

Ligne de commande sous Linux (Suite)

420-1S6 Systèmes d'exploitation

Plan de la séance :

- La manipulation de fichiers/répertoires
- Les commandes utilisées pour les informations système
- SUDO
- Chemins absous & relatifs
- Les caractères génériques

Le terminal (navigation)



Après avoir saisi les premières lettres d'une commande, on peut utiliser l'**autocomplétion** à l'aide de la touche Tab qui permet d'ajouter la suite de la commande.

On peut retrouver les **commandes précédentes** à l'aide des flèches directionnelles Haut et Bas ou encore effectuer une **recherche** parmi les commandes précédentes avec Ctrl + R.

Il existe de nombreux autres raccourcis clavier qu'il est recommandé de connaître pour pouvoir profiter pleinement de la console.

Anatomie d'une commande (Rappel)



Commande

Paramètres

rm **-R** repertoire "autre repertoire"

Switch Paramètre

Paramètre qui comprend des espaces

Une ligne de commande commence par le nom de la commande

La commande peut être suivie d'un ou plusieurs paramètres, selon ce qu'on veut faire et ce que la commande accepte

Contrairement à Windows, l'utilisation d'un switch requiert un tiret (-), ou deux tirets (--), selon le cas, mais pas la barre oblique (/).

Chaque élément de la ligne est séparé par un espace

Ne pas oublier que Linux est un système d'exploitation sensible à la casse, donc qui fait une différence entre les majuscules et les minuscules.



La commande `mv` (move) permet de déplacer des fichiers/répertoires. Elle permet aussi de renommer les fichiers et répertoires en changeant simplement leur nom.

Déplacer un fichier

```
christian@ubuntu:~$ ls
Desktop Documents Downloads Music Pictures Public Templates Videos
christian@ubuntu:~$ touch Fichier1
christian@ubuntu:~$ ls
Desktop Downloads Music Public Videos
Documents Fichier1 Pictures Templates
christian@ubuntu:~$ mv Fichier1 Documents
christian@ubuntu:~$ ls
Desktop Documents Downloads Music Pictures Public Templates Videos
christian@ubuntu:~$ ls Documents
Fichier1
christian@ubuntu:~$
```

Renommer un fichier

```
christian@ubuntu:~$ ls
Desktop Documents Downloads Music Pictures Public Templates Videos
christian@ubuntu:~$ touch Fichier1
christian@ubuntu:~$ ls
Desktop Downloads Music Public Videos
Documents Fichier1 Pictures Templates
christian@ubuntu:~$ mv Fichier1 Fichier2
christian@ubuntu:~$ ls
Desktop Downloads Music Public Videos
Documents Fichier2 Pictures Templates
christian@ubuntu:~$
```



La commande **cp** (copy) permet de faire une copie d'un fichier. Sa fonction est presque similaire à la commande **mv**. Cependant, le fichier source n'est pas supprimé du répertoire.

```
christian@ubuntu:~$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos
christian@ubuntu:~$ touch Fichier1
christian@ubuntu:~$ cp Fichier1 Fichier2
christian@ubuntu:~$ ls
Desktop      Downloads  Fichier2  Pictures  Templates
Documents  Fichier1  Music      Public    Videos
christian@ubuntu:~$
```

grep



La commande **grep** est utilisée pour rechercher un texte dans le fichier / répertoire spécifié.

```
christian@ubuntu:~$ cat Fichier_Output
Ceci est le contenu de Fichier 1
Ligne de commande Linux (Semaine 4)
A B C
D E F
G H I
J K L
Ligne de commande Linux (Semaine 4)
christian@ubuntu:~$ grep Ligne Fichier_Output
Ligne de commande Linux (Semaine 4)
Ligne de commande Linux (Semaine 4)
christian@ubuntu:~$ █
```

La commande ci-dessus recherchera le mot Ligne dans le texte du fichier Fichier_Output

history



La commande **history** affiche la liste de toutes les commandes entrées avec les numéros de lignes depuis le démarrage de la session par l'utilisateur.

