

ls : Affiche le contenu du dossiers courant

ls -a : Affiche les dossiers cachés

ls -l : Affiche les droits des dossier cachés.

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

head

2. Afficher les 10 premières lignes du fichier “.bashrc”

head -n 10 .bashrc : Affiche les 10 première ligne du répertoire .bashrc

3. Afficher les 10 dernières lignes du fichier “.bashrc”

tail -n 10 .bashrc : Affiche les 10 dernière ligne du répertoire .bashrc

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

head -n 20 .bashrc : Affiche les 20 première ligne .bashrc

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

tail -n 20 .bashrc : Affiche les 20 dernière ligne

nano users.txt : Ouvre l'éditeur de fichier et crée le fichier User.txt

6. Créer un groupe appelé “Plateformeurs”

groupadd Plateformeurs : Créer le groupe Plateformeurs

7. Créer un utilisateur appelé “User1”

useradd User1 : Crée l'utilisateur User1

8. Créer un utilisateur appelé “User2”

useradd User2 : Crée l'utilisateur User2

9. Ajouter “User2” au groupe Plateformeurs

usermod Plateformeurs User2 : intègres User2 dans le groupe les plateformeurs

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

cat users.txt > droits.txt ou **less users.txt > droits.txt** : permet de rediriger le flux de données stdout de users.txt dans droits.txt

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

cat users.txt > groupes.txt ou **less users.txt > groupes.txt** : permet de rediriger le flux de données stdout de users.txt dans groupes.txt

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

chown -R User1:User1 droits.txt : Modifie le propriétaire du dossier.

13. Changer les droits du fichier "droits.txt" pour que "User2" l'accès seulement en lecture

chmod 004 droits.txt : modifie les droits de tous les utilisateur dont User2 en lecture seule.

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

chmod 004 groupes.txt : modifie les droits du répertoire groupe et autorise la lecture seule pour tous les utilisateurs.

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

chown -R User2:Plateformeurs groupes.txt | chmod 064 groupes.txt : Définit un utilisateur ou un groupe comme propriétaire d'un dossier.

touch une_commande.txt | echo "Je suis votre fichier texte" > une_commande.txt : crée un répertoire une_commande.txt et envoie de le flux stdout Je suis votre fichier texte dans le répertoire une_commande.txt

wc -l /etc/apt/sources.list > nb_lignes.txt : permet de compter le nombre de ligne contenu dans le dossier /etc/apt/sources.list qui contient l'adresse des différents repository.

cat /etc/apt/sources.list > save_sources : affiche le contenu de /etc/apt/sources.list et redirige le flux de données save_sources.

find "." | grep -w 'alias' : Cherche les fichiers commençant par "." et récupère le fichier qui contient la chaîne de caractère alias

apt-get install tree : installe l'utilitaire tree

tree / > tree_save & : Affiche l'arborescence depuis la racine et redirige le flux de données dans le répertoire tree_save et lance le tout en arrière plan.

ls -l | wc -l : Liste le répertoire courant sous forme de ligne et envoie le résultat de la commande précédente dans wc qui permet de compter les lignes.

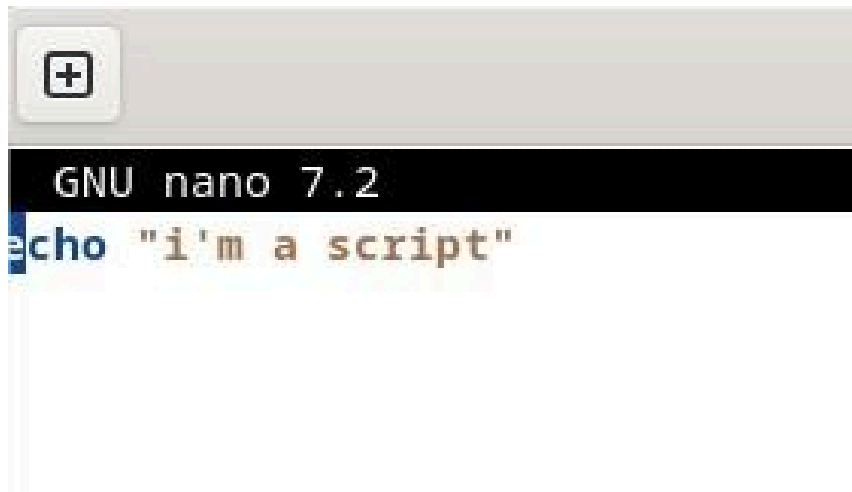
apt-get update && apt-get upgrade : mais à jour les paquets utilitaire et mis à jour la distribution.

Parti 2 Scripting

La commande pour accorder les droit a chaque script est chmod u+x

Job 12

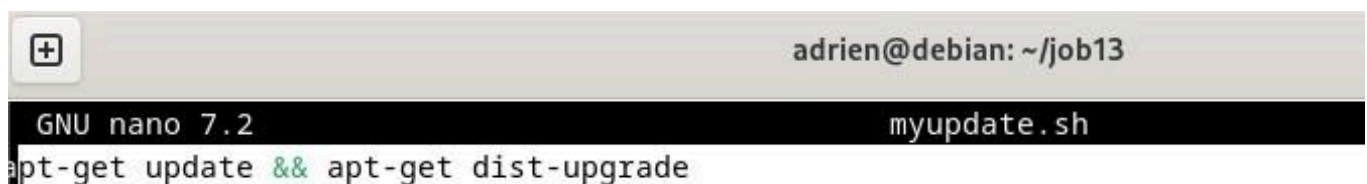
./myfirstscript.sh



```
GNU nano 7.2
echo "i'm a script"
```

Job 13

./myupdate.sh



```
adrien@debian: ~/job13
GNU nano 7.2 myupdate.sh
apt-get update && apt-get dist-upgrade
```

Job 14

./add.sh



```
adrien@debian: ~/job14
GNU nano 7.2 add.sh
var1=$1
var2=$2

som=$((var1+var2))

echo $som
```

Job 15

./argument.sh



```
adrien@debian: ~/job15
GNU nano 7.2 argument.sh
var1=$1
var2=$2
```

Job 16

./hello_bye.sh



```
adrien@debian: ~/job16
GNU nano 7.2 hello_bye.sh
var1=$1

if [ $var1 == Hello ]
then
    echo "Bonjour, je suis un script !"
elif [ $var1 == Bye ]
then
    echo "Au revoir et bonne journée !"
else
    echo "erreur 404"
fi
```

Job 17

./my_calculator.sh

```
adrien@debian: ~/job17
GNU nano 7.2 my_calculator.sh
var1=$1
var2=$2
var3=$3

if [ $var2 = + ]
then
    som=$(( $var1+$var3 ))
    echo $som
elif [ $var2 = - ]
then
    som=$(( $var1-$var3 ))
    echo $som
elif [ $var2 = x ]
then
    som=$(( $var1*$var3 ))
    echo $som
elif [ $var2 = / ]
then
```

Job 18

./myloop.sh

```
adrien@debian: ~/job18
GNU nano 7.2 myloop.sh
var=0

while [ $var -le 10 ]
do
    echo "je suis un script qui arrive à faire une boucle" $var
    ((var++))
done
```