



# A la découverte du “SHELL”

```
Kubuntu 64-bit - VMware Workstation 16 Player (Non-commercial use)
Player
~: bash — Konsole
File Edit View Bookmarks Plugins Settings Help
rastree@kubuntu-rastree-vm:~/Music$ cd ~
rastree@kubuntu-rastree-vm:~$ ls -la
.          Desktop      .kde       snap
..         Documents   .lesshtst .sudo_as_admin_successful
.bash_history Downloads  .local     Templates
.bash_logout .face      Music      Videos
.bashrc      .face.icon Pictures   .Xauthority
.cache       .fonts.conf .profile   .xsession-errors
.config      .gtkrc-2.0 Public
rastree@kubuntu-rastree-vm:~$ ls -la
total 608
drwxr-x--- 15 rastree rastree 4096 sept. 22 16:00 .
drwxr-xr-x  3 root    root   4096 sept. 19 07:28 ..
-rw-----  1 rastree rastree 2685 sept. 22 16:00 .bash_history
-rw-r--r--  1 rastree rastree  220 sept. 19 07:28 .bash_logout
-rw-r--r--  1 rastree rastree 3844 sept. 22 11:13 .bashrc
drwx----- 16 rastree rastree 4096 sept. 22 15:55 .cache
drwxr-xr-x 18 rastree rastree 4096 sept. 22 16:00 .config
drwxr-xr-x  2 rastree rastree 4096 sept. 19 09:26 Desktop
drwxr-xr-x  2 rastree rastree 4096 sept. 19 09:26 Documents
drwxr-xr-x  2 rastree rastree 4096 sept. 19 09:26 Downloads
-rw-r--r--  1 rastree rastree 14965 sept. 19 07:28 .face
lrwxrwxrwx  1 rastree rastree    5 sept. 19 07:28 .face.icon -> .face
-rw-rw-r--  1 rastree rastree  110 sept. 20 12:59 .fonts.conf
-rw-rw-r--  1 rastree rastree  265 sept. 22 09:43 .gtkrc-2.0
drwxrwxr  1 rastree rastree 4096 sept. 19 09:26 .kde
-rw-----  1 rastree rastree  20 sept. 19 12:35 .lesshtst
drwxrwxr  1 rastree rastree 4096 sept. 19 09:26 .local
drwxr-xr  1 rastree rastree 4096 sept. 19 09:26 Music
```

Roger LATIERE

## Table des matières

<b>Introduction</b>	<b>2</b>
<b>JOB 1 : la commande LS</b>	<b>2</b>
Comment ajouter des options à une commande et quelles sont les deux syntaxes d'écriture des options pour une commande ?	3
<b>JOB 2 : lecture d'un fichier</b>	<b>3</b>
"CAT"	3
"HEAD" & "TAIL"	4
<b>JOB 3 : installer, mettre à jour, télécharger</b>	<b>4</b>
Installer et lancer un paquet "cmatrix"	4
Mettre à jour le gestionnaire de paquets et les différents logiciels	5
Télécharger "Google"	6
Redémarrer ma machine*	6
Pour arrêter la machine	6
<b>JOB 4 : créer/ajouter un "user", créer/changer les droits d'un fichier</b>	<b>6</b>
Création d'un utilisateur/groupe et déplacement	6
Changement de propriétaire	8
Changement des droits (ou permissions)	8
<b>JOB 5 : Ajout d'alias et de variable et autres</b>	<b>9</b>
Alias	9
Variable d'environnement	10
<b>JOB 6 =&gt; JOB 7 : Ghost in the Shell</b>	<b>11</b>
Réaliser des actions en une seule commande	11
<b>POUR ALLER PLUS LOIN</b>	<b>13</b>
Réaliser des commandes en une seule ligne	13
<b>ANNEXE</b>	<b>14</b>
Soucis rencontrés	14
Liste de commande utiles	15
On creuse un peu plus !	16

# Introduction

Nous allons découvrir plus en profondeur **le terminal, ou le “shell”** .

On va apprendre à se servir de différentes commandes Linux essentielles, et comment cumuler les commandes pour éviter de trop “écrire”.

Le “**Shell**” qu’on considère comme une **interface système** ou **coque** (*shell* en anglais) **logicielle** est une couche logicielle qui fournit l'interface utilisateur d'un système d'exploitation. Il correspond à la couche la plus externe de ce dernier. L'interface système est utilisée comme diminutif de l'interface utilisateur du système d'exploitation.

Mon système installé est Ubuntu (précisément KUbuntu).

## JOB 1 : la commande LS

**Ls** est l’une **des commandes de base** que tout utilisateur Linux devrait savoir.

La commande “**ls**” répertorie les fichiers et les répertoires dans le système de fichiers et affiche des informations détaillées sur eux.

Il fait partie du package GNU Core Utilities qui est installé sur toutes les distributions Linux.

