

Le système d'exploitation

Un ordinateur est formé d'une partie matérielle et d'une partie logicielle.

Les logiciels qui sont classés en deux catégories :

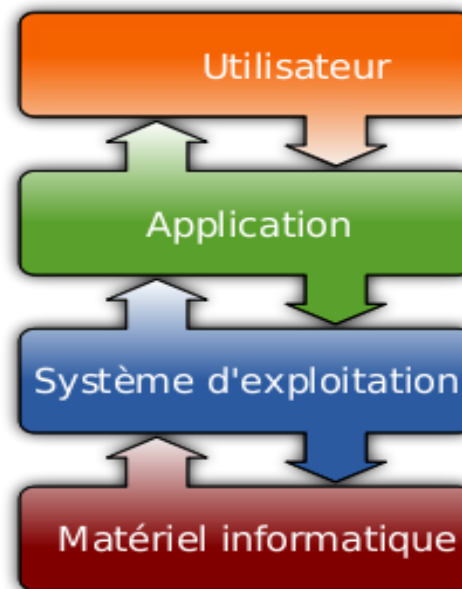
- les programmes d'application des utilisateurs
- les programmes systèmes qui permettent le fonctionnement de l'ordinateur.

(Parmi ceux-ci, le système d'exploitation (SE))

Le système d'exploitation

Le système d'exploitation (SE) est le logiciel qui prend en charge les fonctionnalités élémentaires du matériel et qui propose une plateforme plus efficace en vue de l'exécution des programmes.

Il gère les ressources matérielles, offre des services pour accéder à ces ressources et crée des éléments abstraits de niveau supérieur (des fichiers, des répertoires et des processus).



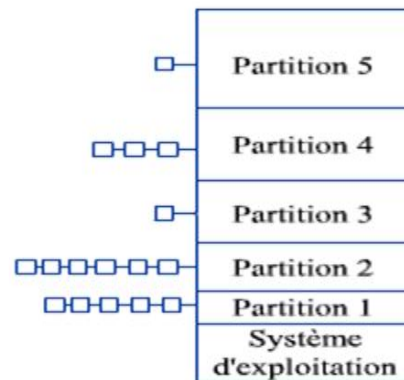
Rôle d'un OS

1. Gestion de la mémoire :

La gestion de la mémoire est le partage de la mémoire entre l'**OS** et les diverses **applications**, ses principaux rôles sont:

- Suivre les parties de la mémoire qui sont utilisées ou non utilisées.
- Allouer/libérer espace mémoire aux processus.
- Contrôler le swapping entre la mémoire principale et le disque.

En cas d'insuffisance de mémoire physique (RAM), le système d'exploitation peut créer une zone mémoire sur le disque dur, appelée mémoire virtuelle.

