

Job 1

Comment ajouter des options à une commande ?

afin d'ajouter une option à une commande il suffit de faire [commande] -{option voulu} (-{seconde option})

Ex :

```
ubuntu@ubuntu-VirtualBox:~$ ls -a -l
```

la commande **ls** permet d'afficher l'ensemble des répertoires

l'option **-a** ajoute à la commande l'affichage des contenus des répertoires l'option **-l** ajoute à la commande l'affichage des répertoire sous forme de liste

```
ubuntu@ubuntu-VirtualBox:~$ ls -a -l total 76

drwxr-x--- 16 ubuntu ubuntu 4096 oct. drwxr-xr-x 3 root root 4096 oct. -rw-r--r-- 1 ubuntu ubuntu 220
oct. -rw-r--r-- 1 ubuntu ubuntu 3771 oct.

3 11:19 . 3 11:12 ..

3 11:12 .bash_logout 3 11:12 .bashrc

3 11:18 Bureau

3 11:19 .cache

3 11:21 .config

3 11:18 Documents 3 11:19 .gnupg

3 11:18 Images 3 11:18 .local

3 11:18 Modèles 3 11:18 Musique

3 11:12 .profile 3 11:18 Public 3 11:18 snap

3 11:19 .ssh

3 11:18 Téléchargements 3 11:18 Vidéos

drwxr-xr-x 2 ubuntu ubuntu drwx----- 11 ubuntu ubuntu drwx----- 10 ubuntu ubuntu
4096 oct. 4096 oct. 4096 oct. 4096 oct. 4096 oct.

4096 oct. 4096 oct. 4096 oct. 4096 oct.

807 oct. 4096 oct. 4096 oct. 4096 oct. 4096 oct. 4096 oct.

drwxr-xr-x drwx----- drwxr-xr-x drwx----- drwxr-xr-x drwxr-xr-x -rw-r--r-- drwxr-xr-x drwx----- drwx-----
drwxr-xr-x drwxr-xr-x
```

```
2 ubuntu ubuntu 2 ubuntu ubuntu 2 ubuntu ubuntu 3 ubuntu ubuntu 2 ubuntu ubuntu 2
ubuntu ubuntu 1 ubuntu ubuntu 2 ubuntu ubuntu 3 ubuntu ubuntu 2 ubuntu ubuntu 2
ubuntu ubuntu 2 ubuntu ubuntu
```

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

Les deux syntaxe principales pour l'écriture d'une commande sont - et --

Job 2

Afin de lire un fichier on peut utiliser les commandes suivantes :

```
ubuntu@ubuntu-VirtualBox:~$ more [nom du fichier]
```

Afin d'afficher les dix première ligne du fichier on utilisera

```
ubuntu@ubuntu-VirtualBox:~$ more head [nom du fichier]
```

 Afin d'afficher les 20 première ligne du fichier on utilisera

```
ubuntu@ubuntu-VirtualBox:~$ more head -20 [nom du fichier]
```

 Afin d'afficher les 20 dernière ligne du fichier on utilisera

```
ubuntu@ubuntu-VirtualBox:~$ more line -20 [nom du fichier]
```

 Afin d'afficher les 10 dernière ligne du fichier on utilisera

```
ubuntu@ubuntu-VirtualBox:~$ more line [nom du fichier]
```

la commande **cat** ainsi que **less** permet de faire cela elles se présent sous cette forme

```
ubuntu@ubuntu-VirtualBox:~$ cat [nom du fichier] ubuntu@ubuntu-VirtualBox:~$ less [nom du fichier]
```

Job 3

afin d'installer le paquet "cmatrix" il faut utiliser la commande

ubuntu@ubuntu-VirtualBox:~\$ apt-get install [nom du paquet] afin d'exécuter le paquet "cmatrix"
il faut utiliser la commande

```
ubuntu@ubuntu-VirtualBox:~$ exec cmatrix
```

