

# JOB 1

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

Pour cette action on utilise la commande **man ls**  
ou commande **ls --help** (--help est une option longue)

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

Pour afficher tous les fichiers y compris les fichiers cachés dans un répertoire on utilise la commande **ls -a**

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

Pour afficher tous les fichiers y compris les fichiers cachés et leurs droits sous forme de liste on utilise la commande **ls -a -l**

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

On ajoute des options après la commande de la manière suivante :

- des options courtes sont formées par un tiret suivi d'une lettre. Exemple : **-a**.
- des options longues formées par deux tirets suivis d'un mot. Exemple : **--size**
- des options courtes ou longues qui exigent un argument. Exemples : **--speed50** ou **-s 50**

Exemple : **ls -a -l**

- **ls** est une commande
  - **-a** et **-l** sont des options. Ce sont des options courtes c'est-à-dire une lettre par option
  - ils existent les options longues comme **home** sera une option longue simple
  - **nb\_fichiers**: sera une option longue attendant un argument en paramètre
  - **--** pour le comportement par défaut
- **Quelles sont les deux syntaxes principales d'écriture des options pour une commande ?**

Les principales syntaxes sont les suivants

1. commande -option
2. commande --option

## JOB 2

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

Pour cette action on utilise la commande **cat** suivi de nom de fichier concerné.

Utilisation : **cat** [option]...[fichier]

Concaténer les fichiers vers la sortie standard (l'écran).

- **Afficher les 10 premières lignes du fichier “.bashrc”**

Pour cette action on utilise la commande **head** qui fait la même chose que **cat** mais affiche seulement les 10 premières lignes d'un fichier.

- **Afficher les 10 dernières lignes du fichier “.bashrc”**

Pour cette action on utilise la commande **tail** qui fait la même chose que **cat** mais affiche seulement les 10 dernières lignes d'un fichier.

- **Afficher les 20 premières lignes du fichier “.bashrc”**

Pour afficher les 20 premières lignes du fichier il faudra ajouter une option **-n** à la commande **head** avec son attribut **20** comme ceci : **head** [options] <fichier>

Pour notre fichier “.bashrc” on utilisera la commande suivante :

**head -n 20 .bashrc**

On pourra toujours modifier l'attribut **20** pour lui donner la valeur en nombre de lignes souhaité à l'afficher.

Sans l'option -n et son attribut la commande head affiche le nombre de lignes par défaut qui est 10.

- **Afficher les 20 dernières lignes du fichier “.bashrc”**

L'option **n** fonctionne aussi avec la commande **tail** qui nous permet d'afficher les dernières lignes d'un fichier.

Pour notre fichier “.bashrc” on utilisera la commande suivante :

**tail -n 20 .bashrc**

Pareil ici, l'attribut **20** est modifiable et la valeur par défaut d'affichage de nombre de ligne en absence de l'option **n** est de 10.

## JOB 3

- **Installer le paquet “cmatrix”**

Pour installer le paquet “cmatrix” il faut saisir la commande suivante :

`sudo apt-get install cmatrix`

- **lancer le paquet que vous venez d’installer**

Pour lancer un logiciel déjà installer on saisi son nom.

Dans notre exemple la commande sera : `cmatrix`

- **Mettre à jour son gestionnaire de paquets**

Pour cette action on utilise :

`sudo apt-get update`

Cette commande met à jour la liste des packages disponibles et leurs versions, mais il n'installe ni ne met à niveau aucun package.

- **Mettre à jour ses différents logiciels**

Pour cette action on utilise :

`sudo apt-get upgrade`

Cette commande installe les nouvelles versions des paquets qu’on a déjà.

