

# Job 1

- **Afficher le manuel de la commande ls** : *man ls*

- **Afficher les fichiers cachés du home de votre utilisateur** :

- *ls* : tous les fichiers non cachés
- *ls -a* (raccourcis clavier : CTRL+H) : tous les fichiers et dossiers y compris ceux cachés
- *ls -d .\** : seulement les fichiers et dossiers cachés
- *ls -d \*/* : seulement les dossiers cachés

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

- *ls -lisa* : liste de tous les fichiers y compris les fichiers cachés avec les droits et la date
- *ls -d .\* -lisa* : liste des fichiers cachés seulement avec les droits et la date

- **Comment ajouter des options à une commande ?**

Pour commencer, une commande n'est rien d'autre qu'une liste d'arguments dont le premier représente l'exécutable. Les arguments ne sont que des chaînes de caractères.

Dans ce cas, comment reconnaître une option (qui définit le comportement de la commande) d'un paramètre (un nom de fichier, de répertoire etc.) ?

La distinction passe par une convention. Et il en existe plusieurs !

- La position : la plus simple à mettre en œuvre mais limitant les possibilités. Elle est utilisée par des commandes comme tar. Ex. : dans la commande *tar xzvf fichier.tar.gz* le premier paramètre correspond aux options de tar (x=extract, z=gunzip, v=verbose, f=file)
- Le simple tiret : la plus répandue surtout sur les anciens systèmes, elle consiste à précéder d'un tiret une lettre représentant une option. Il est également possible avec cette convention de regrouper plusieurs options dans un seul argument. Par exemple, la commande *ps -ef* est identique à la commande *ps -e -f*. Pour passer des valeurs à ce genre d'options, il faut ajouter un argument supplémentaire. Par exemple, dans la commande *bash -c "echo hello"*, la chaîne "echo hello" est une valeur liée à l'option -c. Cette convention présente des inconvénients de lisibilité du code produit. Il n'est en effet pas évident de connaître l'effet de l'option -e de ps sans regarder l'aide de cette commande.
- Le double tiret : la convention du double tiret permet de corriger le manque de lisibilité du simple tiret. Souvent ces deux conventions coexistent. Avec cette convention, il n'est pas possible de combiner plusieurs options en une seule. Par contre, les options sont beaucoup plus parlantes. Par exemple, *mkdir --parents --verbose* est plus lisible que *mkdir -pv*.

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

- Le simple tiret
- Le double tiret

Chaque commande offre des options qui peuvent modifier son comportement. Les options sont souvent nommées par une seule lettre et précédées d'un tiret (« - »). Rares sont les commandes qui nécessitent absolument des options : la plupart peuvent être lancées sans en spécifier.

La plupart du temps, l'ordre dans lequel on donne les options n'a pas d'importance. Aussi, la plupart des commandes permettent de regrouper les options: `ls -l -a` est équivalent à `ls -la`. Vous verrez plusieurs exemples d'options dans les sections décrivant les commandes courantes.

Une option très courante est `-help` (avec deux tirets); cette option indique à la commande de ne pas s'exécuter et de décrire toutes les options possibles.

## Job 2

**- Lisez un fichier en utilisant une commande qui permet seulement de lire : `cat`**

**- 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`**

## Job 3

**- Installer le paquet `“cmatrix”` :**

- `sudo apt-get update`
- `sudo apt-get install cmatrix`

**- lancer le paquet que vous venez d'installer : `cmatrix`**

**- Mettre à jour son gestionnaire de paquets : `sudo apt update`**

**- Mettre à jour ses différents logiciels : `sudo apt upgrade`**

### - Télécharger les internets : Google :

- Installer l'outil "wget" qui permettra de télécharger le paquet Chrome.
  - saisir : `wget -version` puis appuyer sur la touche Entrée
  - si il y a un message d'erreur car wget n'est pas installé, saisir : `sudo apt install wget` puis appuyer sur la touche Entrée
- Utiliser "wget" pour télécharger le paquet Chrome.
  - taper `wget`  
[https://dl.google.com/linux/direct/google-chrome-stable\\_current\\_amd64.deb](https://dl.google.com/linux/direct/google-chrome-stable_current_amd64.deb) puis appuyer sur la touche Entrée
- Installer le paquet Chrome qui a été téléchargé : `sudo dpkg -i google-chrome-stable_current_amd64.deb` puis appuyer sur Entrée
- Corriger les erreurs qui sont apparues lors de l'installation : `sudo apt-get install -f` puis appuyer sur la touche Entrée
- Saisir : `google-chrome` puis appuyer sur Entrée, Google Chrome se lance.

