

Les commandes Linux

CONTENU	
1. Généralités.....	1
1.1. Forme d'une commande.....	1
1.2. Saisie et exécution d'une commande simple.....	1
1.3. Historique des commandes.	2
1.4. Les noms de fichiers dans les commandes.	2
2. Quelques commandes usuelles.	3
2.1. Quelques commandes de manipulation de répertoires.	3
2.2. Quelques commandes de manipulation de fichiers.	4
2.3. Quelques commandes de connexion et d'administration.	9
2.4. Quelques commandes d'information sur le matériel et le système.....	11
2.4. Commandes d'arrêt du serveur.	12

1. GENERALITES

L'interface utilisateur en mode texte s'appelle le SHELL. Il existe plusieurs SHELL. Le plus courant sous linux s'appelle BASH.

1.1. FORME D'UNE COMMANDE.

Une commande LINUX est de la forme : **nom_de_la_commande** [options] [arguments]

nom_de_la_commande est dans la plupart des cas en minuscules. Les crochets indiquent le caractère optionnel.

Les options sont en général précédées d'un ou deux " - " elles servent à modifier le comportement d'une commande. Les arguments sont les paramètres sur lesquels s'appliquent la commande.

Remarque importante : LINUX fait la distinction entre Majuscules et minuscules pour les commandes, noms de fichiers, répertoires, c'est parfois agaçant et source d'erreurs pour les débutants. Toto.txt, TOTO.txt et ToTo.txt sont des noms différents.

1.2. SAISIE ET EXECUTION D'UNE COMMANDE SIMPLE.

A l'invite du Shell saisissez la commande. L'exécution est lancée après validation par la touche "Entrée". Avant validation vous pouvez corriger votre saisie en utilisant les touches "Flèche Gauche", "Flèche Droite", "Effacement Arrière" et "Suppr" pour modifier le texte.

Ex 1 : [root@linuxserveur /root]# ls

Ex 2 : [root@linuxserveur /root]# ls -l

Ex 3 : [root@linuxserveur /root]# rm -rf mes-docs

Dans l'exemple 1 ci-dessus la commande ls qui liste fichiers, répertoires, du répertoire courant est présentée dans sa forme la plus simple elle n'a pas de paramètre. Dans l'exemple suivant la même commande est présentée avec l'option longue (l) permettant d'obtenir plus d'informations. Dans le troisième exemple la commande rm de suppression est présentée avec deux options combinées r (récursive) et f (force) et elle prend en argument le répertoire à supprimer. Certaines commandes disposent de plusieurs dizaines d'options et en la matière il est quasi impossible d'être exhaustif. Il vaut mieux s'habituer à utiliser les différentes aides en ligne. Un premier niveau d'aide est constitué par l'option -h ou --help selon les commandes. Cette option vous donne la forme de la commande et un court commentaire sur chaque option.

1.3. HISTORIQUE DES COMMANDES.

Les touches flèches haut et bas permettent de parcourir les dernières commandes de l'utilisateur, stockées dans le répertoire personnel dans le fichier `.bash_history`

1.4. LES NOMS DE FICHIERS DANS LES COMMANDES.

Il est possible d'utiliser des métacaractères (jokers) pour référencer des noms de fichiers. On trouve ainsi les métacaractères suivants :

- Une chaîne quelconque de caractères.
- `?` Un caractère quelconque.
- `[. . .]` Un caractère quelconque parmi l'ensemble indiqué
- `[! . . .]` Un caractère quelconque hors de l'ensemble indiqué

L'ensemble peut être une liste de caractères ou un intervalle (indiqué par `"-"`).

Exemples :

```
[root@linuxserveur /dev]# ls sdd*
```

```
sdd sdd10 sdd12 sdd14 sdd2 sdd4 sdd6 sdd8
```

```
sdd1 sdd11 sdd13 sdd15 sdd3 sdd5 sdd7 sdd9
```

```
[root@linuxserveur /dev]# ls sdd?
```

```
sdd1 sdd2 sdd3 sdd4 sdd5 sdd6 sdd7 sdd8 sdd9
```

```
[root@linuxserveur /dev]# ls sdd[1-5]
```

```
sdd1 sdd2 sdd3 sdd4 sdd5
```

```
[root@linuxserveur /dev]# ls sdd[!1-5]
```

```
sdd6 sdd7 sdd8 sdd9
```

```
[root@linuxserveur /dev]# ls sdd[17]
```

```
sdd1 sdd7
```

```
[root@linuxserveur /dev]# ls sd[a-z][01]
```

```
sda1 sdb1 sdc1 sdd1 sde1 sdf1 sdg1
```

2. QUELQUES COMMANDES USUELLES.

Il existe des commandes internes au Shell (elles sont marquées dans la suite d'un astérisque) et des commandes externes. La commande type (voir ci-après) permet de savoir de quel type est une commande.

