



# Introduction aux systèmes d'exploitation

---

D.TOUAZI FAYCAL

MAITRE DE CONFÉRENCES

UNIVERSITÉ M'HAMED BOUGARA – BOUMERDES

[f.touazi@univ-boumerdes.dz](mailto:f.touazi@univ-boumerdes.dz)



# Les commandes de traitement des données

---

f.touazi@univ-boumerdes.dz

**Rappelons que** C'est vrai il y a beaucoup de programmes graphique **qui sont facile à utiliser**.

Par contre les commandes suivantes sont généralement utilisées par **les administrateurs système** pour :

- **Leur efficacité**
- **Leur rapidité**
- **Leur puissance**

# Les commandes de traitement des données (extraire, trier et filtrer)

## La commande **head**

La commande **head** affiche les premières lignes d'un fichier

**\$ head            [options]        fichiers**

**Exemple d'utilisation :**

Si un administrateur système veut **lire juste 10 lignes** du fichier **/etc/default/grub**

```
lcpig1b1@ubuntu-virtual-machine:~$ head -n 10 /etc/default/grub
# If you change this file, run 'update-grub' afterwards to update
# /boot/grub/grub.cfg.
# For full documentation of the options in this file, see:
#   info -f grub -n 'Simple configuration'

GRUB_DEFAULT=0
GRUB_TIMEOUT_STYLE=hidden
GRUB_TIMEOUT=10
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash"
lcpig1b1@ubuntu-virtual-machine:~$
```



**Est-ce que il y a une  
commande qui affiche les  
dernières lignes d'un fichier?**

# Les commandes de traitement des données (extraire, trier et filtrer)

## La commande **tail**



La commande **tail** affiche les dernières lignes d'un fichier

**\$ tail [options] fichiers**

**Exemple d'utilisation :**

Si un administrateur système veut **lire juste les 05 dernières lignes** du fichier **/var/log/syslog**.

```
root@ubuntu-virtual-machine:~# tail -n 5 /var/log/syslog
```

```
Oct 31 23:44:06 ubuntu-virtual-machine NetworkManager[772]: <warn> [15
41025846.6163] error requesting auth for org.freedesktop.NetworkManager
.settings.modify.global-dns: Authorization check failed: The maximum nu
mber of pending replies per connection has been reached
Oct 31 23:44:06 ubuntu-virtual-machine NetworkManager[772]: <warn> [15
41025846.6164] error requesting auth for org.freedesktop.NetworkManager
.reload: Authorization check failed: The maximum number of pending repl
ies per connection has been reached
Oct 31 23:44:06 ubuntu-virtual-machine NetworkManager[772]: <warn> [15
41025846.6164] error requesting auth for org.freedesktop.NetworkManager
.checkpoint-rollback: Authorization check failed: The maximum number of
pending replies per connection has been reached
Oct 31 23:44:06 ubuntu-virtual-machine NetworkManager[772]: <warn> [15
41025846.6165] error requesting auth for org.freedesktop.NetworkManager
.enable-disable-statistics: Authorization check failed: The maximum num
ber of pending replies per connection has been reached
Oct 31 23:44:06 ubuntu-virtual-machine NetworkManager[772]: <warn> [15
41025846.6166] error requesting auth for org.freedesktop.NetworkManager
.enable-disable-connectivity-check: Authorization check failed: The max
imum number of pending replies per connection has been reached
```

**Est-ce que il y a d'autre  
commandes qui affichent le  
contenu de fichier?**

**Les commandes `cat`,  
`more`, `less`, `tac`,  
`hexdump`, `pr`, `nl`, ...**

Ces commandes permettent la visualisation du contenu de fichiers.

- **cat** : affiche tout le contenu du fichier.
- **nl** : c'est la même chose que cat -n affiche le contenu du fichier avec numérotation des lignes
- **more** : affiche le contenu du fichier page par page sans revenir en arrière
- **less** : affiche le contenu du fichier page par page dans lequel on peut revenir en arrière

- **hexdump** : affiche le contenu du fichier en hexadécimal
- **pr** : affiche le contenu du fichier format pour l'impression
- **tac** : affiche le contenu du fichier à l'envers

**Est-ce que il y a une  
commande pour appliquer des  
filtres sur les lignes des  
fichiers?**



# La commande **grep**

La commande **grep** est utilisée pour chercher les lignes des fichiers qui contient une expression.

