

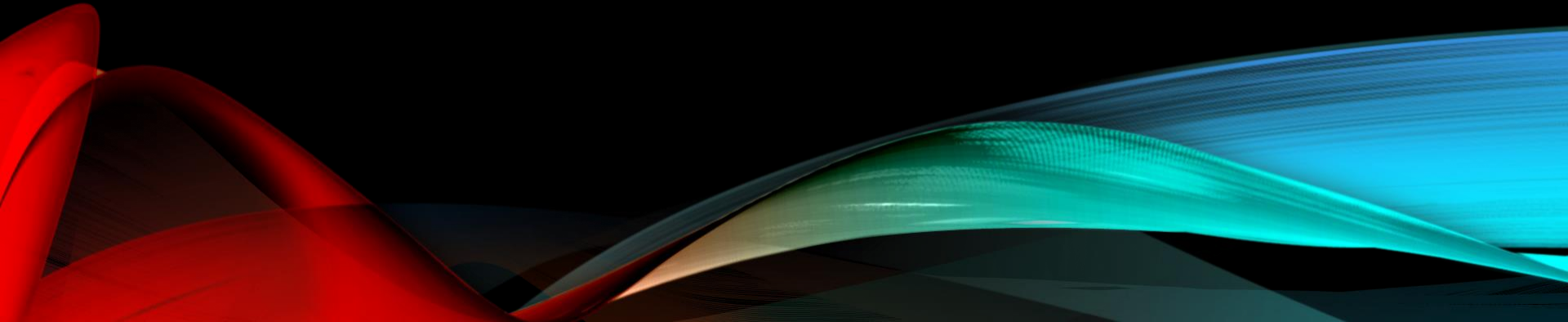
LINUX

Introduction à l'environnement « Linux »

SOMMAIRE

Qu'est ce que Linux ?
La ligne de commande
Quelques commandes
Le système de fichier
Les processus
Utilisateurs et groupes

QU'EST CE QUE LINUX ?



QU'EST CE QUE LINUX ?

Définition

Linux est le nom donné à tout système d'exploitation libre fonctionnant avec le noyau linux et / ou sur le système GNU.

D'où le nom : GNU / Linux

QU'EST CE QUE LINUX ?

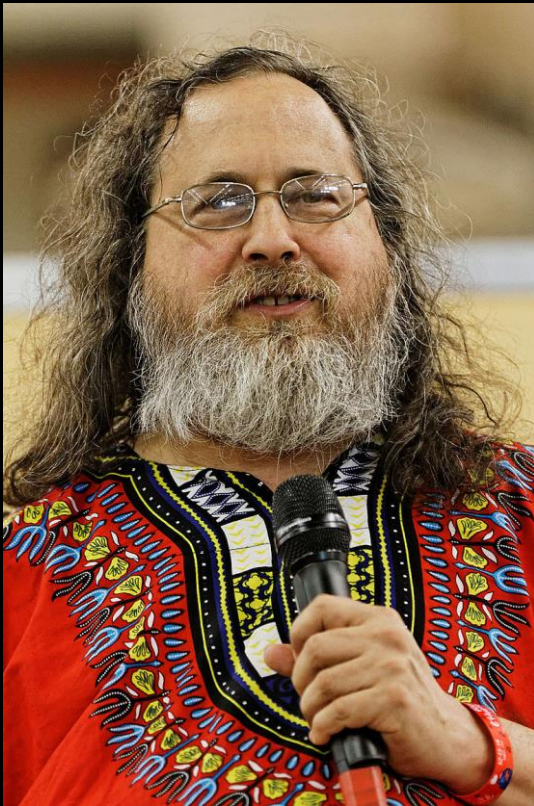
Où sont-ils ?

Quelques chiffres :

- ~2% Ordinateurs
- ~35% Serveurs (~65% en comptant tous les « Unix-like »)
- ~30% Systèmes embarqués
- ~99% Super ordinateurs et mainframes
- ~65,3% Smartphones (2015)

QU'EST CE QUE LINUX ?

