

M04

Fonctionnement des systèmes d'exploitation

Partie Linux

M.MANGAD

PLAN

- 1. Principe de fonctionnement du système Linux**
- 2. Identification des différentes distributions**
- 3. Arborescence du système de fichiers**
- 4. Gestion de base du système de fichiers**
- 5. Commandes de base**
- 6. Droits d'accès et utilisateurs**
- 7. Gestion des paquets et mises à jour**

I. Principe de fonctionnement du système Linux:

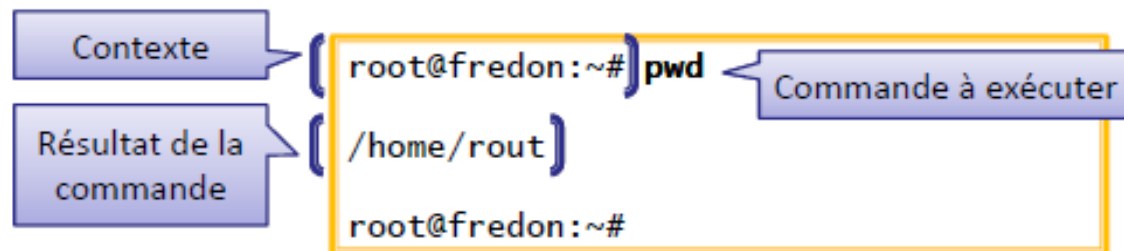
1. Introduction:

- Linux est un système d'exploitation **gratuit et libre** (Open Source), qui peut être utilisé à la place des systèmes d'exploitation commercialisés, tels que Windows, de Microsoft.
- Le système peut être utilisé sur des serveurs ([LAN](#) ou serveurs web), sur des PC ou encore sur des Smartphones.
- Une installation correcte de Linux vous permettra donc de réaliser les opérations les plus classiques comme :
 - Effectuer un travail bureautique,
 - naviguer sur Internet,
 - réaliser la capture et le retraitement d'images,
 - ou encore programmer...

1. Principe de fonctionnement du système Linux:

2. Shell Linux:

- Le Shell est un interpréteur de commandes
 - Permet à l'utilisateur d'interagir avec le système
 - Il lit et exécute les commandes de l'utilisateur
- C'est aussi un véritable langage de programmation
- Il sera possible d'écrire des scripts exécutant des commandes répétitives
- Il en existe plusieurs shell:
 - Le shell « bash » est le plus courant
 - Mais aussi les shells « csh », « ksh », « tcsh »



I. Principe de fonctionnement du système Linux:

3. Prise en charge des disques sous Linux:

- Le pointeur spécial **/dev** permet l'accès aux disques

- Format des pointeurs sur disque :

- **Types de bus**

- hd : Périphériques IDE
 - sc : Périphériques SCSI
 - sd : Périphériques SATA

Diagramme illustrant la structure d'un pointeur de périphérique Linux : **/dev/XYZ**

- Lettre de périphérique** : Indique le type de périphérique (X).
- Type de bus** : Indique le type de bus (Y).
- N° de partition** : Indique le numéro de la partition (Z).

- **Exemples**

- /dev/hda1 :
 - Partition 1 sur le 1^{er} disque IDE
 - /dev/sdb2 :
 - Partition 2 sur le 2^{ème} disque Sata
 - /dev/sca2 :
 - Partition 2 sur le 1^{er} disque SCSI

II. Distributions Linux:

1. Définition:

- Une distribution est constituée :
 - **Du noyau Linux** : se trouve bien évidemment à la base de toutes les autres couches, puisqu'il gère quasiment tout le matériel (mémoire, disques, systèmes de fichiers, réseau, clavier, etc.).
 - De « packages » contenant des logiciels additionnels pour l'interface graphique