**\$ grep [options] expression fichiers**

### Quelques options intéressantes :

- **-v** effectue la recherche inverse
- **-c** ne retourne que le nombre de lignes trouvées
- **-i** pour ne pas faire la différence entre majuscule et minuscule
- **-n** pour indiquer les numéros de lignes
- **-E** pour l'utilisation des expressions régulières

## Exemple d'utilisation 01 :

Si un administrateur système veut chercher dans le fichier **/etc/group** l'expression **sudo** en affichant **numéro de la ligne**.

```
lcpig1b1@ubuntu-virtual-machine:~$ grep -n sudo /etc/group
21: sudo:x:27:guerrou, sehad, khelouat
lcpig1b1@ubuntu-virtual-machine:~$
```

**21** : représente le numéro de la ligne

## Exemple d'utilisation 02 :

Si un administrateur système veut chercher dans le fichier **.bashrc** tous les lignes qui commence avec **alias** en affichant numéro de la ligne.

```
lcpig1b1@ubuntu-virtual-machine:~$ grep -En ^alias .bashrc
91:alias ll='ls -a1F'
92:alias la='ls -A'
93:alias l='ls -CF'
97:alias alert='notify-send --urgency=low -i "$([ $? = 0 ] && echo term
inal || echo error)" "$(history|tail -n1|sed -e '\''s/^\s*[0-9]\+\s*//;
s/[;&|]\s*alert$//'\''")"'
lcpig1b1@ubuntu-virtual-machine:~$
```

représentent les numéros des lignes

**Est-ce que il y a une  
commande pour appliquer des  
filtres sur les colonnes des  
fichiers?**

# La commande **Cut**

La commande **cut** est utilisée pour sélectionner des **colonnes** et des **champs** dans un fichier.

- **\$ cut -c Colonnes [fichiers]**

Une colonne est la position d'un caractère dans la ligne. Le premier caractère est la colonne 1, le deuxième la colonne 2, etc.

- **\$ cut -d Caractère -fChamps [fichiers]**

Les champs sont délimités par défaut par une tabulation, le paramètre -d est utilisé pour spécifier une autre délimiteur



## Exemple d'utilisation 01 :

Si un administrateur système veut sélectionner les colonnes (1-4,11,13-) du fichier **fichier**

```
lcpiglb1@ubuntu-virtual-machine:~$ cat fichier
drwxr-xr-x 5 lcpiglb1 g1      4096 oct.  21 14:31 Bureau
-rwxr-xr-x 1 lcpiglb1 g1    6156288 oct.   7 22:21 core
drwxr-xr-x 2 lcpiglb1 g1      4096 oct.   4 12:10 Documents
-rwxr-xr-x 1 lcpiglb1 g1      8980 oct.   1 15:56 exemples.desktop
-rw-rw---- 1 lcpiglb1 g1         0 oct.  24 09:07 Fich
drwxr-xr-x 2 lcpiglb1 g1      4096 oct.   2 11:51 Vidéos
lcpiglb1@ubuntu-virtual-machine:~$ cut -c1-4,11,13- fichier
drwx  lcpiglb1 g1      4096 oct.  21 14:31 Bureau
-rwx  lcpiglb1 g1    6156288 oct.   7 22:21 core
drwx  lcpiglb1 g1      4096 oct.   4 12:10 Documents
-rwx  lcpiglb1 g1      8980 oct.   1 15:56 exemples.desktop
-rw-  lcpiglb1 g1         0 oct.  24 09:07 Fich
drwx  lcpiglb1 g1      4096 oct.   2 11:51 Vidéos
lcpiglb1@ubuntu-virtual-machine:~$
```

