

le Shell :

Job 1 :

Nous allons devoir afficher le manuel de la commande `ls`. Pour cela, la commande qui permet d'afficher les manuels de commandes spécifiques est la commande **MAN**.

```
pikadmin1@debian:~$ man ls
```

Pour afficher les fichiers qui sont cachés dans un répertoire de l'utilisateur nous allons devoir rajouter une option à notre commande `ls`. L'option **ALL**.

```
pikadmin1@debian:~$ ls -a
```

Pour pouvoir afficher d'une façon spécifique nous allons devoir rajouter une nouvelle option, l'option **LISTE**.

```
pikadmin1@debian:~$ ls -al
```

Comment ajouter des options à une commande ?

Pour pouvoir ajouter des options à une commande sous Linux, il vous faudra ajouter un - devant l'option que nous voudrions pour une commande.

```
pikadmin1@debian:~$ ls -a
```

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

les 2 syntaxes principales d'écriture pour des options pour une commande sont :

- " - "

```
pikadmin1@debian:~$ ls -a
```

- " -- "

```
pikadmin1@debian:~$ ls --all
```

Job 2 :

*Pour pouvoir lire un fichier depuis le terminal Linux vous aurez besoin de la commande : **CAT**.*

```
pikadmin1@debian:~$ cat .bashrc
```

*Maintenant nous voulons uniquement afficher les Dix premières lignes de notre fichier. Ici cela sera la commande **HEAD** qui nous sera utile.*

```
pikadmin1@debian:~$ head .bashrc
```

*À noter que la commande **HEAD** par défaut vous affichera les 10 premières lignes si vous voulez par exemple afficher les 20 premières lignes il faudra ajouter une option.*

```
pikadmin1@debian:~$ head -20 .bashrc
```

*Maintenant nous voulons par exemple afficher uniquement les 10 dernières lignes de notre fichier. Il nous faudra donc utiliser la commande **TAIL**.*

```
pikadmin1@debian:~$ tail .bashrc
```

*Comme pour la commande **HEAD**, la commande **TAIL** pour défaut affichera les 10 dernières lignes du fichier. Pour afficher donc par exemple 20 lignes il faudra aussi rajouter une option.*

```
pikadmin1@debian:~$ tail -20 .bashrc
```

À savoir que nous pouvons aussi faire comme ça :

```
pikadmin1@debian:~$ head -n 20 .bashrc
```

```
pikadmin1@debian:~$ tail -n 20 .bashrc
```

Job 3 :

Maintenant nous allons installer un logiciel supplémentaire **CMATRIX**. Pour cela nous allons utiliser une commande nécessitant un droit d'administrateur, voilà pourquoi notre commande commencera par **SUDO (Super Utilisateur DO)**.

```
pikadmin1@debian:~$ sudo apt-get install cmatrix
```

Il vous sera alors demandé un mot de passe pour pouvoir installer le logiciel.

```
[sudo] Mot de passe de pikadmin1 :
```

Maintenant pour vérifier que nous avons la dernière version de **cmatrix** nous allons le mettre à jour avec la commande **UPGRADE** qui nécessite aussi un droit d'administrateur.

```
pikadmin1@debian:~$ sudo apt-get upgrade cmatrix
```

Maintenant nous voulons installer un Nouveau Navigateur Internet car nous préférons utiliser Google Chrome plutôt que Firefox. Pour cela nous allons taper plusieurs commandes à la suite.

Par exemple tout d'abord nous aurons besoin de **WGET** qui est une commande permettant de télécharger des fichiers en ligne depuis le Terminal Linux.

ATTENTION tous les terminaux ne l'ont pas forcément d'installer au préalable.

Pour cela nous allons devoir vérifier s'il y est.

```
pikadmin1@debian:~$ wget --version
```

Si une version apparaît c'est que vous avez **WGET**, sinon nous allons devoir l'installer.

```
pikadmin1@debian:~$ sudo apt-get install wget
```

Maintenant que nous avons **WGET**, il nous faut donc taper la commande pour aller télécharger notre logiciel Google Chrome. Pour cela nous utiliserons comme dit précédemment **WGET**.

```
pikadmin1@debian:~$ wget https://dl.google.com/linux/direct/google-chrome-stable_current_amd64.deb
```