II. Distributions Linux:

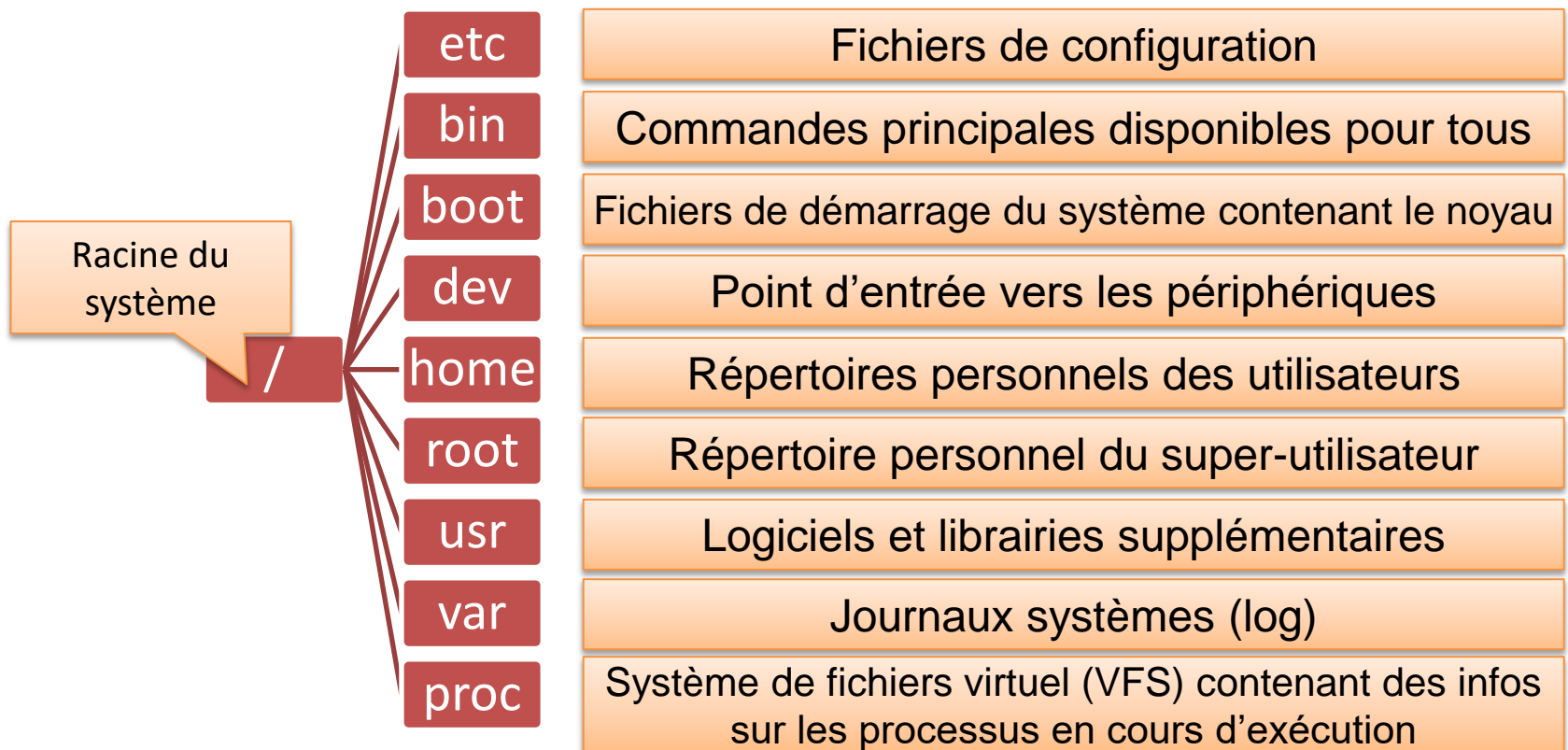
2. Principales distributions sous Linux

Il existe plusieurs distributions sous Linux:

- IPCOP : Sécurité des réseaux
- Backtrack : Piratage
- Kali Linux : Cyber sécurité
- Red Hat ou Cent OS : Administration des réseaux
- Debian
- Mandriva
- Fedora
- ubuntu...

III. Arborescence du système de fichiers Linux:

- Une arborescence est l'Organisation logique des fichiers sur un ou plusieurs systèmes de fichiers
- Il s'agit d'une structure de données hiérarchique de type arbre



IV. Gestion de base du système de fichier Linux

1. Les systèmes de fichiers sous Linux:

Sous Linux

Ext2

Ext3

Ext4

jfs, xfs

ext est le système de fichiers le plus courant pour Linux

Linux peut lire la plupart des formats. Notamment Ceux de Windows : NTFS, FAT, FAT32

2. Le format SWAP

Le format « swap » est utilisé comme « mémoire virtuelle » dans le cas où la mémoire vive est saturée → Améliorer les performance

La taille du « swap » est fixée au double de la mémoire vive

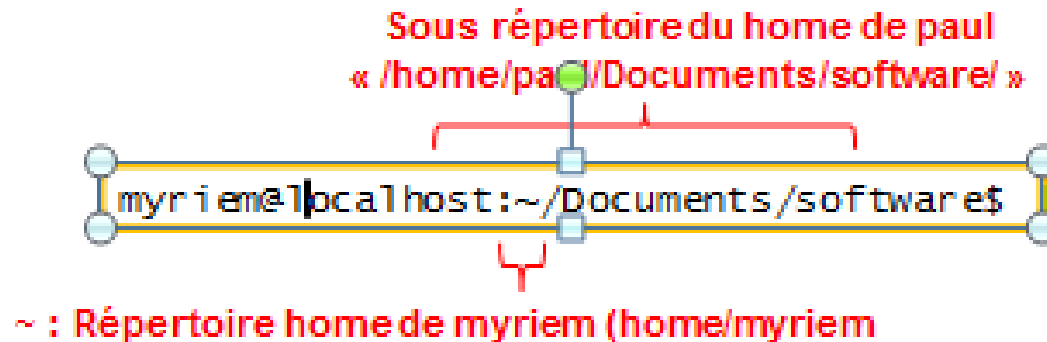
IV. Gestion de base du système de fichier Linux

3. Se repérer dans le système de fichiers:

La ligne de commande donne des informations :



Autre exemple



IV. Gestion de base du système de fichier Linux

3. Se repérer dans le système de fichiers:

La commande « **pwd** » permet de savoir quel est le répertoire courant

La commande « **ls** » permet de lister les fichiers contenus dans un répertoire

La commande « **cd** » permet de changer de répertoire

Les symboles suivants ont une signification particulière :

« . » : Le point désigne le répertoire courant

Exemple : Exécuter un script depuis le répertoire courant

```
root@fredon:/home/paul# ./script.sh
```

« .. » : Les 2 points désignent le répertoire parent

Exemple : Se déplacer dans le répertoire parent

```
root@fredon:/home/paul/Docs# cd ..
```

« ~ » : Désigne le répertoire home de l'utilisateur courant

```
root@fredon:/home/paul/Docs# cd ~
```

V. Commandes de base:

1. Login et déconnexion:

→ Se connecter : Login + password:

Vous devez taper votre nom d'utilisateur, ou « root » si vous désirez vous connecter en tant qu'administrateur. Le système vous demande alors le mot de passe .