La syntaxe de la commande LS est la suivante :

`ls [options] [fichiers]`

Lorsqu’il est utilisé sans options et arguments, “**LS**” affiche une liste des noms de tous les fichiers du répertoire de travail actuel.

- Pour afficher le manuel de la commande ls, il faudra taper sur le terminal :

`man ls`

- Pour afficher et lister le contenu du fichier “**home**” de mon utilisateur, il faudra taper :

`ls /home`

- Pour afficher les fichiers cachés\* du “**home**” de mon utilisateur :

`ls -a $home`

\*(En interface graphique dans l’explorateur de fichiers, il faudra faire [Control]+[h])

On peut remarquer qu’il y a un code couleur (qui peut varier selon le Shell utilisé) :

→ en bleu, les dossiers de bases tel “*Desktop*”, “*Documents*”, “*Downloads*”... etc

→ en blanc , les dossiers cachés tel “*.bash\_history*”, “*.bash\_logout*”... etc

→ en vert, les fichiers, ici il y en a qu’un “*.face.icon*”

- Pour afficher les fichiers cachés et les informations sur les droits sous forme de liste :  
`ls -a $home -l`  
ou `ls -al`

## Comment ajouter des options à une commande et quelles sont les deux syntaxes d'écriture des options pour une commande ?

La syntaxe générale sur le terminal :

- Le 1er mot tapé est une commande.
- Les lettres tapées après un tiret, et les mots tapés après 2 tirets, sont des options.

→ [nom-de-la-commande] -[options] --[options-longues] [parametre1] [parametre2]

Par exemple :

`man -h` → connaître les options de la commande "*man*" (manuel)

`man --help` → idem que ci-dessus

▲ Attention à bien respecter les espaces !

Il existe ainsi plusieurs options selon ce qu'on veut faire et selon la lettre ou le mot que l'on utilise. On pourra aussi les cumuler en une seule commande.

## **JOB 2 : lecture d'un fichier**

### **"CAT"**

**Cat** est une commande qui **affiche le contenu d'un fichier dans la sortie du terminal**.  
C'est la façon la plus simple pour lire le contenu d'un fichier en ligne de commandes.

Lire un fichier en utilisant une commande qui permet seulement de lire :

`cat` [nom de fichiers]

par exemple : `cat pictures`

Cette commande peut offrir beaucoup d'options, par exemple pour afficher les numéros de lignes ou encore pouvoir concaténer des fichiers textes.

(voir <https://www.malekal.com/exemples-pour-utiliser-la-commande-cat-sur-linux/>)

## “HEAD” & “TAIL”

- La commande **Head** est une autre façon de consulter un fichier texte.

Elle **affiche les 10 premières lignes d'un fichier texte par défaut.**

Mais avec des options on peut choisir le nombre de lignes à afficher.

(<https://www.malekal.com/exemples-utiliser-commande-head/>)

- La commande **Tail** génère les dernières parties d'un seul fichier ou plusieurs fichiers.

Par défaut, la commande **“tail” imprime les dix dernières lignes des fichiers d'entrée.**

Enfin on peut aussi l'utiliser pour la lecture de fichiers journaux en temps réel.

(<https://www.malekal.com/exemples-pour-utiliser-la-commande-tail-sur-linux/>)

Afficher les 10 premières lignes du fichier “.bashrc” :

`head .bashrc` (ne pas oublier de mettre le “.”)

Afficher les 10 dernières lignes du fichier “.bashrc” :

`tail .bashrc`

Afficher les 20 premières lignes du fichier “.bashrc” :

`head -20 .bashrc`

Afficher les 20 dernières lignes du fichier “.bashrc”

`tail -20 .bashrc`

## **JOB 3 : installer, mettre à jour, télécharger**

### Installer et lancer un paquet “cmatrix”

➤ On peut le faire à l'aide d'un script :

```
while true;
do printf "\e[32m%X\e[0m" $((RANDOM%2));
  for ((i=0; i<$((RANDOM%128)); i++))
do printf " ";
done;
done;
```

Cela sera une suite de 0 et de 1 affiché aléatoirement.

On arrêtera le script avec [Control + C].

- A partir du Terminal, il va falloir installer le paquet “**cmatrix**” :

`sudo apt install cmatrix`

Une fois installer pour lancer l’effet “matrix”, il faudra juste écrire :

`cmatrix` (et valider bien sur)

## Mettre à jour le gestionnaire de paquets et les différents logiciels

Pour rappel, **un paquet est une archive** contenant les fichiers nécessaires à l’installation d’un programme sur Ubuntu Linux. Cela peut être un logiciel, un utilitaire, un noyau Linux...

Mettre à jour ses paquets, c’est donc **mettre à jour son système**.