Historique

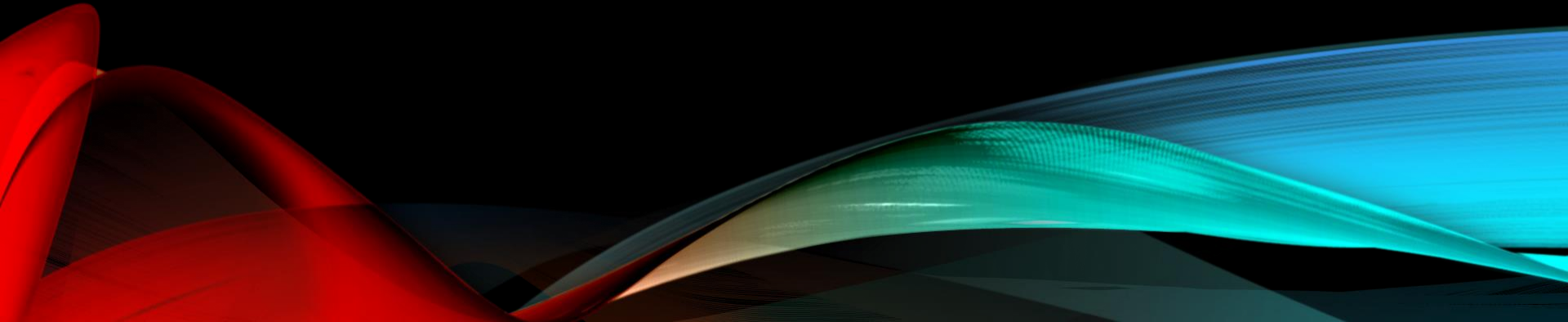


QU'EST CE QUE LINUX ?

Les distributions



LA LIGNE DE COMMANDE



LA LIGNE DE COMMANDE

Qu'est ce que c'est ?

[florentin@dubois ~] \$

[rootfs ~] #

florentin@dubois ~\$

LA LIGNE DE COMMANDE

Pourquoi ?

4 raisons :

- Moins consommateur en ressources
- Toujours présent
- Plus puissant
- Permet le pilotage à distance

LA LIGNE DE COMMANDE

La syntaxe

Syntaxe générale d'une commande :

commande [options] paramètres

Les options sont précédés par des « - » (le plus souvent)

Attention ! Bien respecter la casse

Remarque : l'interpréteur est nommé « shell »

LA LIGNE DE COMMANDE

Exemple

```
tar -xzvf fichier.tgz
```

```
tar -x -z -v -f fichier.tgz
```

Commande : tar

Options : x, z, v et f

Paramètre : fichier.tgz

LA LIGNE DE COMMANDE

Edition

Les touches « ← » et « → » : déplacement (ligne)

Les touches « ↓ » et « ↑ » : déplacement (historique)

Les touches « backspace » et « suppr » : suppression

La touche « tab » : complétion

LA LIGNE DE COMMANDE

Raccourcis

Copier : « ctrl + shift + c »

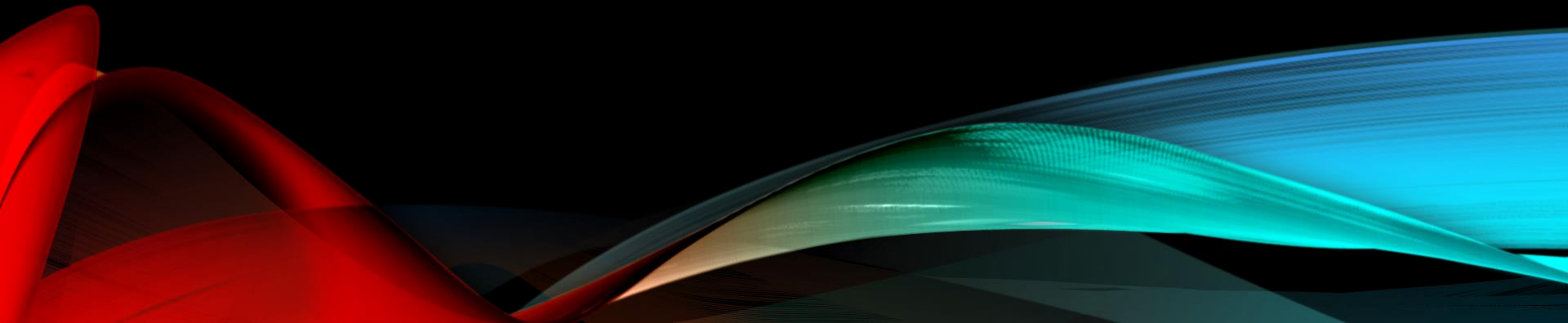
Coller : « ctrl + shift + v »

tty : « ctrl + alt + f[1-7] »

