

memento-linux

Durant le TP de system d'exploitation, j'étais l'utilisateur linux. Je vais donc présenter les divers commandes que j'ai utilisé : (les exemples, captures d'écrans viennent de mon système d'ou un affichage légèrement différent du à zsh)

Commandes de base

Commande pour lister les dossier dans le répertoire ou nous nous trouvons ?

```
ls
```

Une "variante" existe pour ce dernier afin d'afficher les dossier cacher et les permissions pour chaque utilisateurs qui est :

```
ls -la
```

```
> ls -la
total 1608
drwxr-xr-x 1 alan alan    110  9 nov.  22:31 .
drwx----- 1 alan alan  1366  3 déc.  18:08 ..
-rwxrwxrwx 1 alan alan     0 31 mars  2023 archive.tar.gz
drwxr-xr-x 1 alan alan   206 18 avril  2023 book
drwxrwxrwx 1 alan alan    42  1 mars  2023 codage
drwxr-xr-x 1 alan alan    58  1 sept.  12:58 cyber
drwxrwxrwx 1 alan alan   156 24 oct.   08:06 ENSIES
drwxr-xr-x 1 alan alan    88  1 nov.   00:28 games
drwxr-xr-x 1 alan alan   246 25 juin   21:16 .private
-rwxr-xr-x 1 alan alan 1643022 31 oct.   2020 zsh.exe
```

Nous pouvons également utiliser cette même commande suivie d'un nom de repertoire pour afficher ce qu'il s'y trouve (et ceci marche aussi avec le -la)

```
ls nom_du_repertoire
```

```
> ls cyber
crypto  dove  diskimageVirtualBox
```

Se déplacer de répertoire en répertoire

Pour changer de repertoire, nous utilisons la commande suivante :

```
cd nom_du_repertoire
```

Il y a aussi trois répertoire "spéciaux" pour lesquelles nous pouvons y retourner en une commande :

- notre home (fichier où nous nous trouvons au lancement du terminal)

```
cd ~
```

ou

```
cd
```

- la root (racine, la base où se trouvent tous les fichiers)

```
cd /
```

- se déplacer au répertoire précédent :

```
cd .. #nous pouvons ajouter des ../ pour revenir plusieurs répertoire avant
```

Suppression fichier ou répertoire

Pour supprimer un fichier nous utilisons la commande :

```
rm nom_du_fichier
```

Pour un répertoire nous pouvons utiliser si il est vide :

```
rmdir nom_du_repertoire
```

Sinon nous pouvons utiliser :

```
rm -r nom_du_repertoire
```

Droit super utilisateur

Parfois, nous n'avons pas les droits pour exécuter une action, comme mettre à jour le système ou modifier des fichiers root. Pour se faire, nous utilisons la commande suivante :

```
sudo
```

qui nous donne les permission administrateurs sur toutes nos commandes comme pour mettre à jour notre système :

```
sudo apt update      #met à jour les répots de debian de le but de mettre à
jour
sudo apt upgrade     #met à jour le système
```

Cette commande n'est pas relative à tout les OS linux. Elle n'est relative que à ceux utilisant apt. Par exemple Archlinux (le meilleur OS) utilise pacman et pour mettre à jour le système nous utilisons :

```
sudo pacman -Suy
```

Ou pour ceux l'ayant [installé](#) nous pouvons utiliser la commande suivante :

```
yay -Suy
```

(Il y a d'autre gestionnaires de packet mais je n'ai fait que prendre un autre exemple)

Modification des droits pour un fichier

En parlant de droits super utilisateurs, nous pouvons aussi une commande pour changer les divers droits (exécution, read, write) avec la commande suivante :

```
chmod code_des_droits nom_du_fichier
```

Pour obtenir les code à entrer, je vous recommande de regarder sur ce [site](#). Si nous voulons faire au plus simple, nous pouvons mettre le code 777 afin de donner toutes les permission à tout les utilisateur (**ce qui peu créer de problèmes de sécurité !!**).

afficher le contenu d'un fichier

Pour afficher le contenu d'un fichier, nous utilisons la commande suivante :

```
cat nom_du_fichier
```

