



Shell

Nous allons voir au travers de cette documentation, des commandes Linux que l'on peut rentrer dans son terminal.

Nous allons donc lister quelques commandes et expliquer leurs fonctionnalités.

Pour afficher le manuel d'une commande, nous avons qu'à rajouter "man" devant la commande pour afficher tout ce qu'on peut faire avec la commande et également les conditions de syntaxe pour que la commande fonctionne.

A screenshot of a Linux desktop environment showing a terminal window. The window title is "Terminal". The status bar at the top shows the date and time as "27 sept. 10:14" and the user as "valux@debian: ~". The terminal itself has a dark background and contains the command "valux@debian:~\$ man ls". A red arrow points from the bottom left towards the "ls" part of the command. The window has standard window controls (minimize, maximize, close) in the top right corner.



Nous arrivons donc sur les détails de la commande “ls”:

```
LS(1)                               Commandes de l'utilisateur      LS(1)

NOM
    ls - Afficher le contenu de répertoires

SYNOPSIS
    ls [OPTION]... [FICHIER]...

DESCRIPTION
    Afficher les informations des FICHIERS (du répertoire courant par défaut). Les entrées sont triées alphabétiquement si aucune des options -cftuvSUX ou --sort n'est indiquée.

    Les paramètres obligatoires pour les options de forme longue le sont aussi pour les options de forme courte.

    -a, --all
        inclure les entrées débutant par « . »

    -A, --almost-all
        omettre les fichiers « . » et « .. »

    --author
        avec -l, afficher l'auteur de chaque fichier

    -b, --escape
        afficher les caractères non graphiques sous la forme de caractères d'échappement de style C

    --block-size=TAILLE
        avec -l, ajuster les tailles avec TAILLE quand elles sont affichées ; par exemple « --block-size=M » ; voir le format de TAILLE ci-dessous

    -B, --ignore-backups
        omettre les entrées se terminant par « ~ »

    -c
        avec -lt, trier selon la date de modification « ctime » en l'affichant ; avec -l, trier selon le nom et afficher la date de modification ; sinon, trier selon la date de modification, de la plus récente à la plus ancienne

    -c
        afficher en colonnes

    --color[=QUAND]
        colorer la sortie QUAND ; plus d'informations ci-dessous
```



Pour afficher les fichiers cachés du home de notre Linux, nous pouvons entrer la commande : ls -d .*

Le terminal va donc nous afficher les fichiers que l'on ne voit pas sur le home.

```
Activités Terminal 27 sept. 10:35 valux@debian:~  
+ valux@debian:~$ ls -d .*  
.bash_history .bashrc .config .face.icon .lessht .pki .ssh  
.bash_logout .cache .face .gnupg .local .profile  
valux@debian:~$
```

A red arrow points from the text "Afficher les fichiers cachés" in the previous slide to the command "ls -d .*" in the terminal window.

Pour afficher les informations sur les droits de ces fichiers, nous pouvons rentrer 2 commandes, soit “ls -d .* -al” ou alors “ls -ld .*”

```
Activités Terminal 27 sept. 10:45 valux@debian:~  
+ valux@debian:~$ ls -ld .*  
valux@debian:~$
```

A red arrow points from the text "Afficher les droits des fichiers" in the previous slide to the command "ls -ld .*" in the terminal window.



Nous avons donc les informations sur les droits des fichiers cachés
A noter qu'un tiret du 6 est un fichier et la lettre "d" indique qu'il s'agit d'un dossier.

The screenshot shows a terminal window with three tabs, all titled 'valux@debian: ~'. The middle tab is active. A red bracket is drawn around the file names in the output of the 'ls -ld .' command, specifically '.cache', '.config', '.face.icon', '.gnupg', '.local', '.pki', '.profile', and '.ssh'. The output is as follows:

```
valux@debian:~$ ls -ld .*
-rw----- 1 valux valux 2143 27 sept. 10:32 .bash_history
-rw-r--r-- 1 valux valux 220 20 sept. 18:17 .bash_logout
-rw-r--r-- 1 valux valux 3526 20 sept. 18:17 .bashrc
drwx----- 13 valux valux 4096 27 sept. 10:41 .cache
drwxr-xr-x 12 valux valux 4096 25 sept. 08:39 .config
-rw-r--r-- 1 valux valux 5290 20 sept. 18:17 .face
lrwxrwxrwx 1 valux valux 5 20 sept. 18:17 .face.icon -> .face
drwx----- 2 valux valux 4096 20 sept. 18:27 .gnupg
-rw----- 1 valux valux 20 27 sept. 10:26 .lessht
drwx----- 4 valux valux 4096 20 sept. 18:25 .local
drwx----- 3 valux valux 4096 25 sept. 08:39 .pki
-rw-r--r-- 1 valux valux 807 20 sept. 18:17 .profile
drwx----- 2 valux valux 4096 20 sept. 18:27 .ssh
valux@debian:~$
```



Quelques questions pour accompagner notre documentation :

Comment ajouter des options à une commande ?