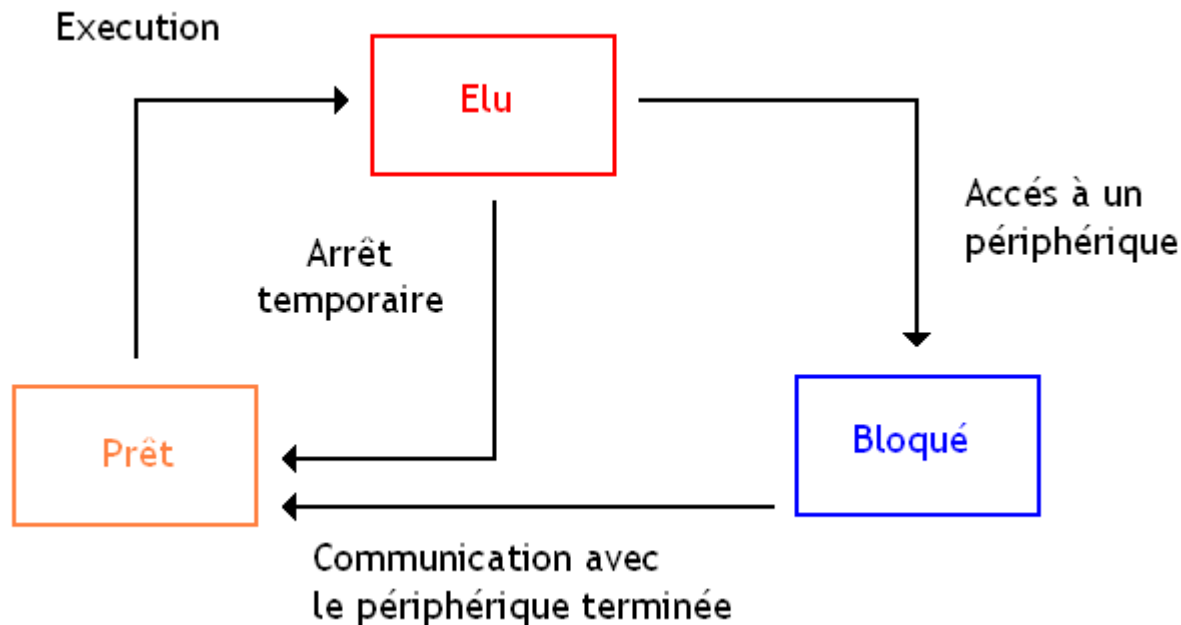


Rôle d'un OS

2. Gestion du processus :

Un processus est un programme en cours d'exécution.

Le SE gère l'allocation du processeur entre les différentes applications grâce à un algorithme d'ordonnancement. IL organise les tâches et les fait commuter tout à tour pour donner l'impression qu'elles s'exécutent toutes simultanément



Rôle d'un OS

3. Gestion du Réseau:

Le rôle du système exploitation réseau est multiple :

☐ L'Accès des utilisateurs au réseau:

- Créer et gérer les comptes des utilisateurs qui ont accès au réseau.
- Définir les permissions des utilisateurs et des groupes (lire, enregistrer, supprimer, exécuter, ...)

☐ Le partage des ressources:

- Des documents
- Des périphériques: imprimantes, Coordonner les accès simultanés à la même ressource

☐ La surveillance du réseau:

- Les performances/La sécurité

Rôle d'un OS

4. Gestion du stockage :

Ils permettent de recevoir des informations, d'en envoyer et de stocker des informations.

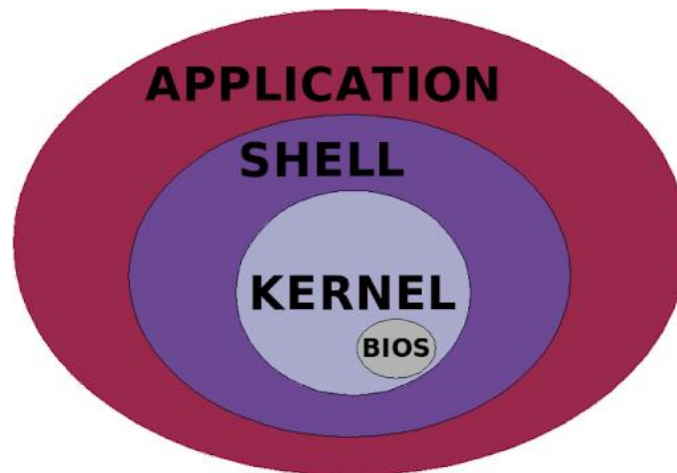
- La manipulation de fichiers (sur une unité de stockage : partition, disque, CD,..)
- Suivre l'état d'utilisation (libre ou réservée) de tout le matériel du système informatique

5. Gestion de l'interface:

Gérer l'accès des programmes aux ressources matérielles par l'intermédiaire des pilotes

Les composants d'un système d'exploitation

- Le **noyau** (*kernel*) représente les fonctions fondamentales du système d'exploitation telles que la gestion de la mémoire, des processus, des fichiers, des entrées-sorties principales, et des fonctionnalités de communication.
- L'**interpréteur de commande** (*shell*) permet la communication avec le système d'exploitation par l'intermédiaire d'un langage de commandes.