2.1. QUELQUES COMMANDES DE MANIPULATION DE REPERTOIRES.

- **Pwd :**

La commande pwd affiche le nom absolu du répertoire courant, elle vous permet de connaître votre position dans l'arborescence.

Syntaxe : pwd

```
root@Debian12:/# cd usr/local/bin
root@Debian12:/usr/local/bin# pwd
/usr/local/bin
```

- **Ls :**

La commande ls liste le contenu de répertoires.

Syntaxe : ls [-options] [répertoire1] [répertoire2]

Cette commande accepte un très grand nombre d'options pouvant être combinées.

ls liste les entrées du répertoire courant.

ls -R liste récursivement tous les répertoires.

ls -a liste toutes les entrées y compris les entrées cachées.

ls -l liste les entrées et affiche toutes les informations.

ls -F liste les entrées et ajoute / derrière les noms de répertoires.

ls --color liste les entrées avec une couleur différente suivant le type (fichier exécutable, texte, répertoire, liens ...)

- **cd (*) :**

La commande cd permet le positionnement sur un répertoire

syntaxe : cd [répertoire]

cd / positionnement à la racine

cd .. positionnement sur le répertoire parent

cd /usr/X11R6/bin positionnement depuis la racine sur le répertoire /usr/X11R6/bin .

cd ../lib positionnement à partir du répertoire parent sur le répertoire /lib.

cd cd~ ou cd \$HOME positionnement depuis n'importe où sur le répertoire personnel.

cd - positionnement sur le répertoire précédent.

- **mkdir :**

La commande mkdir permet la création de répertoires. Chaque répertoire contient deux entrées : le répertoire lui-même (.) et le répertoire parent (..).

Syntaxe : mkdir répertoire1 [répertoire2] [répertoireN]

mkdir perso créé le répertoire perso dans le répertoire courant

mkdir ../perso créé le répertoire perso dans le répertoire parent

mkdir /root/docs/perso créé le répertoire perso dans le répertoire /root/docs

- **rmdir :**

rmdir perso supprime le répertoire perso du répertoire courant

rmdir ../perso supprime le répertoire perso du répertoire parent

rmdir /root/docs/perso supprime le répertoire perso du répertoire /root/docs

La commande rmdir permet la suppression de répertoires vides.

Syntaxe : rmdir répertoire1 [répertoire2] [répertoireN]

N.B. pour supprimer un répertoire non vide voir la commande rm

2.2. QUELQUES COMMANDES DE MANIPULATION DE FICHIERS.

- **cat :**

La commande cat permet de visualiser le contenu d'un fichier texte.

Syntaxe : cat fichier fichier1 [fichier2]

Exemple d'affichage d'un fichier :

```
adeline@Debian12:~$ cat addition2.java
class addition2
{
    public static void main(String arg[])
    {
        int a = 10;
        a = a*2;
        System.out.println("Le résultat est égal à "+a);
    }
}
```

N.B. Pour savoir si un fichier peut être affiché voir la commande file.

- **more ou less :**

Ces commandes permettent d'afficher un fichier page par page. On fait défiler le fichier ligne par ligne en appuyant sur la touche "Entrée", page par page avec "Espace", on revient en arrière avec "b" et on quitte l'affichage avec "q". Le pourcentage qui apparaît sur la dernière ligne indique la position actuelle par rapport à la totalité du fichier.

```
adeline@Debian12:~/Documents$ more readme.txt
/*****/
Hello world
/*****/
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce non odio in ex bi
bendum pulvinar in in felis. Aenean nulla lacus, sagittis non tincidunt sit amet
, iaculis ac leo. Curabitur bibendum consectetur ultricies. Ut at ligula id veli
t vestibulum finibus. Orci varius natoque penatibus et magnis dis parturient mon
tes, nascetur ridiculus mus. Proin finibus viverra erat non congue. Sed euismod
tellus at tortor pellentesque, rhoncus tincidunt dolor elementum.
/*****/

Duis tempor lobortis rutrum. Integer eu efficitur lorem. Etiam ac est eros. In s
ed tristique orci. Curabitur et quam lorem. Etiam hendrerit condimentum porta. D
onec porta maximus ligula ut pellentesque. Sed id luctus libero, non imperdiet p
urus. Integer eu ante tristique, venenatis nunc quis, euismod nisi. Integer ut o
dio pharetra risus blandit venenatis a at tortor. Pellentesque pretium aliquam c
ommodo. Vivamus interdum mollis tristique. Praesent id libero iaculis magna orna
--More-- (72%)
```

