



```
(kali㉿kali)-[~]
$ ls -a
.          .face.icon      Templates
..         .gnupg          .vboxclient-clipboard.pid
.bash_history .ICEauthority  .vboxclient-display-svg-x11.pid
.bash_logout .java          .vboxclient-draganddrop.pid
.bashrc      .lessht        .vboxclient-seamless.pid
.bashrc.original .local
.cache       .mozilla
.config      Music
Desktop      Pictures
.dmrc        .pki
Documents    .profile
Downloads    Public
.face        .sudo_as_admin_successful
```

# SHELL BASH PRÉSENTATION

LIGNE DE COMMANDE LINUX

Pour afficher les fichiers et répertoires cachés, plus les informations sur les droits sous forme de liste on utilise la commande **ls -la**

```
(kali@kali)-[~]
$ ls -la
total 168
drwxr-xr-x 17 kali kali 4096 Oct 3 17:34 .
drwxr-xr-x 3 root root 4096 Aug 8 06:10 ..
-rw-r--r-- 1 kali kali 42 Oct 3 11:03 .bash_history
-rw-r--r-- 1 kali kali 220 Aug 8 06:10 .bash_logout
-rw-r--r-- 1 kali kali 5551 Aug 8 06:10 .bashrc
-rw-r--r-- 1 kali kali 3526 Aug 8 06:10 .bashrc.original
drwxr-xr-x 10 kali kali 4096 Oct 3 10:25 .cache
drwxr-xr-x 13 kali kali 4096 Oct 3 08:52 .config
drwxr-xr-x 2 kali kali 4096 Oct 3 12:21 Desktop
-rw-r--r-- 1 kali kali 35 Oct 3 06:52 .dmrc
drwxr-xr-x 2 kali kali 4096 Oct 3 06:52 Documents
drwxr-xr-x 2 kali kali 4096 Oct 3 10:39 Downloads
-rw-r--r-- 1 kali kali 11759 Aug 8 06:10 .face
lrwxrwxrwx 1 kali kali 5 Aug 8 06:10 .face.icon -> .face
drwxr-xr-x 3 kali kali 4096 Oct 3 06:52 .gnupg
-rw-r--r-- 1 kali kali 0 Oct 3 06:52 .ICEauthority
drwxr-xr-x 3 kali kali 4096 Aug 8 06:10 .java
-rw-r--r-- 1 kali kali 20 Oct 3 17:34 .lessht
drwxr-xr-x 3 kali kali 4096 Oct 3 06:52 .local
drwxr-xr-x 4 kali kali 4096 Oct 3 07:22 .mozilla
drwxr-xr-x 2 kali kali 4096 Oct 3 06:52 Music
drwxr-xr-x 2 kali kali 4096 Oct 3 06:52 Pictures
drwxr-xr-x 3 kali kali 4096 Oct 3 10:25 .pki
-rw-r--r-- 1 kali kali 807 Aug 8 06:10 .profile
```

On peut rajouter des options dans les commandes en faisant  
**[commande] -[option]** ou **[commande] --[option\_longue]**  
Exemple de la commande =