→ Se déconnecter Logout :

Si d'aventure cette commande ne fonctionnait pas, vous pourrez utiliser la commande exit ou la combinaison de touches CTRL+d

→ Arrêt et redémarrage du système

halt, qui permet d'arrêter le système.

reboot, qui permet de le redémarrer.

shutdown [Option][minutes] , où l'option:

-r : redémarrer

-h : Arrêter

-c : annuler

V. Commandes de base:

2. Page manuel:

- Chaque commande Unix a une page de manuel qui la décrit. Pour afficher la page de manuel d'une commande, il suffit d'utiliser la commande suivante :

man nom_commande

- Par exemple, pour afficher l'aide sur la commande **cp**, il suffit de taper :

man cp

- Lorsqu'une page de man est affichée, il est possible de faire défiler son texte à l'aide des touches du curseur. Pour quitter l'aide, il suffit d'appuyer sur la touche **q**.

V. Commandes de base:

3. ls:

- La commande **ls** permet de lister le contenu d'un répertoire.
- Syntaxe: **ls [options] [répertoire]**
 - **[répertoire]** : est le nom du répertoire que l'on désire lister. Si ce paramètre est absent, **ls** affichera tous les fichiers du répertoire courant.
 - Les principales **[options]** sont:
 - **-l**, qui permet d'afficher des informations étendues (notamment les propriétaires, les groupes, les droits, la taille et éventuellement les liens),
 - **-a**, qui permet d'afficher tous les fichiers, y compris les fichiers cachés

V. Commandes de base:

4. cd:

- La commande **cd** permet de changer de répertoire courant.
- Syntaxe: **cd [chemin]** :
 - **[chemin]** est un chemin de répertoire Unix valide.
 - Ce chemin est constitué des noms des répertoires et sous-répertoires successifs, séparés par des barres obliques « / ».

Exemple:

cd /home/id103 : aller au répertoire personnel de 'utilisateur id103

V. Commandes de base:

5. mkdir:

- La commande **mkdir** permet de créer un répertoire :
- **Syntaxe : mkdir [répertoire]**
 - L'option -p permet de créer une arborescence
- **Exemple :** créer un répertoire IstaTAZA , et à l'intérieur un répertoire ID
- **réponse:** mkdir -p IstaTAZA/ID

6. rmdir:

- La commande **rmdir** permet de supprimer un répertoire vide
- il faut qu'il soit vide (c'est-à-dire qu'il ne contienne ni fichier, ni répertoire).
- **Syntaxe: rmdir [répertoire]**
- **rm -r [répertoire]** pour supprimer un répertoire et son contenu (*r* pour récursif)
- **rm -ri [répertoire]** demande une confirmation avant chaque effacement

V. Commandes de base:

7. touch:

- La commande touch permet de créer un fichier vide.
- **Syntaxe:** touch [fichier]

8. Less & cat:

- Les commandes less et cat permettent d'afficher le contenu d'un fichier
- **Syntaxe:**
 - Less [fichier]
 - cat [fichier]

rm:

- rm [fichier] pour supprimer un fichier

V. Commandes de base:

9. cp

- La commande « cp » copie des fichiers source vers une ou plusieurs destinations
- **Syntaxe :** cp [OPTION] [SOURCE] [Destination]
- Exemples:

```
cp tri.txt /home/paul
```

Copie tous les fichiers du répertoire « /home/jean » vers « /home/paul »

```
cp /home/jean/* /home/paul
```

Copie le rep « /home/jean » tout entier vers « /home/paul »

```
cp -r /home/jean/ /home/paul
```

V. Commandes de base:

10. mv

- La commande mv permet de déplacer les fichiers et les répertoires
- Syntaxe: **mv [source] [destination]**
 - [source] est le nom du fichier source
 - [destination] est le nom du répertoire destination.
- La commande mv permet également de renommer des fichiers et des répertoires.
- Pour renommer un fichier ou un répertoire, il suffit d'indiquer le nouveau nom de ce fichier ou de ce répertoire à la place de destination.
- **Exemple:**
- On est dans «/home», Renommer le fichier tri.txt (qui se trouve dans /home/ista) en id.txt
- **Réponse:**
- mv ista/tri.txt i ista/id.txt

V. Commandes de base:

11. gzip

La compression d'un fichier se fait soit avec **gzip** ou **bzip2** :

- **gzip [fichier]**

[fichier] est le fichier à compresser. Après avoir effectué son travail, gzip renomme le fichier compressé en « fichier.gz ».

