

Projet Shell

Sommaire :

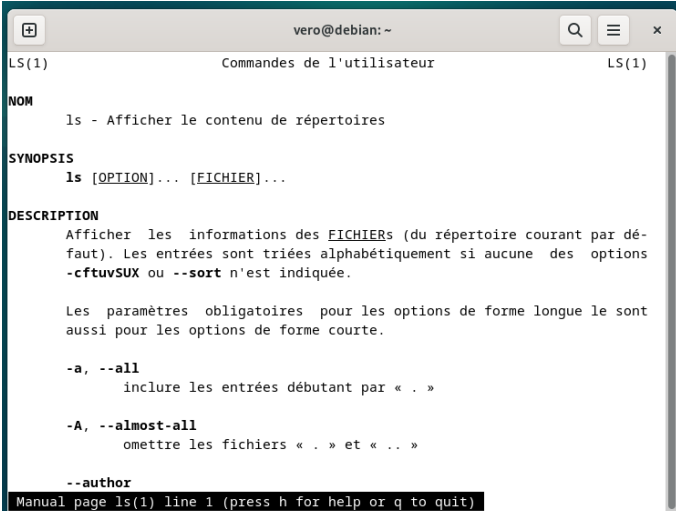
I.	<u>Job 1</u>	<u>1</u>
II.	<u>Job 2</u>	<u>3</u>
III.	<u>Job 3</u>	<u>5</u>
IV.	<u>Job 4</u>	<u>9</u>
V.	<u>Job 5</u>	<u>12</u>
VI.	<u>Job 6</u>	<u>16</u>
VII.	<u>Job 7</u>	<u>17</u>

I. Job 1

A. Afficher le manuel de la commande `ls`

La commande pour afficher le manuel d'une commande est la commande **man** suivi de la commande dont on souhaite le manuel.

Pour avoir le manuel de commande **ls** nous devons donc entrer la commande suivante **man 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

Manual page ls(1) line 1 (press h for help or q to quit)
```

B. Afficher les fichiers cachés du home de votre utilisateur

Nous pouvons ajouter des options à nos commandes. Les **options** permettent de **contrôler divers aspects de la commande**, tel que le format de la sortie, les permissions, les filtres, etc...

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

Il existe **deux syntaxes** principales d'écriture des options pour une commande :

- **Syntaxe à tirets courts** (ou options courtes) : Cette syntaxe **utilise un ou plusieurs tirets courts** (généralement un seul) **suivis d'une lettre** ou d'une **série de lettres** pour représenter une **option**. Elle permet d'entrer les commandes plus **rapidement**.
- **Syntaxe à tirets longs** (ou options longues) : Cette syntaxe utilise **deux tirets longs suivis d'un nom d'option complet** pour représenter une option. Elle permet de **faciliter la relecture** des commandes entrées.

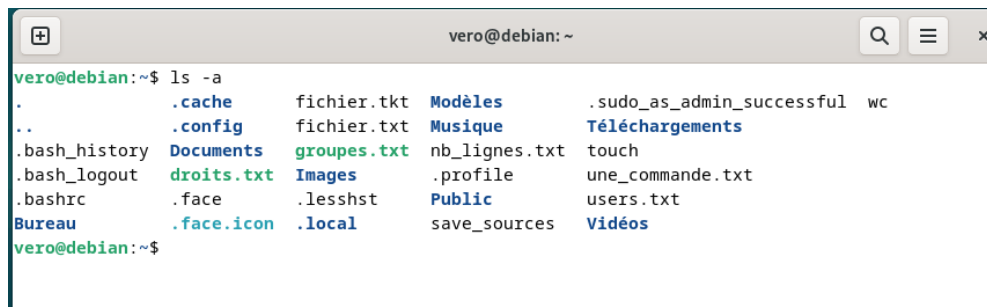
Nous pouvons utiliser les commandes ci-dessous pour afficher les fichiers et dossiers cachés du home de votre utilisateur :

`ls -a` (en syntaxe à tirets courts)

ou

`ls --all` (en syntaxe à tirets longs)

- ❖ `ls` est la commande utilisée pour lister les fichiers et répertoires dans un répertoire donné
- ❖ `-a` (ou `--all`) est l'option qui nous permet de voir tous les fichiers même cachés



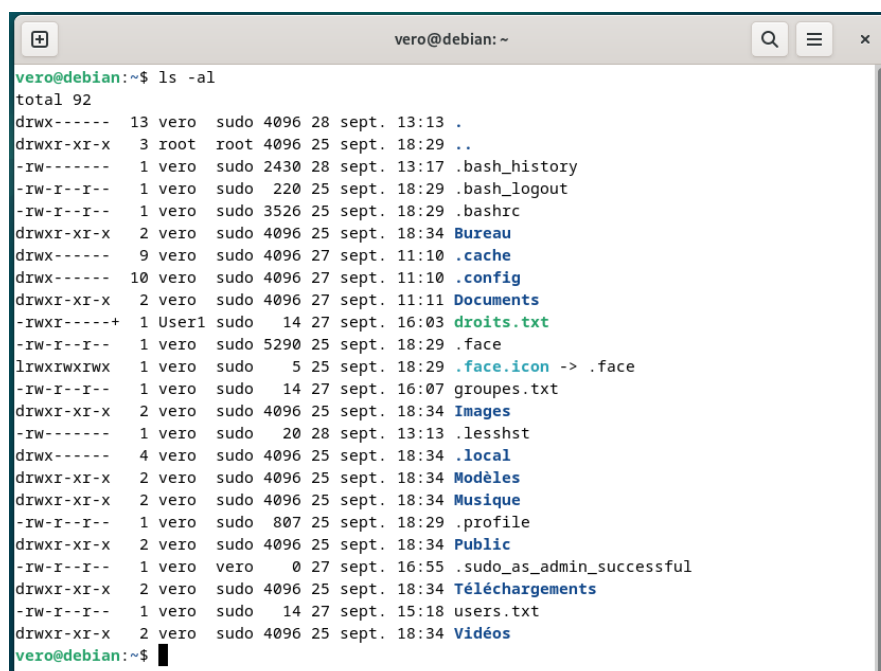
```
vero@debian:~$ ls -a
.          .cache      fichier.tkt  Modèles    .sudo_as_admin_successful  wc
..         .config     fichier.txt  Musique     Téléchargements
.bash_history Documents  groupes.txt nb_lignes.txt touch
.bash_logout droits.txt Images      .profile   une_commande.txt
.bashrc    .face      .lessht    Public      users.txt
Bureau     .face.icon .local     save_sources Vidéos
vero@debian:~$
```

C. Afficher les fichiers cachés plus les informations sur les droits sous forme de liste

La commande pour afficher les fichiers cachés avec les informations sur les droits sous forme de liste est :

`ls -al`

- ❖ `-l` est l'option qui **affiche les informations** sur les fichiers et répertoires **sous forme de liste détaillée** en donnant des informations telles que les autorisations, le propriétaire, le groupe, la taille, la date de modification et le nom du fichier.