Nous obtenons ceci :

```
> cat mqtt-envoie.js
const { error } = require('console');
const mqtt = require('mqtt');

const broker = 'https://test.mosquitto.org';
const topic = 'MASTER';

const client = mqtt.connect(broker);

function publishMessage() {
  const message = 'Salut mon pote ^^';
  client.publish(topic, message);
  console.log('le message ' + message + ' à été envoyé')
}

client.on('connect', () => {
  console.log('on se connect');
  setInterval(publishMessage, 10000)
});

client.on('error', (err) => {
  console.error('NOOOOONNNNNNNN une erreur');
});

client.on('close', () => {
  console.log("Et c'est fermé ... dommage, une prochaine fois peu être ^^");
});
```

Vim

Comme autre commande majeur que j'ai utilisé, il y a vim, plus précisément neovim. Ceci est un éditeur de texte qui s'installe avec la commande suivante:

```
sudo apt install vim neovim
```

Pour l'utiliser suite à l'installation, il nous suffit de faire :

```
nvim nom_du_fichier
```

Si ce dernier n'existe pas, alors il le créeras, donc nous ne sommes pas obligés d'utiliser la commande :

```
touch nom_du_fichier
```

Pour en créer un.

Pour son utilisation, je vous envoie vers une documentation de [vim](#) mais les commandes les plus utilisées (en mode normale en faisant ctrl + c) sont :

commande	effet
yy	copier la ligne
dd	couper la ligne
i	basculer en mode insertion
p	coller
u	annuler la dernière action (comme un ctrl + z)
ctrl + r	refaire la dernière action annulé (comme un ctrl + y)
:w	sauvegarder le fichier
:q	quitter le fichier

Copier et déplacer un fichier (ou répertoire)

Pour copier un fichier (ou un répertoire), nous utilisons la commande suivante :

```
cp nom_du_fichier nom_de_la_copie
```

Nous pouvons aussi le copier dans un autre répertoire :

```
cp nom_du_fichier path/du/fichier/puis_son_nom
```

pour déplacer un fichier (ou répertoire) nous utilisons la commande suivante :

```
mv nom_du_fichier edroit/ou/va/se/trouver/le/fichier
```

Nous pouvons aussi renommer les fichier grâce à cette commande avec :

```
mv nom_du_fichier nouveau_nom
```

Documentation de la commande

Pour obtenir la documentation d'une commande, nous pouvons :

- demander à chatgpt qui feras les recherche à notre place et nous diras quoi faire exactement (ce qui est ~~toujours~~ rarement utilisé par les étudiants)
- chercher sur internet
- utiliser une commande

Cette comamnde est la commande suivante :

```
man la_commande
```

Réseau et IP

Pour afficher notre ip, notre adresse mac et l'interface correspondant à cette dernière, nous utilisons la commande :

```
ip a
```

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: enp3s0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc fq_codel state DOWN group default qlen 1000
    link/ether 08:8f:c3:19:1b:f0 brd ff:ff:ff:ff:ff:ff
3: wlan0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether [redacted] brd ff:ff:ff:ff:ff:ff
    inet [redacted] brd 192.168.0.255 scope global dynamic noprefixroute wlan0
        valid_lft 32550sec preferred_lft 32550sec
    inet6 [redacted] scope global dynamic noprefixroute
        valid_lft 86189sec preferred_lft 86189sec
    inet6 [redacted] scope link noprefixroute
        valid_lft forever preferred_lft forever
```

Exécution fichier en C

Pour se faire, dans un premier temps, nous allons installer gcc avec la commande suivante :

```
sudo apt install gcc
```

Une fois ceci fait et un fichier en C crée et fonctionnel comme :

```
#include <stdio.h>

int main()
{
    printf("Hello World !!");
    return 0;
}
```

(un programme d'une grande complexité)

Ce programme est nommé **helloworld.c** et pour le compiler, nous allons utiliser la commande suivante :

```
gcc -o helloworld.o helloworld.c
```

et allons obtenir un exécutable qui est **helloworld.o**. Pour l'exécuter, nous utiliserons la commande suivante :

```
./helloworld.o
```