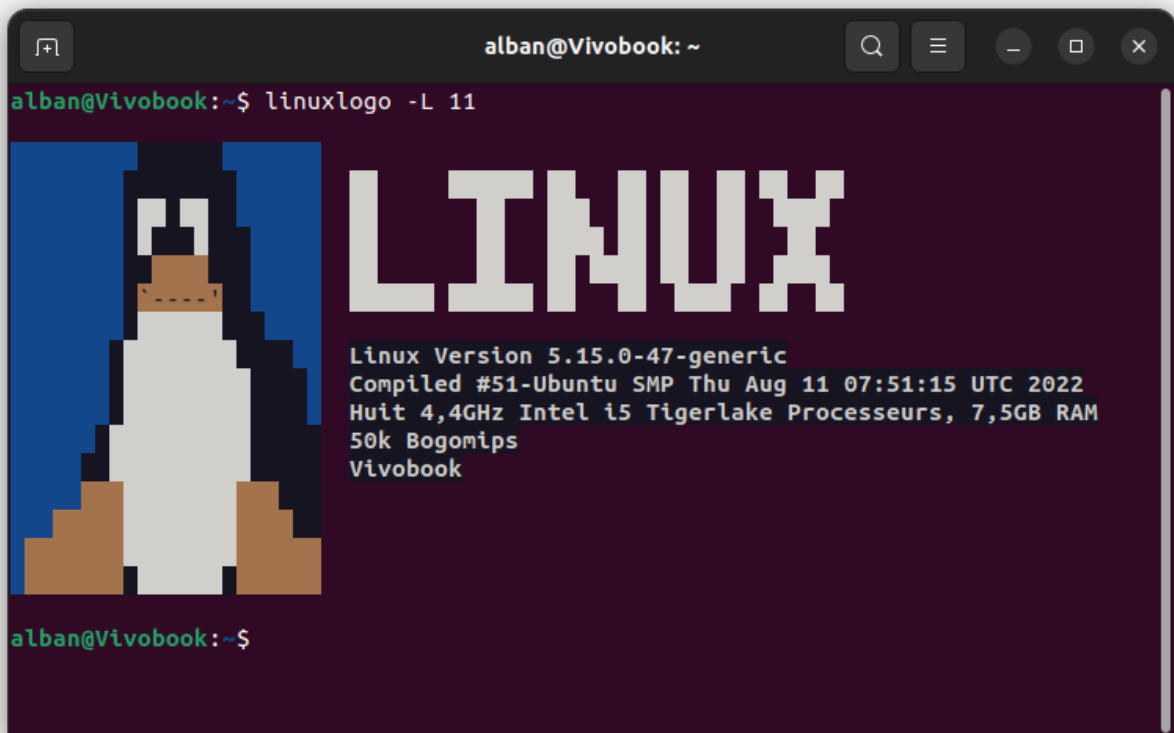


# GHOST IN THE SHELL



**MARTINANT Alban**

LA PLATEFORME

## JOB 1 :

Pour afficher le manuel de la commande **ls** :

```
man ls
```

Afficher les fichiers cachés du home utilisateur, il faut utiliser l'option :

```
ls -a
```

Pour afficher seulement les fichiers cachés, on peut filtrer le résultat de **ls -a** avec l'expression régulière "**^\.**". Le "**^**" permet d'indiquer que c'est le début, le "**\.**" permet de faire un échappement sur le ".", car c'est le caractère que l'on recherche.

```
ls -a | grep "^\."
```

Pour afficher les fichiers cachés et les informations sur les droits :

```
ls -la
```

## Comment ajouter des options à une commande ?

Pour ajouter des options à une commande il faut ajouter des flags.

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

Pour écrire les options d'une commande il y a deux syntaxes principales, la version longue avec deux tirets ( **--** ) et un mot. Exemple :

```
ls --all
```

la version courte avec un tiret ( **-** ) et une lettre. Exemple :

```
ls -a
```

Certaines options existent seulement en version longue ou en version courte alors que d'autres existent dans les deux syntaxes.

## JOB 2 :

Pour lire un fichier, la commande principale est **cat** qui affiche tout le contenu d'un fichier :

```
cat .bashrc
```

Pour naviguer plus facilement dans un fichier, on utilise plutôt la commande **more** :

```
more .bashrc
```

ou **less** (plus récent) :

```
less .bashrc
```

Pour afficher les 10 premières lignes d'un fichier, on peut utiliser **head** :

```
head .bashrc
```

Pour afficher les 10 dernières lignes d'un fichier, on peut utiliser **tail** :

```
tail .bashrc
```

Pour afficher les 20 premières lignes d'un fichier, on doit préciser le nombre de lignes avec l'option **-n** :