N.B. Pour savoir si un fichier peut être affiché voir la commande file.

- **file :**

La commande file permet d'obtenir la nature du ou des fichiers spécifiés.

syntaxe : file fichier1 [fichier2] [fichierN]

```
adeline@Debian12:~$ file moyenne.java
moyenne.java: Java source, ASCII text
adeline@Debian12:~$ file moyenne.class
moyenne.class: compiled Java class data, version 61.0 (Java SE 17)
adeline@Debian12:~$ file *
addition2.class: compiled Java class data, version 61.0 (Java SE 17)
addition2.java: C++ source, Unicode text, UTF-8 text
Desktop:          directory
Documents:        directory
Downloads:        directory
moyenne.class:    compiled Java class data, version 61.0 (Java SE 17)
moyenne.java:     Java source, ASCII text
Music:            directory
Pictures:         directory
Public:           directory
Templates:        directory
Videos:           directory
adeline@Debian12:~$
```

- **cp :**

La commande cp sert à copier des fichiers.

Syntaxe : cp [options] [répertoire1/]fichier1 [répertoire2/]fichier2

cp [options] fichier1 [fichier2] ... [fichierN] répertoire

Avec la première syntaxe le fichier1 du répertoire1 est copié sous le nom fichier2 dans le répertoire2. Notons que répertoire1 et 2 peuvent être le même répertoire. Avec la seconde syntaxe les fichiers sont copiés dans le répertoire de destination à partir du répertoire en cours.

Exemples :

Dans l'exemple ci-dessous le fichier profile est copié du répertoire courant /etc vers le répertoire /root.

```
root@Debian12:/# cd /etc
root@Debian12:/etc# cp profile /root
```

Dans l'exemple ci-dessous le fichier profile est copié du répertoire courant /etc vers le répertoire /root et renommé.

```
root@Debian12:/# cd /etc
root@Debian12:/etc# cp profile /root/ex_de_profile
```

N.B. si le ou les fichiers cible existent déjà, ils seront écrasés purement et simplement.

- **mv :**

La commande mv sert soit à déplacer un fichier soit à le renommer.

Syntaxe : mv [répertoire1/]fichier1 [répertoire2/]fichier2

mv fichier1 [fichier2] ... [fichierN] répertoire

Dans la première forme si les répertoires sources et cibles sont les mêmes le fichier est simplement renommé.

Dans l'exemple ci-dessous le fichier readme.txt est renommé en readme1.txt.

```
root@Debian12:/home/adeline/Documents# mv Dossier/readme.txt Dossier/readme1.txt
root@Debian12:/home/adeline/Documents# ls
addition2.java  Dossier  readme.txt
root@Debian12:/home/adeline/Documents# cd Dossier
root@Debian12:/home/adeline/Documents/Dossier# ls
readme1.txt
```

Dans l'exemple suivant le fichier readme.txt est déplacé du répertoire /Documents vers le répertoire /Documents/Dossier

```
root@Debian12:/home/adeline# mv Documents/readme.txt Documents/Dossier/readme.txt
root@Debian12:/home/adeline# ls
addition2.java  Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos
root@Debian12:/home/adeline# cd Documents/
root@Debian12:/home/adeline/Documents# ls
addition2.java  Dossier
root@Debian12:/home/adeline/Documents# cd Dossier/
root@Debian12:/home/adeline/Documents/Dossier# ls
readme1.txt  readme.txt
```

N.B. Si le ou les fichiers cibles existent déjà, ils seront écrasés purement et simplement.