Maintenant que le fichier est téléchargé, nous allons devoir lancer l'installation. Pour cela nous allons utiliser la commande **DPKG**, étant donné que nous allons vouloir installer un nouveau logiciel nous aurons besoin d'une permission administrateur.

```
pikadmin1@debian:~$ sudo dpkg -i google-chrome-stable_current_amd64.deb
```

Il nous sera demandé un mot de passe :

```
[sudo] Mot de passe de pikadmin1 :
```

*Maintenant nous allons voir comment redémarrer votre machine depuis le terminal. Pour cette action, cela sera la commande **REBOOT** qui nous sera utile.*

```
pikadmin1@debian:~$ sudo reboot
```

Vous pouvez aussi programmer un redémarrage à une Heure précise avec cette commande, pour cela il vous suffit de rajouter une option.

```
|pikadmin1@debian:~$ sudo reboot -h hh:mm
```

où ici le hh et mm seront l'heure et les minutes.

*Maintenant pour pouvoir éteindre une machine depuis votre terminal, vous aurez besoin de la commande **SHUTDOWN**.*

```
pikadmin1@debian:~$ sudo shutdown -r now|
```

*Vous pouvez aussi comme **REBOOT** programmer un arrêt de la machine à une heure précise. Il vous faudra ajouter une option.*

```
pikadmin1@debian:~$ sudo shutdown -h hh:mm|
```

Job 4 :

*Pour créer un fichier nous utiliserons la commande **TOUCH**.*

```
pikadmin1@debian:~$ touch users.txt
```

Maintenant nous allons vouloir modifier ce document pour qu'il contienne User1 et User2.

*Nous utiliserons un des nombreux éditeurs de texte des terminaux : **NANO**.*

```
pikadmin1@debian:~$ nano users.txt
```

*On veillera à sauvegarder nos modifications. Quand nous ferons **CTRL+X** pour fermer il nous sera demandé si nous voulons sauvegarder nos modifications.*

*Maintenant nous allons vouloir créer un groupe nommé Plateformeurs. Pour cela nous utiliserons la commande **GROUPADD** qui nécessite une permission administrateur.*

```
pikadmin1@debian:~$ sudo groupadd Plateformeurs
```

Maintenant nous allons créer 2 nouveaux utilisateurs Nommés : User1 et User2.

*Pour cela nous aurons besoin de la commande **USERADD** qui nécessite aussi une permission administrateur.*

```
pikadmin1@debian:~$ sudo useradd User1
```

*Maintenant que notre groupe et nos utilisateurs sont créés, nous allons placer un de nos utilisateurs dans le groupe en question. Vous aurez besoin de la commande **ADDUSER**.*

```
pikadmin1@debian:~$ sudo adduser User2 Plateformeurs
```

*Maintenant nous allons copier notre fichier users.txt dans un fichier droits.txt. Pour cela nous utiliserons la commande **CP**.*

```
pikadmin1@debian:~$ cp users.txt droits.txt
```

Pareil pour copier le fichier user.txt dans groupes.txt

Maintenant nous allons changer l'utilisateur propriétaire du fichier. La commande utilisée est

qui nécessitera un droit administratif.

```
pikadmin1@debian:~$ sudo chown User1 droits.txt|
```

Maintenant que nous avons changé de propriétaire nous allons apprendre comment changer les permissions. Pour toucher aux permissions des fichiers, vous devrez utiliser la commande **CHMOD (Change Mod)** qui elle aussi aura besoin d'une permission administrateur.

A savoir un fichier à 3 types de groupes de permissions.

Le premier ciblera l'utilisateur Propriétaire du fichier : avec l'option **u**

Le Deuxième ciblera le groupe Propriétaire du fichier : avec l'option **g**

Le Troisième cible les autres qui ne sont Ni dans le premier Ni dans le Deuxième : avec l'option **o** (other)

Vous pouvez aussi cibler les 3 catégories avec le **all** : avec l'option **a**.

De plus il y a 3 différentes types de permissions pour un fichier :

La permission de Lire : qui sera écrite **r** (read)

La permission d'Écrire : qui sera écrite **w** (write)

La permission d'Exécuter : qui sera écrite **x** (execute)

Enfin pour vous soit ajouter, soit retirer les permissions.