Pour ajouter une option à une commande, on ajoute un tiret du 6 suivi d'une ou plusieurs lettres. Par exemple pour la commande “ls” on a rajouté l'option “-ld” pour afficher les fichiers cachés. On aurait pu d'ailleurs rentrer l'option “-a” mais cette option nous aurait affiché TOUS les fichiers/dossiers y compris les fichiers cachés.

Quelles sont les deux syntaxe principales d'écriture des options pour une commande ?

Les deux principales syntaxes d'écriture pour les options d'une commande sont une lettre suivie d'un tiret du 6.

```
valux@debian:~$ ls -d .* -al
```

```
valux@debian:~$ ls .* -ld
```



Nous allons à présent réaliser plusieurs actions (lire un fichier, afficher les 10 premières/dernières lignes etc...) via des lignes de commande Linux.

Nous allons d'abord commencer par lire un fichier en utilisant la commande “more” ou “cat” ou encore “nl” qui affiche également le nombre de lignes. On peut ajouter l'option “-n” devant cat pour afficher le nombre de lignes.

```
valux@debian:~/Documents$ more 'projet shell'
```

```
valux@debian:~/Documents$ cat 'projet shell'
```

```
valux@debian:~/Documents$ nl 'projet shell'
```



Après avoir entré la commande, on devrait se retrouver avec le contenu de notre fichier.

```
valux@debian:~/Documents$ cat 'projet shell'
Salut je m'appelle Valentin et je rentre des commandes sur Linux.
Je vais donc écrire n'importe quoi pour avoir le plus de lignes possibles.
a
b
c
d
e
f
g
h
i
j
k
l
m
n
o
p
q
r
s
t
u
v
w
x
y
z
```

```
valux@debian:~/Documents$ nl 'projet shell'
 1 Salut je m'appelle Valentin et je rentre des commandes sur Linux.
 2 Je vais donc écrire n'importe quoi pour avoir le plus de lignes possibles.
 3 a
 4 b
 5 c
 6 d
 7 e
 8 f
 9 g
10 h
11 i
12 j
13 k
14 l
15 m
16 n
17 o
18 p
19 q
20 r
21 s
22 t
23 u
24 v
25 w
26 x
27 y
28 z
```



Pour lire les 10 premières lignes du fichier “.bashrc”, nous allons rentrer la commande “head .bashrc”.

```
valux@debian:~$ head ".bashrc"
```

Le terminal va donc nous afficher les 10 premières lignes du fichier à savoir :

```
valux@debian:~$ head ".bashrc"
# ~/.bashrc: executed by bash(1) for non-login shells.
# see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)
# for examples

# If not running interactively, don't do anything
case $- in
    *i*) ;;
    *) return;;
esac
```



Pour lire les 10 dernières lignes du fichier “.bashrc”, nous allons rentrer la commande “tail .bashrc”.

```
valux@debian:~$ tail .bashrc
```

Le terminal va donc nous afficher les 10 dernières lignes du fichier à savoir :

```
valux@debian:~$ tail .bashrc
# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
    if [ -f /usr/share/bash-completion/bash_completion ]; then
        . /usr/share/bash-completion/bash_completion
    elif [ -f /etc/bash_completion ]; then
        . /etc/bash_completion
    fi
fi
```



Pour lire les 20 premières lignes du fichier “bashrc”, nous allons rentrer la commande “head -20 .bashrc”

```
valux@debian:~$ head -20 .bashrc
```

Le terminal va donc nous afficher les 10 dernières lignes du fichier à savoir :

```
valux@debian:~$ head -20 .bashrc
# ~/.bashrc: executed by bash(1) for non-login shells.
# see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)
# for examples

# If not running interactively, don't do anything
case $- in
    *i*) ;;
    *) return;;
esac

# don't put duplicate lines or lines starting with space in the history.
# See bash(1) for more options
HISTCONTROL=ignoreboth

# append to the history file, don't overwrite it
shopt -s histappend

# for setting history length see HISTSIZE and HISTFILESIZE in bash(1)
HISTSIZE=1000
HISTFILESIZE=2000
```



Pour lire les 20 dernières lignes du fichier “bashrc”, nous allons rentrer la commande “tail -20 .bashrc”

```
valux@debian:~$ tail -20 .bashrc
```

Le terminal va donc nous afficher les 10 dernières lignes du fichier à savoir :

```
valux@debian:~$ tail -20 .bashrc

# Alias definitions.
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
    if [ -f /usr/share/bash-completion/bash_completion ]; then
        . /usr/share/bash-completion/bash_completion
    elif [ -f /etc/bash_completion ]; then
        . /etc/bash_completion
    fi
fi
```



Nous allons à présent réaliser différentes actions via notre terminal Linux.