## Exemple d'utilisation 02 :

Si un administrateur système veut sélectionner les champs (1,3,4) du fichier **fichier**

```
lcpiglb1@ubuntu-virtual-machine:~$ cat fichier
drwxr-xr-x 5 lcpiglb1 g1      4096 oct. 21 14:31 Bureau
-rwxr-xr-x 1 lcpiglb1 g1    6156288 oct. 7 22:21 core
drwxr-xr-x 2 lcpiglb1 g1      4096 oct. 4 12:10 Documents
-rwxr-xr-x 1 lcpiglb1 g1      8980 oct. 1 15:56 exemples.desktop
-rw-rw---- 1 lcpiglb1 g1         0 oct. 24 09:07 Fich
drwxr-xr-x 2 lcpiglb1 g1      4096 oct. 2 11:51 Vidéos
lcpiglb1@ubuntu-virtual-machine:~$ cut -d" " -f1,3,4 fichier
drwxr-xr-x lcpiglb1 g1
-rwxr-xr-x lcpiglb1 g1
drwxr-xr-x lcpiglb1 g1
-rwxr-xr-x lcpiglb1 g1
-rw-rw---- lcpiglb1 g1
drwxr-xr-x lcpiglb1 g1
lcpiglb1@ubuntu-virtual-machine:~$
```

**Est-ce que il y a une  
commande pour effectue le  
tri ?**

# La commande **Sort**

La commande **sort** permet de trier les lignes d'un fichier, par défaut il est en ordre croissant.

**\$ sort      [options]      [fichiers]**

**Quelques options intéressantes :**

- **-k** pour spécifier le numéro de colonne utilisé pour le tri
- **-o** pour écrire le résultat dans un fichier
- **-r** trier en ordre inverse (tri en ordre décroissant)
- **-n** tri numérique, idéal pour les colonnes de chiffres

# Les droits d'accès

## Exemple d'utilisation :

```
user1@PC:~$ cat fichier
-rw-rw-r-- 1 user1 user1 76473 11:33 présentation.pptx
-rw-rw-r-- 1 user1 user1 75075316 21:29 linux64-8.0.3_en-US.tar.xz
-rw-rw-r-- 1 user1 user1 105631957 14:24 linux-en.tar.gz
-rw-rw-r-- 1 user1 user1 130956158 11:17 112.mp4
user1@PC:~$ sort -n -r -k 5 fichier
-rw-rw-r-- 1 user1 user1 130956158 11:17 112.mp4
-rw-rw-r-- 1 user1 user1 105631957 14:24 linux-en.tar.gz
-rw-rw-r-- 1 user1 user1 75075316 21:29 linux64-8.0.3_en-US.tar.xz
-rw-rw-r-- 1 user1 user1 76473 11:33 présentation.pptx
user1@PC:~$
```

Colonne numéro 05

**Est-ce que il y a une  
commande pour supprimer  
les doublons?**



# La commande **uniq**

La commande **uniq** est permet de supprimer les doublons des **fichiers triés** c'est-à-dire elle ne repère que les lignes successives qui sont identiques

● **\$ uniq [options] [source [destination]]**

Exemple :

```
user1@PC:~$ cat fichier
mohammed
ali
ali
ali
bob
bob
user1@PC:~$ uniq fichier
mohammed
ali
bob
user1@PC:~$ □
```

**Est-ce que il y a une  
commande pour remplacer  
une expression par une autre?**

# La commande **sed**

La commande **sed** est utilisée pour remplacer une **expression** par une **autre**.

```
$ sed -i s/ancien/nouveau/[g] [fichiers]
```

### Exemple d'utilisation :

Changer le nom **ubuntu-virtual-serveur** par **srv** qui se trouve dans **/etc/hostname**

```
root@ubuntu-virtual-machine:~# cat /etc/hostname
```

```
ubuntu-virtual-machine
```

```
root@ubuntu-virtual-machine:~# sed -i s/ubuntu-virtual-machine/srv/g /etc/hostname
```

```
root@ubuntu-virtual-machine:~# cat /etc/hostname
```

```
srv
```

```
root@ubuntu-virtual-machine:~# █
```

**Est-ce que il y a une  
commande pour compter les  
mots ?**



# La commande WC

La commande **wc (word count)** est utilisée pour compter les **lignes**, les **mots**, les **octets** et les **caractères**.

