

LES COMMANDES DE BASE

SOUS LINUX

Notation

Une ligne de commande commençant par un `$` indique que la commande peut être lancée en tant qu'utilisateur normal. En revanche, une commande précédée d'un `#` indique que celle-ci ne peut-être lancée que par le superutilisateur (root).

Commandes indispensables

<code>ls</code>	Affiche les caractéristiques des fichiers
<code>cp</code>	Copie des fichiers ou des arbres
<code>rm</code>	Détruit des fichiers ou des arbres
<code>mv</code>	Renomme ou déplace des fichiers
<code>mkdir, rmdir</code>	Crée ou supprime des répertoires
<code>grep</code>	Recherches des chaînes dans un fichier

1 Commandes de manipulation de fichiers

1.1 Répertoire en cours

Commande : `pwd`

```
$ pwd
/home/tux
```

1.2 Copier un fichier

Commande : `cp`

```
$ cp /etc/group .
```

Pour demander la confirmation avant la copie :

```
$ ls -l group
-rw-r--r--1 tux tux 699 jul 31 12:17 group
$ cp -i /etc/group .
cp: écraser './group'? 0
$ cp group group.old
```

Il peut arriver que la copie de certains fichiers échoue :

```
$ cp /etc/g* .
cp: omission du répertoire '/etc/gconf'
cp: Ne peut ouvrir '/etc/group-' en lecture: Permission non accordée
```

En revanche, les autres fichiers ont bien été copiés :

```
$ ls g*
gnome-vfs-mime-magic gpm-root.conf group group.old
```

1.3 Protéger un fichier contre les modifications

Commande : `chmod`

```
$ chmod 444 group
$ cp /etc/group .
cp: ne peut créer le fichier régulier './group': permission non accordée
```

Pour forcer la copie (par exemple), il faut utiliser l'option de forçage `-f` :

```
$ cp -f /etc/group .
```

1.4 Supprimer un fichier

Commande : `rm`

```
rm group
```

Pour demander la confirmation avant la suppression, il faut utiliser l'option `-i` :

```
$ rm -i group.old
rm: détruire fichier régulier 'group.old'? 0
$ cp /etc/group group.new
```

1.5 Renommer un fichier

Commande : `mv`

```
$ mv group.new group.old
```

La commande est exécutée même si le nouveau nom existe déjà :

```
$ cp /etc/group group.new
$ mv group.new group.old
```

Si l'on veut obtenir une confirmation :

```
$ mv -i group.new group.old
mv: écraser 'group.old'? 0
```

2 Commandes de gestion de répertoire

2.1 Créer un répertoire

Commande : `mkdir`

```
$ mkdir repertoire
```

2.2 Supprimer un répertoire

Commande : `rmdir`

```
$ rmdir repertoire
```

Si le répertoire n'est pas vide, il est impossible de le supprimer :

```
$ rmdir repertoire
rmdir: 'repertoire': Le répertoire n'est pas vide.
```

2.3 Copier un arbre

```
$ ls -R /tmp/r1
/tmp/r1:
f1 f2 r11 r12
/tmp/r1/r11:
f11
/tmp/r1/f12:
f12
$ cp -r /tmp/r1 .
$ ls -R r1
r1:
f1 f2 r11 r12
r1/r11:
f11
r1/f12:
f12
```

2.4 Supprimer un arbre

```
$ rm -rf r1
$ ls r1
ls: r1: Aucun fichier ou répertoire de ce type
```

2.5 Connaître la taille d'un arbre et chacun de ses sous-répertoire

Commande : `du`

```
$ du .
4,0K  ../kde/Autostart
16K   ../kde/
196K  ./rep
512K  .
```

Pour connaître le total :

```
$ du -s .
5,2G  .
```

Il peut arriver que l'unité ne soit pas compréhensible, pour que celle-ci le soit, utiliser l'option `-h`.

3 Autres commandes utiles

3.1 Afficher la fin d'un fichier

Pour les 5 dernières lignes :

```
# tail -5 /var/log/messages
Jul 31 12:25:52 alban1 kernel: 3rdparty/at76c503a/at76c503.c: wlan0:
DeAuth in state 6 ignored
Jul 31 12:29:04 alban1 nmbd[2494]: [2005/07/31 12:29:04, 0] nmbd/nmb
d_incomingrequests.c:process_name_release_request(118)
Jul 31 12:29:04 alban1 nmbd[2494]: process_name_release_request:
Attempt to release name MSHOME<1d> from IP 192.168.0.4 on subnet
192.168.0.5 being rejected as it is one of our names.
Jul 31 13:01:00 alban1 CROND[5062]: (root) CMD (nice -n 19 run-parts
/etc/cron.hourly)
Jul 31 13:53:01 alban1 su(pam_unix)[5405]: session opened for user root
by (uid=501)
```

3.2 Comparer deux fichiers

```
# cp /etc/group /etc/group.old
# cmp /etc/group /etc/group.old
# groupadd nouveaugroupe
# cmp /etc/group /etc/group.old
cmp: Fin-de-fichier (EOF) sur /etc/group/old
```

3.3 Afficher les lignes qui diffèrent entre deux fichiers

```
# diff /etc/group /etc/group.old
58d57
< nouveaugroupe:x:1004:
```

4 Trucs et astuces

4.1 Changer la date de dernière modification

```
$ ls -l group
-rw-r--r--1 alban alban 699 jul 31 13:10 group
$ touch group
$ ls -l group
-rw-r--r--1 alban alban 699 jul 31 14:04 group
```

4.2 Créer un fichier vide avec touch

```
$ ls -l fichier
ls: fichier: Aucun fichier ou répertoire de ce type
$ touch fichier
$ ls -l fichier
-rw-r--r--1 alban alban 0 jul 31 14:06 fichier
```

4.3 Créer un fichier vide avec une redirection en sortie

```
$ > fichier2
$ ls -l fichier2
-rw-r--r--1 alban alban 0 jul 31 14:07 fichier2
```

4.4 Vider un fichier

```
$ ls -l group
-rw-r--r--1 alban alban 699 jul 31 14:04 group
$ > group
$ ls -l group
-rw-r--r--1 alban alban 0 jul 31 14:04 group
```

5 La commande grep

La commande **grep** affiche les lignes d'un fichier qui contiennent une chaîne de caractères donnés en argument. L'utilisation de tubes permet de l'appliquer aussi à la sortie d'une commande.