1. On va commencer par mettre à jour la liste des paquets :  
`sudo apt update`  
(un mot de passe admin nous est demandé)
2. Ensuite, on va afficher les paquets pour lesquels une mise à jour est disponible :  
`apt list --upgradable`
3. Puis mettre à jour les paquets installés sur le système avec 2 possibilités :  
`sudo apt upgrade`  
(un mot de passe admin sera demandé)

**upgrade** : les paquets seront remplacés par des versions plus récentes, mais sans qu’aucun autre paquet ne soit ajouté ou supprimé. Par exemple, une nouvelle version de Firefox sera installée avec `apt upgrade`.

ou `sudo apt full-upgrade`

**full-upgrade** : même chose que “**apt upgrade**”, mais supprime des paquets si cela est nécessaire pour installer les nouvelles versions des paquets.

4. **Supprimez tous les paquets devenus obsolètes** et qui ne sont plus nécessaires :  
`sudo apt autoremove`
5. **Supprimez le cache** des paquets qui ne peuvent plus être téléchargés :  
`sudo apt autoclean`

## Télécharger “Google”

Pour télécharger Google, on va se servir de la commande “**wget**” suivi du lien internet, comme suit :

```
wget https://dl.google.com/linux/direct/google-chrome-stable_current_amd64.deb  
→ “.deb” car je suis sur une distribution Debian
```

Puis la commande “**dpkg -i**” (option “**-i**” → installer) pour l’installer, “**sudo**” précédera cette commande car cela exige des droits élevés :

```
sudo dpkg -i google-chrome-stable_current_amd64.deb
```

## Redémarrer ma machine\*

```
sudo shutdown now -r  
(“-r” pour redémarrer, c’est une option)
```

\*Pour aller plus loin : <https://forum.ubuntu-fr.org/viewtopic.php?id=2050374>

## Pour arrêter la machine

```
sudo shutdown now
```

## **JOB 4 : créer/ajouter un “user”, créer/changer les droits d’un fichier**

### Création d’un utilisateur/groupe et déplacement

1. On va commencer par créer le fichier “*users.txt*”, avec la commande “**touch**”:  

```
touch users.txt
```

2. Ce fichier doit contenir “*User1*” et “*User2*”.

Pour cela nous allons utiliser la commande “**echo**” et appliquer le retour à la ligne concernant le “*User2*” avec le symbole “**>**”:

```
echo User1 > users.txt  
echo User2 >> users.txt
```

→ la commande “**>**” pour copier “*User1*” dans le fichier “*users.txt*”

→ la commande “**>>**” **pour copier “*User2*” à la suite** dans le fichier “*users.txt*”

**NB** : si on se sert que d’un chevron pour la 2ème commande, elle effacera le contenu du fichier.

Pour vérifier si cela à bien marché :

```
cat users.txt
```

→ tout est OK !

3. Nous devons maintenant créer un groupe appelé “*Plateformeurs*”, avec la commande “**groupadd**” (et “**sudo**” pour avoir les droits de création en tant que super-admin sans me logger) :

```
sudo groupadd Plateformeurs
```

Pour vérifier si le groupe a bien été créer :

```
cat /etc/group
```

→ le groupe apparaîtra en fin de liste

4. Création d’un utilisateur\* “User1” et d’un utilisateur “User2” :

```
sudo adduser user1
```

→ la majuscule n’est pas possible pour les noms d’utilisateurs

```
sudo adduser user2
```

\* ▲ Il y a deux commandes pour ajouter : “**useradd**” et “**adduser**” :

- **useradd** → qui autorise la majuscule pour le nom de l’user et sans mdp
- **adduser** → qui n’autorise pas la majuscule (Linux ne l’accepte de base), et demande un mdp “user” et d’autres infos utilisateur

Cette dernière est recommandée et créera moins de problèmes par la suite.

En effet avec “**useradd**”, il ne m’a pas été demandé de créer un mdp... et de ce fait cela peut-être gênant (pour ajouter un mdp à l’User1 : **sudo passwd User1**).

Pour aller d’un utilisateur à un autre : **su User1** ; **su User2**

Pour vérifier si les “user” ont bien été créer et les lister :

```
cat /etc/passwd | awk -F: '{print $ 1}'
```

→ à noter que cela marche aussi pour les groupes, on écrira “**group**” à la place de “**passwd**”

5. Ajoutons “User2” au groupe Plateformeurs :

```
sudo adduser User2 Plateformeurs
```

6. On doit maintenant copier “*users.txt*” dans le fichier “*droits.txt*”. On se servira de la commande “**cp**” :

```
cp users.txt droits.txt
```

(on vérifiera avec “**cat droits.txt**”)

→ à noter que le fichier “*droits.txt*” n’était pas créé. Il s’est créé automatiquement avec la commande de copie : en effet, si il n’existe pas, le Terminal le créera.

7. Et on copie “*users.txt*” dans “*groupes.txt*” :

```
cp users.txt groupes.txt
```



## Changement de propriétaire

Il faut maintenant changer le propriétaire de “*droits.txt*” pour mettre “*user1*”. On va se servir de la commande “**chown**” (**change owner**, changer le propriétaire) :

```
sudo chown User1 droits.txt
```

→ Pour info, Les droits des fichiers d'un répertoire peuvent être affichés par la commande “**ls -l**”, on pourra ainsi voir que “*User1*” est bien propriétaire de “*droits.txt*”.