**\$ wc [Options] [fichiers]**

**Quelques options intéressantes :**

- **-l** compte le nombre de lignes
- **-c** compte le nombre d'octets
- **-w** compte le nombre de mots
- **-m** compte le nombre de caractères

## Exemple d'utilisation :

Administrateur système veut faire une comparaison entre deux fichiers **/etc/passwd** et **/etc/shadow**

```
root@PC:~# wc /etc/passwd /etc/shadow
```

45	73	2597	/etc/passwd
45	45	1680	/etc/shadow
90	118	4277	total

Diagram illustrating the output of the `wc` command comparing `/etc/passwd` and `/etc/shadow`:

- lignes** (lines): 45 for `/etc/passwd`, 45 for `/etc/shadow`, 90 total.
- mots** (words): 73 for `/etc/passwd`, 45 for `/etc/shadow`, 118 total.
- caractères** (characters): 2597 for `/etc/passwd`, 1680 for `/etc/shadow`, 4277 total.

**Est-ce que il y a une  
commande pour traitement  
des fichiers ?**

**Les commandes paste,  
split, ...**

- **Jointure Ligne à ligne**

**\$ paste [options] [fichiers]**

elle concatène les lignes fichiers entre eux, ligne1 de fichier1 avec ligne1 de fichier2, etc.

l'**option -d** est utilisé pour spécifier le **délimiteur**

- **Découpage d'un fichier en morceaux**

**\$ split [options] [fichier [préfixe]]**

Elle permet de découper un gros fichier en plusieurs morceaux.

## Exemple d'utilisation 01 :

Administrateur système veut faire une jointure **ligne à ligne** des fichiers **fichier1**, **fichier2** et **fichier3**

```
user1@PC:~$ cat fichier1
-rwxr-sr-x
-rw-rw-r--
-rwxr-sr-x
user1@PC:~$ cat fichier2
user1 user1 0
user1 user1 158
user1 user1 8560
user1@PC:~$ cat fichier3
19:55 fichier2
19:58 fichier3
10:50 fichier.txt
user1@PC:~$ paste -d" " fichier1 fichier2 fichier3
-rwxr-sr-x user1 user1 0 19:55 fichier2
-rw-rw-r-- user1 user1 158 19:58 fichier3
-rwxr-sr-x user1 user1 8560 10:50 fichier.txt
user1@PC:~$
```

## Exemple d'utilisation 02 :

Administrateur système veut découper un gros fichier **fichier** en **10** morceaux chacun dans un fichier

```
user1@PC:~$ split -n 10 fichier fich
```

```
user1@PC:~$ ls -l fich*
```

```
-rw-rw-r-- 1 user1 user1 364275212  
-rw-rw-r-- 1 user1 user1 364275212  
-rw-rw-r-- 1 user1 user1 364275212  
-rw-rw-r-- 1 user1 user1 364275212  
-rw-rw-r-- 1 user1 user1 364275212  
-rw-rw-r-- 1 user1 user1 364275212  
-rw-rw-r-- 1 user1 user1 364275212  
-rw-rw-r-- 1 user1 user1 364275212  
-rw-rw-r-- 1 user1 user1 364275212  
-rw-rw-r-- 1 user1 user1 364275218  
-rw-rw-r-- 1 user1 user1 3642752126
```

```
user1@PC:~$
```

```
16:44 fichaa  
16:44 fichab  
16:44 fichac  
16:45 fichad  
16:45 fichae  
16:45 fichaf  
16:45 fichag  
16:45 fichah  
16:46 fichai  
16:46 fichaj  
16:38 fichier
```



---

**Autres commandes**  
**join, diff, cmp**

- **Jointure sur des champs communs**

**\$ join [options] fichier1 fichier2**

Elle effectue une jointure de deux fichiers en fonction d'un champ commun.

- **Comparaison de fichiers**

**\$ diff [options] fichier1 fichier2**

**\$ cmp [options] fichier1 fichier2**

Elles permettant de comparer le contenu de deux fichiers

**etc ●●●**