afin de mettre à jour le gestionnaire de paquets

afin de mettre à jour les différents logiciels

afin d'installer Chrome il faudra d'abord posséder **wget**

Maintenant que l'on a vérifié que wget était présent ou non nous pouvons installer chrome

```
ubuntu@ubuntu-VirtualBox:~$ sudo apt update [sudo] Mot de passe de ubuntu :  
  
#après avoir écrit votre mot de passe l'update se lancera  
root@ubuntu-VirtualBox:/home/ubuntu# sudo apt update  
  
ubuntu@ubuntu-VirtualBox:~$ sudo apt upgrade [sudo] Mot de passe de ubuntu :  
  
#après avoir écrit votre mot de passe l'upgrade se lancera  
root@ubuntu-VirtualBox:/home/ubuntu# sudo apt upgrade  
  
ubuntu@ubuntu-VirtualBox:~$ wget --version  
  
#si un message d'erreur s'affiche il faudra l'installer sinon on peut déjà installer chrome  
  
ubuntu@ubuntu-VirtualBox:~$ sudo apt install wget root@ubuntu-VirtualBox:/home/ubuntu# sudo  
apt install wget  
  
ubuntu@ubuntu-VirtualBox:~$ wget  
https://dl.google.com/linux/direct/google-chrome-stable_current_amd64.deb #afin de télécharger le  
paquet de chrome  
  
ubuntu@ubuntu-VirtualBox:~$ sudo dpkg -i google-chrome-stable_current_amd64.deb #afin d'installer  
le paquet de chrome  
  
ubuntu@ubuntu-VirtualBox:~$ sudo apt-get install -f  
  
#si des messages sont apparue alors effectuer cette commande
```

Afin de redémarrer l'ordinateur deux commande sont disponible

```
ubuntu@ubuntu-VirtualBox:~$ shutdown -r ubuntu@ubuntu-VirtualBox:~$ reboot
```

Afin d'éteindre l'ordinateur il suffit de faire

Job 4

Pour créer le groupe "Plateformeurs" on utilise la commande suivante

Ensuite pour créer les utilisateurs nous entrons cette commande

```
root@ubuntu-VirtualBox:/home/ubuntu# useradd [nom d'utilisateur]
```

Puis pour ajouter "User2" au groupe "Plateformeurs" nous utilisons

```
root@ubuntu-VirtualBox:/home/ubuntu# adduser [nom_utilisateur] [nom_groupe]
```

Nous devons ensuite copier notre fichier "user.txt" dans un fichier "droits.txt"

Et dans un fichier "groupes.txt"

```
root@ubuntu-VirtualBox:/home/ubuntu# cp user.txt groupes.txt
```

Afin de changer le propriétaire du fichier "droits.txt" il suffit de faire

```
ubuntu@ubuntu-VirtualBox ~$ shutdown now
```

#now permet de spécifier au système qu'il doit se couper après l'exécution de la commande

```
ubuntu@ubuntu-VirtualBox:~$ sudo groupadd [nom du groupe] [sudo] Mot de passe de ubuntu :
```

#après avoir écrit votre mot de passe le groupe sera créé

```
root@ubuntu-VirtualBox:/home/ubuntu# group add [nom du groupe]
```

```
root@ubuntu-VirtualBox:/home/ubuntu# cp  
commande cp [fichier à copier] [fichier cible]
```

#voici comment marche la

#voici comment marche la commande chown [Utilisateur] {nom du fichier}

Ensuite il faut changer les droits du fichier "droits.txt" pour que "User2" ait seulement accès à la lecture, nous utilisons donc la commande

```
root@ubuntu-VirtualBox:/home/ubuntu# chmod o+r droits.txt
```

Puis nous devons changer les droits du fichier "groupes.txt" pour que les utilisateurs n'ait accès qu'à l'option 'lecture'

```
root@ubuntu-VirtualBox:/home/ubuntu# chmod a+r groupes.txt
```

Enfin nous devons changer les droits du fichier "Plateformeurs" pour que les utilisateurs puissent y accéder en lecture et en écriture

```
root@ubuntu-VirtualBox:/home/ubuntu# chmod a+rw Plateformeurs
```

Job 5

Pour ajouter un alias qui permettra de lancer la commande "ls -la" en tapant "la" nous faisons

```
root@ubuntu-VirtualBox:/home/ubuntu# alias la='ls -la'
```

Puis pour ajouter un alias qui permettra de lancer la commande "apt-get update" en tapant "update" nous faisons

```
root@ubuntu-VirtualBox:/home/ubuntu# alias update='apt-get update'
```

Enfin pour ajouter un alias qui permettra de lancer la commande “apt-get upgrade” en tapant “upgrade” nous faisons

```
root@ubuntu-VirtualBox:/home/ubuntu# alias upgrade='apt-get upgrade'
```

Pour ajouter une variable d’environnement nous utiliserons la commande

Ensuite pour mettre à jour les modifications du bashrc actuel nous tapons la commande

```
root@ubuntu-VirtualBox:/home/ubuntu# exec bash
```

```
root@ubuntu-VirtualBox:/home/ubuntu# USER=("$USER") root@ubuntu-VirtualBox:/home/ubuntu#  
export USER
```

Pour afficher les variables d’environnement nous utilisons la commande

```
root@ubuntu-VirtualBox:/home/ubuntu# printenv
```

