# SHELL
# &&
# GHOST IN THE SHELL

83 72 69 76 76 10 38 38 10 71 72 79 83 84 32 73 78 32
84 72 69 32 83 72 69 76 76 10

# Utilisation du terminal

## JOB 01 :

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

Chaque commande offre des options qui peuvent modifier son comportement. Les options sont souvent nommées par une seule lettre et précédées d'un tiret  - (tiret du 6).
exemple : ls -a  (affichera les fichiers invisibles commençant par un point)
On peut connaître les différentes options d'une commande en écrivant : [nom de la commande] --help
exemple : ls --help
exemple : ls -l
-l : format long d'affichage
Avec un argument :
touch readme.txt
readme.txt est un argument

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

Le premier mot tapé est une commande. Les lettres tapées après un - (tiret du 6), et les mots tapés après 2 tirets, sont des options. Le reste constitue les paramètres (exemple un nom de fichier).
exemple : ls -a -S
ls : affiche tous les fichiers et répertoires
-a : affiche les fichier commençant par un point (non visible)
-S : trie selon la taille du fichier avec les plus grands au début
-l : affiche la version longue
--help : aide sur la commande

## JOB 02 :

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

**head et tail**

- afficher les 10 premières lignes du fichier “.bashrc”
On utilise la commande head [nom du fichier]
exemple : head .bashrc

- afficher les 10 dernières lignes du fichier “.bashrc”
On utilise la commande tail [nom du fichier]
exemple : tail .bashrc
- afficher les 20 premières lignes du fichier “.bashrc”
On utilise la commande head -n20 [fichier]
exemple : head -n20 .bashrc

- afficher les 20 dernières lignes du fichier “.bashrc”
On utilise la commande tail -n20 [fichier]
exemple : tail -n20 .bashrc

## JOB 03 :

Votre fichier de documentation contiendra les actions ci-dessous ainsi que leur équivalent en ligne de commande Linux :

**apt et wget**

- Installer le paquet “cmatrix”

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

- lancer le paquet que vous venez d’installer

```
cmatrix
```

(quitter avec ctrl + C)

- Mettre à jour son gestionnaire de paquets

```
sudo apt update
```

- Mettre à jour ses différents logiciels

```
sudo apt update
```

update : Un fichier d’index est créé pour lister les mises à jour disponibles. Il servira de référence pour l’installation de nouvelles mises à jour.

```
sudo apt upgrade
```

upgrade : Cela installe les mises à jour identifiées avec apt update sans supprimer les paquets installés.

`memo : (dpkg -i : pour installer packet téléchargé)`

- Télécharger les internets : Google

```
wget https://dl.google.com/linux/direct/google-chrome-stable_current_amd6
```

wget : programme en ligne de commande non interactif de téléchargement de fichiers depuis le Web. Il supporte les protocoles HTTP, HTTPS et FTP ainsi que le téléchargement au travers des proxies HTTP

```
sudo apt install ./google-chrome-stable_current_amd64.deb
```

Installe ou met à jour le paquet Google Chrome

4.deb

**reboot et halt**

- Redémarrer votre machine

```
sudo reboot
```

- éteindre votre machine

```
sudo halt
```

## Job 04 :

Votre fichier de documentation contiendra les actions ci-dessous ainsi que leur équivalent en ligne de commande Linux :

**groupadd et adduser**

Créer un fichier users.txt qui contiendra User1 et User2 séparé par un retour à la ligne
touch users.txt
On peut éditer le fichier avec la commande vi ou nano.

- Créer un groupe appelé “Plateformeurs”
sudo groupadd Plateformeurs

- Créer un utilisateur appelé “User1”
sudo adduser user1 (mettre User1 quand fullname demandé)

- Créer un utilisateur appelé “User2”
sudo adduser user2

- Ajouter “User2” au groupe Plateformeurs
La commande pour cela est sudo adduser [nom utilisateur] [nom groupe]
exemple : sudo adduser user2 Plateformeurs

- Copier votre “users.txt” dans un fichier “droits.txt”
cp [OPTIONS] SOURCE... DESTINATION
exemple : cp -a /home/morgane/users.txt /home/morgane/droits.txt

- Copier votre “users.txt” dans un fichier “groupes.txt”
exemple : cp -a /home/morgane/users.txt /home/morgane/groupes.txt

Quelques commandes :
sudo adduser [nom] => ajoute un utilisateur
sudo groupadd [nomGroupe] => ajoute un groupe
id -Gn => connaître ses groupes
id -Gn [nomUtilisateur] => connaître les groupes d’un utilisateur spécifique

usermod => La commande usermod permet de modifier toutes les options fixées par la commande adduser. Nécessite parfois d'utiliser sudo avec.
usermod -aG [nomGroupe] [nomUtilisateur] => ajoute un utilisateur dans un groupe
sudo deluser [user] [group] => supprime un utilisateur d'un groupe
/etc/group => permet de voir tous les groupes avec les utilisateurs associés