La compression d'un fichier avec bzip2 utilise exactement la même syntaxe :

- **bzip2 [fichier]**

Le nom du fichier compressé porte l'extension .bz2 au lieu de .gz. Le fichier obtenu est donc nommé « fichier.bz2 ».

La décompression d'un fichier se fait à l'aide de la commande suivante :

gunzip fichier.gz

ou

bunzip2 fichier.bz2

V. Commandes de base:

12. tar

une archive est un fichier dans lequel se trouve tout le contenu d'un dossier (fichiers, arborescence et droits d'accès).

L'archivage de fichiers se fait classiquement sous Unix avec le programme **tar**.

Ce programme permet simplement de regrouper tous les fichiers qu'il doit archiver dans un seul fichier structuré en blocs.

- La syntaxe de tar est : **tar [options] [nom archive][fichiers à archiver]**

[options] sont les options qui indiquent l'opération à effectuer et comment elle doit être réalisée,

[archive] est le nom de l'archive qui doit être créée

[fichiers] est la liste des fichiers à archiver.

Les options de tar que vous utiliserez le plus souvent sont les suivantes :

cvf pour créer une archive ;

tvf pour lister le contenu d'une archive ;

xvf pour restaurer le contenu d'une archive.

rvf pour Ajouter des fichiers à la fin de l'archive.

Exemple: Archiver le contenu du répertoire courant dans le fichier archive.tar,

Réponse: tar cvf archive.tar *

V. Commandes de base:

13. Le super utilisateur:

- Le compte « root » possède tous les droits
 - Celui qui possède le mot de passe root peut tout faire
- Pour passer au mode administrateur, il suffit de taper la commande « **su** »
- Cette commande demande le mot de passe root
- Pour basculer entre les utilisateurs standards, il faut taper la commande:

su -l nom_utilisateur

- Pour vous déconnecter, taper la commande: exit

```
[myriem@localhost ~]$  
[myriem@localhost ~]$ su  
Password:  
[root@localhost myriem]#  
[root@localhost myriem]#  
[root@localhost myriem]# su -l myriem  
[myriem@localhost ~]$  
[myriem@localhost ~]$  
[myriem@localhost ~]$  
[myriem@localhost ~]$ su -l test  
Password:  
su: incorrect password  
[myriem@localhost ~]$ su -l test  
Password:  
[test@localhost ~]$
```

```
[id@localhost ~]$ su -  
Mot de passe :  
[root@localhost ~]# pwd  
/root  
[root@localhost ~]#  
[root@localhost ~]#  
[root@localhost ~]# exit  
logout  
[id@localhost ~]$  
[id@localhost ~]$
```

V. Commandes de base:

14. Complétion automatique des commandes:

- La plupart des Shell sont capables d'effectuer ce que l'on appelle la *complétion automatique des commandes*. La *complétion automatique* permet de n'écrire qu'une partie des noms de fichiers ou de répertoires et de demander au shell de compléter ces noms. Cela peut se faire de deux manières.
 - → La première solution, consiste à taper le début du nom, puis d'utiliser tabulation.
 - → La deuxième solution est d'utiliser les caractères génériques du shell.
 - « * » : remplace une chaîne de caractères
 - « ? » : remplace un caractère
- **exemples:**
 - `cd /ho*`
 - `cd /?ome`
 - `Ls *.txt`

V. Commandes de base:

14. Complétion automatique des commandes:

Critères de recherche des caractères spéciaux

- Les critères de recherche des caractères spéciaux sont les suivants:
- * : remplace une chaîne de caractère de longueur variable
- ? : remplace un caractère unique
- [] : remplace une plage de caractères
- [!...] : inverse la recherche

Exemples :

- A* : les fichiers dont le nom commence par « A »
- A ?? : les fichiers en 3 lettres qui commencent par « A »
- [aA]* : les fichiers dont le nom commence par « a » ou « A »
- [a-m]* : les fichiers dont le nom commence par une lettre de la plage « a-m »

Remarque :

- Il est parfois utile de désactiver la signification spéciale de ces caractères
- Pour les désactiver, utiliser \ , " ou « »

V. Commandes de base:

Autres commandes utiles:

Affichage du système installé :

- **uname -a**

Distribution installée :

- **[cat (ou bien) less] /etc/issue (Debian)**
- **[cat (ou bien) less] /etc/Centos-release (Centos)**

Utilisateurs en cours :

- **who**

Afficher le contenu trié d'un fichier:

- **sort [fichier]**
 - **Sort -n** : effectue un tri numérique
 - **Sort -r** : affiche le contenu trié dans l'ordre décroissant
 - **Sort -t** : indiquer le délimiteur
 - **Sort -k** : indiquer le numéro du champs à trier

V. Commandes de base:

Autres commandes utiles:

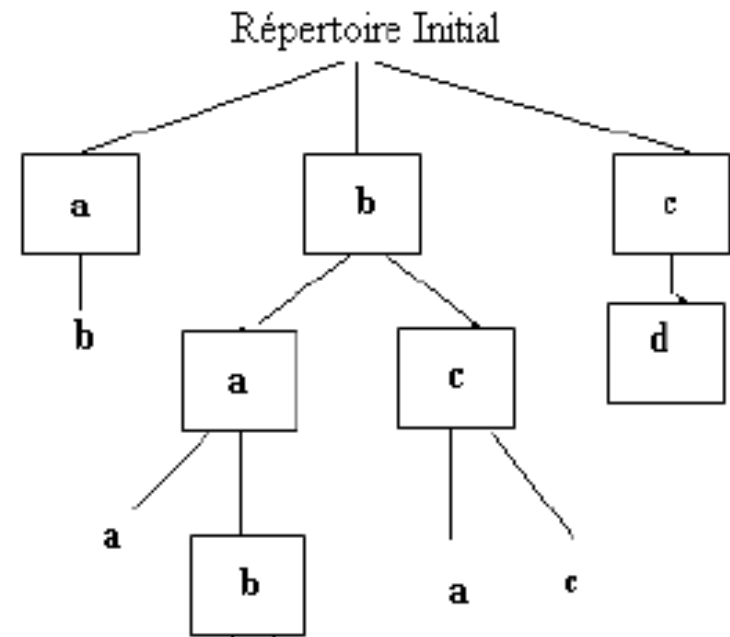
Création d'une arborescence :

Quelle séquence de commandes vous permet de créer l'arborescence suivante?

Réponse:

Mkdir -p a b/a/b b/c c/d

Touch a/b b/a/a b/c/a b/c/c



V. Commandes de base:

15.find:

Rechercher un Fichier ou un répertoire

- Il vous sera sans doute nécessaire de rechercher des fichiers selon un critère donné dans toute une arborescence de répertoires. Pour cela, vous utiliserez la commande : **find [répertoire] -[option] -[print]**

Option	Signification		
-name	Recherche par nom de fichier.	-atime	Recherche par date de dernier accès .
-type	Recherche par type	-mtime	Recherche par date de dernière modification .
-user	Recherche par propriétaire .	-ctime	Recherche par date de création .
-group	Recherche par appartenance à un groupe .	-perm	Recherche par autorisations d'accès .
-size	Recherche par taille de fichier.	-links	Recherche par nombre de liens au fichier.

Exemple : Rechercher le fichier linux-test2 dans toute la racine.

Réponse: **find / -name linux-test2 -print**

V. Commandes de base:

16. grep:

- La commande **grep** permet de chercher une chaîne de caractères dans un Fichier

grep [option] [Model de critères] [fichiers]

[option] :

- i : ne pas tenir les min et maj
- v : toutes les lignes qui ne contiennent pas de critère
- n : avoir les num de lignes
- l : pour lister seulement le fichiers et non pas les lignes contenant la chaîne cherchée
- c : la commande ne retourne que le nombre de lignes trouvées
- E : pour les expressions compliquées

[Model de critères] :

[...] : Plage de caractères permis à cet emplacement

[^...] : Plage de caractères non permis à cet emplacement

* : signe de répétition , agit sur le caractère placé devant le *

. : Caractère quelconque y compris l'espace

\$: fin de ligne

| : ou

^ : début de ligne

\{...\} : caractère de répétition entre accolades

\{nombre\} : nombre exacte

\{nombre.\} : nombre minimum

\{nombre1,nombre2\} : de nombre 1 à nombre2

V. Commandes de base:

16. grep:

- Exemples :

- Chercher les lignes qui commencent par a dans le fichier fich.txt

```
[idl@localhost ~]$ cat fich.txt
Bonjour
Ali
ISTA
Infrastructure digital
ali*
```

```
[idl@localhost ~]$ grep ^a fich.txt
ali*
[idl@localhost ~]$
```

- Calculer le nombre de lignes qui commencent par a dans le fichier fich.txt

```
[idl@localhost ~]$ grep -c ^a fich.txt
1
```

- Afficher les numéros de lignes qui commencent par a dans fich.txt

```
[idl@localhost ~]$ grep -n ^a fich.txt
5:ali*
```

V. Commandes de base:

16. grep:

- Exemples :

- Afficher la liste des utilisateurs créés dans votre machine

```
[id1@localhost ~]$ grep home /etc/passwd  
id1:x:500:500:~/home/id1:/bin/bash  
id2:x:501:501:~/home/id2:/bin/bash  
id3:x:502:502:~/home/id3:/bin/bash
```

- Calculer le nombre de comptes créés dans votre machine.

```
[id1@localhost ~]$  
[id1@localhost ~]$ grep -c home /etc/passwd  
3
```

V. Commandes de base:

17. Les Tubes:

Les tubes permettent d'établir une liaison directe entre la sortie d'une commande avec l'entrée d'une autre commande.