- **rm :**

rm permet de supprimer fichiers et répertoires. Parmi les options de cette commande deux sont intéressantes : r permet la suppression d'un répertoire même non vide et i demande confirmation avant suppression. Cette commande est très dangereuse, c'est pourquoi en général un alias est créé, forçant la demande de confirmation. Vérifiez qu'il est créé sur votre système en tapant alias, vous devriez voir apparaître alias rm='rm -i'. Si cet alias n'existe pas, un bon conseil : créez-le. (Voir la commande alias).

L'exemple suivant montre la suppression de deux fichiers et du répertoire mes-docs. A noter que si le répertoire mes-docs contient des sous répertoires ceux-ci sont également supprimés récursivement (d'où le r).

```
root@Debian12:/home/adeline/Documents/Dossier# ls fich*
fichier1.txt  fichier2.txt
root@Debian12:/home/adeline/Documents/Dossier# ls dossier1
doc1  doc2
root@Debian12:/home/adeline/Documents/Dossier# rm fich*
root@Debian12:/home/adeline/Documents/Dossier# rm -r dossier1
root@Debian12:/home/adeline/Documents/Dossier# ls fich*
ls: cannot access 'fich*': No such file or directory
root@Debian12:/home/adeline/Documents/Dossier# ls dossier1
ls: cannot access 'dossier1': No such file or directory
```

N.B.1. Si un alias "rm=rm-i" a été créé il est tout de même possible de forcer la suppression avec l'option "-f". Ainsi "rm -rf" ne vous demandera aucune confirmation pour la suppression d'un répertoire même non vide.

N.B.2. Les alias sont définis dans les fichiers /etc/bashrc et ~/.bashrc.

- **type (*) :**

La commande type vous permet de connaître la nature d'une commande, d'un mot clé ...

Syntaxe : type [-all|-type|-path] nom [nom]

```
root@Debian12:/home/adeline/Documents/Dossier# type -all ls rm if cd
ls is /usr/bin/ls
ls is /bin/ls
rm is /usr/bin/rm
rm is /bin/rm
if is a shell keyword
cd is a shell builtin
```

2.3. QUELQUES COMMANDES DE CONNEXION ET D'ADMINISTRATION.

- **exit (*) :**

Cette commande permet d'interrompre une session en cours, elle déconnecte l'utilisateur du système. La combinaison de touches

"CTRL+D" joue le même rôle.

- **who et whoami :**

La commande who permet de savoir qui est connecté à votre machine. Avec l'option « -q » le nombre d'utilisateurs connectés est fourni :

```
adeline@Debian12:~$ who -q
adeline
# users=1
adeline@Debian12:~$
```

Sinon la commande fournit, entre autres, l'identité et le type de connexion. Les commentaires (#) ont été ajoutés.

```
adeline@Debian12:~$ who
adeline  tty2          2024-02-27 12:28 (tty2)
adeline@Debian12:~$
```

whoami affiche le nom d'utilisateur associé à l'U-ID effectif en cours. C'est à dire vous donne votre identité. Cette commande peut être très utile en phase d'essais, par exemple, lorsque vous travaillez sur plusieurs comptes.

```
adeline@Debian12:~$ whoami
adeline
adeline@Debian12:~$
```

Remarque : dans le cas ci-dessus le prompt ([adeline@Debian12 :~\$ /]) donne l'identité de l'utilisateur en cours, dans ce cas la commande perd de son intérêt. Le prompt est défini dans le fichier /etc/profile par : PS1="[u@\h \W]\\$ " où "\u" désigne le nom de l'utilisateur, "\h" le nom d'hôte, "\W" le nom de base du répertoire en cours, "\\$" produit l'affichage de "#" dans le cas de root (uid=0) et \$ dans les autres cas.

- **id :**

La commande id permet de connaître les renseignements identifiant l'utilisateur.

```
adeline@Debian12:~$ id
uid=1000(adeline) gid=1000(adeline) groups=1000(adeline),100(users),995(vboxsf)
adeline@Debian12:~$ id root
uid=0(root) gid=0(root) groups=0(root)
adeline@Debian12:~$ id adeline
uid=1000(adeline) gid=1000(adeline) groups=1000(adeline),100(users),995(vboxsf)
```

- **passwd :**

passwd permet à l'utilisateur de changer son mot de passe. Comme on peut le constater dans la séquence reproduite ci-dessous, Linux effectue un contrôle syntaxique du mot de passe.

```
adeline@Debian12:~$ passwd
Changing password for adeline.
Current password:
New password:
Retype new password:
You must choose a longer password.
New password:
Retype new password:
The password has not been changed.
New password:
Retype new password:
You must choose a longer password.
passwd: Authentication token manipulation error
passwd: password unchanged
adeline@Debian12:~$ passwd
Changing password for adeline.
Current password:
New password:
Retype new password:
passwd: password updated successfully
adeline@Debian12:~$ █
```

Trois fichiers essentiels servent à stocker les informations sur les divers utilisateurs. Il s'agit des fichiers /etc/passwd, /etc/shadow et /etc/group. Ce sont des fichiers texte, toutes les informations y apparaissent en clair mais naturellement les mots de passe sont cryptés.