Les composants d'un système d'exploitation

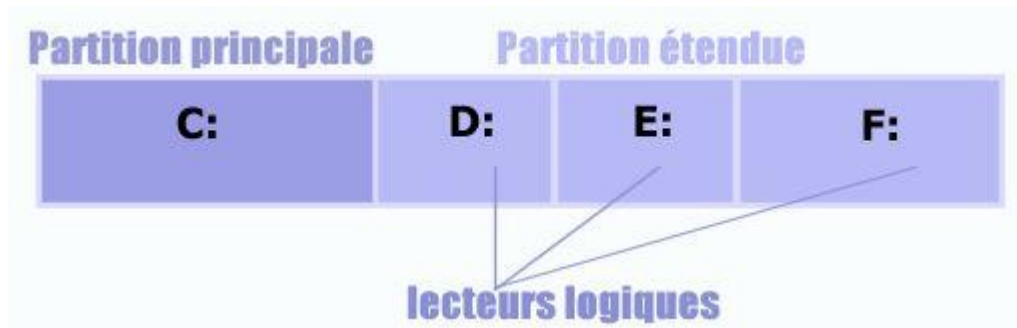
- Le **système de fichiers** (*file system*) permet d'organiser les millions de bits qui contiennent les support de stockage a fin de pouvoir localiser les informations (FAT32/NTFS/Ext3..)

	NTFS	FAT32
Taille maximal de fichier	16TiB	4Go
Nombre maximal de fichiers	4 294 967 295	Supérieur à 250 millions
Taille maximal des noms de fichiers	255 caractères	255 caractères
Taille maximale de volume	256 TiB	2To (16 en théorie)

Les composants d'un système d'exploitation

➤ Partitionnement:

- Partitions principales: peuvent contenir n'importe quel système d'exploitation
 - Sont formatées logiquement pour utiliser un système de fichiers compatible avec le système d'exploitation sur lequel il est installé.
- Partitions étendues: limitées de quatre partitions
- Partitions logiques: existent à l'intérieur d'une partition étendue



Types du SE



Les différents types de licences

Libre:- La liberté d'utiliser le logiciel

- La liberté d'accéder au code source
- La liberté de distribuer des copies du logiciel

Payante:

- Généralement utilisé par les éditeurs commerciaux
- interdit d'accéder au code source , le modifier et le distribuer

Open source: -Accéder au code source et le modifier

- Open Source ne signifie pas non plus “gratuit “



Virtualisation:

La virtualisation est une technologie permettant de créer et d'exécuter une ou plusieurs représentations virtuelles d'un ordinateur sur une même machine physique.

avantage: faire fonctionner plusieurs systèmes d'exploitation sur le même environnement physique/ un environnement de test...

Linux et Unix:

	Linux	Unix
code	Open source	N'est pas disponible
interface	Graphique & ligne de commande	Ligne de commande
portabilité	Sur n'importe disque dur	N'est pas portable
prix	gratuit	payant
installation	très flexible	Dépend de matériel
matériel	économique au niveau matériel	Serveur et pc haute gamme

Linux :

Linux est un système exploitation créé à partir une collection de logiciels utilisant le noyau Linux/GNU. La plupart des les logiciels disponibles dans linux sont libres et open source.

Linux ne contient aucun code provenant de UNIX, il en est juste inspiré, et complètement réécrit.



Les commandes de base du linux



pwd: dans quel répertoire vous êtes à ce moment

ls: lister le contenu du répertoire courant

ls -l: afficher plus d'information

ls -a: afficher les dossiers cachés

ls -lh: afficher plus d'information avec la taille en **Ko**

Déplacement entre les dossiers:

cd, cd . , cd .. , cd ~ , cd /

~: tilde

cd: déplacer vers un dossier

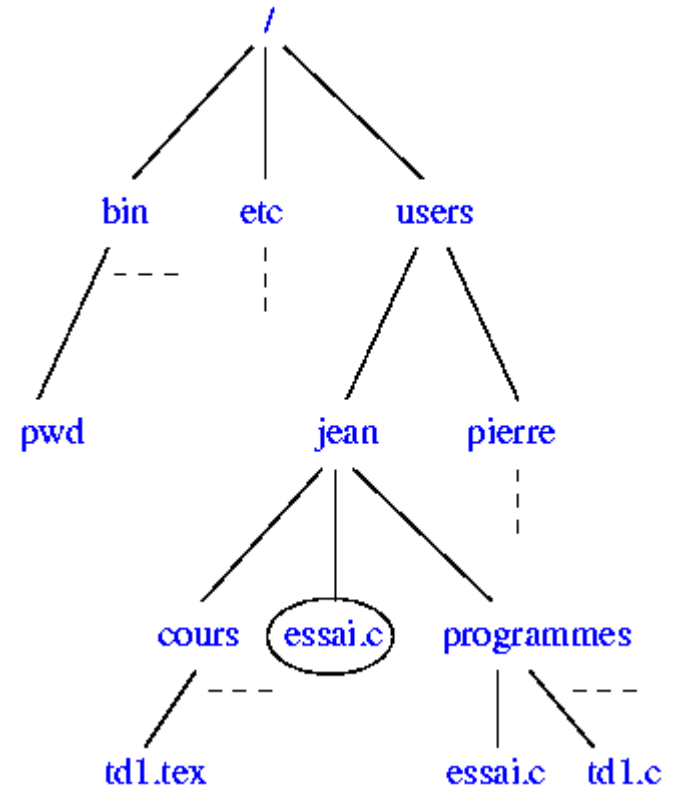
exemple:

cd *nom_du_dossier*

cd *dossier1/dossier2*

cd */dossier1/dossier2*

cd ..: aller au dossier précédent



Création du dossier:

mkdir: créer un dossier(répertoire)