La chaîne argument peut-être une expression régulière. Cette commande se révèle indispensable sachant que la quasi totalité des fichiers de configuration sont de type texte.

5.1 Syntaxe

grep [-ivnc] chaîne [fichier...]

- L'option **-i** permet d'ignorer les différences entre lettres minuscules et lettres majuscules (casse).
- L'option **-v** permet d'afficher les lignes qui ne contiennent pas la chaîne.
- L'option **-n** permet de numéroté les lignes.
- L'option **-c** permet de compter les lignes.

On utilisera le fichier `/etc/passwd` en tant qu'exemple :

```
# cat passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/bin/sh
daemon:x:2:2:daemon:/sbin:/bin/sh
adm:x:3:4:adm:/var/adm:/bin/sh
lp:x:4:7:lp:/var/spool/lpd:/bin/sh
nobody:x:65534:65534:Nobody:/:/bin/sh
alban:x:501:501:Alban:/home/alban:/bin/bash
```

5.2 Lignes contenant un mot

```
# grep bash passwd
root:x:0:0:root:/root:/bin/bash
alban:x:501:501:Alban:/home/alban:/bin/bash
```

5.3 Lignes commençant par un mot

```
# grep ^bin passwd
bin:x:1:1:bin:/bin:/bin/sh
```

5.4 Lignes se terminant par un mot

```
# grep bash$ passwd
root:x:0:0:root:/root:/bin/bash
alban:x:501:501:Alban:/home/alban:/bin/bash
```

5.5 Vérifier qu'un utilisateur est connecté

La commande `who` liste tous les utilisateurs connectés :

```
$ who
root      vc/1          Jul 31 14:21
alban     vc/2          Jul 31 14:21
```

On voit tout l'intérêt de l'utilisation des tubes lorsqu'un nombre important d'utilisateurs sont connectés.

```
$ who | grep root
root      vc/1          Jul 31 14:21
```

6 Archivage

Sous Linux (et Unix en général), l'archivage se fait au moyen de la commande `tar`.

6.1 Création d'une archive

La création d'une archive `projet.tgz` à partir des fichiers se trouvant sous l'arborescence d'un repertoire `projet` se fait en executant la ligne de commande suivante :

```
$ tar czf archive.tgz repertoire
$ cd repertoire
$ ls
archive.tar  file1  file2  sous_repertoire
```

`czf` est une liste d'options de la commande :

- `c` signifie (*create*) que l'on va créer une nouvelle archive ;
- `z` signifie que l'on veut que cette archive soit compressée avec `gzip` ;
- `f` signifie que le paramètre suivant est le nom de l'archive.

6.2 Lister le contenu d'une archive

Pour lister le contenu d'une archive `projet.tgz` il suffit de taper la ligne suivante :

```
$ tar tzf archive.tgz
```

L'option `t` indique à `tar` que l'on souhaite lister l'archive passée en paramètre à l'option `f`. Avec l'option `z` on signale que l'archive est compressée avec `gzip`.

6.3 Extraire le contenu d'une archive

L'option d'extraction du contenu d'une archive est passée avec `x` (pour *eXtract*). Ainsi la ligne de commande suivante permet d'extraire le contenu d'une archive `projet.tgz` dans le répertoire courant :

```
$ tar xzf archive.tgz
```

Attention Les fichiers de l'archive possédant même nom que les fichiers contenus dans le répertoire courant seront écrasés. Il est donc indispensable de vérifier quels fichiers vont être extraits de l'archive avant d'exécuter `tar tzf archive.tgz`.

Table des matières

1	Commandes de manipulation de fichiers	1
1.1	Répertoire en cours	1
1.2	Copier un fichier	1
1.3	Protéger un fichier contre les modifications	2
1.4	Supprimer un fichier	2
1.5	Renommer un fichier	2
2	Commandes de gestion de répertoire	2
2.1	Créer un répertoire	2
2.2	Supprimer un répertoire	3
2.3	Copier un arbre	3
2.4	Supprimer un arbre	3
2.5	Connaître la taille d'un arbre et chacun de ses sous-répertoire	3
3	Autres commandes utiles	4
3.1	Afficher la fin d'un fichier	4
3.2	Comparer deux fichiers	4
3.3	Afficher les lignes qui diffèrent entre deux fichiers	4
4	Trucs et astuces	4
4.1	Changer la date de dernière modification	4
4.2	Créer un fichier vide avec <code>touch</code>	5
4.3	Créer un fichier vide avec une redirection en sortie	5
4.4	Vider un fichier	5
5	La commande <code>grep</code>	5
5.1	Syntaxe	5
5.2	Lignes contenant un mot	6
5.3	Lignes commençant par un mot	6
5.4	Lignes se terminant par un mot	6
5.5	Vérifier qu'un utilisateur est connecté	6
6	Archivage	6
6.1	Création d'une archive	6
6.2	Lister le contenu d'une archive	7
6.3	Extraire le contenu d'une archive	7