- **Télécharger les internets : Google**

`sudo apt-get install google-chrome-install`

- **Redémarrer votre machine**

Pour cela on utilise la commande : `reboot`

- **éteindre votre machine**

Pour éteindre la machine la commande :

`sudo halt`

## JOB 4

- **Créer un groupe appelé “Plateformeurs”**

Pour créer ce groupe on utilise la commande : `groupadd Plateformeurs`

- **Créer un utilisateur appelé “User1”**

Pour créer cet utilisateur on utilise la commande : `sudo useradd User1`

Suivi de la commande de création de son mot de passe : `sudo passwd User1`

Puis il faudra créer le mot de passe et le saisir une 2ème fois pour la vérification.

*On n'utilise pas `adduser` car dans ce cas là la commande n'accepte pas le nom d'utilisateur User1 en disant qu'il n'est pas bon.*

- **Créer un utilisateur appelé “User2”**

On saisit la même commande pour User2 : `sudo useradd User2`

Puis on configure son mot de passe : `sudo passwd User2`

- **Ajouter “User2” au groupe Plateformeurs**

Pour ajouter l'utilisateur “User2” dans le groupe Plateformeurs on doit saisir la commande suivante : `sudo -a -G Plateformeurs User2`

ou la commande : `gpasswd --add User2 Plateformeurs`

(vérifier si on a bien créé les utilisateurs et groupes : `getent groupe`)

- **Copier votre “users.txt” dans un fichier “droits.txt”**

Pour cela on utilise la commande `cp` comme ceci : `cp users.txt droits.txt`

- **Copier votre “users.txt” dans un fichier “groupes.txt”**

Pareil comme la commande précédente : `cp users.txt groupes.txt`

- **Changer le propriétaire du fichier “droits.txt” pour mettre “User1”**

Pour cela on utilise la commande `chown` comme ceci : `chown User1 droits.txt`

- **Changer les droits du fichier “droits.txt” pour que “User2” ai accès seulement en lecture**

Le propriétaire de fichier “droits.txt” est l'User1, pour donner accès seulement en lecture pour User2 qui est dans un groupe il entre dans le 2ème groupe de permissions donc on doit saisir la commande suivante

`chmod 740 droits.txt` qui donne accès à lecture écriture et exécution au propriétaire de fichier et seulement à la lecture au groupes.

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

Pour cette action on utilise la commande : `chmod 744` qui donne les droits de lecture aux utilisateurs qui sont dans les groupes et aux autres (others).

## JOB 5

- **Ajouter un alias qui permettra de lancer la commande “ls -la” en tapant “la”**

Pour créer un nouvel alias , on utilise la commande **alias** de cette façon :

**alias [NomAlias]='[commande avec ses options ou arguments]'**

Pour créer l'alias la qui remplace la commande ls -la on doit saisir :

**alias la='la -a'**

- **Ajouter un alias qui permettra de lancer la commande “apt-get update” en tapant “update”**

Pour cela on tape la commande suivante :

**alias update='sudo apt-get update'**

- **Ajouter un alias qui permettra de lancer la commande “apt-get upgrade” en tapant “upgrade”**

Pour cela on tape la commande suivante :

**alias upgrade='sudo apt-get upgrade'**

- pour voir les alias déjà créé il faut taper la commande : **alias**

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

Pour ajouter cette variable on saisit la commande suivante : **USER=hayk**

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

Pour trouver le fichier **.bashrc** il faut saisir la commande **ls -a** qui nous permet de voir les fichiers cachés, puis il faudra éditer ce fichier avec un éditeur de texte ex. **nano** en tapant : **nano .bashrc**

Dans l'éditeur de texte on doit saisir les alias et enregistrer le fichier.

Ça nous permettra d'avoir nos alias toujours actifs quand on relancera notre terminal ou on redémarre la machine.

