

JOB 1

Comment ajouter des options à une commande ?

Beaucoup de commandes ont des options supplémentaires à ajouter à cette dernière. On ajoute des options à une commande avec un '-' suivi d'une lettre ou alors avec un '--' suivi d'un mot.

`tree --help`

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

Les deux principales syntaxes d'écriture d'options pour une commande sur linux sont les suivantes :

1. [commande] [-] [option] ex: `man -h`
2. [commande] [--] [option_longue] ex: `man --help`

Afficher le manuel de la commande ls :

`man ls`

Afficher les fichiers cachés du home de votre utilisateur:

`ls -a`

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

`ls -la`



```
moufid@debian: ~  
drwxr-xr-x 3 root root 4096 3 oct. 10:12 ..  
-rw-r--r-- 1 moufid moufid 0 4 oct. 09:39 :1  
-rw----- 1 moufid moufid 7008 3 oct. 20:49 .bash_history  
-rw-r--r-- 1 moufid moufid 220 3 oct. 10:12 .bash_logout  
-rw-r--r-- 1 moufid moufid 3552 3 oct. 16:17 .bashrc  
drwxr-xr-x 2 moufid moufid 4096 3 oct. 19:58 Bureau  
drwx----- 14 moufid moufid 4096 3 oct. 16:42 .cache  
drwx----- 12 moufid moufid 4096 3 oct. 16:29 .config  
drwxr-xr-x 2 moufid moufid 4096 3 oct. 10:13 Documents  
drwx----- 2 moufid moufid 4096 3 oct. 16:30 .gnupg  
drwxr-xr-x 2 moufid moufid 4096 3 oct. 10:13 Images  
drwx----- 3 moufid moufid 4096 3 oct. 10:13 .local  
drwxr-xr-x 2 moufid moufid 4096 3 oct. 10:13 Modèles  
drwx----- 4 moufid moufid 4096 3 oct. 10:14 .mozilla  
drwxr-xr-x 2 moufid moufid 4096 3 oct. 10:13 Musique  
drwx----- 3 moufid moufid 4096 3 oct. 16:29 .pki  
-rw-r--r-- 1 moufid moufid 807 3 oct. 10:12 .profile  
drwxr-xr-x 2 moufid moufid 4096 3 oct. 10:13 Public  
drwx----- 2 moufid moufid 4096 3 oct. 10:18 .ssh  
drwxr-xr-x 2 moufid moufid 4096 3 oct. 16:43 Téléchargements  
-rw-r--r-- 1 moufid moufid 679 4 oct. 09:40 tree.save  
drwxr-xr-x 2 moufid moufid 4096 3 oct. 10:13 Vidéos  
-rw----- 1 moufid moufid 9191 4 oct. 09:40 .viminfo  
moufid@debian:~$
```

JOB 2

Pouvoir lire un nombre précis de ligne à partir du début ou fin de fichier pourrait être très pratique, pour ce faire il existe une commande.

On va afficher les 10 première lignes du fichier .bashrc :

- Première étape, se rendre là où se trouve le fichier :

```
moufid@debian:~$ cd ..  
moufid@debian:/home$ cd ..  
moufid@debian:/$ cd etc  
moufid@debian:/etc$
```

le fichier bash.bashrc se trouve dans /etc.

- Une fois au bon endroit, on exécute la commande : `head -n 10 .bashrc`
`head` = pour les première lignes ; `-n` = pour afficher les lignes ; `10` = le nombre de lignes ; fichier

- afficher les 10 premières lignes du fichier “.bashrc” : `head -n 10 bashrc`
- afficher les 10 dernières lignes du fichier “.bashrc” : `tail -n 10 bashrc`
- afficher les 20 premières lignes du fichier “.bashrc” : `head -n 20 bashrc`
- afficher les 20 dernières lignes du fichier “.bashrc” : `tail -n 20 bashrc`

```
moufid@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  
moufid@debian:~$
```

```
moufid@debian:~$ 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 $- in  
  *i*) ;;  
  *) return;;  
esac  
moufid@debian:~$
```

```
moufid@debian:~$ tail -n 10 .bashrc  
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  
  
export USER=/home/moufid  
export PATH=$PATH:/home/moufid/Bureau  
moufid@debian:~$
```

```
moufid@debian:~$ tail -n 20 .bashrc  
# ~/.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  
  
export USER=/home/moufid  
export PATH=$PATH:/home/moufid/Bureau  
moufid@debian:~$
```