```
vero@debian:~$ ls -al
total 92
drwx----- 13 vero  sudo 4096 28 sept. 13:13 .
drwxr-xr-x  3 root  root 4096 25 sept. 18:29 ..
-rw-----  1 vero  sudo 2430 28 sept. 13:17 .bash_history
-rw-r--r--  1 vero  sudo  220 25 sept. 18:29 .bash_logout
-rw-r--r--  1 vero  sudo 3526 25 sept. 18:29 .bashrc
drwxr-xr-x  2 vero  sudo 4096 25 sept. 18:34 Bureau
drwx-----  9 vero  sudo 4096 27 sept. 11:10 .cache
drwx----- 10 vero  sudo 4096 27 sept. 11:10 .config
drwxr-xr-x  2 vero  sudo 4096 27 sept. 11:11 Documents
-rwxr-----+ 1 User1 sudo  14 27 sept. 16:03 droits.txt
-rw-r--r--  1 vero  sudo 5290 25 sept. 18:29 .face
lrwxrwxrwx  1 vero  sudo   5 25 sept. 18:29 .face.icon -> .face
-rw-r--r--  1 vero  sudo  14 27 sept. 16:07 groupes.txt
drwxr-xr-x  2 vero  sudo 4096 25 sept. 18:34 Images
-rw-----  1 vero  sudo   20 28 sept. 13:13 .lessht
drwx-----  4 vero  sudo 4096 25 sept. 18:34 .local
drwxr-xr-x  2 vero  sudo 4096 25 sept. 18:34 Modèles
drwxr-xr-x  2 vero  sudo 4096 25 sept. 18:34 Musique
-rw-r--r--  1 vero  sudo  807 25 sept. 18:29 .profile
drwxr-xr-x  2 vero  sudo 4096 25 sept. 18:34 Public
-rw-r--r--  1 vero  vero   0 27 sept. 16:55 .sudo_as_admin_successful
drwxr-xr-x  2 vero  sudo 4096 25 sept. 18:34 Téléchargements
-rw-r--r--  1 vero  sudo  14 27 sept. 15:18 users.txt
drwxr-xr-x  2 vero  sudo 4096 25 sept. 18:34 Vidéos
vero@debian:~$
```

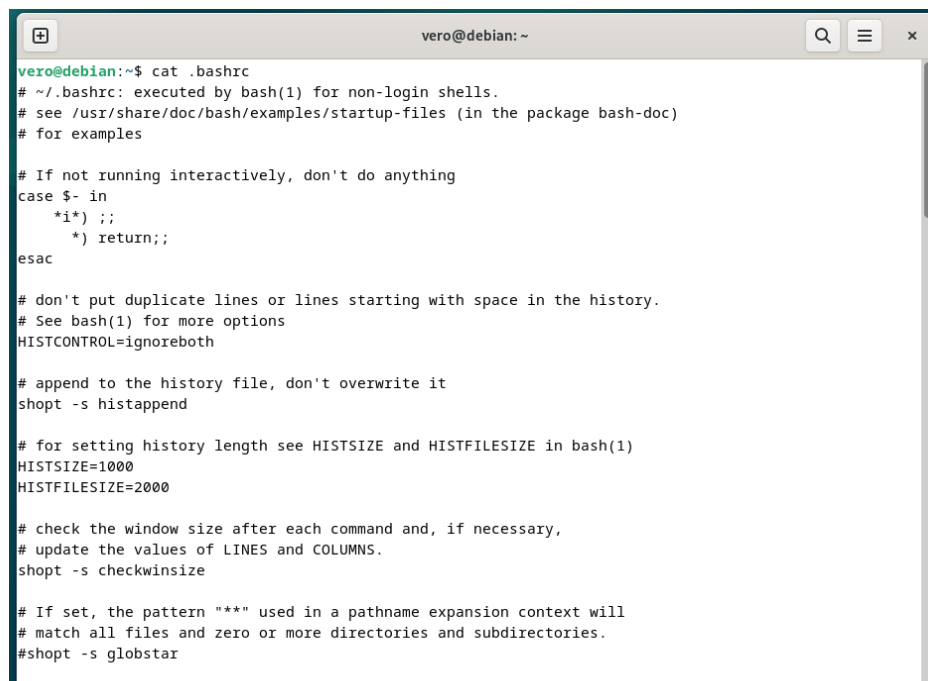
II. Job 2

A. Lisez un fichier en utilisant une commande qui permet seulement de lire

La commande **cat** est une commande utilisée pour **afficher le contenu d'un ou plusieurs fichiers** à l'écran. Le nom "**cat**" est une **abréviation de "concatenate"** (concaténer en français).

Pour afficher le contenu d'un fichier nous devons entrer la commande suivante : **cat** suivi du nom du fichier. Dans notre cas nous allons lire le fichier ".bashrc" donc :

cat .bashrc



```
vero@debian:~$ cat .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

# check the window size after each command and, if necessary,
# update the values of LINES and COLUMNS.
shopt -s checkwinsize