Pour cela utiliser le symbole |

Exemples :

1. afficher le contenu trié de votre répertoire courant
2. chercher la chaine « **Do** » dans le contenu de **/etc**
3. Chercher dans la racine de votre système le fichier **ista**, stocker le résultat de la recherche dans le fichier **resultat** et afficher les erreurs de la recherche à l'écran

Réponses:

1. `ls -l | sort`
2. `ls -l /etc | grep Do`
3. `find / -type f -name ista > resultat`

V. Commandes de base:

18.cut

- La commande **cut** permet d'afficher les caractères d'un fichier
 - Afficher le 5 ème caractère du fichier Ex : **cut -c5 fichier**
 - Afficher du 5 ème au 10 ème caractère : **cut -c5-10 fichier**
 - Afficher le 5 ème et le 10 ème caractère : **cut -c5,10 fichier**
 - Afficher à partir du 5ème caractère jusqu'à la fin du fichier : **cut -c5- ex**

Remarques:

- pour **délimiter les champs**, on utilise l'option **-d**
- pour **indiquer les champs**, on utilise l'option **-f**
- **Exemple:** Copiez le fichier **/etc/passwd** dans votre home directory. Editez uniquement les champs contenant le login et la home directory.

```
[idl@localhost ~]$ cp /etc/passwd ~ [idl@localhost ~]$ cut -d : -f1,6 /etc/passwd | grep home
[idl@localhost ~]$
[idl@localhost ~]$ cat passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
```


V. Commandes de base:

19. wc

- La commande wc permet de calculer le nombre de caractères, de mots ou de lignes dans un fichier

Options:

- l : calculer le nombre de lignes
- c : calculer le nombre de caractères
- w : calculer le nombre de mots

Exemple : Calculer le nombre de fichiers dans votre repertoire d'accueil.

```
[idl@localhost ~]$ ls
Bureau      fl.txt~  Modèles  passwd  Public  Téléchargements  test3.txt~
ex2.txt~   Images  Musique  projet  r_admin  test              Vidéos
[idl@localhost ~]$ ls -l | grep ^- | wc -l
4
```

V. Commandes de base:

20. tr

- La commande **tr** permet de remplacer des chaînes de caractères dans un mot ou un fichier.
- **Syntaxe:** `tr [options] ch1 ch2`
- **Exemple:**

```
[id1@localhost ~]$ cat tp.txt
ista          TAZA
groupe                               ID101
Filière infrastructure digitale
cours Linux
Ville TAZA
```

```
[id1@localhost ~]$ cat tp.txt |tr TAZA CASA
ista          CASA
groupe                               ID101
Filière infrastructure digitale
cours Linux
Ville CASA
```

V. Commandes de base:

20. tr

- La commande **tr** permet de remplacer des chaînes de caractères dans un mot ou un fichier.
- **Syntaxe:** `tr [options] ch1 ch2`
- **Options:**
 - `-d` : supprimer les caractères indiqués

```
[id1@localhost ~]$ cat tp.txt
ista          taza
groupe                                ID101
Filière infrastructure digitale
cours Linux
```

```
[id1@localhost ~]$ cat tp.txt | tr -d " "
istataza
groupeID101
Filièreinfrastructuredigitale
coursLinux
```

V. Commandes de base:

20. tr

- La commande **tr** permet de remplacer des chaînes de caractères dans un mot ou un fichier.
- **Syntaxe:** `tr [options] ch1 ch2`
- **Options:**
 - `-s` : remplace une séquence d'occurrences répétées.

```
[id1@localhost ~]$ cat tp.txt
ista          taza
groupe                               ID101
Filière infrastructure digitale
cours Linux

[id1@localhost ~]$ cat tp.txt | tr -s " " " "
ista taza
groupe ID101
Filière infrastructure digitale
cours Linux
```

V. Commandes de base:

21.ps

- La commande ps permet de lister les processus en cours d'exécution
 - Ps -ef : affiche la liste des processus en cours ainsi l'utilisateur qui a lancé chacun d'entre eux.

```
[idl@localhost ~]$ ps -ef
```

UID	PID	PPID	C	STIME	TTY	TIME	CMD
root	1	0	0	11:46	?	00:00:02	/sbin/init
root	2	0	0	11:46	?	00:00:00	[kthreadd]
root	3	2	0	11:46	?	00:00:00	[migration/0]
root	4	2	0	11:46	?	00:00:00	[ksoftirqd/0]
idl	1965	1937	0	11:51	?	00:00:00	gnome-session
idl	1973	1	0	11:51	?	00:00:00	dbus-launch --sh-syntax --exit-
idl	1974	1	0	11:51	?	00:00:00	/bin/dbus-daemon --fork --print
idl	2048	1	0	11:51	?	00:00:00	/usr/libexec/gconfd-2
idl	2052	1	0	11:51	?	00:00:01	/usr/libexec/gnome-settings-dae
idl	2062	1	0	11:51	?	00:00:00	/usr/libexec/gvfsd
idl	2065	1	0	11:51	?	00:00:00	seahorse-daemon
idl	2073	1965	0	11:51	?	00:00:02	metacity
idl	2079	1	0	11:51	?	00:00:02	/usr/bin/pulseaudio --start --l
idl	2082	1965	0	11:51	?	00:00:02	gnome-panel
idl	2088	2079	0	11:51	?	00:00:00	/usr/libexec/pulse/aconf-helper