Nous allons dans un premier temps installer le paquet “cmatrix”. Pour cela nous entrons la commande “sudo apt-get install cmatrix”.

```
valux@debian:~$ sudo apt-get install cmatrix
```

Cmatrix va se télécharger et s'installer.

```
valux@debian:~$ sudo apt-get install cmatrix
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Paquets suggérés :
  cmatrix-xfont
Les NOUVEAUX paquets suivants seront installés :
  cmatrix
0 mis à jour, 1 nouvellement installés, 0 à enlever et 3 non mis à jour.
Il est nécessaire de prendre 17,5 ko dans les archives.
Après cette opération, 53,2 ko d'espace disque supplémentaires seront utilisés.
Réception de :1 http://deb.debian.org/debian bookworm/main amd64 cmatrix amd64 2.0-3 [17,5 kB]
17,5 ko réceptionnés en 0s (390 ko/s)
Sélection du paquet cmatrix précédemment désélectionné.
(Lecture de la base de données... 154904 fichiers et répertoires déjà installés.)
Préparation du dépaquetage de .../cmatrix_2.0-3_amd64.deb ...
Dépaquetage de cmatrix (2.0-3) ...
Paramétrage de cmatrix (2.0-3) ...
Traitement des actions différées (« triggers ») pour mailcap (3.70+nmu1) ...
Traitement des actions différées (« triggers ») pour desktop-file-utils (0.26-1) ...
Traitement des actions différées (« triggers ») pour gnome-menus (3.36.0-1.1) ...
Traitement des actions différées (« triggers ») pour man-db (2.11.2-2) ...
```



Nous allons ensuite lancer Cmatrix en écrivant tout simplement “cmatrix” dans le terminal.

valux@debian: ~\$ cmatrix

```
Activités Terminal 27 sept. 13:54 valux@debian: ~ valux@debian: ~
p H T c / 7 S b j 0 > @ 6 * 1 K d 3 O % b N 9 i
f F f w X t P D \ # x c 9 & E < c P 2 M ? ^
` 2 m 9 A = M Z # - F < q M X 4 d p L k B 2 A
, V N I = ` Z g J V [ 3 4 ) z 2 F h Y T " W ; 9 :
- * d 6 3 T z b c 9 ) < P 3 i U ) I q W F # Q k )
P x - J 0 Q V v o K p _ : x [ 1 ) N H G d , % &
- m i c s R 0 Q N x n - % - H * . H : d & n p
W 4 , v I w [ S ) F 6 k . A 9 P p g ) j w
* S V < 0 S T c P & ] # g , ? B t o v
e J Q - T \ / T & A j 4 0 Z D L t U ) \ @
H \ b - ) H Y \ 9 E 0 \ ; * ( Z 3 Y q p " 0 S
" % S ) F t G o ( v * x i U 7 w ; 6 V Q y 1
$ 9 0 g h K % , - 0 5 g x ) / [ < ' e z h 9
W P > o W i ! 5 $ 7 H " j 2 m " = e s d & j
H u . K x @ z ! \ N : 2 0 P B \ 3 4 n u z ' !
v q Z : & 4 f ' h ] g i E U ^ J 4 Z u ) ) W d z
/ U W V S 8 d a ^ . l J ? m p b C s M T d # t a 0 F G [
3 \ c D s f ] - v t n v ^ g _ Y 1 l V P w E x
H / x 9 2 F F J 9 + b d c J Y o p / I B $ 6
L H 2 t d Q 7 c > 8 M " d v a c - p . l j
* 4 8 5 ; 4 K - \ 6 N L o i i ] D 0 . X k
F u r Z = W / p g b A S * c t g f c d V 9 > e
n e l - i ^ 3 W t & J 8 > y V i o h c . - F ( w
) i e N j w g 5 y u : & . + L 5 ' Q i = A & % s ) )
# N x D I s R s + & \ U " j = d g S y ? Z H p 6 & >
d W G & L I ? * J $ J * ? M S ^ o 4 K G $ t E - s d
I G ^ 2 e i M j @ F o y g P F j o t 3 ] < ;
, V & u q v ^ v o 5 [ x 5 7 c 3 Q ' ^ J : 0
s n E ^ K W < 0 4 Q U l G b a M 9 k s > $ M @ z d
d i 1 % 5 W & 2 v O G V 4 8 u 6 D r - Z l 1 < # Z =
# 3 x % R I X ) # s ) ? I + X , Q # L ( l l P x 9 v Y x
3 o F # 3 0 t + H e u c . ^ 3 8 + 0 > o 1 o o : N > x 6
! x " \ G t B ) 3 J 0 5 < 4 u K d w J ; P , T l 5 0 )
p x c 5 - 6 D & 7 j 3 G P , W S a r C < x 8 > o a j 0
B @ 7 Q - X X ! @ = [ y S a ) & i V 9 P ' T r z Q ; i 0
% * 9 e 5 c ! T 4 H 7 ) 8 y ) % X H B [ o Z a ; R 5 / @
# L 3 : # x n m 1 H . , : ? " U , f X z ; d $ Z e l R B . ( C
X x J 1 u ) v 4 @ / 8 H q X < K d J ) 8 E m P & + & s 8 Q x 3
& Y @ o x : Z + P ? Q T S 3 V R ? = j 8 X l > E & i U G z 4 % ' i
W z g M H B $ l h R Q F ( * L 3 - 6 J Y w 6 b r ' P v " i S y #
I 8 0 V j P > [ B n > [ N " d I ; V U p 9 : ) x 7 D 8 6 + *
N e 9 % E x $ : ' 4 H # . / i # G v o / 9 " 5 8 ) : o ' Z :
```