## Changement des droits (ou permissions)

Changer les droits de “*droits.txt*” pour que “*User2*” ait accès seulement en lecture. On utilisera la commande “**chmod**” (**change mode**, changer les permissions).

▲ Il faut partir du principe que pour donner les droits, on doit le faire en fonction du propriétaire (ici c'est “*User1*”).

Ainsi les autres utilisateurs concernés par les droits seront considérés comme autres/others → “**o**”.

Les permissions (ou droits) sont “**r**” pour “read”, “**w**” pour “write” et “**x**” pour “execute”.

Lorsque nous faisons “**ls -l**” nous pouvons voir une suite de droits sur le début de ligne :

- à noter au tout début, le “**-**” pour fichier et le “**d**” pour dossier
- une suite de lettres “**rwX**” et/ou de “**-**” vont composés les permissions pour
  - les 3 premiers : le propriétaire (u)
  - les 3 suivants : le groupe (g)
  - les 3 derniers : les autres ou “reste du monde” (o)
- le “**-**” signifiera que la permission correspondante à l'ordre précité n'est pas appliquée

(voir le lien pour plus de détails <https://doc.ubuntu-fr.org/permissions>)

1. Ainsi, la commande qui en découle pour le fichier “*droits.txt*” :

```
sudo chmod o+r-wx droits.txt
```

2. Il faut maintenant changer les droits de “*groupes.txt*” pour que les utilisateurs puissent y accéder en lecture uniquement :

```
sudo chmod u+r-wx groupes.txt
```

3. Enfin changer les droits du fichier “*groupes.txt*” pour que le groupe “*Plateformeurs*” y accède en lecture/écriture.

- On va commencer par changer le groupe propriétaire de “*groupes.txt*”. La commande **chgrp** (pour **change group**) permet de changer le groupe auquel appartient le fichier. Tous les membres de ce groupe seront concernés par les permissions du groupe de la 2ème série de **rwX**. Encore une fois, seuls le super-utilisateur ou le *propriétaire actuel* d'un fichier peut utiliser **chgrp** (un membre du groupe ne peut pas changer le groupe propriétaire).

- Ainsi en découle la ligne de commande suivante :  
`sudo chgrp Plateformeurs groupes.txt`  
(**"ls -l"** pour vérifier le changement de groupe)
- Et ensuite on va changer les droits en lecture/écriture pour "Plateformeurs":  
`sudo chmod g+rw-x groupes.txt`

**NB** : **"su"** est supérieur à **"sudo"**

▼ **Pour faciliter la gestion des droits dans son ensemble, on créera des groupes avec certains droits et on mettra les utilisateurs à l'intérieur du groupe qui lui correspond.**

Ainsi, lorsqu'on doit donner les droits à "user2" seulement en lecture

→ en plus du groupe "Plateformeurs" il faudra créer un autre groupe (le nom sera inventé) et y placer "user2" en donnant les droits voulus, en fonction du groupe donc !

## **JOB 5 : Ajout d'alias et de variable et autres**

### **Alias**

Les **alias** sont des **commandes** qu'un utilisateur Linux peut créer dans un terminal Linux.

C'est un **moyen très puissant pour gagner en productivité** en personnalisant le shell Bash.

Créez vos propres commandes Linux à l'aide de fonctions d'alias pour des tâches répétitives, raccourcir de longue commandes et configurez des commandes standards avec les options que vous utilisez très souvent.

1. Il faut ajouter un alias qui permettra de lancer la commande **"ls -la"** en tapant **"la"**. La commande dont on se servira est **"alias"**:  
`alias la='ls -la'`  
→ `alias [NomAlias]='[commande avec ses options ou arguments]'`  
→ **ls -la** : afficher tous les fichiers (même cachés) et leurs permissions.
2. Ensuite , ajoutons un alias **"apt-get update"** en tapant **"update"** :  
`alias update='apt-get update'`
3. Puis , un alias pour **"apt-get upgrade"** en tapant **"upgrade"** :  
`alias upgrade='apt-get upgrade'`

**NB** : pour supprimer un alias → **unalias** [NomAlias]  
pour voir les alias actifs : **alias**

## Variable d'environnement

Les variables d'environnement constituent un moyen d'influencer le comportement des logiciels sur votre système. Par exemple, la variable d'environnement « LANG » détermine la langue que les logiciels utilisent pour communiquer avec l'utilisateur.

Les variables sont constituées de **noms** auxquels on assigne des **valeurs**. Ainsi, le système d'un utilisateur français devrait avoir la valeur « fr\_FR.UTF-8 » assignée à la variable « LANG ».

La signification d'une variable d'environnement et le type de valeur qui peut lui être assignée sont déterminés par l'application qui utilise celle-ci. Il existe un petit nombre de variables d'environnement bien connues, dont le sens et le type de valeur sont bien déterminés, et qui sont utilisées par de nombreuses applications.