Sur les mot de passe :
Accéder au fichier avec les mots de passe (cryptés) :
sudo cat /etc/shadow
Pour avoir des informations sur le mot de passe d'un utilisateur :
sudo chage -l [nomUtilisateur] (chage affiche les caractéristiques de validité du mot de passe d'un utilisateur)
On peut aussi faire les deux commandes en une :
sudo cat /etc/shadow | sudo chage -l user2

**chown et chmod**

- Changer le propriétaire du fichier "droits.txt" pour mettre "User1"
On utilise : sudo chown [nouveau propriétaire] [nom_fichier_ ou du dossier]
exemple : sudo chown User1 droit.txt

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

Explication des commandes de permissions utilisateurs:
Les types d'utilisateurs:
u (pour user) : l'utilisateur auquel appartient le fichier/dossier
g (pour group) : le groupe auquel appartient le fichier/dossier
o (pour other) : les autres utilisateurs
a (pour all) : représente l'ensemble des trois catégories

```
Les modifications à faire:
+ : ajouter
- : supprimer
= : affectation

Les types de droits:
r (pour read) : droit de lire le fichier/dossier
w (pour write) : droit de modifier le fichier/dossier
x (pour execute) : droit d'exécuter le fichier
- (pour rien) : aucun droit sur le fichier/dossier

Connaître les droits d'un fichier ou d'un dossier :
Faire : ls -l
Les fichiers et dossiers s'affichent sous cette forme :
drwxr-xr-x  2 morgane morgane     4096 22 sept. 13:12  Documents

(-rwxrw-r--) : les permissions du fichier/dossier. De gauche à
droite, permission du propriétaire, puis du groupe, puis des autres
utilisateurs
(2) : nombre de liens
(morgane) : nom du propriétaire
(morgane) : nom du groupe propriétaire
(4096): taille du fichier en byte
(22 sept. 13:12) : date de la dernière modification
(Documents) : nom du dossier ou du fichier
```

Dans la commande `sudo chmod o+r droits.txt` nous avons donc o pour les autres utilisateurs hormis le propriétaire, le + pour ajouter et r pour uniquement lire le fichier droits.txt.

- Changer les droits du fichier “groupes.txt” pour que les utilisateurs puissent accéder au fichier en lecture uniquement
exemple : `sudo chmod a=r groupes.txt`
- Changer les droits du fichier pour que le groupe “Plateformeurs” puisse y accéder en lecture/écriture.
2 étapes :
  - On met en propriétaire du fichier groupe.txt un membre du groupe Plateformeurs avec chown. Sachant que user2 est le seul

membre du groupe plateformeurs. Il va devenir propriétaire du fichier.
exemple : `sudo chown user2 groupes.txt`

- On ajoute une permission g (pour groupe auquel appartient le fichier) = (pour affectation) et rw (pour lecture/écriture)
  exemple : `sudo chmod g=rw groupes.txt`

## JOB 05 :

Votre fichier de documentation contiendra les actions ci-dessous ainsi que leur équivalent en ligne de commande Linux :

**alias**

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

- Ajouter un alias qui permettra de lancer la commande “apt-get update” en tapant “update”
exemple : `alias update='sudo apt-get update'`
(sudo est là pour éviter problème de permission)

- Ajouter un alias qui permettra de lancer la commande “apt-get upgrade” en tapant “upgrade”
exemple : `alias upgrade='sudo apt-get upgrade'`

- Ajouter une variable d’environnement qui se nommera “USER” et qui sera égale à votre nom d’utilisateur
exemple : `USER='morgane'`
On peut voir si la variable a été prise en compte en tapant `env`
On peut appeler la variable avec : `$USER`

- Mettre à jour les modifications de votre bashrc dans votre shell actuel
`source ~/.bashrc`
ou écrire directement dans le fichier .bashrc avec la commande :
`nano .bashrc`

- Afficher les variables d’environnement
exemple :
`env`

- Ajouter à votre Path le chemin "/home/'votre utilisateur'/Bureau"
Dans le fichier /home/user/.bashrc, avec la commande nano écrire :
`PATH=$PATH:/home/morgane/Bureau`

## JOB 06 :

Vous devez télécharger l'archive suivante et la désarchiver seulement avec le terminal.
Cette manipulation vous permettra d'accéder à la suite du sujet.
https://drive.google.com/file/d/11dSelXQuH4tih6zesbv-6OMEpr-sT77X/view?usp=sharing
Théoriquement:
wget -c https://drive.google.com/file/d/11dSelXQuH4tih6zesbv-6OMEpr-sT77X/view?usp=sharing -O - | sudo tar -xz

