Documentation $HELL

JOB 7

Commande man

La commande man permet d'obtenir des détails sur une commande précise et s'utilise de la façon suivante :

```
man <commande>
```

Commande ls

La commande ls permet de "lister" les fichiers et dossiers présents dans le dossier actif. Elle s'utilise de la façon suivante :

```
ls
```

L'option **-a** permet d'afficher les fichiers cachés.
L'option **-l** permet d'afficher les informations sur les droits.

Les permissions se présentent sous la forme d'une chaîne de 10 caractères comme ceci : **drwxrwxrwx**

- le 1er caractère correspond au type d'élément, généralement - pour un fichier ou d pour un dossier
- 3 groupes de 3 caractères (dans l'ordre) :
  - **r** : lire
  - **w** : écrire
  - **x** : éxecuter
  - si pas de permission, la lettre est remplacée par un : **-**
- *1er groupe* : **permissions de l'utilisateur propriétaire**
- *2ème groupe* : **permissions du groupe propriétaire**
- *3ème groupe* : **permissions des autres utilisateurs**

7.1 Afficher le manuel de la commande ls :

```
man ls
```

7.2 Afficher les fichiers cachés du home de votre utilisateur :

```
ls -a
```

7.3 Afficher les fichiers cachés plus les informations sur les droits sous forme de liste :

```
ls -la
```

JOB 8

Commande more

La commande more permet d'afficher le contenu d'un fichier sans pouvoir modifier. Elle s'utilise de la façon suivante :

```
more <fichier>
```

Commande cat

La commande cat permet d'afficher un fichier en mode lecture directement depuis le terminal. Elle s'utilise de la façon suivante :

```
cat <fichier>
```

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

```
more .bashrc
```

ou

```
cat .bashrc
```

Commande head

La commande head permet d'afficher les premières lignes d'un fichier. Par défaut, la commande affiche les 10 premières lignes. Elle s'utilise de la façon suivante :

```
head <fichier>
```

> L'option **-X** permet d'afficher les X premières lignes du fichier.

Commande tail

La commande tail permet d'afficher les dernières lignes d'un fichier. Par défaut, la commande affiche les 10 dernières lignes. Elle s'utilise de la façon suivante :

```
tail <fichier>
```

> L'option **-X** permet d'afficher les X dernières lignes du fichier.

8.2 Afficher les 10 premières lignes du fichier ".bashrc"

```
head .bashrc
```

8.3 Afficher les 10 dernières lignes du fichier ".bashrc"

```
tail .bashrc
```

8.4 Afficher les 20 premières lignes du fichier “.bashrc”

```
head -20 .bashrc
```

8.5 Afficher les 20 dernières lignes du fichier “.bashrc”

```
tail -20 .bashrc
```

JOB 9

Commande groupadd

La commande groupadd permet de créer un groupe. Elle s'utilise de la façon suivante :

```
sudo groupadd <nom_groupe>
```

9.1 Créer un groupe appelé "Plateformeurs"

```
sudo groupadd Plateformeurs
```

Commande useradd

La commande useradd permet de créer un utilisateur. Elle s'utilise de la façon suivante :

```
sudo useradd <nom_utilisateur>
```

9.2 Créer un utilisateur appelé "User1"

```
sudo useradd User1
```

9.3 Créer un utilisateur appelé "User2"

```
sudo useradd User2
```

9.4 Ajouter "User2" au groupe Plateformeurs

```
sudo usermod -aG Plateformeurs User2
```

Commande cp

La commande cp permet de copier des fichiers d'un emplacement vers un autre. Elle s'utilise de la façon suivante :

```
cp <fichier_à_copier> <emplacement_copie>
```

9.5 Copier votre "users.txt" dans un fichier "droits.txt"

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

9.6 Copier votre "users.txt" dans un fichier "groupes.txt"

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

Commande chown

La commande chown permet de modifier le propriétaire d'un fichier. Elle s'utilise de la façon suivante :

```
sudo chown <propriétaire> <fichier>
```

9.7 Changer le propriétaire du fichier "droits.txt" pour mettre "User1"

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

Commande chmod

La commande chmod permet de modifier les permissions des utilisateurs sur un élément.

```
chmod <u g o a><+ - =><r w x><file/directory>
```

u : utilisateur propriétaire
g : groupe propriétaire
o : autres utilisateurs
a : tous les utilisateurs
+ : ajouter les permissions
- : enlever les permissions
= : définir les permissions
r : permission de lire
w : permission d'écrire
x : permission d'exécuter

9.8 Changer les droits du fichier "droits.txt" pour que "User2" ai accès seulement en lecture

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

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

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

9.10 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
```

JOB 10

Commande echo

La commande echo permet de renvoyer en sortie ce qu'on lui a donné en entrée pour l'afficher. Elle s'utilise de la façon suivante :

```
echo <chaîne_de_caractère>
```

10.1 Créer un fichier "une_commande.txt" avec le texte suivant "Je suis votre fichier texte"

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

Commande wc

La commande wc renvoie des informations sur un fichier tel que le nombre de lignes ou de mots qu'il contient ou encore le nombre d'octet.
Elle s'utilise de la façon suivante :

```
wc <fichier>
```

L'option **-l** permet de renvoyer uniquement le nombre de lignes.

10.2 Compter le nombre de lignes présentes dans votre fichier de source apt et les enregistrer dans un fichier nommé "nb_lignes.txt"

```
wc -l < /etc/apt/sources.list > nb_lignes.txt
```

10.3 Afficher le contenu du fichier source apt et l'enregistrer dans un autre fichier appelé "save_sources"

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

Commande find

La commande find permet de rechercher tous les fichiers qui correspondent à certains critères comme le nom, la taille, le type de ficher, la date de création/modification...
Elle s'utilise de la façon suivante :

```
find <directory> [option]
```

L'option **-name <nom_à_rechercher>** permet de rechercher en fonction du nom du fichier.

Commande pipe

La commande pipe permet d'effectuer plusieurs commandes la suite en une seule ligne
Elle s'utilise de la façon suivante :

```
<commande_1> | <commande_2>
```

Commande grep

La commande grep recherche une chaîne de caractère à l'intérieur même d'un fichier et affiche les lignes qui contiennent cette chaîne.
Elle s'utilise de la façon suivante :

```
        grep <"caractères_à_chercher"> <fichier>
```

10.4 Faites une recherche des fichiers commençant par "." tout en cherchant le mot alias qui sera utilisé depuis un fichier

```
find / -name .\* | grep "alias"
```

JOB 11

Commande apt install

La commande apt install est une commande de gestion de paquets permettant, entre autre, d'installer des paquets.
Elle s'utilise de la façon suivante :

```
apt install <paquets_à_installer>
```

11.1 Installer la commande tree

```
sudo apt install tree
```

11.2 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 &
```

L'utilisation de & permet d'effectuer la commande qui précède en arrière-plan

11.3 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 -1 | wc -l
```

L'option -1 de la commande ls permet de lister les éléments présents dans le dossier courant en affichant 1 élément par ligne.

11.4 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

```
apt update && apt upgrade -y
```

L'utilisation de && correspond au ET logique, la seconde commande ne s'exécute que si la première commande s'est réalisée avec succès