

JOB 1

Afficher le manuel de la commande ls

`man ls`

Afficher les fichiers cachés du home de votre utilisateur

`ls -a /Users/dondada/`

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

`ls -al`

Comment ajouter des options à une commande ?

Pour ajouter des options on utilise le “-” et les options disponibles dans le manuel.

Exemple : `ls -al`, “ls” est la commande et “al” sont les options.

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

Les deux syntaxes possibles sont de regrouper les options après le “-” ou bien juste les écrire associées à un “-”.

Exemple :

- `ls -al`
- `ls -a -l`

JOB 2

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

`cat exemple.txt`

afficher les n premières lignes du fichier “.bashrc”

`head -n ~/.bashrc`

afficher les n dernières lignes du fichier “.bashrc”

`tail -n ~/.bashrc`

/* n = 10 ou 20

.bashrc ou .zshrc (sous

mac*/

JOB 3

Installer le paquet "cmatrix"

sudo apt-get install cmatrix

brew install cmatrix (macOS)

lancer le paquet que vous venez d'installer

cmatrix

Mettre à jour son gestionnaire de paquets

sudo apt update

Mettre à jour ses différents logiciels

sudo apt-update && sudo apt-upgrade

Télécharger les internets : Google

wget <https://www.google.com/>

Redémarrer votre machine

shutdown -r 0

éteindre votre machine

shutdown -h

JOB 4

Créer un groupe appelé "Plateformeurs"

```
groupadd Plateformeurs
```

Créer un utilisateur appelé "User1" et un autre appelé "User2"

```
useradd User1 && useradd User2
```

Ajouter "User2" au groupe Plateformeurs

```
gpasswd -a Users2 Plateformeurs
```

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

```
cp users.txt droit.txt
```

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

```
cp users.txt groupe.txt
```

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

```
chown User1 droit.txt
```

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

```
chmod 744 droit.txt
```

/ binaire pour 744 (ugo : u = 7, g = 4, o = 4)*

7 = rwx (tous les droits)

*4 = r- (lire uniquement) */*

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

```
chmod u-wx groupes.txt
```

Changer les droits du fichier pour que le groupe "Plateformeurs" puissent y accéder en lecture/écriture.

```
chmod g+rw groupes.txt
```

JOB 5

Ajouter un alias qui permettra de lancer la commande "ls -la" en tapant "la"

```
alias la='ls -la'
```

Ajouter un alias qui permettra de lancer la commande "apt-get update" en tapant "update"

```
alias update='apt-get update'
```

Ajouter un alias qui permettra de lancer la commande "apt-get upgrade" en tapant "upgrade"

```
alias upgrade='apt-get upgrade'
```

Ajouter une variable d'environnement qui se nommera "USER" et qui sera égale à votre nom d'utilisateur

```
export USER='dondada'
```

/* export crée la variable USER pour cette session uniquement
de manière permanente la commande est :

```
echo 'export USER='dondada' >> ~/.bashrc ou .zshrc */
```

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

```
source ~/.bashrc
```

Afficher les variables d'environnement

```
printenv
```

Ajouter à votre Path le chemin "/home/'votre utilisateur'/Bureau"

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

JOB 6

Décompresser une archive

```
tar -xzf Ghost\ in\ the\ Shell.tar.gz
```

JOB 7

Créer un fichier “une_commande.txt” avec le texte suivant “Je suis votre fichier texte”

```
alias “Je_suis_votre_fichier_texte”=’touch une_commande.txt’
```

ou

```
echo “Je suis votre fichier texte” > “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”

```
alias “j-ai_été_cré”=”grep -v -E "(^#|^$)" /ect/apt/souce.list | wc -l > nb_lignes.txt”
```

ou

```
grep -v -E "(^#|^$)" /ect/apt/souce.list | wc -l > nb_lignes.txt
```

Afficher le contenu du fichier source apt et l’enregistrer dans un autre fichier appelé “save_sources”

```
alias “par_quelqu-un_de_G ”=”cat /ect/apt/souce.list > save_sources”
```

ou

```
cat /ect/apt/souce.list > save_sources
```

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

```
alias “NIAL”=”grep -i “alias” .*”
```

ou

```
grep -i “alias” .*
```

Commande du JOB7 en une ligne :

```
Je_suis_votre_fichier_texte && j-ai_été_cré && par_quelqu-un_de_G && NIAL
```

ou

```
echo “Je suis votre fichier texte” > “une_commande.txt” && grep -v -E "(^#|^$)"  
/ect/apt/souce.list | wc -l > nb_lignes.txt && cat /ect/apt/souce.list > save_sources && grep -i  
“alias” .*
```

Pour aller plus loin...

Installer la commande tree

```
brew install tree
```

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 et utilisez directement le résultat de votre première commande pour compter le nombre d'éléments trouvé.