Pour effacer tout l'historique, utilisez l'option -c avec history

```
christian@ubuntu:~$ history
 1  ls
 2  df
 3  df -Th
 4  cd /
 5  ls -ltr
 6  cd
 7  ls -ltr
 8  df
 9  cd /
10  ls -ltr
11  cd mnt
12  ls -ltr
13  ps -p $$
14  clear
15  cd
```

pwd



La commande **pwd** (Print working directory) affiche le chemin du répertoire actuel.

```
christian@ubuntu:~$ pwd
/home/christian
christian@ubuntu:~$ █
```



La commande **clear** consiste à effacer l'écran de Terminal.
Le raccourci clavier : **ctrl + l**

Avant la commande clear

```
fonts          nanorc        ucf.conf
fprintd.conf  netplan        udev
fstab          network        udisks2
fuse.conf     networkd-dispatcher ufw
fwupd          NetworkManager update-manager
gai.conf       networks       update-motd.d
gamemode.ini  newt          update-notifier
gdb            nsswitch.conf UPower
gdm3          openvpn        usb_modeswitch.conf
geoclue        opt           usb_modeswitch.d
ghostscript    os-release    vim
glvnd          PackageKit   vmware-tools
gnome          pam.conf     vtrgb
groff          pam.d        vulkan
group          papersize    wgetrc
group-         passwd       wpa_supplicant
grub.d         passwd-      X11
gshadow        pcmcia       xattr.conf
gshadow-       perl         xdg
gss            pkts         xml
gtk-2.0        pm           zsh_command_not_found
gtk-3.0        pnm2ppa.conf
hdparm.conf    polkit-1
```

christian@ubuntu:~\$ clear

Après la commande clear

```
christian@ubuntu:~$
```

Le pipeline >



On peut aussi créer un nouveau fichier avec le contenu de deux fichiers avec l'opérande «>>».

```
christian@ubuntu:~$ cat Fichier1
Ceci est le contenu de Fichier 1
Ligne de commande Linux (Semaine 4)
christian@ubuntu:~$ cat Fichier2
A B C
D E F
G H I
J K L
christian@ubuntu:~$ cat Fichier1 Fichier2 > Fichier_Output
christian@ubuntu:~$ cat Fichier_Output
Ceci est le contenu de Fichier 1
Ligne de commande Linux (Semaine 4)
A B C
D E F
G H I
J K L
christian@ubuntu:~$
```



Les commandes utilisées pour les informations système

date



La commande **date** affiche la date et l'heure actuelles (y compris le jour de la semaine, le mois, l'heure, le fuseau horaire et l'année).

```
achraf@achraf-virtual-machine:~$ date
mer. 31 août 2022 01:02:24 CEST
achraf@achraf-virtual-machine:~$ date -d yesterday
mar. 30 août 2022 01:02:31 CEST
achraf@achraf-virtual-machine:~$ date -d tomorrow
jeu. 01 sept. 2022 01:02:58 CEST
achraf@achraf-virtual-machine:~$ date -d "2019-09-07"
sam. 07 sept. 2019 00:00:00 CEST
achraf@achraf-virtual-machine:~$ █
```



La commande `df -h` affiche l'espace disque disponible sur tous les systèmes de fichiers.

```
achraf@achraf-virtual-machine:~$ df -h
Sys. de fichiers Taille Utilisé Dispo Uti% Monté sur
udev              1,9G      0  1,9G  0% /dev
tmpfs             393M  1,6M  392M  1% /run
/dev/sda5          47G   11G   34G  25% /
tmpfs             2,0G      0  2,0G  0% /dev/shm
tmpfs             5,0M      0  5,0M  0% /run/lock
tmpfs             2,0G      0  2,0G  0% /sys/fs/cgroup
/dev/loop0         128K   128K      0 100% /snap/bare/5
/dev/loop1         56M    56M      0 100% /snap/core18/2409
/dev/loop4         219M   219M      0 100% /snap/gnome-3-34-1804/77
/dev/loop8         241M   241M      0 100% /snap/gnome-3-34-1804/24
/dev/loop9         50M    50M      0 100% /snap/snap-store/433
/dev/loop11        82M    82M      0 100% /snap/gtk-common-themes/1534
/dev/loop12        55M    55M      0 100% /snap/snap-store/558
/dev/sda1          511M  4,0K  511M  1% /boot/efi
```



La commande **uptime** fournit des informations sur la durée d'exécution du système sur une ligne.