Pour mettre à jour les paquets, nous entrons la formule “sudo apt update”

```
valux@debian:~$ sudo apt update
```

```
valux@debian:~$ sudo apt update
Atteint :1 http://security.debian.org/debian-security bookworm-security InRelease
Atteint :2 http://deb.debian.org/debian bookworm InRelease
Atteint :3 http://deb.debian.org/debian bookworm-updates InRelease
Atteint :4 https://dl.google.com/linux/chrome/deb stable InRelease
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
4 paquets peuvent être mis à jour. Exécutez « apt list --upgradable » pour les voir.
```



Nous allons en profiter pour mettre à jour nos logiciels. Pour cela nous rentrons la commande ‘sudo apt upgrade’

```
valux@debian:~$ sudo apt upgrade
```

```
valux@debian:~$ sudo apt upgrade
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Calcul de la mise à jour... Fait
Les paquets suivants seront mis à jour :
  bind9-dnsutils bind9-host bind9-libs google-chrome-stable
4 mis à jour, 0 nouvellement installés, 0 à enlever et 0 non mis à jour.
Il est nécessaire de prendre 0 o/106 Mo dans les archives.
Après cette opération, 5 241 ko d'espace disque seront libérés.
Souhaitez-vous continuer ? [0/n] o
Lecture des fichiers de modifications (« changelog »)... Terminé
```



La prochaine étape consiste à télécharger les internets, ce qui revient à entrer la commande “wget google.fr”

valux@debian:~\$ wget google.fr

```
valux@debian:~$ wget google.fr
--2023-09-27 14:28:46-- http://google.fr/
Résolution de google.fr (google.fr)... 172.217.171.227, 2a00:1450:4006:802::2003
Connexion à google.fr (google.fr)|172.217.171.227|:80... connecté.
requête HTTP transmise, en attente de la réponse... 301 Moved Permanently
Emplacement : http://www.google.fr/ [suivant]
--2023-09-27 14:28:46-- http://www.google.fr/
Résolution de www.google.fr (www.google.fr)... 172.217.19.35, 2a00:1450:4006:802::2003
Connexion à www.google.fr (www.google.fr)|172.217.19.35|:80... connecté.
requête HTTP transmise, en attente de la réponse... 200 OK
Taille : non indiqué [text/html]
Sauvegarde en : « index.html »

index.html [ <=> ] 23,58K --.-KB/s ds 0,004s

2023-09-27 14:28:46 (5,20 MB/s) - « index.html » sauvegardé [24150]
```



Pour redémarrer la machine via le terminal, il suffit d'entrer la commande “reboot” et la machine redémarrera.

```
valux@debian:~$ reboot
```

Si nous souhaitons éteindre la machine via le terminal, on rentrera alors la commande “sudo halt”.

```
valux@debian:~$ sudo halt
```

```
Broadcast message from root@debian on pts/2 (Wed 2023-09-27 14:36:48 CEST):  
The system will halt now!
```

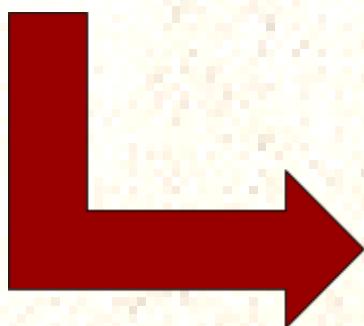


Nous allons continuer à réaliser différentes actions sur le terminal Linux

Nous allons tout d'abord créer un groupe appelé “Plateformeurs”. Pour cela, nous entrons la commande “`sudo groupadd nom du groupe`”.

```
valux@debian:~$ groupadd kgb
```

Pour vérifier que le groupe a bien été créé, on entre la commande “`less /etc/group`”. Si le nom du groupe s'affiche dans la liste, c'est que le groupe a bien été créé.



```
list:x:38:  
irc:x:39:  
src:x:40:  
shadow:x:42:  
utmp:x:43:  
video:x:44:valux  
sasl:x:45:  
plugdev:x:46:valux  
staff:x:50:  
games:x:60:  
users:x:100:valux, valix  
nogroup:x:65534:  
systemd-journal:x:999:  
systemd-network:x:998:  
crontab:x:101:  
input:x:102:  
sgx:x:103:  
kvm:x:104:  
render:x:105:  
netdev:x:106:valux  
tss:x:107:  
systemd-timesync:x:997:  
messagebus:x:108:  
_ssh:x:109:  
ssl-cert:x:110:  
bluetooth:x:111:valux  
avahi:x:112:  
lpadmin:x:113:valux  
pipewire:x:114:  
fwupd-refresh:x:115:  
scanner:x:116:saned, valux  
saned:x:117:  
geoclue:x:118:  
polkitd:x:996:  
rtkit:x:119:  
colord:x:120:  
Debian-gdm:x:121:  
valux:x:1000:  
gnome-initial-setup:x:995:  
valix:x:1001:  
User1:x:1003:  
User2:x:1004:  
Plateformeurs:x:1002:User1
```