**man -h ↓**

**man --help ↓**

```
(kali@kali)-[~]
$ man -h
Usage: man [OPTION...] [SECTION] PAGE...

-C, --config-file=FILE    use this user configuration file
-d, --debug               emit debugging messages
-D, --default              reset all options to their default values
--warnings[=WARNINGS]    enable warnings from groff

Main modes of operation:
-f, --whatis               equivalent to whatis
-k, --apropos              equivalent to apropos
-K, --global-apropos      search for text in all pages
-l, --local-file           interpret PAGE argument(s) as local filename(s)
-w, --where, --path, --location
                           print physical location of man page(s)
-W, --where-cat, --location-cat
                           print physical location of cat file(s)

-c, --catman               used by catman to reformat out of date cat pages
-R, --recode=ENCODING      output source page encoded in ENCODING

Finding manual pages:
-L, --locale=LOCALE        define the locale for this particular man search
-m, --systems=SYSTEM       use manual pages from other systems
-M, --manpath=PATH         set search path for manual pages to PATH
```

```
(kali@kali)-[~]
$ man --help
Usage: man [OPTION...] [SECTION] PAGE...

-C, --config-file=FILE    use this user configuration file
-d, --debug               emit debugging messages
-D, --default              reset all options to their default values
--warnings[=WARNINGS]    enable warnings from groff

Main modes of operation:
-f, --whatis               equivalent to whatis
-k, --apropos              equivalent to apropos
-K, --global-apropos      search for text in all pages
-l, --local-file           interpret PAGE argument(s) as local filename(s)
-w, --where, --path, --location
                           print physical location of man page(s)
-W, --where-cat, --location-cat
                           print physical location of cat file(s)

-c, --catman               used by catman to reformat out of date cat pages
-R, --recode=ENCODING      output source page encoded in ENCODING

Finding manual pages:
-L, --locale=LOCALE        define the locale for this particular man search
-m, --systems=SYSTEM       use manual pages from other systems
-M, --manpath=PATH         set search path for manual pages to PATH
```

Ce sont les deux syntaxes principales d'écriture des options pour une commande ?

# SHELL BASH PRÉSENTATION

LIGNE DE COMMANDE LINUX



## JOB 2

Les commandes **head** et **tail** sont très utiles. Avec ces deux commandes on peut avoir les premières lignes et dernières lignes de nos choix.

Exemple des commandes =

### **head -n 10 .bashrc**

(affiche les 10 premières  
ligne du fichier .bashrc)

```
(kali@kali)-[~]
$ head -n 10 .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
```

### **tail -n 10 .bashrc**

(affiche les 10 dernières  
lignes du fichier .bashrc)

```
(kali@kali)-[~]
$ tail -n 10 .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
```

### **head -n 20 .bashrc**

(affiche les 20 premières  
ligne du fichier .bashrc)

```
(kali@kali)-[~]
$ head -n 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
```

### **tail -n 20 .bashrc**

(affiche les 20 dernières  
lignes du fichier .bashrc)

```
(kali@kali)-[~]
$ tail -n 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
```

On utilisera **head** pour les premières lignes et **tail** pour les dernières puis on rajoute l'option **-n** puis le **chiffre** qui correspond au nombres de lignes qu'on souhaite. Après tout cela, on met le **nom du fichier** que l'on veut.

# SHELL BASH PRÉSENTATION

LIGNE DE COMMANDE LINUX

## JOB 3

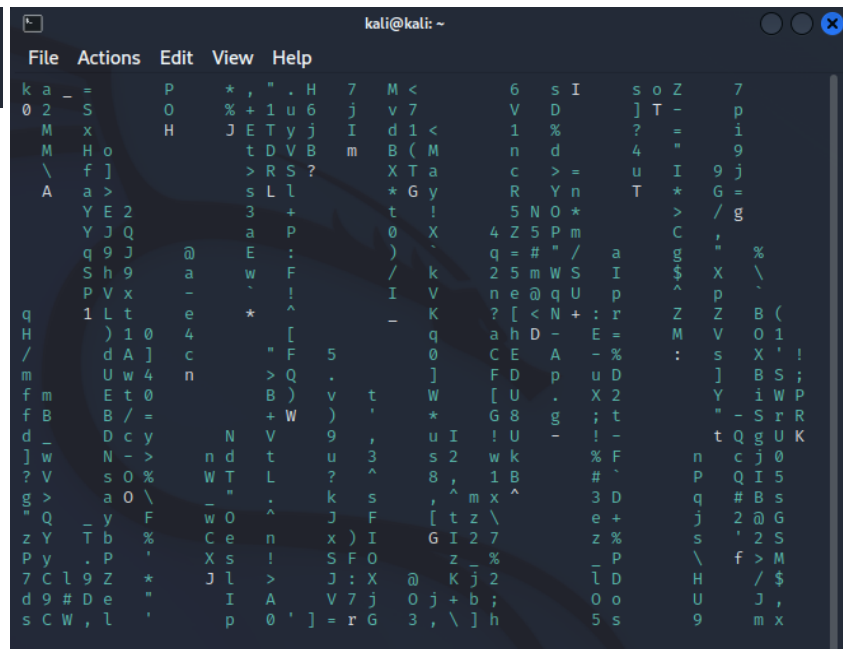
-Pour installer le paquet **cmatrix** on va utiliser la commande **sudo apt-get install cmatrix**

```
(kali㉿kali)-[~]  
$ sudo apt-get install cmatrix
```