Le résultat comprend l'heure actuelle, la durée d'utilisation du système, le nombre d'utilisateurs actuellement connectés et les charges moyennes du système au cours des 1, 5 et 15 dernières minutes, respectivement.

```
christian@ubuntu:~$ uptime
14:06:05 up 2:27, 1 user, load average: 0.04, 0.07, 0.03
christian@ubuntu:~$
```



Le chemin est une représentation textuelle de l'emplacement d'un élément de système de fichiers.

- > Il peut commencer par la **racine** du volume (ou pas)
- > Les **répertoires** sont séparés par des **/** (slash)
- > Le **fichier** est en dernier dans le chemin et se termine généralement par une **extension**.

/Dossier1/Dossier2/nomdefichier.ext

*Racine du
volume*

Chemins absolus et relatifs



```
/dossier1
└ dossier2
    └ fichier.txt
```

Un chemin est **absolu** s'il commence par la racine du volume

Il pointe au même endroit peu importe notre répertoire de travail

```
etudiant@ubuntu:/$ ls /dossier1/dossier2/fichier.txt
```

Un chemin **relatif** ne commence pas par la racine du volume

Il pointe à un endroit du système relatif à notre répertoire de travail

```
etudiant@ubuntu:/dossier1$ ls dossier2/fichier.txt
```



Les caractères génériques



Les caractères génériques permettent de faire des recherche dans les fichiers et répertoires en filtrant certains résultats qui ne correspondent pas au *patron* indiqué.

Il existe deux caractères génériques : ***** et **?**

L'astérisque ***** remplace n'importe quel nombre de caractères.

Le point d'interrogation **?** remplace un seul caractère.

Ce n'est pas clair? Voyez les exemples aux pages suivantes!



Caractères génériques (*)

```
christian@christian-virtual-machine:/etc$ ls -l *.conf
-rw-r--r-- 1 root root 3028 avr 23 2020 adduser.conf
-rw-r--r-- 1 root root 433 oct  1 2017 apg.conf
-rw-r--r-- 1 root root 769 jan 18 2020 appstream.conf
-rw-r--r-- 1 root root 26916 mar  3 2020 brltty.conf
-rw-r--r-- 1 root root 6569 aoû 24 23:12 ca-certificates.conf
-rw-r--r-- 1 root root 2969 aoû  3 2019 debconf.conf
-rw-r--r-- 1 root root 604 sep 15 2018 deluser.conf
-rw-r--r-- 1 root root 685 fév 14 2020 e2scrub.conf
-rw-r--r-- 1 root root 20 fév 10 2020 fprintd.conf
-rw-r--r-- 1 root root 280 jun 20 2014 fuse.conf
-rw-r--r-- 1 root root 2584 fév  1 2020 gai.conf
-rw-r--r-- 1 root root 5060 aoû 21 2019 hdparm.conf
-rw-r--r-- 1 root root 92 déc  5 2019 host.conf
-rw-r--r-- 1 root root 110 aoû 24 22:57 kernel-img.conf
-rw-r--r-- 1 root root 1308 déc  1 2017 kerneloops.conf
-rw-r--r-- 1 root root 34 avr 14 2020 ld.so.conf
-rw-r--r-- 1 root root 27 jan 18 2018 libao.conf
-rw-r--r-- 1 root root 191 fév 18 2020 libaudit.conf
-rw-r--r-- 1 root root 533 jan 21 2019 logrotate.conf
-rw-r--r-- 1 root root 14867 jan 31 2019 ltrace.conf
-rw-r--r-- 1 root root 808 fév 14 2020 mke2fs.conf
-rw-r--r-- 1 root root 624 mar 22 2020 mtools.conf
```