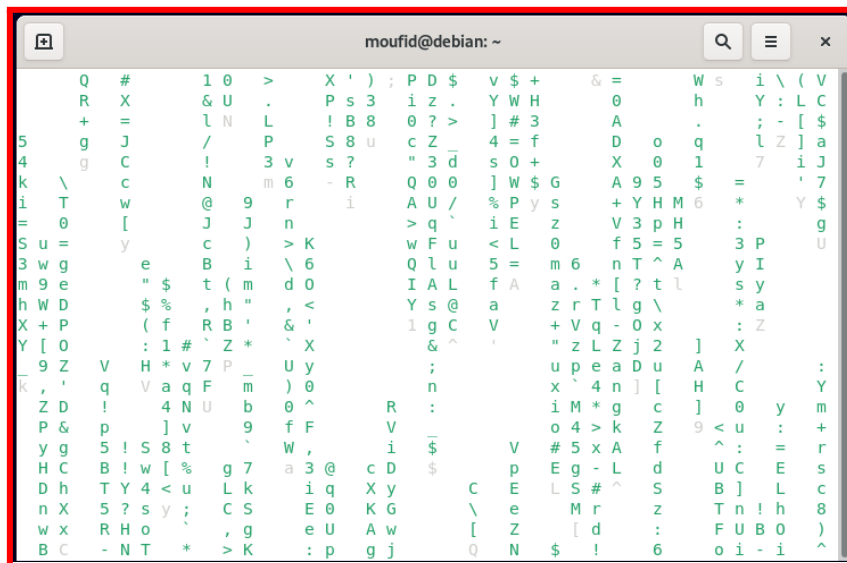
JOB 3

Installer le paquet “cmatrix” :

`sudo apt-get install cmatrix`

Lancer le paquet qu’on vient d’installer :


`cmatrix`



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

Mettre à jour son gestionnaire de paquets et ses logiciels :

`sudo apt-get update` pour lister les mises à jour disponibles et `sudo apt-get upgrade` pour les installer



```
moufid@debian: ~  
moufid@debian:~$ sudo apt-get update && sudo apt-get upgrade
```

Redémarrer sa machine :

`sudo reboot`

Éteindre sa machine :

`sudo shutdown -h [temps]`

exemple : `sudo shutdown -h now`

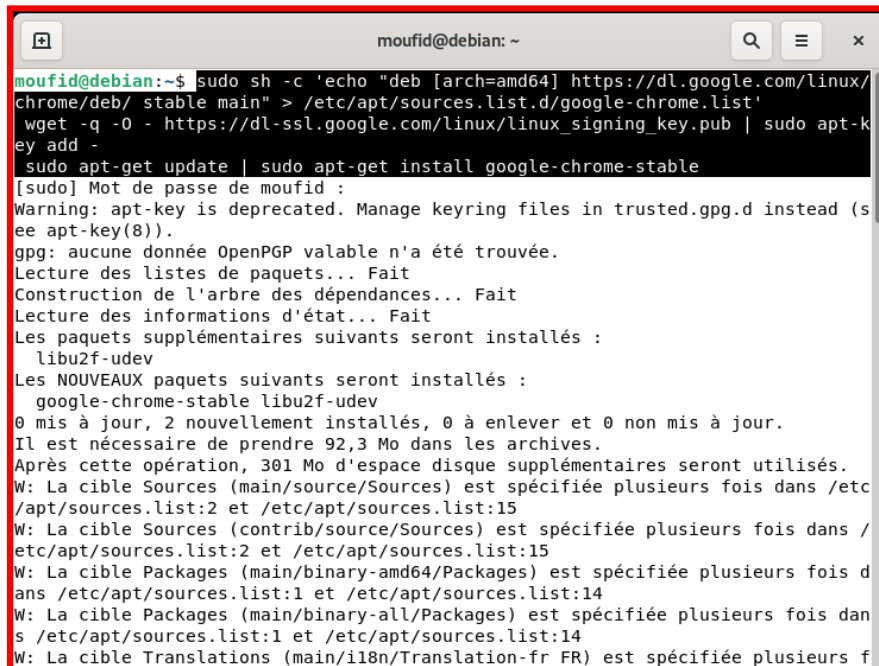


```
moufid@debian: ~  
moufid@debian:~$ sudo shutdown -h 500  
[sudo] Mot de passe de moufid :  
Shutdown scheduled for Thu 2022-10-06 19:06:07 CEST, use 'shutdown -c' to cancel
```