- **su :**

su permet à tout utilisateur de passer en mode Super Utilisateur. Avec l'option "-" les fichiers profile sont pris en compte.

```
adeline@Debian12:~$ id
uid=1000(adeline) gid=1000(adeline) groups=1000(adeline),100(users),995(vboxsf)
adeline@Debian12:~$ su -
Password:
root@Debian12:~# id
uid=0(root) gid=0(root) groups=0(root)
root@Debian12:~# █
```

2.4. QUELQUES COMMANDES D'INFORMATION SUR LE MATERIEL ET LE SYSTEME.

- **df :**

df (disk free) affiche l'espace libre sur les systèmes de fichiers montés.

```
adeline@Debian12:~$ df
Filesystem      1K-blocks      Used Available Use% Mounted on
udev            2480020         0    2480020   0% /dev
tmpfs           502384       1240     501144   1% /run
/dev/sda1       56780936 6889704 46974472  13% /
tmpfs           2511900         0    2511900   0% /dev/shm
tmpfs           5120          8        512   1% /run/lock
tmpfs           502380       2532     499848   1% /run/user/1000
adeline@Debian12:~$
```

- **free :**

free affiche les quantités de mémoire et de swap libres et utilisées, ainsi que la mémoire partagée et les buffers utilisés par le noyau.

```
adeline@Debian12:~$ free
              total        used        free      shared  buff/cache   available
Mem:          5023804      2716524       846996        49716     1766500     2307280
Swap:          999420           0       999420
adeline@Debian12:~$
```

- **uname :**

La commande uname fournit un certain nombre d'informations système.

syntaxe : uname [options]

uname ou uname -s affiche le nom du système d'exploitation.

uname -r affiche le numéro de version du système d'exploitation.

uname -a affiche les informations affichés par le commandes ci-dessus et d'autres.

```
adeline@Debian12:~$ uname -sra
Linux Debian12 6.1.0-17-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.69-1 (2023-12-30) x86_64 GNU/Linux
adeline@Debian12:~$
```

A noter : le répertoire proc est constitué de fichiers contenant des informations sur le système. Ces fichiers sont visibles. Il y a parexemple cpuinfo qui contient des infos sur les processeurs, meminfo qui contient des infos sur la mémoire, partitions sur les disques et partitions. On peut visualiser par exemple ce dernier fichier par cat /proc/partitions.

```
adeline@Debian12:~$ cat /proc/partitions
major minor #blocks name
 8         0   59009504 sda
 8         1   58007552 sda1
 8         2         1 sda2
 8         5    999424 sda5
11         0   1048575 sr0
adeline@Debian12:~$
```

2.4. COMMANDES D'ARRET DU SERVEUR.

- **halt ou shutdown -n now :**

Ces deux commandes sont équivalentes à un arrêt immédiat du système.

Une fois le système arrêté le message System halted s'affiche. C'est à ce moment que vous pouvez mettre hors tension, pas avant !

N.B. Ces commandes sont en principe réservées à l'administrateur.

- **reboot ou shutdown -r now :**

Ces deux commandes sont équivalentes à un redémarrage du système. Elles ne sont accessibles par défaut que par l'administrateur.

N.B : Vous pouvez également utiliser la combinaison de touches "CTRL+ALT+DEL", cette combinaison est, par défaut, accessible par tous. Le rôle de cette séquence de touches est défini dans le fichier /etc/inittab .

3. COMMANDES DIVERSES

3.1 LA COMMANDE ECHO

Utilisez echo pour afficher une chaîne de caractères ou le contenu d'une variable sur la sortie standard.

```
adeline@Debian12:~$ echo Hello world
Hello world
adeline@Debian12:~$ █
```

3.2 LA COMMANDE FIND

La commande find recherche dans l'arborescence des répertoires pour localiser des groupes particuliers de fichiers qui correspondent aux critères indiqués en argument à l'aide des options -name, -type, -exec, -size, -mtime et -user.

