

# Shell

Job - 01

- Afficher le manuel de la commande **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**
- Comment ajouter des options à une commande ?

Options courtes (une lettre) : Les options courtes sont souvent précédées d'un seul tiret suivi d'une lettre.

Exemple, dans la commande **ls -l**, l'option **-l** est utilisée pour afficher les fichiers sous forme de liste détaillée.

Options longues (mots-clés) : Les options longues sont précédées de deux tirets.

Exemple, dans la commande **ls --all**, l'option **--all** est utilisée pour afficher tous les fichiers, y compris les fichiers cachés.

Combinaison d'options : Vous pouvez souvent combiner plusieurs options courtes d'un coup.

Exemple, **ls -la** affichera les fichiers sous forme de liste détaillée en incluant les fichiers cachés.

Position des options : Les options peuvent être placées n'importe où dans la commande, tant qu'elles suivent la commande proprement dite.

Exemple, **ls -al** et **ls -la** font la même chose.

Options avec arguments : Certaines options nécessitent un argument supplémentaire.

Exemple, dans la commande **cp -r dossier\_source dossier\_destination**, l'option **-r** (pour récursif) a besoin de deux arguments : le **dossier source** et le **dossier destination**.

Aide et documentation des commandes : Pour obtenir de l'aide sur les options d'une commande spécifique, vous pouvez généralement utiliser l'option **--help**.

Exemple, **ls --help** fournira des informations sur les options de la commande **ls**.

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

1. Syntaxe à un tiret court :

Dans cette syntaxe, les options sont spécifiées avec un seul tiret (-) suivi d'une lettre. Chaque option est généralement une lettre qui représente un paramètre spécifique pour la commande.

Exemple : `ls -l -a` ou de manière abrégée `ls -la`

2. Syntaxe à deux tirets longs :

Cette syntaxe est plus descriptive et permet souvent d'utiliser des noms complets ou des mots-clés pour spécifier les options. Les options sont précédées de deux tirets (--). Par

Exemple : `ls --all --long`

Dans de nombreux cas, les options courtes (à un seul tiret) peuvent être regroupées.

Exemple, si une commande prend les options `-a`, `-b` et `-c`, vous pouvez les combiner sous forme de `-abc`. Cela est souvent utilisé pour réduire la quantité de saisie nécessaire.

Les options longues fournissent généralement des noms descriptifs pour les paramètres, ce qui les rend plus faciles à comprendre pour les utilisateurs.

Exemple, `--all` au lieu de `-a`

Il est important de noter que toutes les commandes n'utilisent pas forcément les deux types de syntaxes, et certaines commandes peuvent n'utiliser que l'une ou l'autre. La meilleure façon de connaître les options spécifiques à une commande est de consulter sa documentation ou d'utiliser l'option `--help` ou `-h` si elle est disponible.

Exemple, pour obtenir de l'aide sur la commande `ls`, vous pouvez utiliser `ls --help` ou `ls -h`. Cela affichera les options et une brève description de ce qu'elles font.

## Job 02

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

`cat nom_du_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`

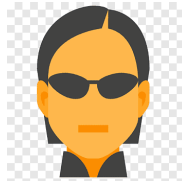
- afficher les 20 premières lignes du fichier “.bashrc”

`head -n 20 ~/.bashrc`

- afficher les 20 dernières lignes du fichier “.bashrc”

`tail -n 20 ~/.bashrc`

## Job 03



**sudo** ("SuperUser Do")

0. Installer le package sudo en utilisant la commande suivante :

```
sudo apt update
```

1. Installer le paquet "cmatrix"

```
sudo apt install cmatrix
```

2. lancer le paquet que vous venez d'installer pour un effet cool

```
cmatrix
```

Ctrl + C (pour fermer l'animation cmatrix)