Bien que quelques applications de configuration en mode graphique manipulent en réalité des variables d'environnement, la ligne de commande offre un maximum de flexibilité pour créer et modifier ces variables.

(voir ici : [https://doc.ubuntu-fr.org/variables\\_d\\_environnement](https://doc.ubuntu-fr.org/variables_d_environnement))

(et ici :

<https://www.digitalocean.com/community/tutorials/how-to-read-and-set-environmental-and-shell-variables-on-linux-fr>)

Nous devons maintenant ajouter une variable d'environnement qui se nommera "USER" et qui sera égale à notre nom d'utilisateur.

Pour cela nous devons créer la variable SHELL et ensuite l'exporter en tant que variable d'environnement.

```
USER=rastree
```

→ le symbole "=" amène à créer la variable SHELL

```
export USER=rastree
```

→ la commande "**export**" permet de l'exporter en tant que variable d'environnement.

→ on peut le faire directement car on veut cette variable d'environnement dès le départ.

Pour afficher les variables d'environnement :

```
printenv
```

→ "**print**" pour afficher et "**env**" pour environnement

Pour être plus précis dans la recherche, on peut faire avec "**printenv**" des valeurs de variables individuelles (il y a SHELL, TERM, PWD...etc), par exemple :

```
printenv SHELL
```

→ pour le SHELL

`printenv USER`

→ pour l'utilisateur...

Nous devons ajouter dans la variable d'environnement "PATH" une valeur ayant le chemin "/home/'votre utilisateur'/Bureau".

`export PATH=$PATH:/home/rastree/Desktop`

→ pour vérifier si tout est bon, on tapera : "**`printenv PATH`**"

## **JOB 6 => JOB 7 : Ghost in the Shell**

On ne pourra pas télécharger via le Terminal le lien car le Cloud du Drive est sécurisé. Il faudra le télécharger en passant par le navigateur sur ma VM et ensuite le décompresser via le Terminal (la commande est "**`tar`**", se renseigner).

Toutes les actions sont à réaliser en une seule commande.

**On va approfondir les commandes, avec les caractères suivant ">< >> <<".**

### **Réaliser des actions en une seule commande**

1. Créer un fichier "*une\_commande.txt*" avec le texte suivant "*Je suis votre fichier texte*".

La commande "**`echo`**" permet d'afficher un texte dans le terminal.

En utilisant redirection ">", il est possible de créer un fichier (ou écraser un existant) avec le contenu souhaité.

Ainsi la commande suivante qui en découle :

`echo "Je suis votre fichier texte" > une_commande.txt`

2. Compter le nombre de lignes présentes dans votre fichier de source "apt" et les enregistrer dans un fichier nommé "nb\_lignes.txt"

**Définition** : Le fichier "**`sources.list`**" est un document au format texte brut contenant une liste successive de lignes **APT**, elles-mêmes donnant de l'information à propos de dépôts de paquets pour votre système d'exploitation Ubuntu.

(voir <https://doc.ubuntu-fr.org/sources.list>)

Le fichier "**`sources.list`**" se trouve dans le dossier **`/etc/apt/`**

Allons dans le dossier source "**`apt`**" pour voir ce qu'il y a à l'intérieur :

`find /etc/apt/`

→ il y a plusieurs fichiers de listés

La commande **"wc"** de Linux vous **permet de compter le nombre de lignes, de mots, de caractères et d'octets** de chaque fichier donné ou d'entrée standard et d'imprimer le résultat.

La commande **"wc"** accepte un ou plusieurs noms de fichiers mais peut aussi lire la sortie de la console.

Ainsi on peut l'imbriquer avec d'autres commandes avec un pipe.

Dans sa forme la plus simple lorsqu'elle est utilisée sans options, la commande **"wc"** imprimera quatre colonnes :

- le nombre de lignes (**-l**, **--lines**)
- de mots (**-w**, **--words**)
- les nombres d'octets (**-c**, **--bytes**)
- le nombre de caractères (**-m**, **--chars**)
- la longueur de la ligne la plus longue (**-L**, **--Max-Line-Longueur**)

Lorsque vous utilisez l'entrée standard, la quatrième colonne (nom de fichier) n'est pas affichée. Par exemple, la commande suivante affichera des informations sur le fichier virtuel.

La syntaxe de la commande **wc** est :

```
wc OPTION [FICHIER]
```

Ainsi, pour aller chercher et lister les lignes du fichier **"sources.list"** dans **"apt"**, on utilisera **"wc"** (word count) suivi de l'option **"-l"**(lines, nbre de lignes) et de l'arborescence **"sources.list"** :

```
wc -l /etc/apt/sources.list
```

Il faut maintenant créer dans la même commande la création du fichier **"nb\_lignes.txt"** et y placer le résultat, ainsi la commande en résultant :

```
wc -l /etc/apt/sources.list > nb_lignes.txt
```

→ le symbole **">"** me permet d'envoyer le résultat de la commande dans un fichier texte. J'aurai très bien pu placer le fichier texte dans **"Document"** : dans ce cas je dois écrire l'arborescence complète de l'endroit où doit se situer le fichier texte.