Exemple :

- Rechercher par nom avec l'option -name :

Pour rechercher dans le répertoire courant tous les fichiers .txt utiliser le modèle générique *.txt :

```
adeline@Debian12:~/Documents/Dossier$ find . -name "*.txt"
./readme.txt
./readme1.txt
adeline@Debian12:~/Documents/Dossier$ █
```

- . fait référence au répertoire courant
- -name « *.txt » spécifie les critères à respecter

Cette recherche avec -name est sensible à la casse dans l'exemple ci-dessous le fichier ReadMe.txt est ignoré. Pour que la recherche est insensible à la casse utilisez l'option -iname :

```
adeline@Debian12:~/Documents/Dossier$ find . -iname "readme.txt"
./ReadMe.txt
./readme.txt
adeline@Debian12:~/Documents/Dossier$ find . -name "readme.txt"
./readme.txt
adeline@Debian12:~/Documents/Dossier$ █
```

- Rechercher par type de fichier avec l'option -type :

Les types de fichiers peuvent être :

- f plain files
- d directories
- l symbolic link
- b block devices
- c character devices
- p named pipes
- s sockets

Par exemple -type d ne listera que les répertoires :

```
adeline@Debian12:~/Documents$ find . -type d
.
./Dossier
adeline@Debian12:~/Documents$
```

- Rechercher par taille de fichier avec l'option -size :

Exemple, rechercher les fichiers de plus de 1Mo :

```
adeline@Debian12:~$ find -size +1M
./mozilla/firefox/51pbuenm.default-esr/security_state/data.safe.bin
./mozilla/firefox/51pbuenm.default-esr/storage/permanent/chrome/idb/3870112724rsegmnoittet-es.sqlite
./mozilla/firefox/51pbuenm.default-esr/favicons.sqlite
./mozilla/firefox/51pbuenm.default-esr/places.sqlite
./mozilla/firefox/51pbuenm.default-esr/gmp-gmpopenh264/2.3.2/libgmpopenh264.so
./cache/mozilla/firefox/51pbuenm.default-esr/safebrowsing/google-trackwhite-digest256.vlpset
./cache/mozilla/firefox/51pbuenm.default-esr/safebrowsing/google4/goog-phish-proto.vlpset
./cache/mozilla/firefox/51pbuenm.default-esr/startupCache/scriptCache-child.bin
./cache/mozilla/firefox/51pbuenm.default-esr/startupCache/scriptCache.bin
./cache/mozilla/firefox/51pbuenm.default-esr/startupCache/startupCache.8.little
./cache/mozilla/firefox/51pbuenm.default-esr/cache2/entries/DC8D71CF5CE210129C76A80EF70B3266831C0671
./cache/mozilla/firefox/51pbuenm.default-esr/cache2/entries/C39A2E50DF4EF7013B29EF7E948C0557F8CB08DA
./cache/gstreamer-1.0/registry.x86_64.bin
./cache/tracker3/files/http%3A%2F%2Ftracker.api.gnome.org%2Fontology%2Fv3%2Ftracker%23FileSystem.db
./cache/tracker3/files/http%3A%2F%2Ftracker.api.gnome.org%2Fontology%2Fv3%2Ftracker%23Software.db
./cache/tracker3/files/http%3A%2F%2Ftracker.api.gnome.org%2Fontology%2Fv3%2Ftracker%23Documents.db
./cache/tracker3/files/http%3A%2F%2Ftracker.api.gnome.org%2Fontology%2Fv3%2Ftracker%23Audio.db
./cache/tracker3/files/http%3A%2F%2Ftracker.api.gnome.org%2Fontology%2Fv3%2Ftracker%23Pictures.db
./cache/tracker3/files/meta.db
./cache/tracker3/files/http%3A%2F%2Ftracker.api.gnome.org%2Fontology%2Fv3%2Ftracker%23Video.db
./cache/tracker3/files/http%3A%2F%2Ftracker.api.gnome.org%2Fontology%2Fv3%2Ftracker%23FileSystem.db-w
1
./cache/mesa_shader_cache/index
./cache/gnome-software/appstream/components.xmlb
./cache/gnome-software/odrs/ratings.json
adeline@Debian12:~$
```

Le signe + signifie rechercher les fichiers plus grands que le nombre suivant. Le signe – peut être utilisé pour indiquer plus petit que. N'utiliser aucun signe signifierait correspondre exactement à la taille.