Stopper une commande : « ctrl + c »

Mettre en « pause » une commande : « ctrl + z »

QUELQUES COMMANDES



QUELQUES COMMANDES

Gestion des fichiers

pwd

mkdir

ls

cd

rmdir

touch

vi

cat

more

less

cp

mv

rm

QUELQUES COMMANDES

pwd

pwd : permet de connaître l'endroit où l'on se situe

Exemple :

```
florentin@dubois ~ $ pwd
```

Sortie :

```
/home/florentin
```

QUELQUES COMMANDES

mkdir

mkdir: créer un dossier

Exemple :

```
florentin@dubois ~ $ mkdir test
```

Option :

p : créer récursivement et / ou ne recréer pas

QUELQUES COMMANDES

cd

cd : se déplacer dans les dossiers

Exemple :

```
florentin@dubois ~ $ cd test
```

Remarque :

. : est le dossier courant

.. : est le dossier parent

~ : est le dossier utilisateur

QUELQUES COMMANDES

touch

touch : créer un fichier

Exemple :

```
florentin@dubois ~ $ touch fichier
```

QUELQUES COMMANDES

ls

ls: permet de lister les fichiers et dossiers

Exemple :

```
florentin@dubois ~ $ ls
```

Options :

a : afficher les fichiers et dossiers cachés

l : afficher les détails

QUELQUES COMMANDES

vi

vi : éditer un fichier

Exemple :

```
florentin@dubois ~ $ vi fichier
```


QUELQUES COMMANDES

cat

cat : permet de visualiser le contenu d'un fichier

Exemple :

```
florentin@dubois ~ $ cat fichier
```

Sortie :

[contenu]

QUELQUES COMMANDES

more

more : permet de visualiser le contenu d'un fichier

Exemple :

```
florentin@dubois ~ $ more fichier
```

Sortie :

[contenu]

QUELQUES COMMANDES

less

less : permet de visualiser le contenu d'un fichier

Exemple :

```
florentin@dubois ~ $ less fichier
```

Sortie :

```
[contenu]
```

QUELQUES COMMANDES

cp

cp : copier un fichier

Exemple :

```
florentin@dubois ~ $ cp fichier destination
```

QUELQUES COMMANDES

mv

mv : déplacer un fichier

Exemple :

```
florentin@dubois ~ $ mv fichier destination
```

QUELQUES COMMANDES

rm

rm : supprimer un fichier

Exemple :

```
florentin@dubois ~ $ rm fichier
```

QUELQUES COMMANDES

rm dir

rm dir : supprimer un dossier

Exemple :

```
florentin@dubois ~ $ rm dir dossier
```

Remarque :

```
florentin@dubois ~ $ rm -r dossier
```

Permet de supprimer un dossier avec son contenu

QUELQUES COMMANDES

La commande sudo

La commande sudo permet de lancer une commande en étant un autre utilisateur.

Syntaxe générale :

```
sudo [-u root] commande [options] paramètres
```

```
sudo -u user commande [options] paramètres
```

Remarque : il faut être listé dans les sudoers.