→ je vérifie : **"ls"** pour voir si mon fichier est bien créé.

3. Ici il va falloir afficher le contenu du fichier source apt et l'enregistrer dans un autre fichier appelé **"save\_sources"**

Ainsi la commande qui en découle :

```
cat /etc/apt/sources.list > save_sources
```

→ je vérifie : **"ls"** ; et je regarde à l'intérieur **"cat save\_sources"** ⇒ tout est OK

4. Faire une recherche des fichiers commençant par **"."** et ayant le mot **"alias"** dans son contenu.

Il va y avoir plusieurs principes de syntaxe concernant cette commande. Commençons par l'écrire :

```
find ~ -type f -iname "*.*
```

⇒ **find** : commande pour rechercher/trouver un fichier

⇒ **~** : pour chercher dans l'arborescence "home", on peut mettre **/** pour chercher à la racine (tout les fichiers du système)

⇒ **-type** : on aura compris que le tiret amène une commande, ici "type" suivi d'une lettre ("f" pour des fichiers normaux, "d" pour des répertoires, "l" pour des liens symboliques... "c" et "b"... ) : c'est la seule option où il y aura un espace pour spécifier ce qu'on recherche

⇒ **f** : expliqué ci-dessus

⇒ **-iname** : j'annonce la recherche par nom. "iname" ignore la casse (maj ou min) ; "name" sera sensible à la casse, ce dernier servira à être plus dans la recherche.

⇒  **"\*.\*** : les guillemets annoncent le mot, ou une partie du mot à rechercher, ici le  **"."**

L'astérisque est placé après le point car on veut rechercher à partir du point et ce qu'il y a après.

En plaçant l'astérisque avant le point on aurait recherché, tout ce qu'il y a avant le point (le point étant inclus dans la recherche bien évidemment) et qui finit donc par le point, c'est-à-dire pas grand chose :)

## POUR ALLER PLUS LOIN

### Réaliser des commandes en une seule ligne

Pour bien comprendre la démarche, je vais faire les actions les unes après les autres pour ensuite les transformer en une seule ligne de commande.

**1.** Installer la commande tree

```
sudo apt install tree
```

**2.** Lancer la commande **"tree"** qui aura pour but d'afficher toute l'arborescence de votre **/** ("root) en enregistrant dans un fichier "tree.save" en arrière-plan.

```
tree ~ > tree.save &
```

→ pour tester, ici nous irons dans l'arborescence /home **"~"**

**3.** Lister les éléments présents dans le dossier courant (le dossier courant correspond à l'endroit où l'on se trouve : le "working directory") et utiliser directement le résultat pour compter le nombre d'éléments trouvés.

```
ls -a | wc -l
```

→ le symbole **" | "** (pipeline ou "pipe") permet de se servir du résultat de la 1ère commande pour faire la 2ème.

4. Lancer une commande pour “update” les paquets, si l’update réussit alors, on doit lancer un upgrade des paquets. Si l’update échoue, l’upgrade ne se lancera pas.

`sudo apt update && sudo apt upgrade`

→ ici le “&&” permet de faire la commande après ces caractères SI et SEULEMENT SI la commande précédente a fini sans erreur.

- Ainsi en découle la commande finale :

`sudo apt install tree && tree / > tree.save & ls -a | wc -l & sudo apt update && sudo apt upgrade`

→ ici le “&” permet de réaliser la 1ère commande (avant le “&” donc) en arrière plan en même temps que la 2ème, même si elle retourne une erreur.

Concernant la 2ème utilisation de “&” dans la commande, on aurait très bien pu mettre “&&” car dans ce cas cela n’est pas gênant.

→ à noter que j’ai créé des alias : ici je mets les commandes complètes, j’aurai très bien pu écrire “*update*” et “*upgrade*” seulement.

**Rappel** : pour supprimer un programme/logiciel/fonction installée dans le terminal

→ `sudo apt remove` [nom du fichier]

---

# **ANNEXE**

## **Soucis rencontrés**

- Mon PC :

- MSI I7-1165 G7 (4 Coeurs avec HT → 8 coeurs)
- 24Go de RAM
- SSD M2 512 Go

### **Problèmes particuliers :**

- **le 19 septembre** : de gros soucis de stabilité avec VMWare 16.2.4 : lorsque je navigue entre ma VM et mon Windows natif (10 Pro x64) pour aller sur le web ou écrire sur ma documentation, au bout d'environ 5mn, je reviens sur ma VM et elle est figée.  
De plus mon PC s'emballe (ventilation à fond) et mon CPU est entre 80 à 85% d'utilisation.

