus (Défaut TERM). signal peut être un nom ou un numéro.

-i *signal* Demande à l'utilisateur de confirmer l'arrêt du processus.

-l Affiche la liste des signaux.

Exemples :

```
$ kill -i X
```

```
$ killall -9 processus_fou
```

Historique des modifications

Version	Date	Commentaire
0.1	2002	Thierry COLAS , auteur original du document « Lignes de commandes - le pense bête de COAGUL »
0.2	2004	Marc POIROUD pour la conversion en LATEX , PDF et HTML. Gnunix , pour sa relecture
0.3	24/03/05	Tony pour la conversion au format OOo / Spip et le découpage en 7 articles.