Nous allons à présent ajouter un nouvel utilisateur grâce à la commande “sudo useradd User”

```
valux@debian:~$ sudo useradd User1
```

Pour vérifier que le compte utilisateur s'est bien ajouté on entre la commande “sudo cat /etc/passwd”

```
valux@debian:~$ sudo cat /etc/passwd
```

Puis on regarde si le nom de l'utilisateur apparaît.

```
valux:x:1000:1000:Valux sur Linux,,,,:/home/valux:/bin/bash
valix:x:1001:1001:valix,,,,:/home/valix:/bin/bash
User1:x:1002:1003:::/home/User1:/bin/sh ←
User2:x:1003:1004:::/home/User2:/bin/sh ←
valux@debian:~$
```

Nous avons en parallèle créé le fichier “users.txt” avec la commande “nano users.txt”. Puis nous avons ajouté “User1” et “User2” séparé d'un retour à la ligne avec la commande suivante “cat >users.txt”.



Nous allons maintenant ajouter l'User2 au groupe "Plateformeurs" grâce à la commande "sudo usermod -G Plateformeurs User2"

```
valux@debian:~$ sudo usermod -G Plateformeurs User2
```

On peut vérifier le contenu du groupe en entrant la commande "getent group Plateformeurs"

```
valux@debian:~$ getent group Plateformeurs  
Plateformeurs:x:1002:User2
```

Nous allons à présent copier notre "users.txt" dans un fichier "droits.txt" en entrant la commande "cat users.txt > droits.txt"

```
valux@debian:~$ cat users.txt > droits.txt
```

```
valux@debian:~$ cat droits.txt  
User1  
User2
```



Pour vérifier que le fichier s'est bien copié on entre la commande “cat droits.txt” pour voir le contenu du fichier, si c'est le même que dans le premier.

On copie le contenu du fichier “users.txt” dans le fichier “groupes.txt”, on répète l'opération en utilisant la même commande.

```
valux@debian:~$ cat users.txt > droits.txt
```

Nous allons à présent changer le propriétaire du fichier “droits.txt”. Pour savoir qui est le propriétaire actuel on entre la commande “ls -l droits.txt”.

```
valux@debian:~$ ls -l droits.txt  
-rw-r--r-- 1 User1 valux 12 27 sept. 15:55 droits.txt
```

```
valux@debian:~$ cat droits.txt  
User1  
User2
```

Pour modifier les droits sur les fichiers, nous allons rentrer la commande “sudo chmod -- nom du fichier”. Les tirets sont en fait des lettres qui influencent les permissions accordées.

```
valux@debian:~$ ls -l droits.txt  
-rw-r--r-- 1 valux valux 12 27 sept. 15:55 droits.txt
```



Chaque lettre désigne un utilisateur, un groupe, le droit de lire, écrire, exécuter etc...

Voici un tableau résumant la signification de ces lettres.

Lettres pour les droits d'accès	Signification
r	Droit de lecture ; aussi indiqué R-Bit
w	Droit d'écriture ; aussi indiqué W-Bit
x	Droit d'exécution ; aussi indiqué X-Bit
Lettres pour les catégories d'utilisateurs	Signification
u	Catégorie d'utilisateur « user », propriétaire
g	Catégorie d'utilisateur « group », groupe
o	Catégorie d'utilisateur « others », autres utilisateurs
a	« all » : la commande concerne toutes les catégories d'utilisateurs.



Par exemple, si je rentre la commande suivante “sudo chmod u+rwx droits.txt” cela signifie que la catégorie utilisateur obtient le droit de lire, écrire et d'exécuter le fichier. Le “+” dans la formule signifie que l'on ajoute des droits, réciproquement si on met un “-” cela signifie que l'on enlève des droits.

+	L'opérateur « + » est utilisé pour attribuer de nouveaux droits de fichier à une catégorie d'utilisateurs. Seuls les droits des fichiers concernés sont écrasés.
-	L'opérateur « - » supprime un droit de fichier d'une catégorie d'utilisateurs.
=	Si les droits sur les fichiers doivent être définis pour une catégorie d'utilisateurs indépendamment de ses droits antérieurs, l'opérateur « = » est utilisé.