Je suis allée là

<https://www.informatiweb-pro.net/virtualisation/vmware/vmware-workstation-15-resoudre-problemes-connus.html> et là

<https://unix.stackexchange.com/questions/458893/vmware-on-linux-host-causes-regu-lar-freezes>

J'ai fait les réglages recommandés, vérifier sur un PC d'un ami qui marche bien (un peu moins récents que le mien, un peu moins puissant aussi) et rien n'y fait. J'ai perdu énormément de temps pour essayer de trouver une solution.

La création de la VM via VMWare avec l'OS Debian et Ubuntu se fige aussi durant l'installation (vers la fin).

Le fait de mettre en pause la VM et de relancer VMWare et la VM solutionne le problème, l'installation s'est finie et comme chaque instance, 5mn après ma VM se fige... et la boucle continue.

Je n'ai pourtant eu aucun soucis lors de l'activité "Dual Boot" en VM et l'installation de "Manjaro" ainsi que l'utilisation des 2 VM, vendredi passé.

**Le problème a été résolu ce jour** (le 20 septembre) : cela venait de mon anti-virus (ESET Internet Security).

Après avoir essayé en le désactivant totalement (en effet il y a 2 couches de protections et en l'occurrence le HIPS) cela marche sans problème.

J'ai donc configuré des règles d'exceptions pour pouvoir me permettre d'utiliser VMWare avec mon anti-virus en fonction.

Tout marche impeccablement aujourd'hui ! :)

- **le 22 septembre** : encore des soucis de stabilité, alors que tout semblait réglé... j'ai désactivé le HIPS de mon AV et tout va bien. Je vais essayer de trouver une solution viable à ce problème.



- **le 23 septembre** : j'ai considérablement augmenté les exceptions concernant le programme VMWare : en effet il y a beaucoup d'exécutables qui peuvent se lancer durant l'utilisation de VMWare = tout est rentré dans l'ordre ce jour :)

---

## Liste de commande utiles

**top** → équivalent au gestionnaire de tâches.

**cd /** → pour aller à la racine du terminal

**cd** → retour dans mon "Home"

**pwd** → pour savoir où on se trouve dans l'arborescence du terminal

**source [nom de fichier]** → permet de prendre en compte les modifications faites dans le fichier modifié, ici ".bashrc" au sein du terminal (alias créer "en dur" avec l'aide de Kamel → ne pas oublier d'ajouter "sudo" aux alias "update" et "upgrade")

**ls /usr/bin/** → afficher toutes les commandes

**getent group** → visualiser directement les groupes disponibles

## On creuse un peu plus !

1. Afin de pouvoir refaire le TP du début, il va falloir supprimer les "user" et autres dossiers ou fichiers créés :
  - pour supprimer les "user"  
→ **sudo deluser User1**
  - pour supprimer "cmatrix"  
→ **sudo apt remove cmatrix**
  - pour désinstaller "google chrome"  
→ **sudo apt remove google-chrome-stable**
  - pour supprimer le fichier "users.txt" et tous les fichiers en ".txt"  
→ **sudo rm users.txt**
  - pour supprimer le groupe Plateformeurs  
→ **sudo groupdel Plateformeurs**  
→ **getent group** (pour vérifier)

Je refais ainsi le TP, et je vais créer des alias pour simplifier la mise à jour des paquets (certains ont déjà été créés).

Concernant le script de "cmatrix" (simplifié, je l'ai nommé "cmatrixbeta"), on peut le faire directement sur le terminal, et il se lancera automatiquement après la fin du script.

En passant par **gedit**, il faudra taper la commande "**bash**" suivi du nom du script.

## 2. GEDIT

Pour modifier un fichier (par exemple le ".bashrc" pour intégrer définitivement les alias dans le bash), on se servira de "**gedit**" qui est un éditeur de texte (il y a aussi "Nano") : pour sauvegarder [Control + S] et pour fermer "**gedit**" [Control + Q]

La 1ère ligne d'un script doit comporter : **#!/bin/bash**

▲ Ne pas oublier de mettre à jour le fichier modifier : en effet le fait de sauvegarder la modification ne suffit pas, il faut mettre à jour les modifications dans le bash avec la commande "source", ici donc :

`source .bashrc`

## 3. Pour installer le navigateur "BRAVE"\* :

- **Pré-requis** : pour une installation via un dépôt, il faut ajouter les paquets suivants :  
`sudo apt install apt-transport-https curl gnupg`
- Installation via Snap, plus simple mais faille de sécurité très critique détectée en Février 2022, toujours pas corrigée → cet outil est normalement installé de base sur les versions bureautiques récentes, mais on ne l'utilisera pas ici car faille de sécurité.
- Installation classique via un dépôt :  
`curl -s https://brave-browser-apt-release.s3.brave.com/brave-core.asc | sudo apt-key --keyring /etc/apt/trusted.gpg.d/brave-browser-release.gpg add -`  
Et :  
`echo "deb [arch=amd64] https://brave-browser-apt-release.s3.brave.com/ stable main" | sudo tee /etc/apt/sources.list.d/brave-browser-release.list`

Puis :

`sudo apt update && sudo apt install brave-browser`

L'installation classique n'a pas fonctionné, je suis passé par Snap.