exemple: **mkdir** *nom_du_dossier*

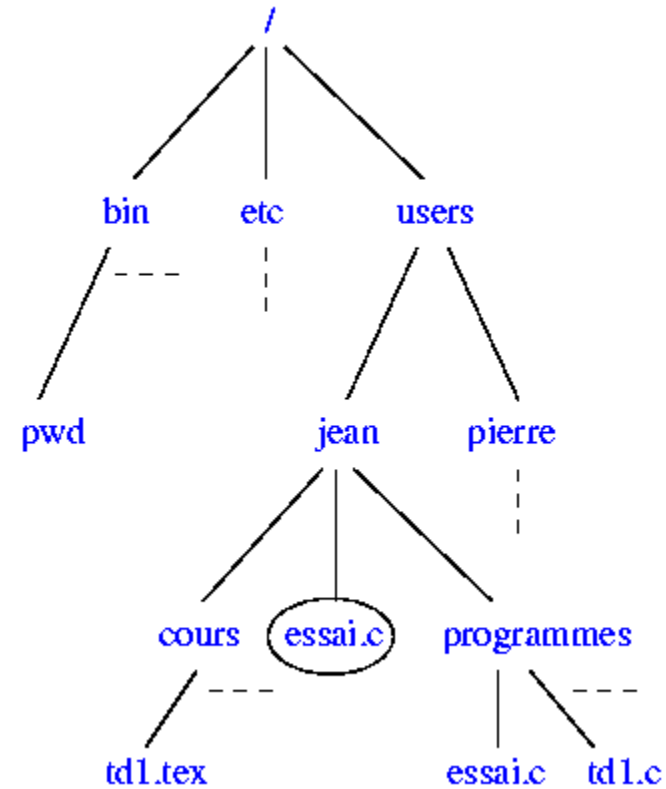
rmdir: supprimer un dossier

exemple: **rmdir** *nom_du_dossier*

touch: créer un fichier

exemple: **touch** *fichier*

touch *fichier1* *fichier2*



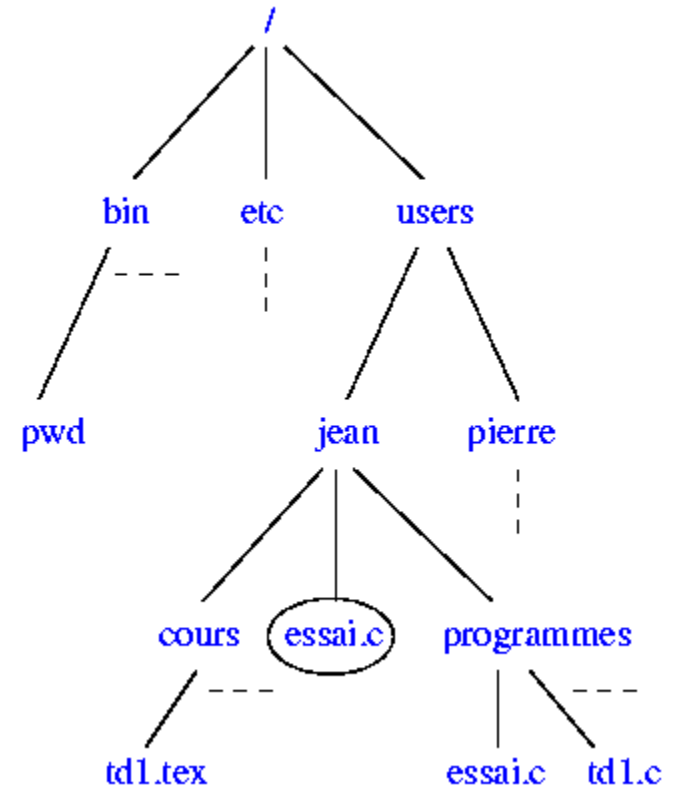
Création du dossier:

mkdir: créer un dossier(répertoire)

exemple:

mkdir *nom_dossier*

mkdir *nom_dossier1 nom_dossier2*



rmdir: supprimer un dossier

exemple: **rmdir** *nom_du_dossier*

rmdir *nom_dossier1 nom_dossier2*

mv: déplacer un fichier ou répertoire

Exemple:

mv *source* *destination*

mv : aussi **renommer** un fichier (non répertoire)

Exemple:

mv *ancien_nom* *nouveau_nom*

rm : supprimer un fichier

Exemple:

rm *nom_fichier*

cp: copier un fichier ou répertoire

Exemple:

cp *nom_fichier* *destination*

cp **-R** *nom_dossier* *destination*

mv: déplacer un fichier ou répertoire

Exemple:

mv *source* *destination*

mv : aussi renommer un fichier (non répertoire)

Exemple:

mv *ancien_nom* *nouveau_nom*

rm : supprimer un fichier

Exemple:

rm *nom_fichier*

rm -r: supprimer un dossier (et son contenu)

rm -r: supprimer un dossier (et son contenu)

Les options de la commande **rm** :

- i** : demander la confirmation
- f** : forcer la suppression, quoi qu'il arrive
- r** : supprimer un dossier et son contenu

confirmation:

- o : signifie « Oui ».
- n : signifie « Non ».

Exemples: `rm -i nom_fichier`
`rm -f nom_fichier`
`rm -r nom_fichier`

cat : afficher le contenu d'un fichier

exemple: **cat** *nom_du_fichier*

gedit (ou nano): éditer le contenu d'un fichier

head : afficher les premières lignes du fichier

exemple: **head** -3 *nom_du_fichier*

(afficher les 3 premières lignes)

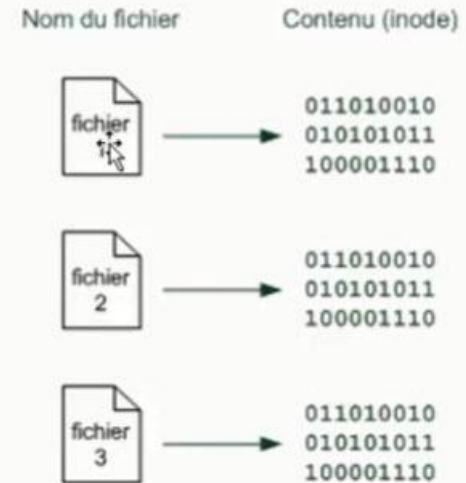
tail: afficher les dernière lignes

exemple: **tail** -2 *nom_du_fichier*

(afficher les 2 dernières lignes)