**Décompresser une archive**

Pour décompresser si on a téléchargé le fichier :
`tar -xzvf 'Copie de Ghost in the Shell.tar.gz'`
On obtient le fichier 'Ghost in the Shell.pdf'

## JOB 07

Toutes les actions sont à réaliser en une seule commande

**> , >> et |**

Maintenant, vous allez approfondir les commandes, avec les caractères suivants “> < >> <<|”, votre fichier de documentation contiendra les actions ci-dessous ainsi que leur équivalent en ligne de commande Linux :

- Créer un fichier “une_commande.txt” avec le texte suivant “Je suis votre fichier
texte”. On écrit echo “votre texte”>nom_du_fichier.txt
exemple : echo “Je suis votre fichier”>une_commande.txt
Un seul > permet de réécrire le fichier de 0.
Deux >> permet d’ajouter des lignes au fichier.

- Compter le nombre de lignes présentes dans votre fichier de source apt et les enregistrer dans un fichier nommé “nb_lignes.txt”
commande qui compte + commande qui enregistre nombre de ligne
wc : commande qui compte
-l : le nombre de ligne
wc -l /etc/apt/sources.list>/home/morgane/nb_lignes.txt

head -20 /etc/apt/sources.list>/home/morgane/nb_lignes.txt (permet de copier un nombre de lignes d’un fichier prédéfinis dans un autre fichier qui est créé)

- Afficher le contenu du fichier source apt et l’enregistrer dans un autre fichier appelé “save_sources”
commande qui affiche le contenu + commande qui enregistre l’affichage dans un autre fichier
cat /etc/apt/sources.list>/home/morgane/save_sources.txt

Recherche des fichiers commençant par un point :
find -name .*
find sert à faire une recherche de fichier

- Faites une recherche des fichiers commençant par “.” tout en cherchant le mot alias qui sera utilisé depuis un fichier
La commande :
```
ls -a | grep .* | grep alias
```
Ou avec l’alias la:
```
la | grep .* | grep alias
```

## Pour aller plus loin ...

Votre fichier de documentation contiendra les actions ci-dessous ainsi que leur équivalent en ligne de commande Linux en utilisant seulement les caractères suivants “| || & &&” :

- Installer la commande tree
apt-get update && apt-get install tree

- Lancer la commande tree en arrière-plan qui aura pour but d’afficher toute l'arborescence de votre / en enregistrant le résultat dans un fichier “tree.save”

  - Pour lancer la commande mais pas en arrière plan :
    tree />/home/morgane/tree.save
  - Pour placer une tâche de premier plan en arrière plan :
    bg [job_spec ...]
  - Pour lancer une commande en arrière plan sans job_spec on utilise ‘&’ à la fin
    tree />/home/morgane/tree.save $
  - Pour lancer la commande mais en arrière plan on ajoute &:
    tree />/home/morgane/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.
Pour le fichier tree.save :
wc -l /home/morgane/tree.save>/home/morgane/nb_elements_tree.txt && cat home/morgane/nb_elements_tree.txt
wc -l /home/morgane/tree.save : comptage de ligne de tree.save (sachant que une ligne = un élément)
/home/morgane/nb_elements_tree.txt : création d’un fichier nb_elements_tree.txt avec le nombre total de ligne dans tree.save
&& : pour que la deuxième commande s'exécute si la première a été effectuée avec succès

cat home/morgane/nb_elements_tree.txt : affiche sur le terminal ce qu'il y a d'écrit sur le fichier nb_elements_tree.txt soit le nombre de ligne dans tree.save

- Lancer une commande pour update vos paquets, si l'update réussi 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

## Opérateurs de chaînage linux

### & esperluette

Permet d'exécuter une commande en arrière-plan. Il se met à la fin de la commande
exemple : ping c5 www.2error.net &

### ; point virgule

L'opérateur point-virgule permet d'exécuter, plusieurs commandes en une seule fois et l'exécution de commande se fait séquentiellement.
exemple : /home/2error# apt-get update ; apt-get upgrade ; mkdir test
La combinaison de commandes ci-dessus exécutera d'abord update instruction, alors upgrade instruction et enfin créera un 'test' sous le répertoire de travail actuel.

### && et

Le AND Operator (&&) n'exécute que la deuxième commande, si l'exécution de la première commande est un succès.

### || ou

Permet d'exécuter la deuxième commande uniquement si l'exécution de la première commande échoue.