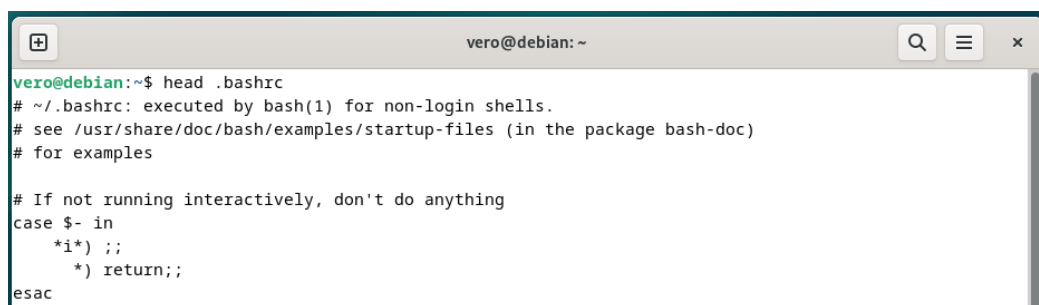
# If set, the pattern "*" used in a pathname expansion context will
# match all files and zero or more directories and subdirectories.
#shopt -s globstar
```

B. Afficher les 10 premières lignes du fichier ".bashrc"

La commande **head** ("tête" en français) nous permet de **visualiser les 10 premières lignes** d'un fichier. Pour afficher les 10 premières lignes d'un fichier nous devons entrer la commande suivante : **head** suivi du nom du fichier.

Afin de lire les 10 premières lignes du fichier .bashrc nous allons donc entrer la commande :

head .bashrc



```
vero@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

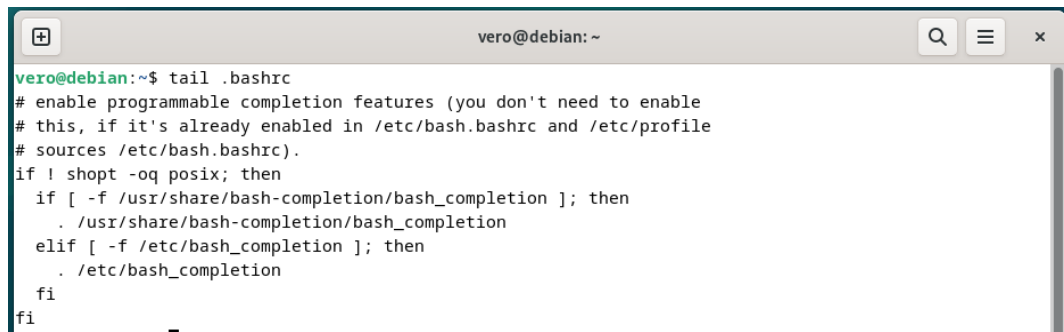
# If not running interactively, don't do anything
case $- in
  *i*) ;;
  *) return;;
esac
```

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

La commande **tail** (“queue” en français) nous permet de **visualiser les 10 dernières lignes** d’un fichier. Pour afficher les 10 dernière lignes d’un fichier nous devons entrer la commande suivante : **tail** suivi du nom du fichier.

Afin de lire les 10 dernières lignes du fichier .bashrc nous allons donc entrer la commande :

```
tail .bashrc
```

A terminal window titled 'vero@debian: ~' showing the output of the command 'tail .bashrc'. The output displays the last few lines of the .bashrc file, which are related to bash completion features. The text is as follows:

```
vero@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
```

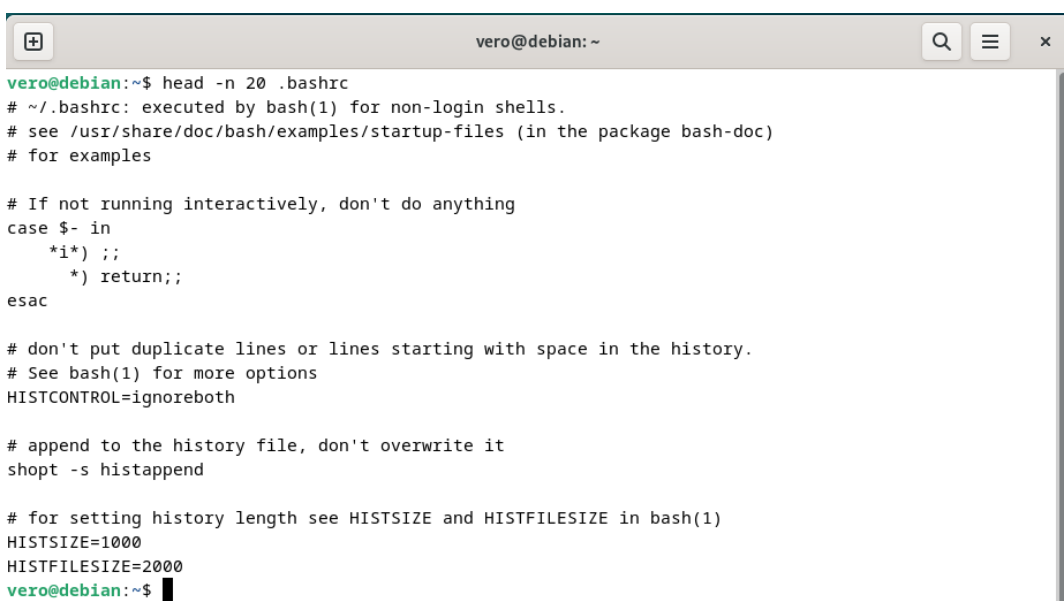
D. Afficher les 20 premières lignes du fichier “.bashrc”

Nous allons utiliser l’option **-n** pour **déterminer le nombre de lignes** que nous souhaitons visualiser.

La commande pour afficher les 20 premières lignes d’un fichier est : **head** suivi de **-n** puis du nombre de lignes que nous souhaitons afficher et du nom du fichier.

Pour lire les 20 première lignes du fichier .bashrc nous allons donc entrer la commande :

```
head -n 20 .bashrc
```

A terminal window titled 'vero@debian: ~' showing the output of the command 'head -n 20 .bashrc'. The output displays the first 20 lines of the .bashrc file, which include comments and configuration for bash completion and history. The text is as follows:

```
vero@debian:~$ 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
vero@debian:~$
```

E. Afficher les 20 dernières lignes du fichier “.bashrc”

La commande pour afficher les 20 dernière lignes d’un fichier : **tail** suivi de **-n** puis du nombre de lignes que nous souhaitons afficher et du nom du fichier

Pour lire les 20 dernière lignes du fichier .bashrc nous allons donc entrer la commande :