-Pour le lancer le paquet **cmatrix** on utilisera la commande **cmatrix**

```
(kali㉿kali)-[~]  
$ cmatrix
```

cmatrix lancé →



-Pour mettre à jour son gestionnaire de paquet on utilise la commande

-Pour mettre à jour ses logiciels on utilise d'abord la commande **sudo apt update** pour **lister à la machine les mise à jour disponible**. Après on utilise la commande **sudo apt upgrade** pour **faire les mises à jour disponible**. On peut faire les deux commandes sur une seule ligne en faisant en sorte que si la première commande ne fonctionne pas, la deuxième aussi. Cette commande c'est **sudo apt update && sudo apt upgrade**

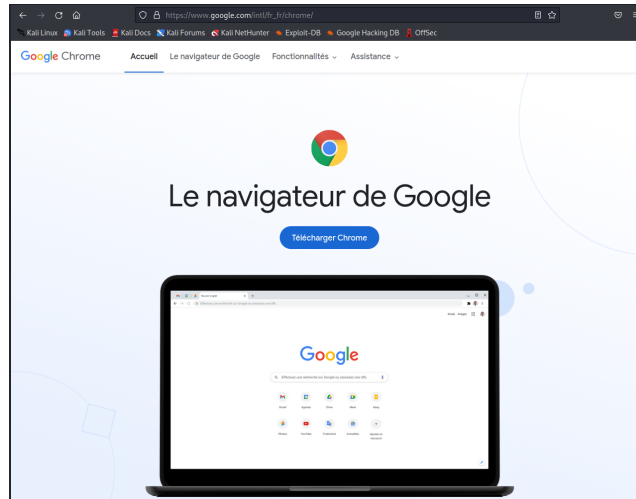
```
(kali㉿kali)-[~]  
$ sudo apt update && sudo apt upgrade
```

# SHELL BASH PRÉSENTATION

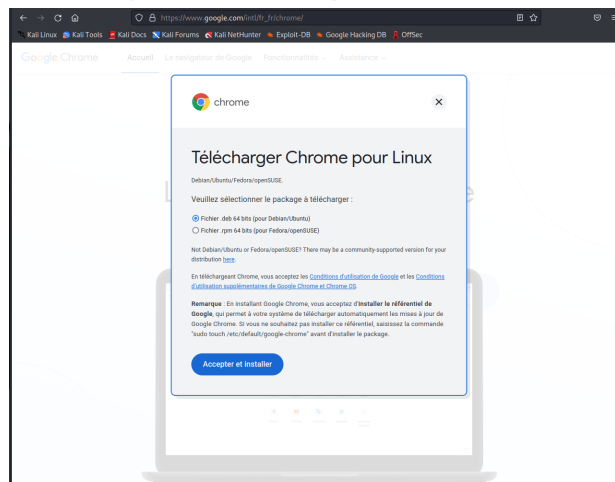
LIGNE DE COMMANDE LINUX

-Pour télécharger **Google Chrome**, on va sur firefox (installer predefinement sur la machine) puis on va sur le site

[https://www.google.com/intl/fr\\_fr/chrome/](https://www.google.com/intl/fr_fr/chrome/)