3. Mettre à jour son gestionnaire de paquets

```
sudo pacman -Sy
```

4. Mettre à jour ses différents logiciels

```
sudo pacman -Syu
```

5. Télécharger les internets : Google

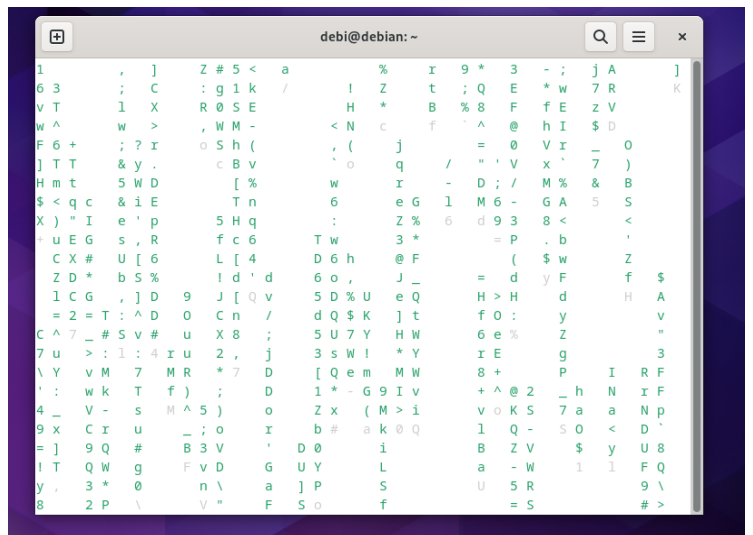
```
wget https://www.google.com
```

6. Redémarrer votre machine

```
sudo reboot
```

7. Éteindre votre machine

```
sudo shutdown now
```



## Job 04

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

```
echo -e "User1\nUser2" > users.txt
```

Créer un groupe appelé "Plateformeurs" :

```
sudo groupadd Plateformeurs
```

Créer un utilisateur appelé "User1" :

```
sudo useradd -m User1
```

Créer un utilisateur appelé "User2" :

```
sudo useradd -m User2
```

Ajouter "User2" au groupe Plateformeurs :

```
sudo 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" :

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

Changer les droits du fichier "droits.txt" pour que "User2" ait accès seulement en lecture :

```
sudo chmod 400 droits.txt
```

Changer les droits du fichier "groupes.txt" pour que les utilisateurs puissent y accéder en lecture uniquement :

```
sudo chmod 444 groupes.txt
```

Changer les droits du fichier pour que le groupe "Plateformeurs" puisse y accéder en lecture/écriture :

```
sudo chgrp Plateformeurs droits.txt
```

```
sudo chmod 664 droits.txt
```

## Job 05

Vous pouvez exécuter toutes ces étapes dans le terminal en une seule fois à l'aide de commandes séparées par des points-virgules (;).

```
echo 'alias la="ls -la"' >> ~/.bashrc ; echo 'alias update="sudo apt-get update"' >> ~/.bashrc ; echo 'alias upgrade="sudo apt-get upgrade"' >> ~/.bashrc ; echo 'export USER="votre_nom"' >> ~/.bashrc ; source ~/.bashrc ; env ; export PATH="/home/votre_utilisateur/Bureau:$PATH" >> ~/.bashrc ; source ~/.bashrc
```

Explications des étapes :

`echo 'alias la="ls -la"' >> ~/.bashrc` : Ajoute l'alias pour ls -la dans le fichier .bashrc

`echo 'alias update="sudo apt-get update"' >> ~/.bashrc` : Ajoute l'alias pour apt-get update

`echo 'alias upgrade="sudo apt-get upgrade"' >> ~/.bashrc` : Ajoute l'alias pour apt-get upgrade.

`echo 'export USER="votre_nom"' >> ~/.bashrc` : Définit une variable d'environnement USER.

`source ~/.bashrc` : Applique les modifications actuelles au shell actuel.

`env` : Affiche les variables d'environnement.

`export PATH="/home/votre_utilisateur/Bureau:$PATH" >> ~/.bashrc` : Ajoute le chemin du Bureau à la variable PATH.

`source ~/.bashrc` : Applique les dernières modifications du fichier .bashrc.

Remplacer "votre\_nom" et "votre\_utilisateur" par vos propres informations, comme ici sur Manjaro.