```
tail -n 20 .bashrc
```



```
vero@debian:~$ 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
vero@debian:~$
```

III. Job 3

A. Installer le paquet “cmatrix”

Pour installer le paquet **cmatrix** il faut dans un premier temps être en utilisateur “root” en entrant la commande **su** - puis entrer la commande :

```
sudo apt-get install cmatrix cmatrix-xfont vlock
```

- ❖ **sudo** : commande utilisée pour exécuter une commande avec des privilèges administrateur
- ❖ **apt-get** : commande de gestion de paquets APT, utilisée pour installer, mettre à jour et supprimer des logiciels sur des systèmes basés sur Debian
- ❖ **install** : action que nous demandons à **apt-get** d’effectuer (installer des paquets logiciels)
- ❖ **cmatrix** : programme qui affiche une simulation de pluie de caractère ASCII (comme on le voit dans le film “Matrix”)

- ❖ **cmatrix-xfont** : paquet logiciel qui est une police de caractère pour cmatrix qui améliore l'apparence des caractères lors de l'exécution de **cmatrix**
- ❖ **vlock** : paquet logiciel qui est un utilitaire qui nous permet de verrouiller notre terminal avec un mot de passe pour sécuriser notre session

```

vero@debian: ~
vero@debian:~$ su -
Mot de passe :
root@debian:~# sudo apt-get install cmatrix cmatrix-xfont vlock
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Les NOUVEAUX paquets suivants seront installés :
  cmatrix cmatrix-xfont vlock
0 mis à jour, 3 nouvellement installés, 0 à enlever et 0 non mis à jour.
Il est nécessaire de prendre 60,2 ko dans les archives.
Après cette opération, 211 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]
Réception de :2 http://deb.debian.org/debian bookworm/main amd64 cmatrix-xfont all 2.0-3 [8 952 B]
Réception de :3 http://deb.debian.org/debian bookworm/main amd64 vlock amd64 2.2.2-11+b1 [33,8 kB]
60,2 ko réceptionnés en 0s (803 ko/s)
Sélection du paquet cmatrix précédemment désélectionné.
(Lecture de la base de données... 154873 fichiers et répertoires déjà installés.)
Préparation du dépaquetage de .../cmatrix_2.0-3_amd64.deb ...
Dépaquetage de cmatrix (2.0-3) ...
Sélection du paquet cmatrix-xfont précédemment désélectionné.
Préparation du dépaquetage de .../cmatrix-xfont_2.0-3_all.deb ...
Dépaquetage de cmatrix-xfont (2.0-3) ...
Sélection du paquet vlock précédemment désélectionné.
Préparation du dépaquetage de .../vlock_2.2.2-11+b1_amd64.deb ...
Dépaquetage de vlock (2.2.2-11+b1) ...
Paramétrage de cmatrix (2.0-3) ...
Paramétrage de vlock (2.2.2-11+b1) ...
Paramétrage de cmatrix-xfont (2.0-3) ...
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) ...
Traitement des actions différées (« triggers ») pour mailcap (3.70+nmul) ...
Traitement des actions différées (« triggers ») pour fontconfig (2.14.1-4) ...
root@debian:~#

```

B. Lancer le paquet que vous venez d'installer

Pour lancer le paquet **cmatrix**, il suffit d'entrer la commande :

cmatrix

```

vero@debian: ~
3 D&1Q $D3)@) @ , m5Uo= ^ [ x V`>
/ F P C %32pym A z , ) J X m $ t ` 2 _ x e
Y \ y Y ( < j X @ ! C : Y i Z k 8 ) c Z & k $ 9 r *
H Q A W k H ! q 7 6 ! S g H [ f p U % 4 o " 8
$ k " C ; j x x 5 f N l = p 0 ] 4 q ) 9 L 0
P Y o H ` g ! . Q 0 t ? J m : [ z p / 0 * c 4
% S ! [ F n K ' @ a F b 3 _ A u 8 > ) f o o N h l
" p h 4 ( : y F m 8 * U n Y o 7 # ^ x f ' z l r
Z < A O P . Y z P i & ] E R < Z ^ $ x l p t u Q
7 C f K ' 4 s _ t g 4 5 1 X j " [ 7 x _ r &
k E , p I z * , : ? o o o ! & + E C X + K
4 I % \ L f P _ ] ^ \ t ` ' ] m Q 0 - [ [
_ . h [ N T C 0 h e 1 o S S n u z s D @ -
L ' m 0 e " r o D 5 0 I w 7 9 j F Q d k @ Y
B P 8 P g A B : Q m Q B ' ; 4 h o 9 K ?
L y b Y ; > f & S h ^ I , D w E R = 9
J b t 8 : ' \ n u / ( 8 K C ) E l l R 4 Y j Z
g j Y I " C ' I ) P k 5 \ B 6 u o a E )
l $ B / m + S A x & u o 6 s C 4 ? b t G k
4 Z 4 z < F ^ C 9 s v X T - X j j ' u c J J
D E s m P M ^ @ ) ] H 3 k f @ x M u c d j >
n . r d N r l d M d f ` ; x Y Y \ k w j S
y q N s " c _ = D r q 0 ; 6 6 % 2 ) l L * z
K ( q z h ? I x w 8 Y m < _ 7 Z 0 + I 6 i *
m $ I o F Q N X L " V 2 x / Z = 0 1 W $ j E V

```

C. Mettre à jour de paquets

son gestionnaire

Pour la mise à jour des paquets il faut entrer la commande (en tant qu'utilisateur "**root**") :

apt update

```
root@debian:~# apt 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
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Tous les paquets sont à jour.
root@debian:~#
```

D. Mettre à jour ses différents logiciels

Pour mettre à jour ses différents logiciels il faut entrer la commande (en tant qu'utilisateur "**root**") :

apt upgrade