Télécharger Google Chrome :

Pour télécharger Google Chrome à l'aide du terminal de commande uniquement et avec une seule commande, on va faire :

```
sudo sh -c 'echo "deb [arch=amd64] https://dl.google.com/linux/chrome/deb/ stable main" > /etc/apt/sources.list.d/google-chrome.list'
wget -q -O - https://dl-ssl.google.com/linux/linux_signing_key.pub | sudo apt-key add -
sudo apt-get update | sudo apt-get install google-chrome-stable
```

A terminal window titled 'moufid@debian: ~' showing the execution of the commands to install Google Chrome. The output shows the addition of the Google Chrome repository, the download of the signing key, and the successful installation of 'google-chrome-stable' and 'libu2f-udev'.

```
moufid@debian:~$ sudo sh -c 'echo "deb [arch=amd64] https://dl.google.com/linux/chrome/deb/ stable main" > /etc/apt/sources.list.d/google-chrome.list'
moufid@debian:~$ wget -q -O - https://dl-ssl.google.com/linux/linux_signing_key.pub | sudo apt-key add -
moufid@debian:~$ sudo apt-get update | sudo apt-get install google-chrome-stable
[sudo] Mot de passe de moufid :
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
gpg: aucune donnée OpenPGP valable n'a été trouvée.
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
  libu2f-udev
Les NOUVEAUX paquets suivants seront installés :
  google-chrome-stable libu2f-udev
0 mis à jour, 2 nouvellement installés, 0 à enlever et 0 non mis à jour.
Il est nécessaire de prendre 92,3 Mo dans les archives.
Après cette opération, 301 Mo d'espace disque supplémentaires seront utilisés.
W: La cible Sources (main/source/Sources) est spécifiée plusieurs fois dans /etc/apt/sources.list:2 et /etc/apt/sources.list:15
W: La cible Sources (contrib/source/Sources) est spécifiée plusieurs fois dans /etc/apt/sources.list:2 et /etc/apt/sources.list:15
W: La cible Packages (main/binary-amd64/Packages) est spécifiée plusieurs fois dans /etc/apt/sources.list:1 et /etc/apt/sources.list:14
W: La cible Packages (main/binary-all/Packages) est spécifiée plusieurs fois dans /etc/apt/sources.list:1 et /etc/apt/sources.list:14
W: La cible Translations (main/i18n/Translation-fr FR) est spécifiée plusieurs f
```

JOB 4

On va maintenant s'intéresser aux commandes pour gérer les utilisateurs, les permissions ...

Afin de créer un fichier nommé users.txt qui contiendra le texte "User1" et "User2" séparé par un retour à la ligne, on va utiliser la commande echo : `echo -e "User1\nUser2" >users.txt`

```
moufid@debian:~$ echo -e "User1\nUser2" >users.txt
```