```
[manjaro@manjaro ~]$ echo 'alias la='ls -la'' >> ~/.bashrc ; echo 'alias update='sudo apt-get update'' >> ~/.bashrc ; echo 'alias upgrade='sudo apt-get upgrade' >> ~/.bashrc ; echo 'export USER='manjaro'' >> ~/.bashrc ; source ~/.bashrc ; export PATH='/home/manjaro/Bureau:$PATH' >> ~/.bashrc ; source ~/.bashrc
```

## Job 06

```
tar -xzvf "/home/debi/Documents/Ghost in the Shell.tar"
```

Cette commande est utilisée pour extraire le contenu d'une archive au format tar.gz (ou .tgz)

Voici ce que font les différents arguments de la commande :

**tar** : C'est le programme qui est utilisé pour manipuler les archives.

**-x** : Cela indique à tar d'extraire les fichiers de l'archive.

**-z** : Cela spécifie que l'archive est compressée avec gzip.

**-v** : Cela active le mode verbeux, ce qui signifie que tar affichera les fichiers qu'il extrait pendant le processus.

**-f** : Cela spécifie le nom de l'archive à traiter.



## Job 07

```
echo "Je suis votre fichier texte" > une_commande.txt && sudo apt-get update > /dev/null  
&& apt-cache show apt > save_sources && grep -r "alias" .* > fichiers_avec_alias.txt
```

Explications :

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

Créer un fichier appelé "une\_commande.txt" avec le texte spécifié.

```
sudo apt-get update > /dev/null
```

Met à jour la liste des paquets apt et redirige la sortie vers /dev/null pour supprimer les messages de mise à jour. Cela compte le nombre de lignes modifiées, mais ne les affiche pas.

```
apt-cache show apt > save_sources
```

Affiche les informations sur le paquet "apt" et enregistre la sortie dans un fichier appelé "save\_sources".

```
grep -r "alias" .* > fichiers_avec_alias.txt
```

Recherche les fichiers (y compris les fichiers cachés commençant par ".") à la recherche de lignes contenant le mot "alias".

Le résultat est enregistré dans un fichier appelé "fichiers\_avec\_alias.txt".

```
debi@debian:~$ cowsay Hello, je suis une vache !  
  
  _____  
 < Hello, je suis une vache ! >  
  -----  
      \   ^__^  
       \  (oo)\_____  
        (__)\\       )\\/\  
           ||----w |  
           ||     ||
```

Pour aller plus loin...

```
sudo apt install tree | (tree / > tree.save &) || ((ls | wc -l) > nb_elements.txt) && (sudo apt update && sudo apt upgrade)
```

Explications :

Installer la commande tree :

```
sudo apt install tree
```

Lancer la commande tree en arrière-plan pour afficher l'arborescence de votre répertoire racine et enregistrer le résultat dans "tree.save" :

```
tree / > tree.save &
```

Lister les éléments présents dans le dossier courant et utiliser le résultat pour compter le nombre d'éléments trouvés :

```
ls | wc -l > nb_elements.txt
```

Lancer une commande pour mettre à jour vos paquets, et si la mise à jour réussit, lancer une mise à jour des paquets :

```
sudo apt update && sudo apt upgrade
```

Le caractère `&&` assure que la seconde commande ne s'exécute que si la première a réussi. Si la première commande (update) échoue, alors la seconde (upgrade) ne se lancera pas.

Le caractère `|` permet d'envoyer la sortie de la première commande comme entrée à la deuxième commande.

Le caractère `||` permet d'exécuter la deuxième commande uniquement si la première échoue.

Le caractère `"&"` permet d'exécuter une commande en arrière-plan.

Les parenthèses `()` permettent de regrouper les commandes pour les exécuter en tant que groupe.

```
terminal 22 sept. 17:36
debi@debian: ~
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.
debi@debian:~$ tree
.
├── Bureau
├── Documents
├── droits.txt
├── fichiers_alias.txt
├── fichiers_avec_alias.txt
├── Ghost in the Shell.pdf
├── google-chrome-stable_current_amd64.deb
├── groupes.txt
├── Images
├── index.html
├── jesuisunbg
├── Modèles
├── Musique
├── nb_lignes.txt
├── Public
├── save_sources
├── Téléchargements
│   └── Ghost in the Shell.tar
├── tree.save
├── une_commande.txt
├── users.txt
└── Vidéos

9 directories, 14 files
debi@debian:~$
```