```
root@debian:~# 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 :
  firefox-esr firefox-esr-l10n-fr
2 mis à jour, 0 nouvellement installés, 0 à enlever et 0 non mis à jour.
Il est nécessaire de prendre 0 o/63,6 Mo dans les archives.
Après cette opération, 13,8 Mo d'espace disque supplémentaires seront utilisés.
Souhaitez-vous continuer ? [O/n] o
Lecture des fichiers de modifications (« changelog »)... Terminé
(Lecture de la base de données... 154919 fichiers et répertoires déjà installés.)
Préparation du dépaquetage de .../firefox-esr-l10n-fr_115.3.0esr-1~deb12u1_all.deb ...
Dépaquetage de firefox-esr-l10n-fr (115.3.0esr-1~deb12u1) sur (102.15.1esr-1~deb12u1) ...
Préparation du dépaquetage de .../firefox-esr_115.3.0esr-1~deb12u1_amd64.deb ...
Conservation de « détournement de /usr/bin/firefox en /usr/bin/firefox.real par firefox-esr »
Dépaquetage de firefox-esr (115.3.0esr-1~deb12u1) sur (102.15.1esr-1~deb12u1) ...
Paramétrage de firefox-esr (115.3.0esr-1~deb12u1) ...
Paramétrage de firefox-esr-l10n-fr (115.3.0esr-1~deb12u1) ...
Traitement des actions différées (« triggers ») pour desktop-file-utils (0.26-1) ...
Traitement des actions différées (« triggers ») pour hicolor-icon-theme (0.17-2) ...
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) ...
Traitement des actions différées (« triggers ») pour mailcap (3.70+nmu1) ...
```

E. Télécharger les internets : Google

Pour installer Google via le terminal linux, il faut dans un premier temps être en utilisateur "**root**" :

- installer **wget** (un programme en ligne de commande non interactif de téléchargement de fichiers depuis le Web) en utilisant la commande : **apt install wget**
- pour installer google il faut utiliser la commande :

wget apt install google.com

```
vero@debian:~$ su -
Mot de passe :
root@debian:~# wget apt install google.com
--2023-09-27 15:05:53-- http://apt/
Résolution de apt (apt)... google
échec : Nom ou service inconnu.
wget : impossible de résoudre l'adresse de l'hôte « apt »
--2023-09-27 15:06:00-- http://install/
```


F. Redémarrer votre machine

Pour redémarrer notre machine nous pouvons utiliser 2 commandes en utilisateur “**root**” :

```
sudo reboot
```

```
sudo shutdown -r now
```

G. Éteindre votre machine

Pour éteindre notre machine machine nous pouvons utiliser 2 commandes en étant en utilisateur “**root**” :

```
poweroff
```

```
shutdown -h now
```

IV. Job 4

Créer un fichier users.txt qui contiendra User1 et User2 séparé par un retour à la ligne :

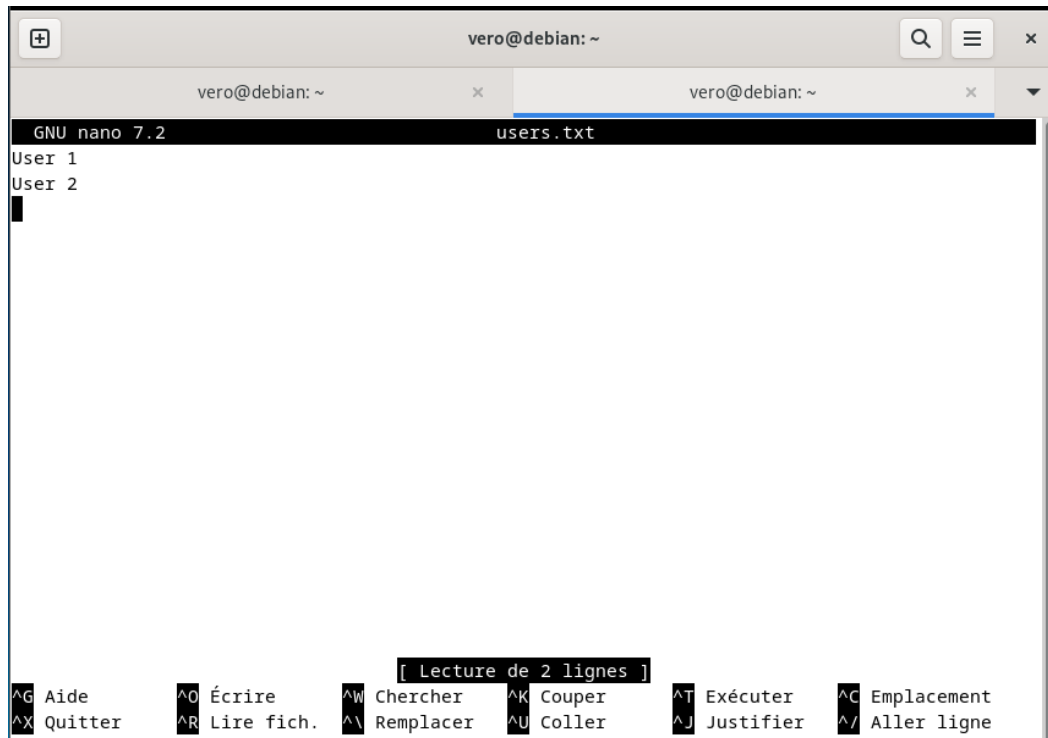
Pour créer un fichier il faut utiliser la commande **touch** suivi du nom du fichier, dans nos cas nous appellerons notre fichier “users.txt” :

```
touch users.txt
```

Pour éditer notre fichier “users.txt”, nous allons utiliser la commande nous allons utiliser la commande **nano** suivi du nom du fichier :

nano users.txt

L'interface de traitement de texte nous permettra d'écrire "User 1" et "User 2" espacer d'un retour à la ligne.



```
GNU nano 7.2 users.txt
User 1
User 2

```

[Lecture de 2 lignes]

^G Aide	^O Écrire	^W Chercher	^K Couper	^T Exécuter	^C Emplacement
^X Quitter	^R Lire fich.	^E Remplacer	^U Coller	^J Justifier	^L Aller ligne

A. Créer un groupe appelé "Plateformeurs"

Il faut se mettre en utilisateur sudo avec la commande **su -**

Pour créer un groupe d'utilisateurs il faut utiliser la commande : **groupadd -g groupeID nomdugroupe**

Le **groupe ID (ou GID)** est une **valeur numérique** représentant un **groupe spécifique**. L'étendue de valeurs pour un GID varie entre les différents systèmes; un GID **peut être compris entre 0 et 32767**, avec une restriction : le groupe pour le super-utilisateur doit avoir le GID 0.

Dans notre cas notre GID se nom de notre groupe sera "Plateformeurs" donc :

groupadd -g 2000 Plateformeurs

B. Créer un utilisateur appelé “User1”

Pour créer un utilisateur appelé User1 nous utiliserons la commande `useradd` suivie du nom de l'utilisateur en étant en utilisateur “**root**” donc :