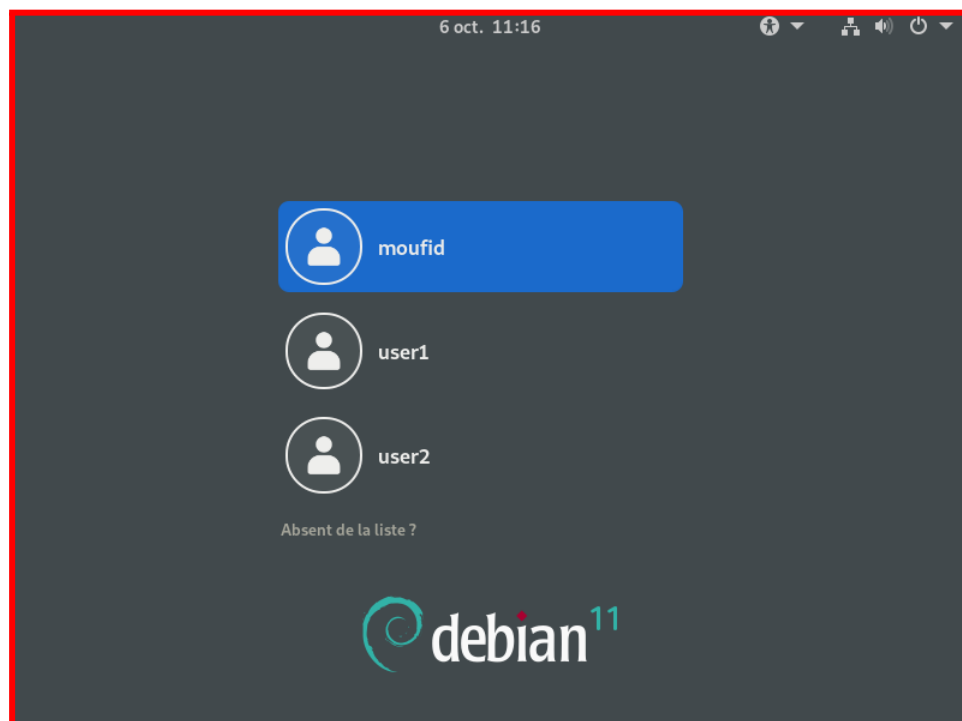
Pour créer un groupe qu'on appellera 'Plateformeurs', on utilise la commande suivante : `sudo groupadd Plateformeurs`

```
moufid@debian:~$ sudo groupadd Plateformeurs
[sudo] Mot de passe de moufid :
```

Pour créer des utilisateurs qu'on appellera User1 and User2, on utilise la commande suivante : `sudo adduser User1 ; sudo adduser User2`

```
moufid@debian:~$ sudo adduser User1
```

```
moufid@debian:~$ sudo adduser User2
```



Ajouter l'utilisateur qu'on vient de créer dans le groupe 'Plateformeurs' : `sudo usermod -a -G Plateformeurs User2`

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

On va maintenant copier les éléments présents dans le fichier `users.txt` dans deux autres fichiers qu'on appellera '`droits.txt`' et '`groupes.txt`' avec la commande `cp` :

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

Le propriétaire par défaut d'un fichier est celui qui le crée, on peut néanmoins le changer par le biais de la commande `chown` :

```
moufid@debian:~$ sudo chown User1 droits.txt
```

Pour changer les droits d'un fichier pour un utilisateur ou un groupe entier, on utilise la commande `chmod` :

```
moufid@debian:~$ sudo chmod o+r,o-w,o-x droits.txt
```

ou alors

```
moufid@debian:~$ sudo chmod 004 droits.txt
```

Pourquoi 004 ?

Chmod 004

	Owner Rights (u)	Group Rights (g)	Others Rights (o)
Read (4)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Write (2)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Et cette fois on va changer les droits du fichier `groupes.txt` pour que tous les utilisateurs puissent accéder au fichier en lecture uniquement :

```
moufid@debian:~$ sudo chmod a+r,a-w,a-x groupes.txt
```

ou alors

```
moufid@debian:~$ sudo chmod 444 groupes.txt
```

Pourquoi 444 ?