La commande enverra une réponse de ce type :

```
-rw-r----- 1 User1 valux 12 27 sept. 15:55 droits.txt
```

Le premier tiret indique la nature de l'élément, si c'est un fichier, un répertoire, un dossier etc... Ici nous avons un fichier (le - le définit). Juste après, nous avons une suite de 9 tirets que nous allons pouvoir séparer en 3 blocs :

- Les 3 premiers tirets désigne le droits du propriétaire
- Les 3 suivants désigne les droits de la catégorie d'utilisateurs groupe
- Les 3 derniers informe les droits de tous les autres utilisateurs

Par exemple, sur la commande ci-dessus, nous avons un fichier (-), le propriétaire a le droit de lire et écrire (rw-), les utilisateurs groupe (y compris User2) ont uniquement le droit de lecture et les autres utilisateurs n'ont aucun droit sur le fichier “droits.txt”.



Pour changer les droits du fichier “groupes.txt” afin que les utilisateurs puissent accéder au fichier en lecture uniquement, on entre la commande “sudo chmod o+r groupes.txt”.

```
valux@debian:~$ sudo chmod o+r groupes.txt  
valux@debian:~$ ls -l groupes.txt  
-rwx---r-- 1 valux valux 12 27 sept. 16:10 groupes.txt
```

Nous allons à présent changer les droits du fichier pour que le groupe “Plateformeurs” puisse y accéder en lecture/écriture. On entre donc la commande “sudo chgrp Plateformeurs groupes.txt”.

```
valux@debian:~$ sudo chgrp Plateformeurs groupes.txt
```

```
valux@debian:~$ ls -l groupes.txt  
-rwxrw-r-- 1 valux Plateformeurs 12 27 sept. 16:10 groupes.txt
```



Pour ajouter un alias on entre la commande alias *la commande* = “une autre commande”
soit → alias la=”ls -la” permet d’entrer la commande.
“ls -la” en entrant la commande “la”

```
valux@debian:~$ alias la="ls -la"
valux@debian:~$ la
total 136
drwx----- 17 valux valux          4096 27 sept. 16:10 .
drwxr-xr-x  4 root  root          4096 21 sept. 18:50 ..
-rw-----  1 valux valux         5664 28 sept. 10:14 .bash_history
-rw-r--r--  1 valux valux         220 20 sept. 18:17 .bash_logout
-rw-r--r--  1 valux valux        3526 20 sept. 18:17 .bashrc
drwxr-xr-x  2 valux valux          4096 20 sept. 18:25 Bureau
drwx----- 13 valux valux          4096 27 sept. 10:41 .cache
drwxr-xr-x 13 valux valux          4096 27 sept. 14:29 .config
drwxr-xr-x  2 valux valux          4096 27 sept. 11:38 Documents
-rw-r----  1 User1 valux          12 27 sept. 15:55 droits.txt
-rw-r--r--  1 valux valux         5290 20 sept. 18:17 .face
lrwxrwxrwx  1 valux valux             5 20 sept. 18:17 .face.icon -> .face
drwx-----  3 valux valux          4096 27 sept. 14:29 .gnome
drwx-----  2 valux valux          4096 20 sept. 18:27 .gnupg
-rwxrw-r--  1 valux Plateformeurs      12 27 sept. 16:10 groupes.txt
drwxr-xr-x  2 valux valux          4096 20 sept. 18:25 Images
-rw-r--r--  1 valux valux        24150 27 sept. 14:28 index.html
-rw-----  1 valux valux             20 27 sept. 15:45 .lessht
drwx-----  4 valux valux          4096 20 sept. 18:25 .local
drwxr-xr-x  2 valux valux          4096 20 sept. 18:25 Modèles
drwxr-xr-x  2 valux valux          4096 20 sept. 18:50 Musique
drwx-----  3 valux valux          4096 25 sept. 08:39 .pki
-rw-r--r--  1 valux valux             807 20 sept. 18:17 .profile
drwxr-xr-x  2 valux valux          4096 20 sept. 18:25 Public
drwx-----  2 valux valux          4096 20 sept. 18:27 .ssh
-rw-r--r--  1 valux valux              0 27 sept. 13:46 .sudo_as_admin_successful
drwxr-xr-x  2 valux valux          4096 20 sept. 18:25 Téléchargements
-rw-r--r--  1 valux valux             12 27 sept. 15:18 users.txt
drwxr-xr-x  2 valux valux          4096 20 sept. 18:25 Vidéos
```



On répète la même opération pour la commande “apt-get update” pour la lancer avec “update” à la différence qu'on ajoutera “sudo” car c'est une commande demandant des permissions. On a donc alias update=”sudo apt-get update”.

```
valux@debian:~$ alias update="sudo apt-get update"
valux@debian:~$ update
Atteint :1 http://deb.debian.org/debian bookworm InRelease
Atteint :2 http://security.debian.org/debian-security bookworm-security InRelease
Atteint :3 http://deb.debian.org/debian bookworm-updates InRelease
Atteint :4 https://dl.google.com/linux/chrome/deb stable InRelease
Lecture des listes de paquets... Fait
```