```
ls -a > buff && wc -l buff && rm buff
```

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

```
brew update && brew upgrade
```

commande enchaînée :

```
brew install tree && tree > tree.save && ls -a > buff && wc -l buff && brew update && brew upgrade
```

Pour aller ENCORE plus loin...

Installer SSH

SSH est installé de base sur MacOS, sinon sous Debian ou autre la commande est :
`sudo apt install openssh-client`

Générer une clé SSH

```
+---[RSA 3072]---+
|..... *==      |
| oo. = o        |
|... + O +       |
|.. * B B        |
|... # * S        |
|o.oB B E        |
|+o...+ o        |
|o. + o          |
| . .            |
+---[SHA256]-----+
```

Se connecter à une VM ou l'ordinateur d'un camarade via SSH

commande :

```
ssh [user]@[ip de connexion]
```

(sous mac il y a juste un paramètre à activer dans Réglage>Partage>Session à distance)

Configurer SSH pour empêcher le login root (root ne peut pas se connecter en SSH)

```
vi /etc/ssh/sshd_config
```

Ensuite il faut remplacer le PermitRootLogin yes par no

Modifier le port de connexion de SSH (autre que 22)

```
vim /etc/ssh/sshd_config
```

Rechercher port 22 et remplacer le "22" par un chiffre entre 1024 et 65536

Ensuite se connecter en SSH sans avoir à renseigner de mot de passe

Tout d'abord il faut avoir une clé SSH

Pour s'en assurer on peut lancer la commande :

```
ls -al ~/.ssh/id_*.pub
```

Si elle n'existe pas on verra cette ligne en réponse :

```
ls: cannot access /users/appsadm/.ssh/id_*.pub: No such file or directory
```

On utilise pour la créer la commande :

```
ssh-keygen -t rsa
```

La commande ci-dessous va s'occuper de renseigner la clé et la copier :

```
cat ~/.ssh/id_rsa.pub | ssh [user]@[ip de connexion] "mkdir -p ~/.ssh && cat >>
```

```
~/.ssh/authorized_keys"
```

Maintenant on peut essayer de se connecter avec la commande :

```
ssh utilisateur_distant@adresse_IP_distante
```

Uploader un fichier avec SSH (de votre pc ou VM vers le pc ou VM d'un camarade)

```
scp /Users/dondada/Desktop/ root@XXX.XXX.XX.XXX:/home/debian/Bureau/exemple.txt
```

Télécharger un fichier avec SSH (de votre pc ou VM vers le pc ou VM d'un camarade)

```
scp root@XXX.XXX.XX.XXX:/home/debian/Bureau/exemple.txt /Users/dondada/Desktop/
```

Limiter l'utilisation de SSH à un groupe particulier nommé "Plateforme_ssh"

```
vim /etc/ssh/ssh_config
```

Ajouter en ligne finale la ligne :

```
AllowGroups Plateforme_ssh
```

Quel est l'intérêt d'utiliser SSH ?

SSH va permettre de se connecter à un serveur ou un pc à distance dans le but d'en effectuer la gestion, transmission de fichiers, ...

Est-ce que les clés générées par SSH par défaut sont-elles assez sécurisées ? Justifier votre réponse

Est-ce que les clés SSH sont assez sécurisées ?

Je répondrai que oui dans le sens où celles-ci sont plus complexes car non générées par une personne. Ce qui les rend par conséquent plus "sécurisées" qu'un simple mot de passe cependant je pense pas qu'elles soient infaillibles pour autant.

Mais pour renforcer la sécurité il est possible de rajouter un mot de passe supplémentaire.

Citez d'autres protocoles de transfert ? Quelles sont les différences entre ses protocoles ?

Comme autres protocoles de transfert, nous avons :

- FTP : 1er protocole de transfert, aucun des deux canaux n'est crypté donc les données transmises peuvent être détournées.
- HTTP, HTTPS : Il définit le format des messages via lesquels navigateurs et serveurs communiquent, mais également la façon dont le navigateur web doit répondre à une requête Web.