V. Commandes de base:

21.ps

▪ Exemples

- Lister la liste des processus lancés par root:

```
[idl@localhost ~]$ ps -ef | grep root
```

root	1	0	0	11:46	?	00:00:02	/sbin/init
root	2	0	0	11:46	?	00:00:00	[kthreadd]
root	3	2	0	11:46	?	00:00:00	[migration/0]
root	4	2	0	11:46	?	00:00:00	[ksoftirqd/0]
root	5	2	0	11:46	?	00:00:00	[migration/0]
root	6	2	0	11:46	?	00:00:00	[watchdog/0]
root	7	2	0	11:46	?	00:00:00	[events/0]
root	8	2	0	11:46	?	00:00:00	[cgroup]
root	9	2	0	11:46	?	00:00:00	[khelper]

V. Commandes de base:

21.ps

▪ Exemples

- Lister la liste des processus des processus triés par PID dans l'ordre numérique décroissant

```
[id1@localhost ~]$ ps -ef | tr -s " " | sort -nr -t " " -k 2
id1 2566 2544 0 10:26 pts/0 00:00:00 sort -nr -t -k 2
id1 2565 2544 0 10:26 pts/0 00:00:00 tr -s
id1 2564 2544 2 10:26 pts/0 00:00:00 ps -ef
id1 2544 2542 0 10:25 pts/0 00:00:00 bash
id1 2543 2542 0 10:25 ? 00:00:00 gnome-pty-helper
id1 2346 2162 0 10:25 ? 00:00:00 python /usr/share/system-config-printer/applet.py
id1 2343 2162 0 10:25 ? 00:00:00 nm-applet --sm-disable
id1 2338 2162 0 10:25 ? 00:00:00 /usr/libexec/gdu-notification-daemon
id1 2335 2162 0 10:25 ? 00:00:00 bluetooth-applet
id1 2332 2162 0 10:25 ? 00:00:00 /usr/sbin/restorecond -u
id1 2330 2162 0 10:25 ? 00:00:00 abrt-applet
id1 2329 2162 0 10:25 ? 00:00:00 /usr/libexec/polkit-gnome-authentication-agent-1
id1 2324 2162 0 10:25 ? 00:00:00 gnome-volume-control-applet
id1 2322 2162 0 10:25 ? 00:00:00 gnome-power-manager
id1 2319 2162 0 10:25 ? 00:00:00 rhsm-icon
id1 2316 2162 0 10:25 ? 00:00:00 gpk-update-icon
root 2308 2307 0 10:25 ? 00:00:00 udisks-daemon: polling /dev/sr0
```

V. Commandes de base:

21.ps

▪ Exemples

- Lister les PID des processus des processus et les trier par ordre numérique décroissant:

```
[id1@localhost ~]$ ps -ef |tr -s " " | cut -d " " -f2 | sort -nr  
3615  
3614  
3613  
3612  
3572  
3571  
3570  
3495  
3491  
3478
```


V. Commandes de base:

22. Redirection des entrées / sorties:

Pour enregistrer le résultat d'une commande dans un fichier ailleurs qu'à l'écran, utiliser la : **Commande > fichier**

Si le fichier existe déjà, son contenu sera perdu (écrasé)

Si non, il faut utiliser >>

Si le fichier n'existe pas, il est crée automatiquement

Pour enregistrer juste les erreurs de la commande, utiliser 2>

Exemples :

1. Enregistrer le contenu détaillé de **/etc** dans le fichier **tri** sans écraser son contenu
2. Rechercher dans votre répertoire personnel le fichier **tri**, et rediriger les erreurs de la recherche vers le fichier **erreur**.

Réponses:

1. `ls -l /etc >> tri`
2. `Find ~ -name tri 2> erreur`

VI. Droits d'accès et utilisateurs:

1. Introduction:

- Linux est un système multi-utilisateurs
 - Plusieurs utilisateurs se partagent l'espace disque
 - Les fichiers et répertoires d'un utilisateur ne doivent pas être accessibles par les autres
 - Les fichiers de configuration du système doivent être protégés
- Nécessité de spécifier des droits pour chaque fichier/répertoire
 - Plusieurs types de droits : Lecture (R), écriture (W), exécution (X)
 - Ces droits s'appliquent pour 4 groupes d'utilisateurs
 - Le propriétaire (u) du fichier
 - Le groupe (g) propriétaire (Tous les utilisateurs membre du groupe)
 - Les autres (o). Désigne tous les utilisateurs non membres des 2 précédents
 - Tout le monde (a)

VI. Droits d'accès et utilisateurs:

2. Signification des Droits:

- Il existe 3 types de droits : r, w et x
- Ces droits n'ont pas la même signification pour un fichier que pour un répertoire
- Pour un fichier :
 - r : Lecture (afficher)
 - w : Ecriture (modification)
 - x : Exécution (**uniquement le cas d'un script**)
- Pour un répertoire
 - r : Lire le contenu, lister les fichiers (avec ls par exemple)
 - w : Modifier le contenu, créer et supprimer des fichiers (avec les commandes « cp », « mv », « rm »)
 - x : d'accéder aux fichiers du répertoire. Mais aussi de naviguer dans les sous-répertoires (avec « cd »)