Le nombre est suivi de l'unité de taille de fichier. Les unités peuvent être :

Symbole	Unité de taille
b	512 blocs
c	Octets
k	kilobytes
M	Mégaoctets
G	Gigaoctets

3.3 LA COMMANDE GREP

La commande grep sert à extraire des lignes d'un fichier ou du retour d'une commande.

Syntaxe : `grep « chaîne de caractère » nom_du_fichier`

Ainsi toutes les lignes contenant la chaîne de caractère seront affichées.

Les options de la commande grep :

- -i pour ignorer la casse
- -v permet d'ignorer toutes les lignes contenant le motif
- -m num permet d'afficher le nombre de ligne souhaité, exemple : je veux afficher les 3 premières lignes contenant « usr » dans le fichier /etc/passwd :

```
adeline@Debian12:~$ grep -m 3 usr /etc/passwd
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
adeline@Debian12:~$
```

- -n permet d'afficher le numéro de ligne
- -c affiche seulement le nombre de ligne qui contient le motif
- -l permet d'afficher uniquement le nom des fichiers contenant le motif
- -R permet d'utiliser la récursivité et ainsi effectuer une recherche dans plusieurs fichiers ou dossier
- -A num affiche la ligne qui contient le motif + les num lignes suivantes
- -B num affiche la ligne qui contient le motif + les num lignes précédentes
- -E pour effectuer une recherche en utilisant une regex

3.4 LA COMMANDE HISTORY

La commande history permet d'afficher l'historique de toutes les commandes utilisées depuis le début de la session en cours.

Exemple :

```
adeline@Debian12:~$ history
 1  su root
 2  cowsay hello
 3  su -
 4  docker run hello-world
 5  su root
 6  su
 7  ls
 8  cd Documents/
 9  touch addition2.java
10  vim addition2.java
11  apt-get install vim
12  cd ..
13  vim addition2.java
14  cat addition2.java
15  ls
16  cd Documents/
17  touch readme.txt
18  vim readme.txt
19  more readme.txt
20  ls
21  mv /Documents/readme.txt /Documents/readme1.txt
22  cd ..
23  mv /Documents/readme.txt /Documents/readme1.txt
24  mv /Documents/readme /Documents/readme1
25  ls
26  cd Documents/
27  ls
28  cd Dossier/
29  ls
30  mv /Documents/readme /Documents/readme1
31  mv readme.txt /Dossier/readme1.txt
32  mv readme.txt /Documents/Dossier/readme1.txt
33  mv /Dossier/readme.txt /Dossier/readme1.txt
34  mv Dossier/readme.txt Dossier/readme1.txt
35  cd ..
36  mv Dossier/readme.txt Dossier/readme1.txt
37  su
38  who -q
```

- Le nombre d'entrée :

On peut savoir combien history peut stocker de commande en entrant la commande :

```
adeline@Debian12:~$ echo $HISTSIZE
1000
adeline@Debian12:~$
```

Pour changer ce nombre à 300 (par exemple) :

```
adeline@Debian12:~$ export HISTSIZE=300
adeline@Debian12:~$ echo $HISTSIZE
300
adeline@Debian12:~$
```

- Effacer ces traces :

Pour vider votre fichier history afin que root ou un utilisateur avec certains privilèges ne puisse plus voir les commandes que vous avez entrées, utilisez la commande :

```
adeline@Debian12:~$ history -c
adeline@Debian12:~$ history
  1  history
adeline@Debian12:~$
```

3.5 LA COMMANDE TOUCH

La commande touch est utilisé pour créer un nouveau fichier vide.

Syntaxe : touch nom_du_fichier

Exemple :

```
adeline@Debian12:~$ cd Documents/
adeline@Debian12:~/Documents$ touch monFichier.md
adeline@Debian12:~/Documents$ ls
addition2.java  Dossier  monFichier.md
adeline@Debian12:~/Documents$
```

LA COMMANDE CHMOD

La commande `chmod` permet de modifier les droits d'utilisation d'un répertoire ou d'un fichier.