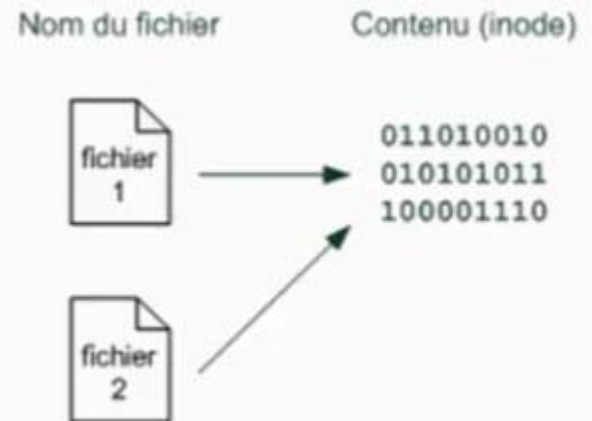
Inode:

Chaque contenu de fichier se voit attribuer un numéro d'identification appelé inode. chaque nom du fichier est donc associé à un inode (figure suivante).



Les liens physiques:

Permet d'avoir plusieurs noms de fichier qui partagent exactement le même contenu, c'est-à-dire le même inode



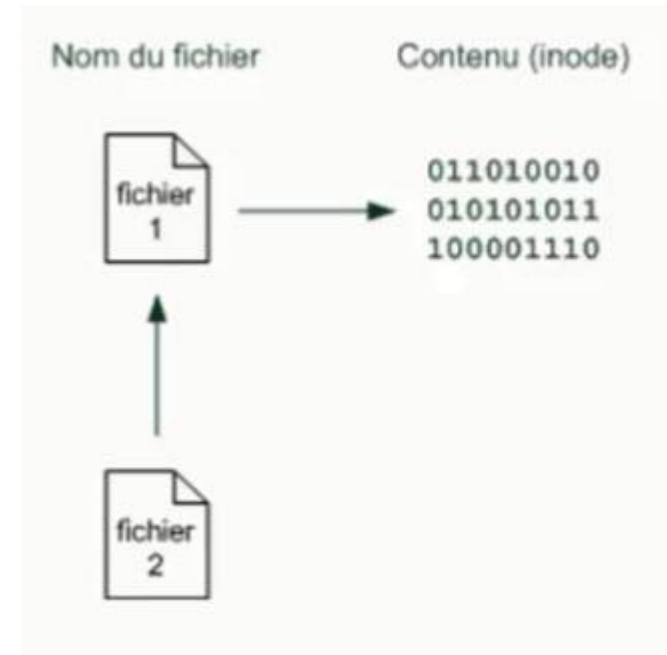
Créer un lien physique:

ln : créer un lien physique

Exemple: **ln** *nom_fichier1* *nom_fichier2*

Créer des liens symboliques:

Les liens symboliques rassemblent plus aux « raccourcis ». Le principe du lien symbolique est que l'on crée un lien vers un autre nom du fichier. Cette fois, pointe vers le nom du fichier et non vers l'inode directement



Créer un lien symbolique:

ln : créer un lien physique

Exemple: **ln -s** nom_fichier1 nom_fichier2

2. Gestion des utilisateurs/groupes

- Un utilisateur est un fichier, tout ce concerne ses identifications, informations, autorisations .ect sont également des fichiers (/etc/passwd)

- Chaque utilisateur a un répertoire
- Chaque utilisateur est identifier par son:

`nom:mot de passe:uid:gid:info:répertoire:shell`

- Linux permet une administration des groupes(création, ajouter un utilisateur au groupe, suppression un utilisateur d'un groupe, suppression de groupe) (/etc/group)