\*(voir ici : <https://doc.ubuntu-fr.org/brave>)

## C'est quoi Snap\*\* ?

Le format **snap** vise à permettre l'installation de nouvelles versions de logiciels dans les systèmes Linux, tout en apportant aux développeurs la facilité de distribution, la fiabilité et la sécurité.

\*\* (voir ici : <https://doc.ubuntu-fr.org/snap>)

### 4. Je veux supprimer certaines lignes dans un fichier texte (ici "users.txt", j'y ai placé plusieurs "User", 7 précisément).

- Une des méthodes est d'utiliser la commande "**sed**" → en effet elle permet de supprimer les lignes que l'on veut supprimer par intervalle ou à partir d'une ligne spécifiée :

**sed -i '2~3d' mon-fichier**

→ cette commande permet de supprimer la 2e ligne, puis la 5e ligne, puis les lignes 2+3k de tout un fichier

➤ si on décortique la commande

- **-i** : option qui permet de modifier directement le fichier
- **2** est le numéro de la première ligne à supprimer
- **3** est l'intervalle entre les deux lignes à supprimer
- **d** signifie "delete"

Dans mon cas, la commande qui suit :

**sed -i '3~d' users.txt** → ici je vais supprimer que la 3ème ligne

**sed -i '3~1d' users.txt** → **c'est la bonne commande**, je vais supprimer toutes les lignes à partir de la 3ème inclus.

En vérifiant avec la commande "**cat**", ça a fonctionné !

### 5. Création de plusieurs groupes, je veux effacer plusieurs groupes à la fois

**sudo groupdel Erase\_me Erase Remove\_me Remove** → ça ne marche pas

J'ai pensé à faire un script pour cela, mais je n'ai pas réussi à le rendre opérationnel. Je me suis servi de ce script\* qui à la base sert à effacer plusieurs utilisateurs:

```
for i in toto titi tutu tata
do
deluser --remove-home $i
done
```

→ en mettant "**groupdel**" à la place de "**deluser**", en mettant le chemin où se trouve les groupes (/etc/group) en essayant diverses options... impossible de faire fonctionner le script : message d'erreur systématique.

J'ai donc supprimé les groupes un à un.

Le ";" permet de séparer et faire plusieurs commandes avant de valider.

\*(voir ici : <https://forum.ubuntu-fr.org/viewtopic.php?id=1705111>)

## 6. Création de plusieurs utilisateurs, je veux effacer les utilisateurs qui ne sont pas utiles en une seule fois.

Je vais me servir ainsi du script ci-dessus, qui représente une boucle.

Je passe par “**gedit**” en tapant : `gedit erase_user_test` → le nom de mon script

```
for i in user3 user4 user5
```

```
do
```

```
sudo deluser --remove-home $i → j'ai rajouté sudo car sinon cela ne marche pas  
done
```

→ Ne pas oublier : pour exécuter le script :

***bash [nom du script]***

→ les 3 utilisateurs ont été effacés en même temps !Parfait !

## 7. Concernant le changement de droits

- Pour l'user2, je vais créer un groupe où je placerai user2, afin de donner les droits particuliers au groupe de user2 et non à tous les “user”.

Le groupe s'appellera “**For\_user2group**” :

```
sudo adduser user2 For_user2group
```

→ je me rends compte que user2 est dans le groupe “Plateformeurs”, et le groupe que je viens de créer, ainsi je vais faire :

```
sudo deluser user2 Plateformeurs
```

→ désormais user2 est que dans le groupe “For\_user2group”

- Je dois maintenant changer le groupe propriétaire du fichier “droits.txt”, tout en sachant que “user1” est le propriétaire :

```
sudo chown user1:For_user2group droits.txt
```

→ Comme on peut le voir il suffit de séparer par le symbole deux-points “:”, le nom de l'utilisateur propriétaire à gauche et le nom du nouveau groupe propriétaire à droite.

- Ainsi pour les droits seulement en lecture, je vais les appliquer au groupe propriétaire

```
sudo chmod g+r-wx droits.txt
```

ou 

```
sudo chmod g-wx droits.txt
```

 (on part du principe qu'à la base on a tous les droits)

→ je vérifie avec mon alias “la” (ls -la), tout est ok !

## Idées de scripts

1. Pourquoi pas automatiser tous les jours à 10h par exemple la mise à jour des paquets (mettre à jour la liste + afficher les paquets pour lesquels la maj est dispo + mettre à jour les paquets installés avec "full-upgradable" c'est mieux + supprimez les paquets obsolète + supprimer le cache des paquets ne pouvant plus être téléchargés)  
⇒ j'essaierai de voir ça plus tard

### A noter :

Pour rendre un script exécutable : **change mod+x** nom-de-fichier