*.conf

Tous les fichiers et répertoires qui commencent par **n'importe quoi** et qui se terminent par **.conf**



Caractères génériques (*)

```
christian@christian-virtual-machine:/etc$ ls -l a*
-rw-r--r-- 1 root root 3028 avr 23 2020 adduser.conf
-rw-r--r-- 1 root root 401 jui 16 2019 anacrontab
-rw-r--r-- 1 root root 433 oct 1 2017 apg.conf
-rw-r--r-- 1 root root 769 jan 18 2020 appstream.conf

acpi:
total 24
-rwxr-xr-x 1 root root 391 jan 31 2019 asus-keyboard-backlight.sh
-rwxr-xr-x 1 root root 180 jan 31 2019 asus-wireless.sh
drwxr-xr-x 2 root root 4096 avr 23 2020 events
-rwxr-xr-x 1 root root 608 jan 31 2019 ibm-wireless.sh
-rwxr-xr-x 1 root root 455 jan 31 2019 tosh-wireless.sh
-rwxr-xr-x 1 root root 238 jan 31 2019 undock.sh

alsa:
total 4
drwxr-xr-x 2 root root 4096 aoû 24 23:08 conf.d

alternatives:
total 8
lrwxrwxrwx 1 root root 23 aoû 24 22:52 arptables -> /usr/sbin/arptables-nft
lrwxrwxrwx 1 root root 31 aoû 24 22:52 arptables-restore -> /usr/sbin/arptables-nft-restore
lrwxrwxrwx 1 root root 28 aoû 24 22:52 arptables-save -> /usr/sbin/arptables-nft-save
```

a*

Tous les fichiers et répertoires qui commencent par un **a** et qui se terminent par **n'importe quoi** (et le contenu des sous-répertoires qui suivent le même patron)

Caractères génériques (*)



```
christian@christian-virtual-machine:/etc$ ls -l *y*
-rw-r---- 1 root root 33 avr 23 2020 brlapi.key
-rw-r--r-- 1 root root 26916 mar 3 2020 brltty.conf
-rw-r--r-- 1 root root 1816 déc 26 2019 ethertypes
-rw-r--r-- 1 root root 711 avr 23 2020 hosts.deny
-rw-r--r-- 1 root root 24546 oct 18 2019 mime.types
-rw-rw-r-- 1 root root 350 aoû 24 22:56 popularity-contest.conf
-rw-r--r-- 1 root root 1382 fév 11 2020 rsyslog.conf
-rw-r--r-- 1 root root 5211 mar 11 2020 rygel.conf
-rw-r--r-- 1 root root 2351 fév 13 2020 sysctl.conf
```

```
brltty:
total 20
drwxr-xr-x 2 root root 4096 avr 23 2020 Attributes
d LibreOffice Writer ot root 4096 avr 23 2020 Contraction
drwxr-xr-x 41 root root 4096 avr 23 2020 Input
drwxr-xr-x 2 root root 4096 avr 23 2020 Keyboard
drwxr-xr-x 2 root root 4096 avr 23 2020 Text
```

```
cron.daily:
total 44
-rwxr-xr-x 1 root root 311 juil 16 2019 0anacron
-rwxr-xr-x 1 root root 376 déc 4 2019 apport
-rwxr-xr-x 1 root root 1478 avr 9 2020 apt-compat
-rwxr-xr-x 1 root root 355 déc 29 2017 bsdmainutils
```

y

Tous les fichiers et répertoires qui commencent par **n'importe quoi**, contiennent un **y** et qui se terminent par **n'importe quoi** (et le contenu des sous-répertoires qui suivent le même patron)