Pour ajouter au “Path” le chemin “/home/user/bureau” nous entrons la commande

```
root@ubuntu-VirtualBox:/home/ubuntu# path='/home/user/bureau/....'
```

Job 6

Nous devons ensuite télécharger une archive et la désarchiver, nous utiliserons les commandes suivantes :

```
root@ubuntu-VirtualBox:/home/ubuntu# .tar zxvf nom_du_fichier.tar.gz
```

Job 7

Chaque actions sera effectuées en une seule commande

Pour créer un fichier “une_commande.txt” avec le texte suivant “Je suis votre fichier texte” nous utilisons donc

```
root@ubuntu-VirtualBox:/home/ubuntu# echo Je suis votre fichier texte >> une_commande.txt
```

Puis pour compter le nombre de lignes présentes dans le fichier de source apt et les enregistrer dans un fichier nommé “nb_lignes.txt” nous faisons

```
root@ubuntu-VirtualBox:/home/ubuntu# touch nb_lignes.txt && wc -l sources.list > nb_lignes.txt
```

Ensuite nous afficherons le contenu du fichier source apt et l’enregistrerons dans un autre fichier appelé “save_sources”

```
root@ubuntu-VirtualBox:/home/ubuntu# touch save_sources && echo sources.list > save_sources
```

Enfin nous ferons une recherche des fichiers commençants par “.” tout en cherchant le mot alias qui sera utilisé depuis un fichier grâce à la commande

```
root@ubuntu-VirtualBox:/home/ubuntu# find -wholename . & find -iname alias
```

Job Bonus

Installer SSH

```
apt-get install ssh
```

Générer une clé SSH

```
ssh-keygen
```

Se connecter à une VM ou l'ordinateur d'un camarade via SSH

```
ssh nom-d'utilisateur@nom-de-domaine
```

Configurer SSH pour empêcher le login root (root ne peut pas se connecter en SSH)

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

```
sudo nano /etc/ssh/sshd_config
```

Ensuite se connecter en SSH sans avoir à renseigner de mot de passe

Uploader un fichier avec SSH (de votre pc ou VM vers le pc ou VM d'un camarade)

```
scp nom-d'utilisateur@nom-de-domaine:/home/user/etc.../fichier-cible  
/home/user(receveur)/destination
```

Télécharger un fichier avec SSH (de votre pc ou VM vers le pc ou VM d'un camarade)

Limiter l'utilisation de SSH à un groupe particulier nommé “Plateforme_ssh”

Quel est l'intérêt d'utiliser SSH ?

SSH est sécurisé cryptant les données échangées et demandant aux deux communicants d'échanger des clés de chiffrement afin de communiquer

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

Les clés SSH constituent une méthode plus sécurisée que les mots de passe pour se connecter à un serveur distant. Elles sont générées en local sur la machine et composées de deux éléments : une clé privée que vous gardez sur l'ordinateur et une publique qui est transmise au serveur.

L'authentification par clé fonctionne grâce à 3 composants :

- Une clé publique : elle sera exportée sur chaque hôte sur lequel on souhaite pouvoir se connecter ;
- Une clé privée : elle permet de prouver son identité aux serveurs ;
- Une passphrase : Permet de sécuriser la clé privée (notons la subtilité, passphrase et pas password... donc « phrase de passe » et non pas « mot de passe »).

La sécurité est vraiment accrue car la passphrase seule ne sert à rien sans la clé privée, et vice-versa

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

1. **SSL** : permet la sécurité en permettant aux applications de chiffrer les données allant d'un client à un serveur correspondant.
2. **IP(Internet Protocol)** : le protocole IP est le protocole par lequel les données sont envoyées d'un ordinateur à un autre sur Internet. Chaque ordinateur sur Internet a au moins une **adresse IP** qui l'identifie de manière unique à partir de tous les autres ordinateurs sur Internet.
3. **FTP(File Transfer Protocol)**: Le protocole FTP (File Transfer Protocol) est un protocole Internet standard pour la **transmission** de fichiers entre ordinateurs sur Internet via des connexions TCP/IP. Le protocole FTP utilise généralement le port 21 comme moyen de communication.
4. **SSL(Secure Sockets Layer)**: SSL permet la sécurité en permettant aux applications de chiffrer les données allant d'un client à un serveur correspondant.
5. **HTTPS(Hypertext Transfer Protocol Secure)**: *Le protocole HTTPS est une variante du protocole HTTP standard qui ajoute une couche de sécurité aux données en transit via le protocole SSL (Secure Socket Layer) ou TLS (Transport Layer Security). HTTPS permet une communication cryptée et une connexion sécurisée entre un utilisateur distant et le serveur Web.*