```
useradd User1
```

Nous procéderons de la même manière pour la création de l'utilisateur User2.

C. Ajouter “User2” au groupe Plateformeurs

Afin d'ajouter notre utilisateur User2 dans le groupe Plateformeurs nous allons utiliser la commande suivante :

```
usermod -G nomgroupe nomutilisateur
```

Nous allons donc entrer la commande :

```
usermod -G Plateformeurs User2
```

```
|root@debian:~# usermod -G Plateformeurs User2
```

D. Copier votre “users.txt” dans un fichier “droits.txt”

Pour **copier le contenu d'un fichier** dans un autre fichier situé dans le même répertoire, nous utiliserons la commande **cp** en entrant : **cp fichier1 fichier2**

“fichier1” est le fichier à copier et “fichier2” est la copie du fichier. Si “fichier2” n'existe pas avant la copie, il sera créé.

Dans notre cas nous allons copier le fichier `users.txt` au fichier `droits.txt` donc nous allons entrer la commande suivante :

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

Nous procéderons de la même manière pour copier le fichier `users.txt` dans `groupes.txt` donc :

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

```
|vero@debian:~$ cp users.txt droits.txt
```

E. Changer le propriétaire du fichier “droits.txt” pour mettre “User1”

La commande qui nous permet de voir qui est propriétaire d'un fichier est : **ls -al**

Nous utiliserons la commande suivante pour **changer le propriétaire du fichier** “droits.txt” pour mettre “User1”.

❖ La commande **sudo chown** (change owner) permet de **changer le propriétaire** et/ou le groupe.

Changer le propriétaire d'un fichier :

```
sudo chown nouvelutilisateur nomfichier
```

Dans notre cas nous allons donc entrer la commande :

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

F. Changer les droits du fichier “droits.txt” pour que “User2” ai accès seulement en lecture

Tout d’abord il faut s’assurer d’avoir bien installer ACL afin de précéder à la suite en utilisant la commande :

```
sudo apt-get install acl
```

Puis utiliser la commande suivante : **setfacl -m u:nomutilisateur:r nomfichier**

Nous allons donc entrer la commande :

```
setfacl -m u:User2:r droits.txt
```

G. Changer les droits du fichier “groupes.txt” pour que les utilisateurs puissent accéder au fichier en lecture uniquement

La commande **chmod** (change mode) permet de **modifier les droits** sur un fichier.

Afin de donner de permettre à tous les utilisateurs d’**accéder en lecture uniquement** au fichier “groupes.txt”,

Il faut utiliser la commande suivante :

```
chmod o+r groupes.txt
```

❖ **o** = signifie “**others**”

❖ **r** = signifie “**read**”

H. Changer les droits du fichier “groupes.txt” pour que le groupe “Plateformeurs” puissent y accéder en lecture/écriture

Nous allons dans un premier temps ajouter le groupe “Plateformeurs” comme nouveau propriétaire à l’aide de la commande **chown propriétaire:groupepropriétaire nomdufichier** donc nous allons entrer la commande suivante :

```
sudo chown vero:Plateformeurs groupes.txt
```

Puis nous allons attribuer les droits de lecture et d’écriture au groupe “Plateformeurs” à l’aide de la commande **chmod** en utilisant le système **octal** (système de notation numérique utilisé pour représenter les autorisations de fichiers de manière concise et compréhensible.

Ce système est **souvent utilisé avec la commande chmod** pour définir les autorisations de fichiers en utilisant des chiffres octaux).

Chaque **chiffre correspond à une autorisation** que nous pouvons **combinaison** pour ajouter des autorisations et la position des chiffres correspond à propriétaire/groupe/others :

- ❖ r (read) = 4
- ❖ w (write) = 2
- ❖ x (execute) = 1
- ❖ - = 0

Nous allons donc entrer la commande :

```
chmod 760 groupes.txt
```

Nous pouvons vérifier que les modifications ont bien été apportées en utilisant la commande **ls -al**

```
| -rwxr--r-- 1 vero Plateformeurs 14 27 sept. 16:07 groupes.txt
```

V. Job 5

A. Ajouter un alias qui permettra de lancer la commande “ls -la” en tapant “la”

La création d’**alias** dans le terminal Linux présente **plusieurs avantages** et peut rendre votre expérience en **ligne de commande plus efficace et pratique** (par exemple en permettant de raccourcir des commandes longues et complexes).

La commande pour ajouter un alias est : **alias raccourci="commande"**

Pour ajouter un alias qui permettra de lancer la commande **ls -la** en tapant **la** il faudra donc entrer la commande suivante :

alias la="ls -al"

```
vero@debian:~$ alias la="ls -al"
vero@debian:~$ la
total 92
drwx----- 13 vero sudo          4096 28 sept. 13:13 .
drwxr-xr-x  3 root root          4096 25 sept. 18:29 ..
-rw-----  1 vero sudo          2473 28 sept. 14:08 .bash_history
-rw-r--r--  1 vero sudo           220 25 sept. 18:29 .bash_logout
-rw-r--r--  1 vero sudo          3526 25 sept. 18:29 .bashrc
drwxr-xr-x  2 vero sudo          4096 25 sept. 18:34 Bureau
drwx-----  9 vero sudo          4096 27 sept. 11:10 .cache
drwx----- 10 vero sudo          4096 27 sept. 11:10 .config
drwxr-xr-x  2 vero sudo          4096 27 sept. 11:11 Documents
-rwxr-----+ 1 User1 sudo           14 27 sept. 16:03 droits.txt
-rw-r--r--  1 vero sudo          5290 25 sept. 18:29 .face
lrwxrwxrwx  1 vero sudo           5 25 sept. 18:29 .face.icon -> .face
-rwxrw----  1 vero Plateformeurs  14 27 sept. 16:07 groupes.txt
drwxr-xr-x  3 vero sudo          4096 25 sept. 18:34 Images
-rw-----  1 vero sudo           20 28 sept. 13:13 .lessht
drwx-----  4 vero sudo          4096 25 sept. 18:34 .local
drwxr-xr-x  2 vero sudo          4096 25 sept. 18:34 Modèles
drwxr-xr-x  2 vero sudo          4096 25 sept. 18:34 Musique
-rw-r--r--  1 vero sudo           807 25 sept. 18:29 .profile
drwxr-xr-x  2 vero sudo          4096 25 sept. 18:34 Public
-rw-r--r--  1 vero vero           0 27 sept. 16:55 .sudo_as_admin_successful
drwxr-xr-x  2 vero sudo          4096 25 sept. 18:34 Téléchargements
-rw-r--r--  1 vero sudo           14 27 sept. 15:18 users.txt
drwxr-xr-x  2 vero sudo          4096 25 sept. 18:34 Vidéos
vero@debian:~$
```

B. Ajouter un alias qui permettra de lancer la commande “apt-get update” en tapant “update”

Pour ajouter un alias qui permettra de lancer la commande **apt-get update** en tapant **update** il faudra donc entrer la commande suivante :

alias update="apt-get update"