Chmod 444

	Owner Rights (u)	Group Rights (g)	Others Rights (o)
Read (4)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Write (2)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Execute (1)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Pour finir on va changer les droits pour que le groupe 'Plateformeurs' puisse accéder à l'écriture et à la lecture sur le fichier groupes.txt :

```
moufid@debian:~$ sudo chmod 464 groupes.txt
```

Pourquoi 464

Chmod 464

	Owner Rights (u)	Group Rights (g)	Others Rights (o)
Read (4)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Write (2)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Execute (1)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

JOB 5

Les **alias** sont des substitutions abrégées de commandes répétitives qui sont plus ou moins longues à taper dans la console. C'est un excellent moyen de **gagner du temps**.

On ajoutera un alias qui nous permettra de lancer la commande 'ls -la' en tapant uniquement 'la' :

```
moufid@debian:~$ alias la='ls -la'
```

On ajoutera également un alias qui nous permettra de lancer la commande 'apt-get update' en tapant uniquement 'update' :

```
moufid@debian:~$ alias update='sudo apt-get update'
```

Et enfin pour finir, un dernier alias pour nous permettre de lancer la commande 'apt-get upgrade' en tapant uniquement 'upgrade' :

```
moufid@debian:~$ alias upgrade='sudo apt-get upgrade'
```

A présent, nous allons ajouter une variable d'environnement qui se nommera 'USER' et qui sera égale à mon nom d'utilisateur, soit 'moufid' :

Premièrement, on fait :

```
moufid@debian:~$ sudo nano /home/moufid/ .bashrc
```

Ensuite on modifie le fichier .bashrc (grâce à la commande nano) en ajoutant au fichier :

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

Et pour mettre à jour les modifications qu'on vient de mettre à notre dans notre shell actuel fichier bashrc, on utilise la commande : 'exec'

```
moufid@debian:~$ exec bash
```

Ajouter à votre Path (une variable d'environnement) le chemin '/home/utilisateur/Bureau' :

```
moufid@debian:~$ export PATH=$PATH:/home/moufid/Bureau
```

Pour finir sur les variables d'environnement, on peut toute les afficher en exécutant une simple commande :

```
moufid@debian:~$ env
```

JOB 6

On doit accéder à la suite du sujet en désarchivant le fichier mis à notre disposition. Pour désarchiver un téléchargement seulement avec le terminal, on utilise 'tar' :

```
moufid@debian:~$ cd Téléchargements/  
moufid@debian:~/Téléchargements$ tar -xvzf Ghost\ in\ the\ Shell.tar.gz
```


JOB 7

Pour ce job, il va falloir (en une seule commande) faire plusieurs actions ;

- En effet, il faudra créer 3 fichiers : une_commande.txt, nb_lignes.txt, save_sources
→ une_commande.txt devrait afficher "Je suis votre fichier texte".
→ nb_lignes.txt contiendra le nombre de lignes présentes dans le fichier sources apt
→ save_sources contiendra une copie du contenu du fichier sources apt
- A la fin on affichera une recherche des fichiers commençant par '.' tout en cherchant le mot alias

```
moufid@debian:~$ echo "Je suis votre fichier texte" >une_commande.txt | wc -l /etc  
/apt/sources.list ./ >/home/moufid/Bureau/nb_lignes.txt | cat /etc/apt/sources.lis  
t ./ >/home/moufid/Bureau/save_sources | grep -rnw /home/ -e alias .
```

Examinons les différentes commande qu'on a utilisé dans le commande :

```
echo "Je suis votre fichier texte" >une_commande.txt | wc -l /etc/apt/sources.list ./  
>/home/moufid/Bureau/nb_lignes.txt | cat /etc/apt/sources.list ./  
>/home/moufid/Bureau/save_sources | grep -rnw /home/ -e alias .
```

- La commande suivante créer le fichier puis insère le texte entre guillemets :

```
moufid@debian:~$ echo "Je suis votre fichier texte" >une_commande.txt
```

- La commande wc -l compte le nombre de ligne et colle les données dans un fichier grâce à la commande ./ > :

```
moufid@debian:~$ wc -l /etc/apt/sources.list ./ >/home/moufid/Bureau/nb_lignes.txt
```

- La commande cat affiche le contenu du fichier sources.list et le colle dans le fichiers save_sources grâce à la commande ./ > :

```
moufid@debian:~$ cat /etc/apt/sources.list ./ >/home/moufid/Bureau/save_sources
```

- Recherche les fichiers commençant par '.' tout en cherchant le mot alias qui sera utilisé depuis un fichier :

```
moufid@debian:~$ grep -rnw /home/ -e alias .
```

- La commande " | " sépare les commandes entre elles :

```
s.txt | cat /
```

POUR ALLER PLUS LOIN...