Pour ajouter une permission il faut ajouter un **+**

Alors que pour supprimer une permission il faut ajouter un **-**

Par exemple, nous voulons que **User2** ait seulement accès au fichier **droits.txt** en lecture seule. Étant donné que qu'il n'est pas l'utilisateur propriétaire du Fichier et qu'il n'appartient pas au groupe propriétaire du Fichier, il sera dans la catégorie **other**.

```
pikadmin1@debian:~$ sudo chmod o-wx droits.txt
```

En deuxième exemple, nous voulons que pour les fichiers **groupes.txt**, nous voulons que tous les utilisateurs aient seulement l'accès en lecture seule.

Étant donné que nous voulons cibler tous les utilisateurs nous utiliserons la catégorie **all**.

```
|pikadmin1@debian:~$ sudo chmod a-wx groupes.txt
```

Enfin nous voulons que tous les utilisateurs dans le groupe **Plateformeurs** puissent avoir la permission de lire et écrire sur le fichier **groupes.txt**.

Étant donné que **Plateformeurs** n'est pas le groupe Propriétaire du fichier nous allons devoir le changer avec la commande **CHGRP (Change Groupe)**.

```
pikadmin1@debian:~$ sudo chgrp Plateformeurs groupes.txt
```

enfin nous devons lui ajouter la permission d'écrire. Nous ciblons alors la catégorie groupe.

```
pikadmin1@debian:~$ sudo chmod g+w groupes.txt
```

Job 5 :

*Tout d'abord nous allons ouvrir le fichier .bashrc avec **NANO** l'éditeur.*

*Ensuite à la fin nous allons voir comment créer des raccourcis de commandes avec la commande **ALIAS**.*

À noter que nous pouvons faire les alias en dehors du .bashrc, mais elles ne seront pas sauvegardées pour les prochaines sessions de terminal.

*Par exemple, prenons en exemple la commande **LS -LA** qui permet d'afficher tous le contenu du répertoire où l'on se trouve en mode liste ainsi que les fichiers cachés.*

*Avec **ALIAS** nous allons pouvoir faire en sorte que lorsqu'on tape **LA** cela tape la commande **LS -LA**.*

```
pikadmin1@debian:~$ alias la="ls -la"
```

*Ensuite pour le deuxième exemple, si l'on veut créer un **ALIAS** qui nous permettrait de faire les mises à jours des paquets, il ne faut pas oublier que comme il nous faut une permission administrateur il faudra rajouter **SUDO**.*

```
pikadmin1@debian:~$ alias update="sudo apt-get update"
```

*Pareil pour **UPGRADE**.*

```
pikadmin1@debian:~$ alias upgrade="sudo apt-get upgrade"
```

Maintenant que nous avons vu comment ajouter des commandes raccourcis, nous verrons comment nous allons ajouter des variables d'environnement.

*Mais tout d'abord **qu'est une variable d'environnement ?***

Les variables d'environnement sont des chaînes qui contiennent des informations sur l'environnement pour le système, et l'utilisateur ayant une session en cours. Certains logiciels utilisent ces informations pour déterminer où ils doivent placer des fichiers (par exemple les fichiers temporaires).

Pour créer une nouvelle variable d'environnement nous aurons besoin de la commande **EXPORT**.

```
pikadmin1@debian:~$ export USER=/home/nomutilisateur
```

Une fois votre modification sur le Bash terminé nous voulons que les modifications faites apparaisse sur notre session du terminal actuelle, Pour cela nous aurons besoin de la commande **SOURCE**.

```
pikadmin1@debian:~$ source .bashrc
```

Lorsque vous lancez le terminal, et que vous vous demandez comment je peux avoir la liste des variables d'environnement de la session, vous pouvez l'avoir en tapant la commande **ENV** ou **PRINTENV**.

Maintenant nous allons changer la valeur d'une variable d'environnement déjà existante.

```
pikadmin1@debian:~$ PATH=$PATH::home/"votre utilisateur"/Bureau|
```

Mais **Qu'es ce qu'un PATH ?**

PATH est la variable système utilisée par le système d'exploitation pour localiser les fichiers exécutables indispensables depuis la ligne de commande ou la fenêtre de terminal.

Job 6 :

Maintenant nous allons vous apprendre à désarchiver un fichier sur le terminal Linux. Pour cela il vous suffit d'abord d'aller chercher un fichier avec la commande **WGÉT**, vu précédemment ensuite nous utiliserons la commande **TAR -XZVF**.

```
pikadmin1@debian:~$ tar -xzvf "archive.tar.gz"
```


Job 7 :

Maintenant nous allons vous apprendre à créer un fichier, avec un texte noté à l'intérieur (je suis votre texte par exemple), qu'on puisse compter le nombre de lignes. Que le résultat du calcul de ligne s'enregistre dans un fichier nommé nb_ligne.txt. Qu'on affiche le contenu du fichier source apt et qu'on l'enregistre dans un autre fichier nommé save_sources. Ainsi que faire une recherche des fichiers commençant par . ainsi qu'en cherchant le mot alias qui sera utilisé depuis ce fichier. Et tout ça dans une même commande.