- Redémarrer votre machine : `sudo reboot` ou `sudo shutdown -r now`

- éteindre votre machine : `sudo halt` ou `sudo shutdown -h now`

## Job 4

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

- Nano est un éditeur de texte préinstallé sur toutes les distributions Linux basées sur Ubuntu. Si Nano n'est pas installé, le téléchargement est possible : `sudo apt install nano`
- Accéder au répertoire dans lequel créer le fichier : `cd Documents`
- Création du fichier texte : `nano users.txt`
- Taper dans le fichier texte : `User1` puis retour à la ligne `User2`
- Enregistrer le fichier : CTRL+O
- Sortir de nano : CTRL+X

- Créer un groupe appelé "Plateformeurs" : `sudo groupadd Plateformeurs`

- Créer un utilisateur appelé "User1" : `adduser user1`

- Créer un utilisateur appelé "User2" : `adduser user2`

- Ajouter "User2" au groupe Plateformeurs : `usermod -aG Plateformeurs user2`

- Copier votre "users.txt" dans un fichier "droits.txt": `cp users.txt droits.txt`

- Copier votre "users.txt" dans un fichier "groupes.txt": `cp users.txt groupes.txt`

- Changer le propriétaire du fichier "droits.txt" pour mettre "User1": `chown user1:user1 droits.txt`

- Changer les droits du fichier "droits.txt" pour que "User2" ai accès seulement en lecture : `chmod o+r droits.txt`

- Changer les droits du fichier "groupes.txt" pour que les utilisateurs puissent accéder au fichier en lecture uniquement : `chmod u-w groupes.txt`

- Changer les droits du fichier pour que le groupe "Plateformeurs" puissent y accéder en lecture/écriture : `chmod g+wr groupes.txt`

## Job 5

- Ajouter un alias qui permettra de lancer la commande "ls -la" en tapant "la" : `alias la='ls -la'`

- Ajouter un alias qui permettra de lancer la commande "apt-get update" en tapant "update" : `alias update='apt-get update'`

- Ajouter un alias qui permettra de lancer la commande "apt-get upgrade" en tapant "upgrade" : `alias upgrade='apt-get upgrade'`

- Ajouter une variable d'environnement qui se nommera "USER" et qui sera égale à votre nom d'utilisateur : `USER=mimi`

- Mettre à jour les modifications de votre bashrc dans votre shell actuel : `exec bash`

- Afficher les variables d'environnement : `env / printenv`

- Ajouter à votre Path le chemin "/home/votre utilisateur/Bureau" : `export PATH=$PATH:/home/mimi/Bureau`

## Job 6

**Vous devez télécharger l'archive suivante**

(<https://drive.google.com/file/d/1s9ZhRhjo0FXcBNRB5khAGK1jVxkZj6Uk/view?usp=ssharing>) et la désarchiver seulement avec le terminal.

Cette manipulation vous permettra d'accéder à la suite du sujet. : /Téléchargements\$  
`tar -xf Ghost\ in\ the\ Shell.tar.gz`

## Job 7

Toutes les actions sont à réaliser en une seule commande

- Créer un fichier "une\_commande.txt" avec le texte suivant "Je suis votre fichier texte"
- Compter le nombre de lignes présentes dans votre fichier de source apt et les enregistrer dans un fichier nommé "nb\_lignes.txt"
- Afficher le contenu du fichier source apt et l'enregistrer dans un autre fichier appelé "save\_sources"
- Faites une recherche des fichiers commençant par "." tout en cherchant le mot alias qui sera utilisé depuis un fichier

```
mimi@mimi:~$ echo 'je suis votre fichier texte' >une_commande.txt | wc -l  
/etc/apt/sources.list ./ >nb_lignes.txt | cat /etc/apt/sources.list ./ >save_sources | grep  
-r alias .
```

(- Télécharger cowsay : *sudo apt-get install cowsay*)

## Pour aller plus loin...

Toutes les actions sont à réaliser en une seule commande

- Installer la commande tree
- Lancer la commande tree en arrière-plan qui aura pour but d'afficher toute l'arborescence en de votre / en enregistrant le résultat dans un fichier "tree.save"
- lister les éléments présents dans le dossier courant est utilisé directement le résultat de votre première commande pour compter le nombre d'éléments trouvés
- Lancer une commande pour update vos paquets, si l'update réussit alors, vous devrez lancer un upgrade de vos paquets. Si l'update échoue, votre upgrade ne se lancera pas

```
sudo apt-get install tree | tree & tree -o tree.save | ls -l | wc -l | sudo apt-get  
update && apt-get upgrade
```