Comme lors du JOB précédent, on va devoir faire plusieurs actions à l'aide d'une seule et même commande :

- Il faudra installer tree
- afficher et enregistrer l'arborescence dans un fichier tree.save
- lister les éléments de votre première commande pour compter le nombre d'éléments trouvés
- lancer l'update de nos paquets et si elle réussit faire l'upgrade sinon ne pas lancer l'upgrade

```
moufid@debian:~$ sudo apt install tree | tree | tree -a / >/home/moufid/Bureau  
| ls -lisa | sudo apt-get update && sudo apt-get upgrade
```

• la commande tree affiche l'arborescence : voici un exemple de l'arborescence de ~\$:

```
moufid@debian:~$ tree  
.  
├── :1  
├── Bureau  
│   ├── abde.txt  
│   ├── droits.txt  
│   ├── groupes.txt  
│   ├── JOB7  
│   ├── KEY  
│   ├── nb_lignes.txt  
│   ├── save_sources  
│   ├── test.save  
│   ├── une_commande.txt  
│   └── users.txt  
├── Documents  
├── Images  
│   └── Capture d'écran de 2022-10-05 10-35-52.png  
├── Modèles  
├── Musique  
├── Public  
├── Téléchargements  
│   ├── Ghost in the Shell.pdf  
│   ├── Ghost in the Shell.tar.gz  
│   └── google-chrome-stable_current_amd64.deb  
├── tout_tree  
├── tree.save  
├── users.txt  
├── Vidéos  
├── wget-log  
└── wget-log.1  
  
10 directories, 18 files  
moufid@debian:~$
```

• La commande ci dessous affiche toute l'arborescence de votre / et enregistre le résultat dans un fichier 'tree.save' :

```
tree -a / >/home/moufid/Bureau
```

• Ces deux commandes servent à premièrement lancer une update, puis grâce à '&&' elle lance l'upgrade si l'update réussit. Si l'update ne réussit pas l'upgrade ne se lancera pas.

```
sudo apt-get update && sudo apt-get upgrade
```

BONUS

→ Quel est l'intérêt d'utiliser SSH ?

L'intérêt principal d'utiliser SSH est que les utilisateurs qui exécutent la commande SSH exécutent des commandes shell sur leur propre ordinateur comme s'ils se trouvaient dans la machine physique d'un autre (ou sa machine virtuelle).

→ Est-ce que les clés générées par SSH par défaut sont assez sécurisées ? Justifier votre réponse

L'avantage du SSH est que c'est une utilisation cryptée pour assurer un transfert d'information sécurisé entre le client et le serveur.

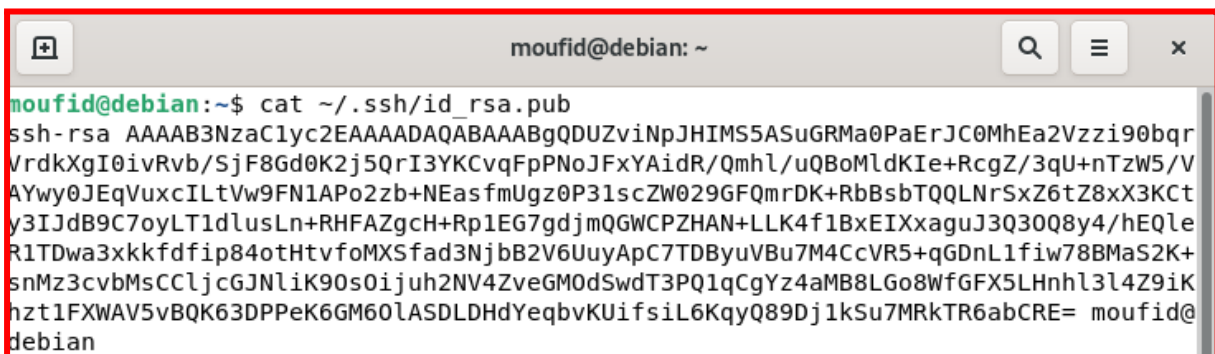
La clé de session est négociée lors de la connexion, puis utilisée avec un algorithme de chiffrement symétrique et un algorithme de code d'authentification de message pour protéger les données.