2. Gestion d'utilisateur

Un utilisateur est enregistré dans le passwd

`cat /etc/passwd`

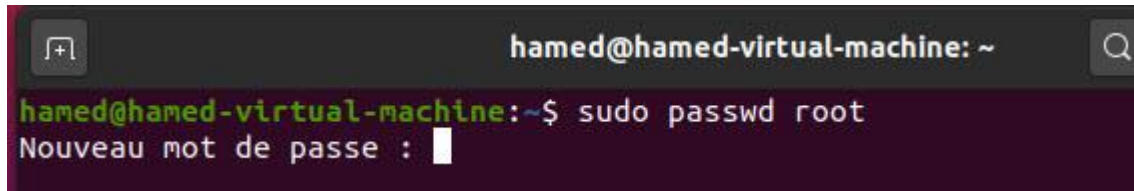
```
hamed@hamed-virtual-machine:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
```

```
colord:x:121:126:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin
geoclue:x:122:127:/:/var/lib/geoclue:/usr/sbin/nologin
pulse:x:123:128:PulseAudio daemon,,,:/var/run/pulse:/usr/sbin/nologin
gnome-initial-setup:x:124:65534:/:/run/gnome-initial-setup:/bin/false
ndm:x:125:130:Gnome Display Manager:/var/lib/ndm3:/bin/false
hamed:x:1000:1000:hamed,,,:/home/hamed:/bin/bash
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
hamed@hamed-virtual-machine:~$
```

2. Gestion des utilisateurs/ groupes

Pour passer en mode **root**, il faut ajouter **sudo** et définir un mot de passe car seulement le super utilisateur (root) est autorisé à ajouter un utilisateur ou un groupe au système.

* définir un mot de passe: **taper:** `sudo passwd root`



```
hamed@hamed-virtual-machine: ~  
hamed@hamed-virtual-machine:~$ sudo passwd root  
Nouveau mot de passe : 
```

`sudo -i` (ou bien `'su'`): Pour accéder au root

`exit`: sortir du mode root

`su` : changer l'utilisateur

exemple: `su name_user`

`whoami`: affiche le nom d'utilisateur courant

sudo adduser: ajouter un utilisateur

exemple: **sudo adduser** *name_user*

sudo addgroup: ajouter un groupe

exemple: **sudo addgroup** *name_group*

sudo userdel : supprimer un utilisateur

exemple: **sudo userdel** *name_user*

sudo groupdel :supprimer un groupe

exemple: **sudo groupdel** *name_group*

passwd: changer le mot du passe

sudo chown : l'utilisateur soit le propriétaire d'un fichier

exemple: **sudo chown** *name_user* *name_file*

sudo chgrp : le groupe soit le propriétaire d'un fichier

exemple: **sudo chgrp** *name_group* *name_file*

Chmod: pour modifier les droits

r: lire

w: écrire

x: exécuter



exemple

r	w	x	total
4	2	1	7

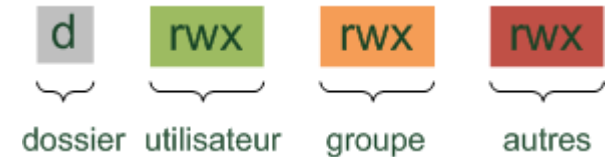
r	-	x	total
4	0	1	5

r	w	-	total
4	2	0	6

pour modifier les droits à un fichier:

Sudo chmod 764 *name_file*

umask: pour modifier les droits



Par défaut:

umask: 022

Fichier: (666) → *rw- rw- r--*

Répertoire : (777) → *rwx r-x r-x*

666-022=644 (rw- rw- r--)

777-022= 755 (rwx r-x r-x)

3. Recherche de fichiers

- **Recherche du chemin:**

`cd ~` : revenir à la racine de Linux

`updatedb`: La commande `updatedb` sous linux permet de mettre à jour la base de données utilisée par la commande `locate`.

`locate`: Chercher le chemin d'accès

Rq: il faut installer le paquet `mlocate`:

`sudo apt install mlocate`

3. Recherche de fichiers

- Recherche du chemin:

Exemple 1:

Sudo updatedb

après taper: locate nom_fichier

Exemple 2: recherche d'un chemin d'accès à tous les fichiers de même extension

locate *extension

locate caractère* : recherche les chemins d'accès à tous les fichiers qui commence par ce caractère

(extensions: .mp3; .mp4; .jpeg; .avi; .flv; .txt; .gif; ...)