```
vero@debian:~$ su -
Mot de passe :
root@debian:~# alias update="apt-get update"
root@debian:~# update
Atteint :1 http://deb.debian.org/debian bookworm InRelease
Atteint :2 http://deb.debian.org/debian bookworm-updates InRelease
Réception de :3 http://security.debian.org/debian-security bookworm-security InRelease [48
,0 kB]
48,0 ko réceptionnés en 1s (62,5 ko/s)
Lecture des listes de paquets... Fait
root@debian:~#
```

C. Ajouter un alias qui permettra de lancer la commande “apt-get upgrade” en tapant “upgrade”

Pour ajouter un alias qui permettra de lancer la commande **apt-get upgrade** en tapant **upgrade** il faudra donc entrer la commande suivante :

```
alias upgrade="apt-get upgrade"
```

```
root@debian:~# alias upgrade="apt-get upgrade"
root@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
0 mis à jour, 0 nouvellement installés, 0 à enlever et 0 non mis à jour.
root@debian:~#
```

D. Ajouter une variable d'environnement qui se nommera “USER” et qui sera égale à votre nom d'utilisateur

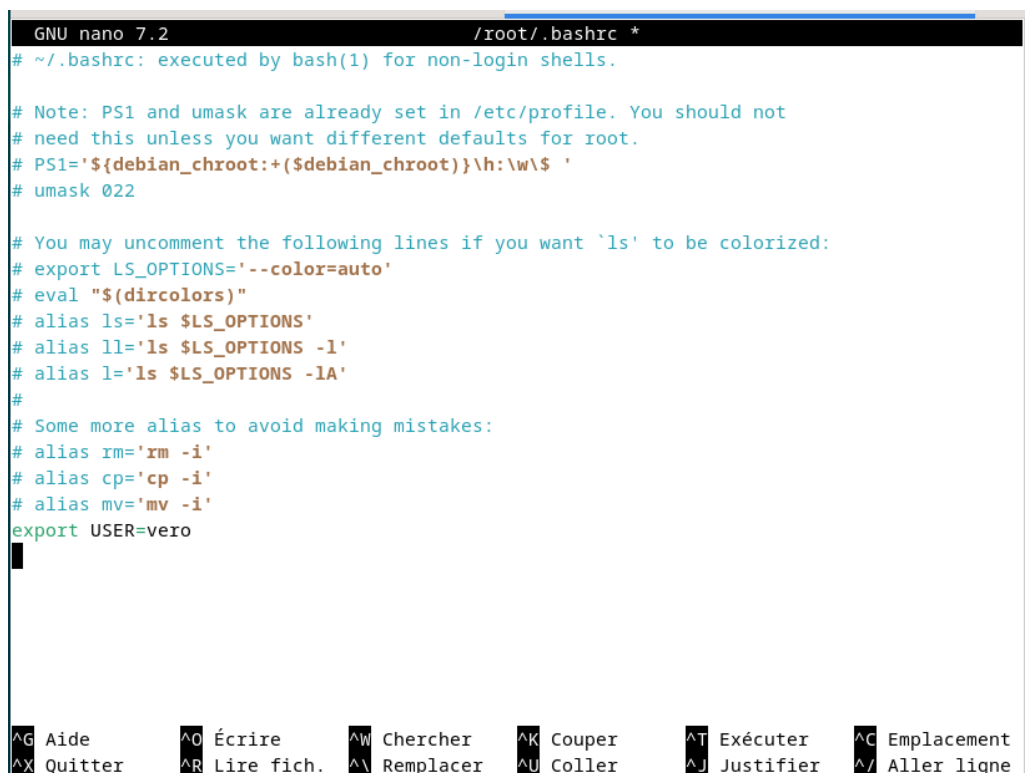
- Afin d'ajouter une variable d'environnement il faudra dans un premier temps apporter des modifications dans le fichier **.bashrc** à l'aide de la commande **nano** :

```
nano ~/.bashrc
```

- Une fois dans l'interface de traitement de texte, nous allons ajouter notre nouvelle variable d'environnement à la fin du fichier de la manière suivante **export NOMVARIABLE=NOMCONTENU**:

Nous entrerons donc la commande suivante :

```
export USER=vero
```



```
GNU nano 7.2 /root/.bashrc *
# ~/.bashrc: executed by bash(1) for non-login shells.

# Note: PS1 and umask are already set in /etc/profile. You should not
# need this unless you want different defaults for root.
# PS1='${debian_chroot:+($debian_chroot)}\h:\w\$ '
# umask 022

# You may uncomment the following lines if you want 'ls' to be colorized:
# export LS_OPTIONS='--color=auto'
# eval "$(dircolors)"
# alias ls='ls $LS_OPTIONS'
# alias ll='ls $LS_OPTIONS -l'
# alias l='ls $LS_OPTIONS -lA'
#
# Some more alias to avoid making mistakes:
# alias rm='rm -i'
# alias cp='cp -i'
# alias mv='mv -i'
export USER=vero
```

E. Mettre à jour les modifications de votre bashrc dans votre shell actuel

- Nous allons maintenant **appliquer les modifications** apportées sur le fichier **.bashrc** à l'aide de la commande suivante :

```
source ~/.bashrc
```

F. Afficher les variables d'environnement

- Nous pouvons afficher la nouvelle variable d'environnement en utilisant la commande :

```
echo $USER
```

G. Ajouter à votre Path le chemin "/home/votre utilisateur/Bureau"

- Pour ajouter un répertoire au chemin **\$PATH** il faudra dans un premier temps apporter des modifications dans le fichier **.bashrc** à l'aide de la commande **nano** :

```
nano ~/.bashrc
```

- Une fois dans l'interface de traitement de texte, nous allons ajouter notre nouveau chemin à la fin du fichier de la manière suivante :

```
export PATH=$PATH:/home/vero/Bureau
```

```
GNU nano 7.2 /root/.bashrc
# ~/.bashrc: executed by bash(1) for non-login shells.