*Pour commencer plusieurs commandes apparaissent. Tout d'abord pour compter le nombre de lignes d'un fichier nous utiliserons la commande **WC**.*

Où wc -l affiche le nombre de ligne d'un fichier,

wc -w affiche le nombre de mots d'un fichier,

wc -m affiche le nombre d'octets d'un fichier,

wc -c affiche le nombre de caractères d'un fichier,

wc -L affiche uniquement la longueur de la plus longue ligne d'un fichier.

*Ensuite la commande pour faire une recherche sera la commande **FIND** suivie du caractère par lequel commencera le fichier.*

*pikadmin1@debian:~\$ find .**

Où l'astérisque sera pour dire tous les fichiers commençant par le caractère avant moi.

*echo "je suis votre fichier" >> une_commande.txt | wc -l /etc/apt/sources.list > nb_lignes.txt
| cat /etc/apt/sources.list > save_sources.txt | find .* -type f print | grep -ri "alias"*

Pour Aller plus loin :

Dans ce pour aller plus nous verrons d'autres éléments pour pouvoir lancer plusieurs commandes à la suite.

Tout d'abord | que l'on appelle le pipe vous permet de faire la distinction entre 2 commandes différents placés sur la même ligne.

le || s'appelle le double pipe, fera en sorte que si la première commande s'exécute sans erreur, la deuxième ne s'exécutera pas.

nous avons aussi le & qui permet d'imbriquer une commande avec une deuxième commande.

Mais que veut dire imbriquer : Disposer des éléments (par exemple, des tuiles, des blocs dans un programme informatique, etc.) de façon qu'ils se chevauchent ou soient les uns dans les autres. Pour faire simple en informatique on parlera de commande/fonction/fichier imbriquer pour des commande/fonction/fichier qui s'exécutent à la suite sans pour autant que cela soit considéré comme une autre commande.

Le && va vous permettre de faire en sorte que la commande suivante ne s'exécutera uniquement si la commande précédente s'est exécutée avec succès.

Donc pour ce sujet nous allons voir comment nous allons installer Tree, puis lancer cette même commande en arrière-plan (elle aura pour but d'afficher toute l'arborescence de votre /), tout en voulant enregistrer le résultat dans un fichier tree.save.

De plus, nous voudrions que cela liste les éléments présents dans le dossier courant et utiliser le résultat obtenu pour compter le nombre d'éléments trouvés.

Enfin nous voudrions lancer la commande pour UPDATE les paquets et faire en sorte que si l'update réussi cela lance l'upgrade sinon cela ne la lancera pas.

Pour Installer la commande tree nous ferons :

```
pikadmin1@debian:~$ sudo apt-get install tree|
```

Ensuite pour afficher l'arborescence avec la commande TREE il faudra utiliser :

```
pikadmin1@debian:~$ tree -o tree.save
```

Du coup la commande finale sera :

```
pikadmin1@debian:~$ sudo apt-get install tree |tree & tree -o tree.save |ls -l |  
wc -l| sudo apt-get update && sudo apt-get upgrade
```

Bonus :

*Maintenant nous allons apprendre comment installer **SSH**. Mais qu'est ce qu'une **SSH** ? **SSH**, ou **Secure Socket Shell**, est un **protocole réseau** qui permet **aux administrateurs d'accéder à distance à un ordinateur**, en toute sécurité. **SSH** désigne également **l'ensemble des utilitaires qui mettent en œuvre le protocole**.*

*Donc nous commençons par installer **SSH** sur notre Machine Virtuelle.*

Pour commencer nous allons vérifier que tout est à jour.

```
pikadmin1@debian:~$ sudo apt-get update
```

*Ensuite nous allons installer le serveur **OPENS****SSH**.*

```
pikadmin1@debian:~$ sudo apt-get install openssh-server
```

*Maintenant nous voulons générer une clé **SSH**.*

Pour cela nous allons utiliser la commande

```
pikadmin1@debian:~$ ssh-keygen -t rsa|
```