On répète exactement la même opération pour la commande “upgrade” en entrant la commande alias upgrade=”sudo apt-get upgrade”.

```
valux@debian:~$ alias upgrade="sudo apt-get upgrade"
valux@debian:~$ upgrade
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Calcul de la mise à jour... Fait
Les paquets suivants seront mis à jour :
  google-chrome-stable
  1 mis à jour, 0 nouvellement installés, 0 à enlever et 0 non mis à jour.
Il est nécessaire de prendre 0 o/104 Mo dans les archives.
Après cette opération, 7 293 ko d'espace disque supplémentaires seront utilisés.
Souhaitez-vous continuer ? [0/n] █
```



Pour ajouter une variables d'environnement qui se nommera USER et qui sera égale à notre nom d'utilisateur, on entre la commande “export USER=valux”.

```
valux@debian:~$ export USER=valux
```

```
valux@debian:~$ echo $USER  
valux
```

Maintenant, nous souhaitons garder nos alias et notre variable pour nos futures connexions. A savoir que tant que ces commandes ne sont pas inscrites dans le bashrc, elles disparaîtront lors de la mise hors tension de l'ordinateur. Nous nous rendons donc dans le .bashrc en entrant la commande “nano .bashrc” afin de pouvoir modifier le contenu.

On entre nos alias et variables que l'on souhaite garder pour chaque connexion et on enregistre.



On entre donc les commandes précédemment introduites

```
# colored GCC warnings and errors
#export GCC_COLORS='error=01;31:warning=01;35:note=01;36:caret=01;32:locus=01:quote=01'

# some more ls aliases
#alias ll='ls -l'
#alias la='ls -A'
#alias l='ls -CF'

# Alias definitions.
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
    if [ -f /usr/share/bash-completion/bash_completion ]; then
        . /usr/share/bash-completion/bash_completion
    elif [ -f /etc/bash_completion ]; then
        . /etc/bash_completion
    fi
fi
alias la="la -la"
alias update="sudo apt-get update"
alias upgrade="sudo apt-get upgrade"
export USER=valux
```



On va maintenant afficher les variables d'environnement de notre Linux en entrant la commande “printenv”

```
valux@debian:~$ printenv
SHELL=/bin/bash
SESSION_MANAGER=local/debian:@/tmp/.ICE-unix/1423,unix/debian:/tmp/.ICE-unix/1423
QT_ACCESSIBILITY=1
COLORTERM=truecolor
SSH_AGENT_LAUNCHER=openssh
XDG_MENU_PREFIX=gnome-
GNOME_DESKTOP_SESSION_ID=this-is-deprecated
SSH_AUTH_SOCK=/run/user/1000/keyring/ssh
XMODIFIERS=@im=ibus
DESKTOP_SESSION=gnome
GTK_MODULES=gail:atk-bridge
PWD=/home/valux
LOGNAME=valux
XDG_SESSION_DESKTOP=gnome
XDG_SESSION_TYPE=wayland
SYSTEMD_EXEC_PID=1446
XAUTHORITY=/run/user/1000/.mutter-Xwaylandauth.E020B2
GDM_LANG=fr_FR.UTF-8
HOME=/home/valux
USERNAME=valux
IM_CONFIG_PHASE=1
LANG=fr_FR.UTF-8
```



Pour ajouter à notre PATH le chemin “/home/’votre utilisateur’/Bureau on entre la commande “PATH=\$PATH:~/Bureau”

```
valux@debian:~$ PATH=$PATH:~/Bureau
```

On vérifie en entre la commande “echo \$PATH”

```
valux@debian:~$ echo $PATH  
/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games:/home/valux/Bureau
```



Nous allons maintenant décompresser un fichier .tar que nous allons télécharger. Le fichier “Ghost in the Shell.tar” va pouvoir être décompressé grâce à la commande tar -zxvf “Ghost in the Shell.tar”

```
valux@debian:~/Téléchargements$ tar -zxvf "Ghost in the Shell.tar"  
.Ghost in the Shell.pdf  
tar: Le mot clé inconnu « LIBARCHIVE.xattr.com.apple.quarantine » pour l'en-tête  
étendu a été ignoré  
tar: Le mot clé inconnu « LIBARCHIVE.xattr.com.apple.provenance » pour l'en-tête  
étendu a été ignoré  
tar: Le mot clé inconnu « LIBARCHIVE.xattr.com.apple.macl » pour l'en-tête étend  
u a été ignoré  
Ghost in the Shell.pdf
```



Nous allons maintenant entrer une commande qui contiendra plusieurs actions. Le but étant de créer, compter, enregistrer, afficher et rechercher dans la même commande en utilisant les points virgules et le pype (|).