Caractères génériques (?)

```
christian@christian-virtual-machine:/etc$ ls -l ????.conf
-rw-r--r-- 1 root root 433 oct  1  2017 apg.conf
-rw-r--r-- 1 root root 2584 fév  1  2020 gai.conf
-rw-r--r-- 1 root root 552 déc 17  2019 pam.conf
-rw-r--r-- 1 root root 1260 déc 14  2018 ucf.conf
christian@christian-virtual-machine:/etc$ █
```

????.conf

Tous les fichiers et répertoires qui commencent par **une lettre**, suivie d'**une lettre** et d'une **autre** et qui se terminent par **.conf**



sudo, c'est l'abréviation de "superuser do", donc super utilisateur fait.

sudo permet à un administrateur système de donner à un utilisateur la possibilité d'exécuter une ou plusieurs commandes en tant que super utilisateur, tout en gardant une trace des commandes tapées et en demandant un mot de passe à l'utilisateur avant d'exécuter sa commande.

sudo peut aussi empêcher l'exécution de commandes critiques (généralement des commandes d'administration) pouvant gravement affecter le système.

```
rej@rej-virtual-machine:~$ apt-get update
Lecture des listes de paquets... Fait
E: Impossible d'ouvrir le fichier verrou /var/lib/apt/lists/lock - open (13: Permission non accordée)
E: Impossible de verrouiller le répertoire /var/lib/apt/lists/
W: Problème de suppression du lien /var/cache/apt/pkgcache.bin - RemoveCaches (13: Permission non accordée)
W: Problème de suppression du lien /var/cache/apt/srcpkgcache.bin - RemoveCaches (13: Permission non accordée)
rej@rej-virtual-machine:~$ sudo apt-get update
[sudo] Mot de passe de rej : ←
Réception de :1 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
```

Comprendre la syntaxe d'une commande : exemples

```
christian@christian-virtual-machine:/etc$ head -5 passwd
```

Visualisation d'un fichier dont on a les permissions

```
root:x:0:0:root:/root:/bin/bash  
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin  
bin:x:2:2:bin:/bin:/usr/sbin/nologin  
sys:x:3:3:sys:/dev:/usr/sbin/nologin  
sync:x:4:65534:sync:/bin:/bin/sync
```

```
christian@christian-virtual-machine:/etc$ head -5 shadow
```

Essaie de visualisation d'un fichier dont nous n'avons pas les permissions

```
head: impossible d'ouvrir 'shadow' en lecture: Permission non accordée
```

```
christian@christian-virtual-machine:/etc$ sudo head -5 shadow
```

Message d'erreur nous donnant un indice ...

```
[sudo] Mot de passe de christian :
```

```
root!:18864:0:99999:7:::  
daemon:*:18375:0:99999:7:::  
bin:*:18375:0:99999:7:::  
sys:*:18375:0:99999:7:::  
sync:*:18375:0:99999:7:::
```

```
christian@christian-virtual-machine:/etc$ █
```

Utilisation de sudo pour éléver nos priviléges et nous permettre de visualiser le fichier

Comprendre la syntaxe d'une commande : exemples

```
christian@christian-virtual-machine:/etc$ apt install tree
E: Impossible d'ouvrir le fichier verrou /var/lib/dpkg/lock-frontend - open (13: Permission non accordée)
E: Impossible d'obtenir le verrou de dpkg (/var/lib/dpkg/lock-frontend). Avez-vous les droits du superutilisateur ?
christian@christian-virtual-machine:/etc$ 
christian@christian-virtual-machine:/etc$ sudo apt install tree
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
tree est déjà la version la plus récente (1.8.0-1).
Les paquets suivants ont été installés automatiquement et ne sont plus nécessaires :
  libfprint-2-tod1 liblvm9
Veuillez utiliser « sudo apt autoremove » pour les supprimer.
0 mis à jour, 0 nouvellement installés, 0 à enlever et 17 non mis à jour.
christian@christian-virtual-machine:/etc$
```

Essaie d'installation de la commande tree

Message d'erreur nous donnant un indice ...

Utilisation de sudo pour éléver nos priviléges et nous permettre d'installer la commande tree