QUELQUES COMMANDES

Exemple

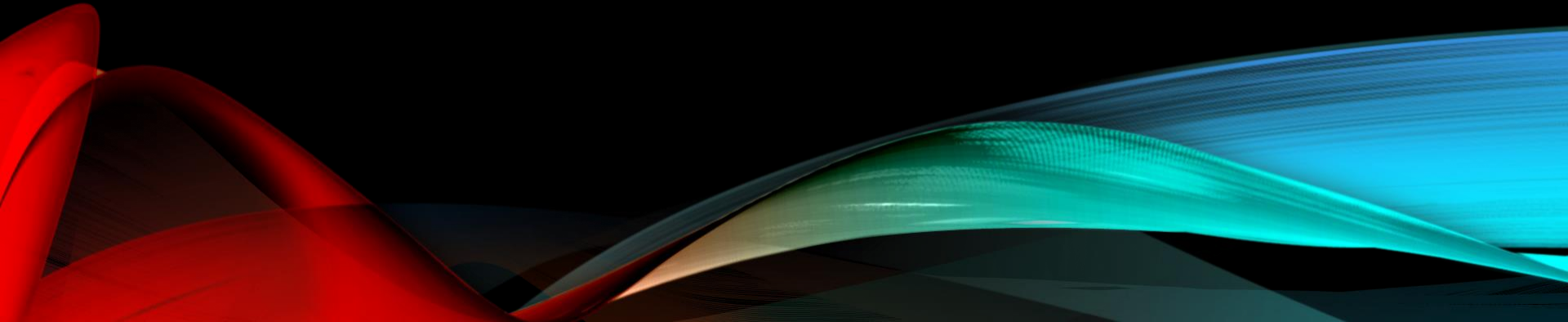
```
sudo rm -rf /*
```

Commande : rm

Options : r et f

Paramètre : /*

LE SYSTÈME DE FICHER



LE SYSTÈME DE FICHER

Architecture

```
florentin@dubois → ~ tree -L 1 /  
/  
├── bin -> usr/bin  
├── boot  
├── dev  
├── etc  
├── home  
├── lib -> usr/lib  
├── lib64 -> usr/lib  
├── lost+found  
├── media  
├── mnt  
├── opt  
├── proc  
├── root  
├── run  
├── sbin -> usr/bin  
├── srv  
├── sys  
├── tmp  
├── usr  
└── var
```

LE SYSTÈME DE FICHER

Répertoire utilisateur

```
florentin@dubois → ~ tree -L 2 /home
```

```
/home
├── florentin
│   ├── Bureau
│   ├── Documents
│   ├── Images
│   ├── Minecraft
│   ├── Modèles
│   ├── Musique
│   ├── Projects
│   ├── Public
│   ├── Qemu
│   ├── Server
│   ├── Téléchargements
│   ├── Vidéos
│   └── Web
└── lost+found [error opening dir]
```

LE SYSTÈME DE FICHER

Les fichiers

Point à connaître :

Un dossier « est » un fichier

Ne contient pas forcément d'extension

Accepte tout les caractères sauf : / et \0

Cependant, on évite les caractères :

* ? [] < > | () { } & ! \ ; % # " ' et l'espace

LE SYSTÈME DE FICHER

Les fichiers cachés

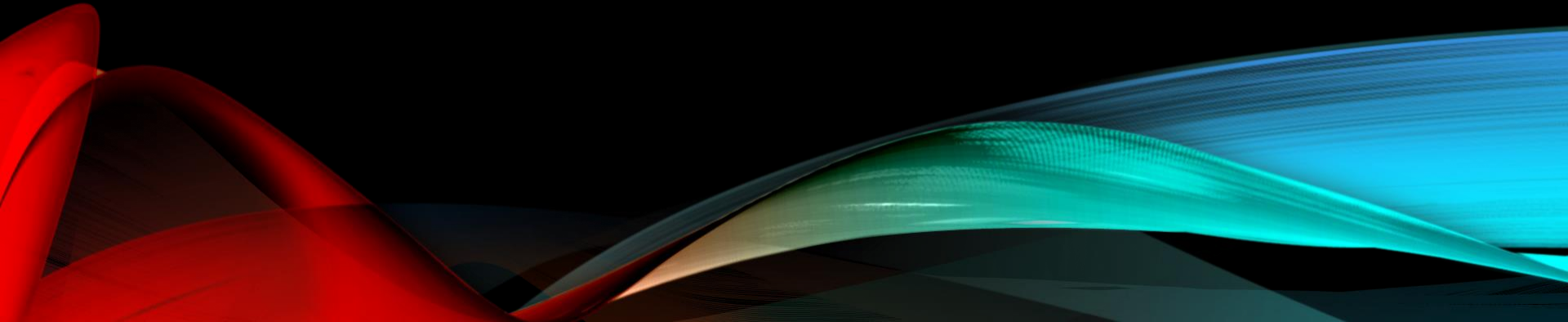
Un fichier caché sous linux :

Respecte les règles ci-avant

Commence par le caractère point

Ou est nommer dans le fichier « .hidden » du dossier

GESTION DES PROCESSUS



GESTION DES PROCESSUS

Qu'est-ce qu'un processus ?