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

Il est possible aussi d'écrire directement le nombre de lignes que l'on souhaite afficher avec un tiret ( - ) :

```
head -20 .bashrc
```

Afficher les 20 dernières lignes d'un fichier :

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

ou

```
tail -20 .bashrc
```

### **JOB 3 :**

( A partir de maintenant j'utiliserai **sudo** chaque fois qu'une commande doit être exécuté avec les droits super utilisateur )

Pour installer le paquet "cmatrix", on peut utiliser le gestionnaire de paquet **apt** ::

```
sudo apt install cmatrix
```

Pour lancer le paquet, il suffit de faire :

```
cmatrix
```

Mettre à jour le gestionnaire de paquets :

```
sudo apt update
```

Mettre à jour ses différents logiciels :

```
sudo apt upgrade
```

Télécharger Google.com :

```
wget -r http://www.google.com/
```

Pour redémarrer la machine plusieurs commandes sont possibles, on peut utiliser **reboot** :

```
sudo reboot
```

**shutdown** et l'option **-r**, on doit lui préciser quand il doit redémarrer. Pour redémarrer immédiatement :

```
sudo shutdown -r now
```

Pour redémarrer dans n minutes, on écrit le nombre de minutes, par exemple pour 10 minutes :

```
shutdown -r 10
```

On peut aussi lui donner une heure, par exemple pour redémarrer à 21h00 :

```
shutdown -r 21:00
```

Il est aussi possible d'utiliser la commande **init** qui est associé au runlevel, donc pour redémarrer on peut faire :

```
sudo init 6
```

Pour éteindre le système, là encore il y a plusieurs possibilités, on peut simplement faire :

```
sudo halt
```

ou la commande **poweroff** qui appelle la commande **halt** avec l'option **-p**, ce qui éteint le système par extinction directe de l'alimentation sans utiliser les signaux habituellement utilisés :

```
sudo poweroff
```

Avec **shutdown** on lui donne l'option **-h** :

```
sudo shutdown -h now
```

Avec **init** et le runlevel associé :

```
sudo init 0
```

## JOB 4 :

Je vais ici utiliser l'option `--force-badname` pour suivre la consigne à la lettre car normalement linux n'accepte pas les noms de groupes ou d'utilisateurs qui commence par une majuscule.

Pour créer un groupe appelé "Plateformeurs" :

```
sudo addgroup --force-badname Plateformeurs
```

Créer un utilisateur "User1" :

```
sudo adduser --force-badname User1
```

Il faudra ensuite définir le mot de passe et remplir les informations demandées (facultatif)

Créer un utilisateur "User2" :

```
sudo adduser --force-badname User2
```

Définir le mot de passe et remplir les informations demandées (facultatif)

Pour ajouter "User2" au groupe "Plateformeurs" :

```
sudo adduser User2 Plateformeurs
```

Créer un fichier users.txt

```
touch users.txt
```

Ajouter User1 a users.txt (avec retour à la ligne) :

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

Ajouter User2 (avec retour à la ligne) :

```
echo -e User2\ >> users.txt
```

Pour copier le fichier users.txt vers droits.txt :

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

ou

```
cat users.txt >> droits.txt
```

Copier votre “users.txt” dans un fichier “groupes.txt” :

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

ou :

```
cat users.txt >> groupes.txt
```

On va maintenant mettre “User1” propriétaire du fichier droits.txt grâce à **chown** :

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

Pour que “User2” ait seulement les droits d’accès en lecture, on va utiliser son groupe. D’abord, on va mettre le groupe “Plateformeurs” propriétaire du fichier droits.txt :

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

ou

```
sudo chown :Plateformeurs droits.txt
```

Pour que “User2” ait accès à droits.txt seulement en lecture, on peut maintenant faire :

```
sudo chmod g=r droits.txt
```

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

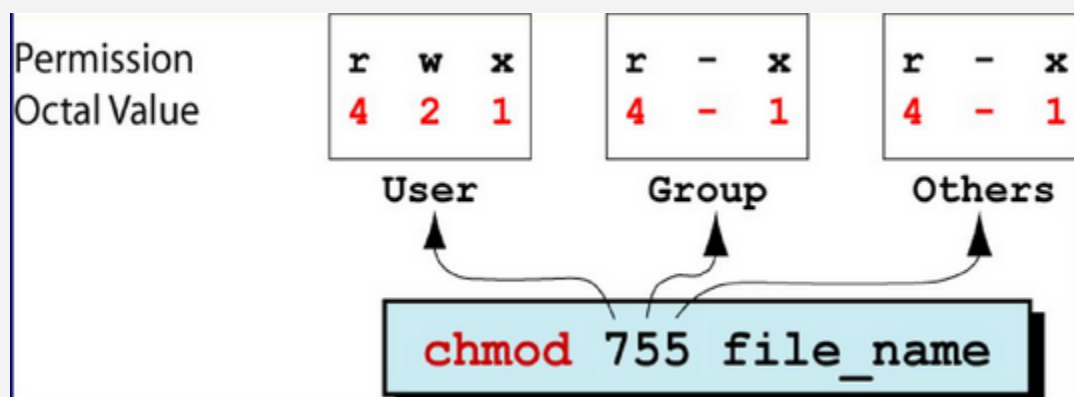
```
sudo chmod o=r groupes.txt
```

Changer les droits du fichier pour que le groupe “Plateformeurs” puissent y accéder en lecture/écriture :

```
sudo chgrp Plateformeurs groupes.txt
```

```
sudo chmod g=rw groupes.txt
```