3. Recherche de fichiers

- Recherche des fichiers et répertoires:

find: rechercher

Exemple: recherche par taille

```
find {directory} -size +1M
```

Exemple: recherche par type

```
find {directory} -type d
```

d=répertoire ou dossier;

c=caractère;

f=fichier normal;

3. Recherche de fichiers

- **Recherche des fichiers et répertoires:**

Option:

- name recherche sur le nom du fichier,
- perm recherche sur les droits d'accès du fichier,
- links recherche sur le nombre de liens du fichier,
- user recherche sur le propriétaire du fichier,
- group recherche sur le groupe auquel appartient le fichier,
- type recherche sur le type (d=répertoire, c=caractère, f=fichier normal),
- size recherche sur la taille du fichier en nombre de blocs,
- atime recherche par date de dernier accès en lecture du fichier,
- mtime recherche par date de dernière modification du fichier,
- ctime recherche par date de création du fichier

4. Extraire, Trier et Filtrer des données

- **Extraire:**

grep: La commande grep permet de rechercher une chaîne de caractères dans un fichier. Les options sont les suivantes

- v : affiche les lignes ne contenant pas la chaîne
- i : ignore la casse
- c : compte le nombre de lignes contenant la chaîne
- n : chaque ligne contenant la chaîne est numérotée
- x : ligne correspondant exactement à la chaîne
- l : affiche le nom des fichiers qui contiennent la chaîne

4. Extraire, Trier et Filtrer des données

- **Extraire:**

Exemple:

```
grep mot nom_fichier
```

```
grep {option} mot nom_fichier
```

4. Extraire, Trier et Filtrer des données

- **Trier:**

Sort: trier par ordre alphabétique

Options:

-o : écrire le résultat dans un fichier

-r : trier en ordre inverse

-R : trier aléatoirement

-n : trier des nombres

Exemple:

```
sort name_file
```

```
sort {option} name_file
```

4. Extraire, Trier et Filtrer des données

- **Filtrer:**

uniq: supprimer les doublons dans un fichier

Options:

-u : Affiche seulement les lignes n'apparaissant qu'une seule fois.

-d : Affiche seulement les lignes répétées.

-c : le nombre d'occurrences .

Exemple:

uniq name_file new_file

uniq {option} name_file new_file

4. Extraire, Trier et Filtrer des données

- **Filtrer:**

wc: Compter le nombre de mots

Option:

-l : nb de lignes

-w : nb de mots

-c : nb d'octets

-m : nb de caractères

Exemple:

wc {option} name_file

4. Extraire, Trier et Filtrer des données

- **Filtrer:**

cut: Couper une colonne

Exemple:

```
cut -d , -f 2 name_file
```

5. Redirection du flux

Au lieu d'afficher le résultat d'une commande dans une console, il est possible de l'enregistrer dans un fichier. Il suffit d'ajouter le symbole `>` suivi du nom du fichier à la fin de la commande.

exemple: `ls > liste_fichiers.txt`

enregistrer la liste des fichiers dans un fichier et écraser les anciens résultats s'il existe déjà.

Le symbole `>>` enregistrer les résultats et garder les anciens s'il existe déjà.

5. Redirection du flux

- Le symbole `>>` enregistrer les résultats et garder les anciens s'il existe déjà.
- Les symboles `2>` et `2>>` permettent de rediriger seulement les erreurs dans un fichier. Quant à `2>&1` il redirige les erreurs dans le même fichier que les messages normaux.

5. Redirection du flux

- `<` permet de lire des données depuis un fichier et de les envoyer à une commande, tandis que `<<` lit les données depuis le clavier.

`tee` : remplace `>`

`echo`: afficher une chaîne de caractère

exemple:

`echo` 'un petit test'

`echo` 'un petit test' `>` test.txt

6. Chainer les commandes

- Le symbole `|` combine des commandes : les données de la commande à sa gauche sont envoyées à la commande à sa droite. Ainsi, `ls | sort` récupère la liste des fichiers avec et l'envoie à `sort` pour qu'il la trie .