Pour vérifier les autorisations et les attributs, utiliser la commande `ls -l` :

```
adeline@Debian12:~$ ls -l
total 36
-rw-r--r-- 1 adeline adeline 145 Feb 27 12:49 addition2.java
drwxr-xr-x 2 adeline adeline 4096 Jan 5 08:43 Desktop
drwxr-xr-x 3 adeline adeline 4096 Feb 28 10:00 Documents
drwxr-xr-x 2 adeline adeline 4096 Jan 5 08:43 Downloads
drwxr-xr-x 2 adeline adeline 4096 Jan 5 08:43 Music
drwxr-xr-x 2 adeline adeline 4096 Jan 5 08:43 Pictures
drwxr-xr-x 2 adeline adeline 4096 Jan 5 08:43 Public
drwxr-xr-x 2 adeline adeline 4096 Jan 5 08:43 Templates
drwxr-xr-x 2 adeline adeline 4096 Jan 5 08:43 Videos
adeline@Debian12:~$ cd Documents/
adeline@Debian12:~/Documents$ ls -l
total 4
-rw-r--r-- 1 adeline adeline 0 Feb 27 12:42 addition2.java
drwxr-xr-x 2 root root 4096 Feb 28 08:51 Dossier
-rw-r--r-- 1 adeline adeline 0 Feb 28 10:00 monFichier.md
adeline@Debian12:~/Documents$
```

Cette commande permet de définir les droits pour 3 types d'utilisateurs différents :

- Les droits concernant le propriétaire du fichier/dossier
- Les droits concernant le groupe propriétaire du fichier/dossier
- Les droits concernant tous les autres utilisateurs

Il y a 3 types de droits :

- `r` = read : droit en lecture
- `w` = write : droit d'écriture
- `x` = exécution : droit d'exécution

Chacun de ces types à une valeur :

- « `r` » à la valeur 4
- « `w` » à la valeur 2
- « `x` » à la valeur 1

Il faut additionner les chiffres entre eux selon les droits que vous souhaitez attribuer. Lorsque l'on veut donner tous les droits, c'est-à-dire lecture – écriture – exécution, on additionne $4 + 2 + 1$, ce qui donne 7 et correspondra au contrôle total pour le type d'utilisateur auquel le 7 sera attribué.

Exemple, je veux que seul le propriétaire du fichier monFichier.md(c'est-à-dire adeline) puisse l'ouvrir et le modifier, il faut que je lui attribue les droits maximum(correspondant à la valeur 7), et pour les autre types d'utilisateurs je doit leur donner aucun droit, la valeur sera donc 0.

```
adeline@Debian12:~/Documents$ chmod 700 monFichier.md
adeline@Debian12:~/Documents$ ls -l
total 4
-rw-r--r-- 1 adeline adeline    0 Feb 27 12:42 addition2.java
drwxr-xr-x 2 root    root    4096 Feb 28 08:51 Dossier
-rwx----- 1 adeline adeline    0 Feb 28 10:00 monFichier.md
adeline@Debian12:~/Documents$
```

4. CREATION ET GESTION DES UTILISATEURS

4.1 CREATION D'UN UTILISATEUR

Pour créer un utilisateur on utilise la commande `adduser`, elle permet de créer un compte utilisateur avec un répertoire personnel, un groupe principal et un shell par défaut. On peut également ajouter des options pour personnaliser d'avantages le compte.

Exemple :

```
root@Debian12:/# sudo adduser utilisateur1
Adding user `utilisateur1' ...
Adding new group `utilisateur1' (1001) ...
Adding new user `utilisateur1' (1001) with group `utilisateur1 (1001)' ...
Creating home directory `/home/utilisateur1' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for utilisateur1
Enter the new value, or press ENTER for the default
    Full Name []: utilisateur1
    Room Number []: 1
    Work Phone []: 0000000000
    Home Phone []: 0101010101
    Other []: .....
Is the information correct? [Y/n] y
Adding new user `utilisateur1' to supplemental / extra groups `users' ...
Adding user `utilisateur1' to group `users' ...
root@Debian12:/#
```

Syntaxe : `sudo adduser [nom_utilisateur]`

- On spécifie un mot de passe
- On renseigne les informations du compte utilisateur
- On valide par Y
- L'utilisateur est créé dans le groupe users

4.2 SUPPRESSION D'UN UTILISATEUR

Pour supprimer un utilisateur on utilise la commande `userdel` :

```
root@Debian12:/# sudo userdel utilisateur1
root@Debian12:/#
```