# Note: PS1 and umask are already set in /etc/profile. You should not
# need this unless you want different defaults for root.
# PS1='${debian_chroot:+($debian_chroot)}\h:\w\$ '
# umask 022

# You may uncomment the following lines if you want 'ls' to be colorized:
# export LS_OPTIONS='--color=auto'
# eval "$(dircolors)"
# alias ls='ls $LS_OPTIONS'
# alias ll='ls $LS_OPTIONS -l'
# alias l='ls $LS_OPTIONS -lA'
#
# Some more alias to avoid making mistakes:
# alias rm='rm -i'
# alias cp='cp -i'
# alias mv='mv -i'
export USER=vero
export PATH=$PATH:/home/vero/Bureau
```

- Nous allons maintenant appliquer les modifications apportées sur le fichier **.bashrc** à l'aide de la commande suivante :

```
source ~/.bashrc
```


- Nous pouvons vérifier que le nouveau chemin a été ajouté en entrant la commande :

```
echo $PATH
```

VI. Job 6

Vous devez télécharger l'archive suivante et la désarchiver seulement avec le terminal

GNU **tar** est un programme d'**archivage** conçu pour **stocker** plusieurs fichiers dans un seul fichier (une archive), et de **manipuler ces archives**.

- Après avoir téléchargé le fichier "Ghost in the Shell.tar", nous allons copier dans nos téléchargement le fichier pour ensuite le coller dans notre dossier "fichier" de notre VM Debian.
- Ensuite nous allons dans notre dossier Téléchargement à l'aide de la commande **cd** afin d'extraire le fichier "Ghost in the Shell.tar" et l'avoir en format **.pdf** à l'aide de la commande **tar** et des options **-xvf** (x = extraire l'archive, v = affiche les informations verbose, f = crée des archives avec le nom de fichier donné) :

```
tar -xvf /home/vero/Téléchargements/Ghost\ in\ the\ Shell.tar ._Ghost in the Shell.pdf
```

Signification options -xvf :

- **x** : extraire l'archive
- **v** : affiche les informations verboses
- **f** : crée des archives avec le nom de fichier donné

```
vero@debian:~$ cd Téléchargements/
vero@debian:~/Téléchargements$ ls
'Ghost in the Shell.tar'
vero@debian:~/Téléchargements$ tar -xvf /home/vero/Téléchargements/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 étendu a été ignoré
Ghost in the Shell.pdf
vero@debian:~/Téléchargements$ ls
'Ghost in the Shell.pdf' 'Ghost in the Shell.tar'
vero@debian:~/Téléchargements$
```

VII. Job 7

- Créer un fichier “une_commande.txt” avec le texte suivant “Je suis votre fichier texte” :

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

- Compter le nombre de lignes présentes dans votre fichier de source apt et les enregistrer dans un fichier nommé “nb_lignes.txt” :

fichier source apt : `/etc/apt/sources.list`

commande pour compter le nbr de lignes dans un fichier : **wc** suivi de l’option **-l** pour imprimer le

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

- Afficher le contenu du fichier source apt et l’enregistrer dans un autre fichier appelé “save_sources” :
- commande **cat** pour afficher le contenu d’un fichier et la redirection **>** va nous permettre de l’enregistrer dans un fichier
- commande pour afficher le contenu du fichier source apt et de l’enregistrer dans un fichier que nous nommerons “save_sources” :

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

- Faites une recherche des fichiers commençant par “.” tout en cherchant le mot alias qui sera utilisé depuis un fichier

La commande **find** (suivi du chemin à partir duquel nous souhaitons commencer la recherche) pour effectuer une recherche de fichiers ou répertoires, nous ajoutons l’option **-type f** pour préciser que nous recherchons des fichiers et l’option **-name “.*”** pour spécifier que nous recherchons des fichiers commençant par un “.”

Nous allons par la ajouter la commande **-exec** qui servira à exécuter la commande **grep**. Celle-ci nous permettra de trouver tous les fichiers qui contiennent le mot “alias”. “{ }” est un espace réservé pour le nom de chaque fichier trouvé et “\;” sert à mettre fin à la commande

```
-exec find /home/vero -type f -name “.*” -exec grep “alias” {} \;
```

Nous pouvons maintenant combiner toutes ces commandes pour qu'elles ne fassent qu'une ligne à l'aide de caractères qui permettront de faire la liaison entre elles :

```
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 && find /home/vero -type f -name ".*" -exec grep "alias" {} \;
```

```
vero@debian:~$ echo "Je suis votre fichier test" > une_commande.txt && wc -l /etc/apt/sources.  
list > nb_lignes.txt && cat /etc/apt/sources.list > save_sources && find /home/vero -type f -n  
ame ".*" -exec grep "alias" {} \;  
# enable color support of ls and also add handy aliases  
alias ls='ls --color=auto'  
#alias dir='dir --color=auto'  
#alias vdir='vdir --color=auto'  
#alias grep='grep --color=auto'  
#alias fgrep='fgrep --color=auto'  
#alias egrep='egrep --color=auto'  
# some more ls aliases  
#alias ll='ls -l'  
#alias la='ls -A'  
#alias l='ls -CF'  
# ~/.bash_aliases, instead of adding them here directly.  
if [ -f ~/.bash_aliases ]; then  
    . ~/.bash_aliases  
alias update="apt-get update"  
alias update="apt-get update"  
alias update="apt-get update"  
alias upgrade="apt-get upgrade"  
alias upgrate="apt-get upgrade"  
alias la="ls -al"
```