On clique sur **“Télécharger Chrome”** puis on choisit le package selon son **système**. Après cela on clique sur **“Accepter et installer”**



Une fois le package installé, il faudra installer **gdebi** qui permettra d'installer le paquet .deb de Google Chrome

```
(kali㉿kali)-[~]  
$ sudo apt install gdebi
```





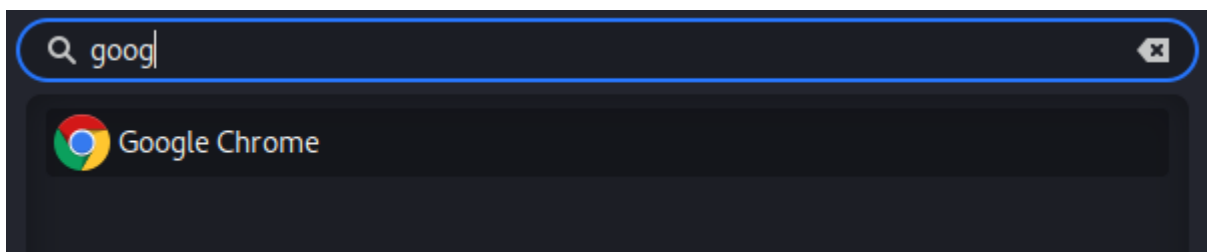
Une fois gdebi installer on va dans le **répertoire de téléchargement** avec le terminal et on cherche le **paquet .deb de Google Chrome**

```
(kali㉿kali)-[~/Downloads]
$ ls
google-chrome-stable_current_amd64.deb
```

Après avoir trouvé le **paquet .deb de Google Chrome** on va l'installer avec **gdebi** et pour cela on va utiliser la commande **sudo gdebi [le paquet .deb de Google Chrome]**

```
(kali㉿kali)-[~/Downloads]
$ sudo gdebi google-chrome-stable_current_amd64.deb
```

**-Maintenant Google Chrome est installé**



-Pour redémarrer la machine avec le terminale on utilise la commande **sudo reboot**

```
(kali㉿kali)-[~]
$ sudo reboot
```

-Pour éteindre la machine avec le terminale on utilise la commande **sudo shutdown [temp]**

```
(kali㉿kali)-[~]
$ sudo shutdown now
```



## JOB 4

Pour créer un fichier texte qui contiendra User1 et User2 séparé par un retour à la ligne, on utilisera la commande **echo -e "User1\nUser2" >users.txt**

```
(kali㉿kali)-[~/Desktop/job4]  
$ echo -e "User1\nUser2" >users.txt
```

-Pour créer un groupe appelé "Plateformeurs" on va utiliser la commande **sudo groupadd Plateformeurs**

```
(kali㉿kali)-[~/Desktop/job4]  
$ sudo groupadd Plateformeurs
```

-Pour créer un utilisateur appelé "User1" on va utiliser la commande **sudo adduser User1**

```
(kali㉿kali)-[~/Desktop/job4]  
$ sudo adduser User1
```

-Pour créer un utilisateur appelé "User2" on va utiliser la commande **sudo adduser User2**

```
(kali㉿kali)-[~/Desktop/job4]  
$ sudo adduser User2
```

-Pour ajouter User2 au groupe Plateformeurs on utilisera la commande **sudo usermod -a -G Plateformeurs User2**

```
(kali㉿kali)-[~/Desktop/job4]  
$ sudo usermod -a -G Plateformeurs User2
```

-Pour copier le fichier **users.txt** dans un fichier **droits.txt** on utilise la commande **cp users.txt droits.txt**

```
(kali㉿kali)-[~/Desktop/job4]  
$ cp users.txt droits.txt
```



-Pour copier le fichier **users.txt** dans un fichier **groupes.txt** on utilise la commande **cp users.txt groupes.txt**

```
(kali㉿kali)-[~/Desktop/job4]  
$ cp users.txt groupes.txt
```