→ Citez d'autres protocoles de transfert ? Quelles sont les différences entre ses protocoles ?

voici une liste de différents protocoles de transfert :

- SMB/CIFS (Common Internet File Service)
- AFP (Protocole de fichiers Apple)
- NFS (Network File Service)
- FTP (File Transfert Protocol)
- FTPS (FTP avec chiffrement SSL)
- WebDAV
- SFTP
- Rsync

• Générer une clé SSH : `cat ~/.ssh/id_rsa.pub`



```
moufid@debian: ~  
moufid@debian:~$ cat ~/.ssh/id_rsa.pub  
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGDUzviNpJHIMS5ASuGRMa0PaErJC0MhEa2Vzzi90bqr  
VrdkXgI0ivRvb/SjF8Gd0K2j5QrI3YKcvqFpPNoJFxAidR/Qmhl/uQBoMldKie+RcgZ/3qU+nTzW5/V  
AYwy0JEqVuxcILtVw9FN1APo2zb+NEasfmUgz0P31scZW029GFQmrDK+RbBsbTQQLNrSxZ6tZ8xX3Kct  
y3IJdB9C7oyLT1dLusLn+RHFAZgCH+Rp1EG7gdjmQGWCpZHAN+LLK4f1BxEIXxaguJ3Q30Q8y4/hEQle  
R1TDwa3xkkfdp84otHtvfoMXSfad3NjbB2V6UuyApC7TDByuVBu7M4CcVR5+qGDnL1fiw78BmaS2K+  
snMz3cvbMsCCLjcGJNlik90s0ijuh2NV4ZveGM0dSwdT3PQ1qCgYz4aMB8LGo8WfGFX5LHnhl3l4Z9iK  
hzt1FXWAV5vBQK63DPPEK6GM60lASDLdHdYeqbvKUiFsiL6KqyQ89Dj1kSu7MRkTR6abCRE= moufid@  
debian
```

-Uploader ou télécharger un fichier avec SSH de votre PC ou VM vers le pc ou VM de quelqu'un d'autre :

```
sudo apt install ssh (pour installer SSH)
sudo systemctl status (Vérifier son statut)
sudo systemctl start ssh (démarrer si notre statut est inactif)
scp [fichier] USER@adresse_ip:/home/USER/Bureau/ (envoyer à quelqu'un)
Are you sure you want to continue connecting : YES
USER@IP's password :
USER reçoit le fichier envoyé
```

-Configurer SSH pour empêcher le login root :

```
vi /etc/ssh/sshd_config
remplacer PermitRootLogin yes par PermitRootLogin no
sudo systemctl start
```

Modifier le port de connexion de SSH (autre que 22) :

```
se connecter au serveur en tant qu'admin
ouvrir le fichier config SSH (sshd_config) via vi : vi /etc/ssh/sshd_config
rechercher l'entrée d'un port et le remplacer (exemple port 9122)
redémarrer le service: sudo systemctl restart ssh
```

Se connecter en SSH sans avoir à renseigner de mot de passe :

```
ssh-keygen -t rsa
ssh-keygen -t rsa -b 4096
copier la clé publique pour activer le SSH sans mot de passe : ssh-copy-id [user]@adresse_ip
```

Limiter l'utilisation de SSH à un groupe particulier :

```
sudo groupadd Plateforme_ssh
editer le fichier /etc/ssh/sshd_config avec la commande sudo nano
→y ajouter AllowGroups Plateforme_ssh
```