Un processus est un programme en cours d'exécution

il possède un identifiant PID et celui de son parent PPID

Le premier processus se nomme init et est lancé par root

GESTION DES PROCESSUS

Lister les processus

ps : lister les processus

Exemple :

```
florentin@dubois ~$ ps -u florentin
```

Options :

e : tout les processus (every)

f : toute les informations (full)

u : processus lancé par (user)

GESTION DES PROCESSUS

Lister les processus

htop : lister les processus

Exemple :

```
florentin@dubois ~$ htop
```

GESTION DES PROCESSUS

Lancer un programme en fond

Ajouter le caractère & à la fin de la commande

Mettre en « pause » puis utiliser la commande bg

Utiliser la commande fg pour mettre au premier plan

GESTION DES PROCESSUS

La commande kill

kill : permet d'envoyer un signal

Exemple :

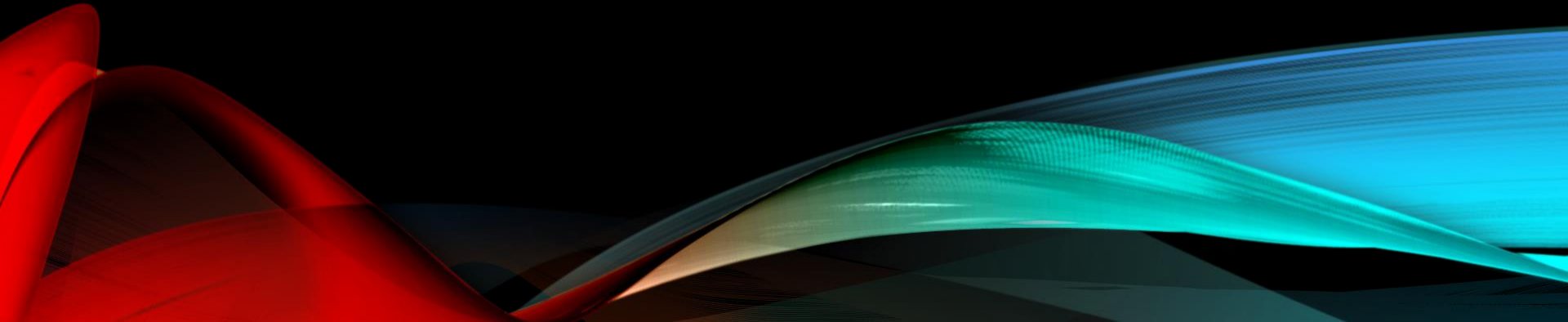
```
florentin@dubois ~$ kill -9 1
```

```
florentin@dubois ~$ kill -KILL 1
```

Option :

l : connaître les différents signaux

UTILISATEURS ET GROUPES



UTILISATEURS ET GROUPES

A savoir

Toute entité interagissant avec le système doit être authentifiée

Un utilisateur possède un nom unique et un numéro unique

Sur tout les systèmes Linux/Unix, il existe un super utilisateur : root

Un utilisateur appartient à au moins un groupe

UTILISATEURS ET GROUPES

Créer un utilisateur

useradd : créer un utilisateur

Exemple :

```
$ useradd -m -g user -G http,ftp -s /usr/bin/zsh <utilisateur>
```

Options :

- m : créer le repertoire
- g : groupe primaire
- G : groupe secondaire
- s : shell par défaut

UTILISATEURS ET GROUPES

Changer ou mettre un mot de passe

passwd : permet de changer de mot de passe

Exemple :

```
florentin@dubois ~$ passwd <utilisateur>
```

UTILISATEURS ET GROUPES

Rajouter un utilisateur dans un groupe

gpsswd : permet de rajouter un utilisateur a un groupe

Exemple :

```
florentin@dubois ~$ gpsswd --add <utilisateur> ssh
```

UTILISATEURS ET GROUPES

Supprimer un utilisateur

userdel : supprime un utilisateur

Exemple :

```
florentin@dubois $ userdel -r <utilisateur>
```

Options :

r : supprime le répertoire de l'utilisateur

UTILISATEURS ET GROUPES

Comment connaître son(ses) groupe(s) ?

groups : commande qui permet de connaître son(ses)
groupe(s)

Exemple :

```
florentin@dubois ~$ groups
```

UTILISATEURS ET GROUPES

Comment connaître son(ses) groupe(s) ?

id : commande qui permet de connaître son(ses)
groupe(s)

Exemple :

```
florentin@dubois ~$ id
```

DES QUESTION ?