-Pour changer le propriétaire du fichier **droits.txt** pour mettre **User1** on va utiliser la commande **sudo chown User1 droits.txt**

```
(kali㉿kali)-[~/Desktop/job4]  
$ sudo chown User1 droits.txt
```

-Pour changer les droits du fichier "**droits.txt**" pour que "**User2**" ai accès seulement en lecture on va changer les droits du fichier pour tous les autres avec la commande **sudo chmod o+r,o-w,o-x droits.txt**

```
(kali㉿kali)-[~/Desktop/job4]  
$ sudo chmod o+r,o-w,o-x droits.txt
```

-Pour changer les droits du fichier "**groupes.txt**" pour que **les utilisateurs puissent accéder au fichier en lecture uniquement** on utilisera la commande **sudo chmod u+r,u-w,u-x groupes.txt**

```
(kali㉿kali)-[~/Desktop/job4]  
$ sudo chmod u+r,u-w,u-x groupes.txt
```

-Pour changer les droits du fichier pour que le groupe "**Plateformeurs**" puisse y accéder en lecture/écriture on va **changer le groupe du fichier** pour mettre le groupe **Plateformeurs**. Après cela on va **changer les droits du fichier** pour mettre que le groupe **Plateformeurs** est les droits qu'en lecture et écriture. Pour cela on va utiliser deux commandes, la première c'est **sudo chgrp Plateformeurs groupes.txt** et la deuxième c'est **sudo chmod g+r,g+w,g-x groupes.txt**

```
(kali㉿kali)-[~/Desktop/job4]  
$ sudo chgrp Plateformeurs groupes.txt
```

```
(kali㉿kali)-[~/Desktop/job4]  
$ sudo chmod g+r,g+w,g-x groupes.txt
```





## JOB 5

-Pour créer un alias qui permettra de lancer la commande "ls -la" en tapant "la" on fait la commande **alias la='ls -la'**

```
(kali㉿kali)-[~/Desktop/job4]  
$ alias la='ls -la' █
```

-Pour créer un alias qui permettra de lancer la commande "apt-get update" en tapant "update" on utilisera la commande **alias update='apt-get update'**

```
(kali㉿kali)-[~/Desktop/job4]  
$ alias update='apt-get update' █
```

-Pour créer un alias qui permettra de lancer la commande "apt-get upgrade" en tapant "upgrade" on utilisera la commande **alias upgrade='apt-get upgrade'**

```
(kali㉿kali)-[~/Desktop/job4]  
$ alias upgrade='apt-get upgrade' █
```

-Pour ajouter une variable d'environnement qui se nommera "USER" et qui sera égale à votre nom d'utilisateur on fait la commande **export USER=/home/kali**

```
(kali㉿kali)-[~]  
$ export USER=/home/kali █
```

-Pour mettre à jour les modifications de votre bashrc dans votre shell actuel on utilisera la commande **exec bash**

```
(kali㉿kali)-[~]  
$ exec bash █
```

-Pour afficher les variables d'environnement on utilisera la commande **env**

```
(kali㉿kali)-[~]  
$ env █
```

-Pour ajouter à votre **Path** le chemin **"/home/[Nom\_D'Utilisateur]/Bureau"** on fera la commande **export PATH=\$PATH:/home/kali/[Bureau]**

```
(kali㉿kali)-[~]  
$ export PATH=$PATH:/home/kali/Desktop █
```



# SHELL BASH PRÉSENTATION

LIGNE DE COMMANDE LINUX



## JOB 6

-Pour désarchiver avec le terminal on utilisera la commande **tar -xzf**  
**[nom\_du\_dossier]**

```
(kali㉿kali)-[~/Downloads]
$ ls
'Ghost in the Shell.tar.gz'
```

```
(kali㉿kali)-[~/Downloads]
$ tar -xzf Ghost\ in\ the\ Shell.tar.gz
```

```
(kali㉿kali)-[~/Downloads]
$ tar -xzf Ghost\ in\ the\ Shell.tar.gz
Ghost in the Shell.pdf
```

```
(kali㉿kali)-[~/Downloads]
$ ls
'Ghost in the Shell.pdf'  'Ghost in the Shell.tar.gz'
```