### | PIPE

La sortie de la première commande agit comme une entrée de la deuxième commande. Cela permet, entre autres, de diviser des problèmes complexes en sous-problèmes plus petits, et de profiter ainsi d'une meilleure vue d'ensemble.
exemple : `cat .bashrc |grep "alias"`

### ! note opérateur

Cette commande exécutera tout sauf la condition fournie.
exemple : `/user rm -r !(*.html)`
Supprime tous les fichiers sauf les fichiers 'html' en une seule fois.

### {} combinaison de commande

Combinez deux ou plusieurs commandes, la deuxième commande dépend de l'exécution de la première commande.
exemple : `mkdir -p test/{coucou/{titi,toto},tata}`

### () priorité

L'Opérateur permet d'exécuter les commandes dans l'ordre de priorité.
exemple : `(Command_x1 && Command_x2) || (Command_x3 && Command_x4)`
Dans la pseudo-commande ci-dessus, si Command_x1 échoue, Command_x2 échoue également mais toujours Command_x3 et Command_x4 s'exécute dépend de l'état de sortie de Command_x3.

### () concaténation

exemple : `nano test(1).txt`

## Programmes fun de terminal linux

### Cow say

Installer cowsay :
```
apt-get update && apt-get install cowsay
```
Puis taper cowsay :
```
cowsay hello df
```
Autres exemples :

- koala :
  ```
  cowsay -f koala Hey, how are you?
  ```
- dragon :
  ```
  cowsay -f dragon I dare you to come a little closer !
  ```

- ```
  cowsay -f /usr/share/cowsay/cows/eyes.cow FromLinux.net
  ```

### Steam locomotive

Permet de faire défiler un train
Installer steam locomotive

```
sudo apt-get install sl
```

Commande :
```
sl
```
Options :
```
-a: mode accident. ...
-l: montre un train plus petit mais avec plus d'autocars.
-F: Un train volant.
-e: permet l'interruption par Ctrl + C.
```

### Lolcat

Fait des couleurs arc-en-ciel sur les lettres
Installer lolcat
```
sudo apt-get install lolcat
```

Exemple utilisation avec un autre programme
```
cmatrix | lolcat
```
```

## Figlet

Ecrire des chose en ASCII
Installer figlet
`sudo apt-get -y install figlet`
Utilisation :
`figlet Hello world !`

Exemple avec les opérateurs de chaînage :

```
(sl | lolcat) && (sl -l | lolcat) && ( sl -F | lolcat) && ( cmatrix
| lolcat)
```

```
sl && (sl -l | lolcat) && ( sl -F | lolcat)
```

```
echo | fortune>>citation.txt && cat citation.txt | lolcat
```

`echo | fortune>>quote.txt` (fortune est une commande particulière qui fait des citations, ici les citations seront écritent dans un fichier quote.txt)

```
cat .bashrc | lolcat
```

Comme c'est long à écrire on peut en faire des alias et les inscrire dans le fichier .bashrc

```
alias justtrain='sl && (sl -l | lolcat) && ( sl -F | lolcat)'
alias rainbow="(sl -l | lolcat) && (cmatrix | lolcat)"
alias promo="figlet La plateforme_Start | lolcat"
```

**Faire des petits scripts**

Dans son /home/user créer un dossier pour les scripts:
exemple : mkdir dossier_script
Créer un fichier pour chaque script avec une extension .sh
exemple : touch monscript.sh
Rendre le fichier exécutable s'il ne l'est pas (vérifiable avec ls -l):
exemple : chmod u+x monscript.sh
Ecrire dans le fichier => #!/bin/bash
Exécuter le script une fois dans le bon répertoire :
exemple : ./monscript.sh
On peut enfermer son script dans un alias
exemple : alias hello="./monscript.sh"

## Exemple de script :

```
#!/bin/bash
fine="It's a good day, I hope everything is fine for you!"
echo "Hello!"
echo $fine
echo "What is your name? Please just tell me it's not Kevin!"
read name
      if [ $name == "Kevin" ]
      then
      echo "This name is so ugly!I'm not sur your mom likes you"
      else
      echo "It's a really nice day "$name" ! And you have such a
      lovely name !"
      fi
echo "By the way, I can tell you than you're amazing. How many time
do you want than I tell you this ?"
read amazingtime
      amazing="You are amazing!"
      amazingcount=0
echo "I'm going to tell this $amazingtime time"
      while [ $amazingcount -lt $amazingtime ]
      do
      echo $amazing
      amazingcount=$((amazingcount + 1))
      done
```

**Autres infos de commande qui peuvent être utiles**

**Différence en commande cat et less**

less : Less est une commande et utilitaire de ligne de commande qui affiche le contenu d’un fichier ou d’une sortie de commande, une page à la fois.

cat : affiche tout le fichier et positionne à la fin du fichier