Nous rentrons alors la commande suivante : echo "Je suis votre fichier texte" > une_commande.txt ; cat /etc/apt/sources.list | wc -l > nb_lignes.txt ; cat /etc/apt/sources.list | tee save_sources ; grep -r --include=".*" "alias"

```
valux@debian:~$ echo "Je suis votre fichier texte" > une_commande.txt ; cat /etc/apt/sources.list | wc -l > nb_lignes.txt ; cat /etc/apt/sources.list | tee save_sources ; grep -r --include=".*" "alias"
```



Voilà la réponse de notre terminal

```
valuedebian:~$ echo "Je suis votre fichier texte" > une_commande.txt ; cat /etc/apt/sources.list | wc -l > nb_lignes.txt ; cat /etc/apt/sources.list | tee save_sources ; grep -r --include=".*" "alias"
#deb cdrom:[Debian GNU/Linux 12.1.0 _Bookworm_ - Official amd64 NETINST with firmware 20230722-10:48]/ bookworm main non-free-firmware

deb http://deb.debian.org/debian/ bookworm main non-free-firmware
deb-src http://deb.debian.org/debian/ bookworm main non-free-firmware

deb http://security.debian.org/debian-security bookworm-security main non-free-firmware
deb-src http://security.debian.org/debian-security bookworm-security main non-free-firmware

# bookworm-updates, to get updates before a point release is made;
# see https://www.debian.org/doc/manuals/debian-reference/ch02.en.html#_updates_and_backports
deb http://deb.debian.org/debian/ bookworm-updates main non-free-firmware
deb-src http://deb.debian.org/debian/ bookworm-updates main non-free-firmware

# This system was installed using small removable media
# (e.g. netinst, live or single CD). The matching "deb cdrom"
# entries were disabled at the end of the installation process.
# For information about how to configure apt package sources,
# see the sources.list(5) manual.
.bashrc:# enable color support of ls and also add handy aliases
.bashrc: alias ls='ls --color=auto'
.bashrc: alias dir='dir --color=auto'
.bashrc: alias vdir='vdir --color=auto'
.bashrc: alias grep='grep --color=auto'
.bashrc: alias fgrep='fgrep --color=auto'
.bashrc: alias egrep='egrep --color=auto'
.bashrc:# some more ls aliases
.bashrc:alias ll='ls -l'
.bashrc:alias la='ls -A'
.bashrc:alias l='ls -CF'
.bashrc:# ~/.bash_aliases, instead of adding them here directly.
.bashrc:if [ -f ~/.bash_aliases ]; then
.bashrc: . ~/.bash_aliases
.bashrc:alias la="la -la"
.bashrc:alias update="sudo apt-get update"
.bashrc:alias upgrade="sudo apt-get upgrade"
.bash_history:alias
.bash_history:alias la=ls-la
.bash_history:alias la="ls - la"
.bash_history:alias la="ls -la"
.bash_history:alias update="apt-get update"
.bash_history:alias update="sudo apt-get update"
.bash_history:alias upgrade="sudo apt-get upgrade"
```



& = Le & sert effectivement à lancer une commande en arrière plan. Ça veut dire que le shell lance la commande et n'attend pas qu'elle se termine. Le & est interprété par le shell.

&& = Il est possible d'assembler 2 conditions entre-elles pour former une condition plus complexe qui n'est valide que si les deux sous-conditions sont vérifiées.

; = Un enchaînement de commande avec un point virgule comme c'est le cas dans l'exemple enchaînera les commandes sans se soucier de la réussite ou de l'échec de chacune

|| = permet de faire la même chose que && mais seulement si au moins une des conditions est correcte



Nous allons entrer une nouvelle commande qui contiendra plusieurs actions. Le but étant d'installer, lancer, enregistrer, lister, compter et upgrade dans la même commande en utilisant les points virgules et le pype (|).



On entre la commande suivante : sudo apt-get install tree || tree > tree.save && cat tree.save ; wc -l tree.save && update ; upgrade



Voila la réponse du terminal

```
valux@debian:~$ sudo apt-get install tree || tree > tree.save && cat tree.save ; wc -l tree.save && update ; upgrade
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
tree est déjà la version la plus récente (2.1.0-1).
0 mis à jour, 0 nouvellement installés, 0 à enlever et 0 non mis à jour.

.
├── Bureau
├── Documents
│   └── projet shell
├── droits.txt
├── groupes.txt
├── Images
├── index.html
├── install.log
├── Modèles
├── Musique
├── nb_lignes.txt
├── Public
├── save_sources
├── Téléchargements
│   ├── Ghost in the Shell.pdf
│   └── Ghost in the Shell.tar
├── tree.save
├── une_commande.txt
└── users.txt
└── Vidéos

9 directories, 12 files
23 tree.save
Atteint :1 http://security.debian.org/debian-security bookworm-security InRelease
Atteint :2 http://deb.debian.org/debian bookworm InRelease
Atteint :3 http://deb.debian.org/debian bookworm-updates InRelease
Atteint :4 https://dl.google.com/linux/chrome/deb stable InRelease
Lecture des listes de paquets... Fait
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Calcul de la mise à jour... Fait
0 mis à jour, 0 nouvellement installés, 0 à enlever et 0 non mis à jour.
```