## JOB 7

-pour créer le fichier "une\_commande.txt" avec le texte suivant "Je suis votre fichier texte" on va utiliser la commande **echo "Je suis votre fichier texte"**

**>une\_commande.txt**. Puis pour compter le nombre de lignes présentes dans votre fichier source apt et l'enregistrer dans un fichier nommé "nb\_lignes.txt" on devra utiliser la commande **wc -l /etc/apt/sources.list >nb\_lignes.txt**. Après cela pour afficher le contenu du fichier source apt et l'enregistrer dans un autre fichier appelé "save\_sources" on va utiliser la commande **cat /etc/apt/sources.list >save.sources**. Et pour finir pour faire une recherche des fichiers commençant par "." tout en cherchant le mot alias qui sera utilisé depuis un fichier on fait la commande **grep -rnw /home/ -e alias** .

Maintenant qu'on a toutes nos commandes on veut toutes les lancer dans une seule commandes on va utiliser la commande **echo Je suis votre fichier texte**

**>une\_commande.txt | wc -l /etc/apt/sources.list >nb\_lignes.txt | cat /etc/apt/sources.list >save.sources | grep -rnw /home/ -e alias** .

```
(kali㉿kali)-[~/Desktop]
└─$ echo Je suis votre fichier texte >une_commande.txt | wc -l
/etc/apt/sources.list >nb_lignes.txt | cat /etc/apt/sources.lis
t >save.sources | grep -rnw /home/ -e alias .
```



## Pour aller plus loin

-Pour installer la commande tree, puis lancer la commande tree en arrière-plan qui aura pour but d'afficher toute l'arborescence de votre / en enregistrant le résultat dans un fichier "tree.save", pour qu'après on liste les éléments présents dans le dossier courant et utilisé directement le résultat de votre première commande pour compter le nombre d'éléments trouvés, pour qu'à la fin on lance une commande pour update vos paquets, si l'update réussit alors, vous devrez lancer un upgrade de vos paquets. Si l'update échoue, votre upgrade ne se lancera pas et tout cela en une seule commande, on va utiliser la commande **sudo apt install tree | tree -a / >tree.save | ls -la | sudo apt-get update && sudo apt-get upgrade**

```
(kali㉿kali)-[~/Desktop]
$ sudo apt install tree | tree -f / >tree.save | ls -la | sudo apt-get update && sudo apt-get upgrade
```



# SHELL BASH PRÉSENTATION

LIGNE DE COMMANDE LINUX

AB

## BONUS

-Pour installer ssh on utilise la commande **sudo apt install ssh**

```
(kali㉿kali)-[~/Desktop]  
$ sudo apt install ssh
```

-Pour générer une clé ssh on utilise la commande **cat ~/.ssh/id\_rsa.pub**

```
(kali㉿kali)-[~/Desktop]  
$ cat ~/.ssh/id_rsa.pub
```

-Pour télécharger un fichier avec SSH (de votre pc ou VM vers le pc ou VM d'un camarade) il faut d'abord savoir si on a le statut "running" apres avoir lancer la commande **sudo systemctl status** si ce n'est pas le statut "running" que vous voyez, activer le avec la commande **sudo systemctl start ssh**. Après cela, on vérifie si la personne à qui on veut envoyer le fichier a le statut "running". Puis on utilise la commande **scp [fichier]**

**User\_camarade@adresse\_camarade:/home/User\_camarade/[destination]**

```
(kali㉿kali)-[~/Desktop]  
$ scp moufidtest.txt moufid@10.10.38.209:/home/moufid/Bureau/
```