Il existe une façon de définir les droits pour tous en une seule commande grâce aux valeurs octales



Pour donner les droits de lecture(r), d’écriture(w) et d’exécution(x) au propriétaire, les droits de lecture(r) au groupe propriétaire et aucuns droits aux autres, on peut faire :

```
chmod 740 droits.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'
```

Les commandes ci-dessus vont créer des alias temporaires, pour les créer durablement il faut modifier le fichier .bashrc ou le fichier .bash\_aliases, si ce dernier n'existe pas, il faut le créer et décommenter les lignes suivantes dans le .bashrc si ce n'est déjà le cas :

```
if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi
```

puis ajouter les alias :

```
alias la='ls -la'
alias update='apt-get update'
alias upgrade='apt-get upgrade'
```

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

```
export USER=alban
```

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

```
source ~/.bashrc
```

Afficher les variables d'environnement

```
env
```

Ajouter à votre Path le chemin "/home/votre utilisateur/Bureau", temporairement :

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

ou dans bashrc :

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

## JOB 6 :

Pour télécharger le fichier :

```
wget https://drive.google.com/file/d/11dSelXQuH4tih6zesbv-6OMEpr-sT77X/view?usp=sharing
```

On obtient un fichier .tar.gz que l'on peut décompresser de la manière suivante :

```
tar xzvf Copie\ de\ Ghost\ in\ the\ Shell.tar.gz
```

- "x" : extraire une archive.
- "z" : permet de décompresser le "gzip".
- "v" : mode verbeux , permet d'afficher dans le détail les fichiers décompressés.
- "f" : indication de la donnée à décompresser.
- Les "\" servent à échapper les espaces qui créeraient une erreur.

## JOB 7 :

Créer un fichier "une\_commande.txt" avec le texte suivant "Je suis votre fichier texte", on redirige la sortie de echo vers un fichier, si le fichier n'existe pas, il sera créé :

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

Compter le nombre de lignes présentes dans votre fichier de source apt et enregistrer le

résultat dans un fichier nommé “nb\_lignes.txt” :

```
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” :

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

ou :

```
cat /etc/apt/sources.list | tee save_sources
```

Pour faire une recherche des fichiers commençant par un point ( . ) et rechercher à l’intérieur de ceux-ci le mot “alias”, on peut utiliser la commande suivante :

```
find ~+ -maxdepth 1 -type f -name ".*" -print -exec grep --color=auto -n "alias" {} \;
```

Cette commande mérite une explication détaillée :

- **find** est la commande qui permet de rechercher.
- **~+** est pour afficher le chemin complet de chaque fichier trouvé pour avoir un résultat plus clair.
- **-maxdepth 1** est pour éviter d’aller chercher en profondeur dans l’arborescence, on veut rester dans le répertoire courant.
- **-type f** pour dire que l’on ne veut que les fichiers.
- **-name “.\*”** pour spécifier que l’on recherche les fichiers commençant par un point ( . ).
- **-print** pour afficher chaque résultat du find, pour plus de clarté.
- **-exec** permet de lancer une commande sur chaque occurrence trouvée.
- **grep** est donné à -exec, il sera donc appelé pour chaque occurrence trouvée.
- **--color=auto** pour colorer les résultats.
- **-n** est l’option est pour afficher le numéro de la ligne où il trouve la chaîne de caractère spécifiée après.
- **“alias”** est la chaîne de caractères à chercher.
- **{}** sera remplacé par le chemin de chaque fichier trouvé.
- **;** est pour indiquer la fin de -exec.

Une autre solution est de faire simplement :

```
grep alias .*
```

## POUR ALLER PLUS LOIN :

Installer la commande tree :

```
sudo apt install tree
```

Lancer la commande tree en arrière-plan qui aura pour but d'afficher toute l'arborescence à partir de la racine ( / ) en enregistrant le résultat dans un fichier "tree.save" :

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

lister les éléments présents dans le dossier courant et utiliser directement le résultat de votre première commande pour compter le nombre d'éléments trouvés :

```
ls | wc -l
```

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 update && sudo apt upgrade
```