

SHELL

Job 1

Afficher le manuel de la commande ls

man ls

Afficher les fichiers cachés du home de votre utilisateur

ls -a

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

ls -al

Récapitulatif des options de commande utilisées précédemment :

Pour lister dans un terminal les éléments non-cachés du dossier en cours, entrez la commande:

ls

Pour afficher tous les éléments, y compris les éléments cachés, il suffit d'ajouter l'argument -a («all» en anglais):

ls -a

Et pour n'afficher que les fichiers et dossiers cachés:

ls -d .*

Si vous ajoutez /, vous ne voyez que les dossiers cachés:

ls -d .*/

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

Les deux syntaxe principales pour l'écriture d'une commande sont - et ==

Job 2

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

More [nom du fichier]

afficher les 10 premières lignes du fichier “.bashrc”

more head [nom du fichier]

afficher les 10 dernières lignes du fichier “.bashrc”

more line [nom du fichier]

afficher les 20 premières lignes du fichier “.bashrc”

more head -20 [nom du fichier]

afficher les 20 dernières lignes du fichier “.bashrc”

more line -20 [nom du fichier]

Attention :

Les commandes “cat” et “less” permettent également de faire cela:

- cat [nom du fichier]
- less [nom du fichier]

“Les commandes renseignées fonctionnent sur la ligne de commande d’un OS Linux comme Debian. Cependant sur macOS le fichier .bashrc est vide et l’affichage des lignes ne peut s’exécuter.”

Job 3

Installer le paquet “cmatrix”

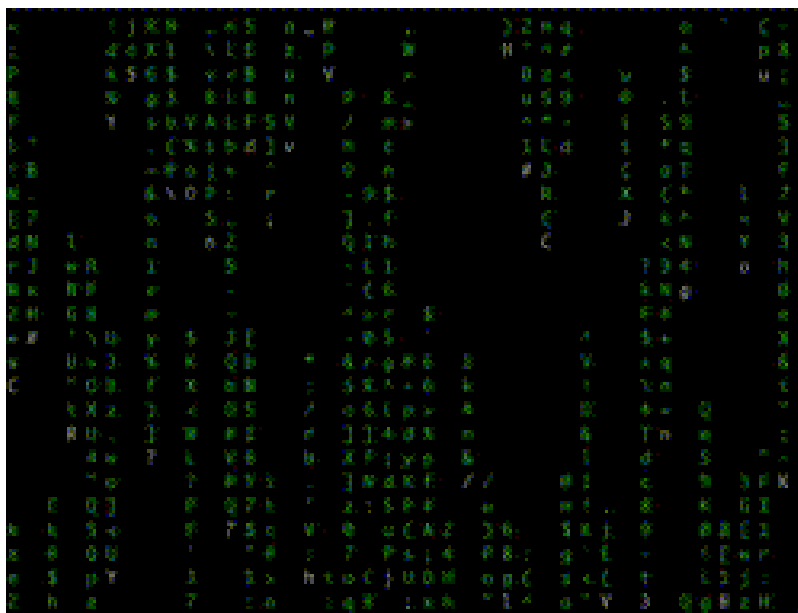
brew install cmatrix

brew install cmatrix

Lancer le paquet que vous venez d’installer

cmatrix

cmatrix



Mettre à jour son gestionnaire de paquets

brew update

Mettre à jour ses différents logiciels

brew upgrade nom du logiciel

Ouvrir Google via Terminal :

```
open -a Google\ Chrome.app
```

Redémarrer l'ordinateur :

```
sudo shutdown -r now
```

Eteindre l'ordinateur

```
sudo shutdown -h now
```

Job 4

Créer un groupe appelé "Plateformeurs"

```
dscl . create /Groups/nomdugroupe
```

Créer un utilisateur appelé "User1"

```
sudo dscl . -create /Users/User1
```

Créer un utilisateur appelé "User2"

```
sudo dscl . -create /Users/User2
```

Ajouter "User2" au groupe Plateformeurs

```
sudo dscl / -append /Groups/plateformeurs4 GroupMembership User2
```

Créer un fichier texte

```
touch droits5.txt
```

Copier "untexte.txt" dans le fichier "droits5.txt" qu'on a précédemment créé:

```
cp untexte.txt droits5.txt
```

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

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

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

```
sudo chown User1 droit.txt
```

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

```
chmod o+r droit.txt
```

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

```
chmod 444 groups.txt
```

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

```
chmod 464 groups.txt
```

Job 5

Pour ajouter un alias qui permettra de lancer la commande “ls -la” en tapant “la” nous faisons

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

Puis pour ajouter un alias qui permettra de lancer la commande “apt-get update” en tapant

“update” nous faisons

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

Enfin pour ajouter un alias qui permettra de lancer la commande “apt-get upgrade” en tapant “upgrade” nous faisons

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

Pour ajouter une variable d’environnement nous utiliserons la commande

```
USER=("$USER")
```

```
export USER
```

Ensuite pour mettre à jour les modifications du bashrc actuel nous tapons la commande

```
exec bash
```

Job 6

Nous devons ensuite télécharger une archive et la désarchiver, nous utiliserons les commandes suivantes :

```
.tar zxvf nom_du_fichier.tar.gz
```

Job 7

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

```
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"

```
touch nb_lignes.txt && wc -l nom_du_fichier.list > nb_lignes.txt
```

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

```
touch nom_du_fichier && echo sources.list > save_sources
```

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

```
find wholname . & find iname alias
```

Job Bonus

Installer SSH

```
brew install ssh ou apt-get install ssh
```

Générer une clé SSH

```
ssh-keygen
```

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

```
ssh nom-d'utilisateur@(adresse IP)
```

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

```
sudo nano /etc/ssh/sshd_config
```

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

```
scp nom-d'utilisateur@adressesIP:/home/user/etc.../fichier-coble
```

```
/home/user(receveur)/destination
```

Quel est l'intérêt d'utiliser SSH ?

SSH est sécurisé car cryptant les données échangées et demandant au deux communiquant d'échanger des clés de chiffrement afin de communiquer.