- Si les alias sont pas pris en compte de suite il faut redémarrer le terminal ou saisir la commande suivante : **source ~/.bashrc**

- **Afficher les variables d'environnement**

Pour examiner les valeurs des variables d'environnement on tape : **printenv**

Pour examiner la valeur d'une variable en particulier, il suffit de spécifier son nom après la commande **printenv** comme ceci : **printenv USER**

- **Ajouter à votre Path le chemin "/home/votre utilisateur/Bureau"**

Pour ajouter ce chemin à notre PATH on doit saisir :

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

Après l'avoir exécuté et en effectuant la commande **echo \$PATH** on obtiendra un chemin de recherche qui va d'abord vérifier ici : **/home/hayk/Bureau**

Notre répertoire est en début de chemin de recherche. Pour le mettre en fin de chemin on saisit la commande suivante :

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

Ce changement disparaîtra une fois le terminal redémarré.

Pour le rendre permanent il faut éditer le fichier **.bashrc** et à la dernière ligne **export** **export PATH=/home/hayk/Bureau:\$PATH** puis enregistrer le fichier.

Pour le tester on ferme et rouvre la fenêtre du terminal ou on utilise le dot commande pour lire le **.bashrc** fichier, comme suit : **. .bashrc**

## **JOB 6**

- **Vous devez télécharger l'archive suivante et la désarchiver seulement avec le terminal.**

Pour cela on utilise la commande **wget** suivi du lien et séparé par une espace.

Puis pour décompresser un paquet on utilise : **tar -xzvf [nom\_de\_fichier.extension]**

## JOB 7

- **Créer un fichier “une\_commande.txt” avec le texte suivant “Je suis votre fichier texte”**

Pour écrire et créer un fichier en même temps on utilise **echo** et le symbole **>** de cette façon : **echo [texte du contenu de fichier] > [nom de fichier.extension]**

Pour notre commande ça sera : **echo Je suis votre fichier texte > une\_commande.txt**

- Pour ajouter du texte dans ce fichier il faudra remplacer **>** par **>>** comme ceci : **echo texte ajouté >> une\_commande.txt**

- **Compter le nombre de lignes présentes dans votre fichier de source apt et les enregistrer dans un fichier nommé “nb\_lignes.txt”**

Pour compter le nombre de lignes dans un fichier on utilise la commande **wc** avec l'option **-l**.

Pour notre cas il faudra saisir la commande suivante :

**wc -l /etc/apt/sources.list > /home/hayk/nb\_lignes.txt**

- **Afficher le contenu du fichier source apt et l'enregistrer dans un autre fichier appelé “save\_sources”**

Pour cela on utilise la commande **cat** qu'on va rediriger vers une autre sortie et pas la sortie standard=écran avec **>** et créer le fichier en même temps comme ceci :

**cat etc/apt/sources.list ; cat etc/apt/sources.list > /home/hayk/save\_sources**

L'opérateur point-virgule (;) vous permet d'exécuter plusieurs commandes successivement

- **Faites une recherche des fichiers commençants par “.” tout en cherchant le mot alias qui sera utilisé depuis un fichier**

Pour cela on utilise la commande **find** pour trouver toutes les fichiers qui contient le ‘.’ (point) dans leurs nom, on passe le résultat de **find** dans une nouvelle commande

**grep** à l'aide de pipe **|** qui pourra faire une recherche selon le contenu du fichier.

“**|**” permet de rediriger la sortie standard de la commande1 vers la commande2

comme entrée standard puis la commande2 traite l'entrée standard de commande1.

**find .\* | grep -rn "alias"**

- L'option **n** de **grep** permet d'afficher le numéro de ligne où apparaît le texte recherché ;
- L'option **r** de **grep** permet d'effectuer cette recherche récursivement ;

## 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**

`sudo apt-get install tree` (pour ubuntu)

- **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”**

`tree > 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**

Le moyen le plus simple de compter les fichiers dans un répertoire sous Linux consiste à utiliser la commande **ls** et à la diriger avec la commande **wc -l**.

Comme ceci : `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**

Si on veut que la deuxième commande ne s'exécute que si la première commande réussit, on les sépare avec l'opérateur logique AND, qui correspond à deux esperluettes ( **&&** ).

`sudo apt-get update && sudo apt-get upgrade`