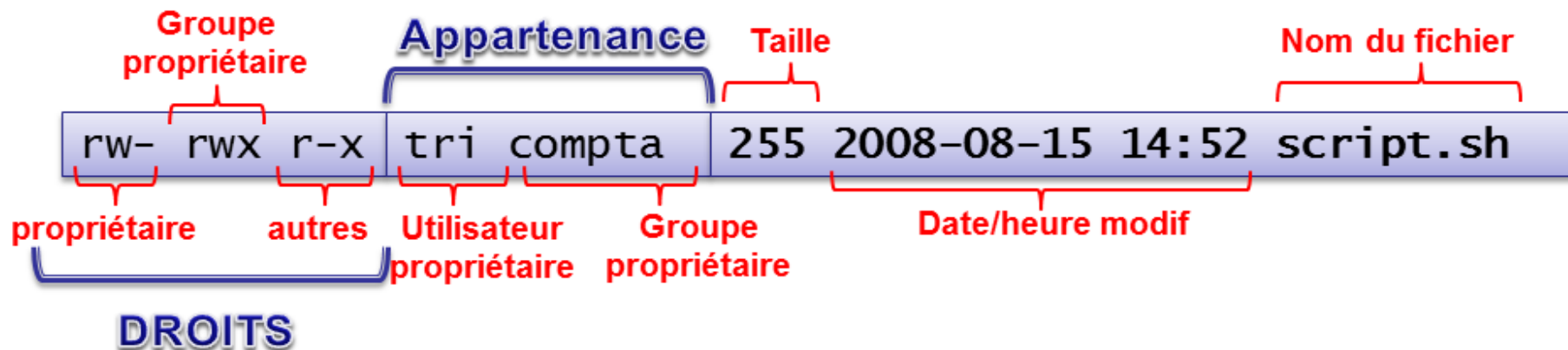
VI. Droits d'accès et utilisateurs:

2. Signification des Droits:

- La commande « ls -l » permet d'afficher les droits qui s'appliquent

```
root@ista:/home/tri/Documents# ls -l
total 20
-rw-r--r-- 1 tri tri      0 2008-08-15 14:42 projet.txt
-rw-rw-r-x 1 tri compta 7406 2008-08-15 14:44 rapport2006.ods
-rw-rw-r- 1 tri tri    7363 2008-08-15 14:44 rapport-activite.odt
-rw-rwxr-x 1 tri compta 255 2008-08-15 14:52 script.sh
```

- Signification des différents champs



Propriétaire : Lecture, écriture

Groupe : Lecture, écriture et exécution

Autres : Lecture et exécution

VI. Droits d'accès et utilisateurs:

3. Commande « chmod »

Droit sous forme symbolique:

- La commande chmod permet de modifier les droits d'accès
chmod [droits] [fichier/repertoire]
- Les droits sont constitués de 2 éléments:
 - La personne sur laquelle on applique le droit:
 - u** pour le champ « utilisateur », c'est-à-dire le propriétaire du fichier ;
 - g** pour le champ « groupe »
 - o** pour le champ « other »
 - a** pour tous les champs sans distinction, donc pour tous les utilisateurs.
 - Le droit qu'on veut affecter
 - r** pour le droit de lecture ;
 - w** pour le droit d'écriture ;
 - x** pour le droit d'exécution ;

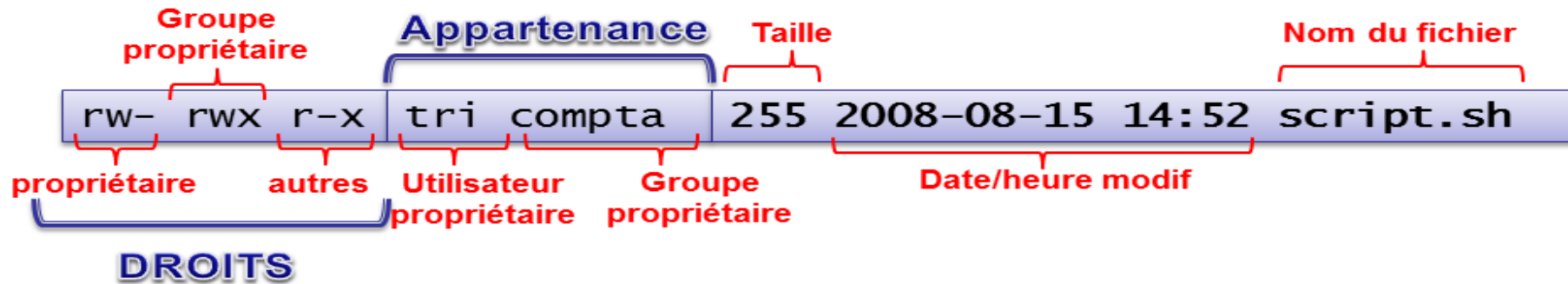
Exemple :

chmod g+w exemple , permet de donner le droit d'écriture sur le fichier exemple à tous les membres du groupe auquel ce fichier appartient.

VI. Droits d'accès et utilisateurs:

3. Commande « chmod »

Droit sous forme octale:



- **r (lecture) = 4**
- **w (Ecriture) = 2**
- **x (Exécution) = 1**

Exemples:

- **777** signifie que tout le monde a tous les droits (lecture, écriture, exécution) sur ce fichier.
- **750** signifie que le propriétaire a tous les droits, le groupe peut lire et exécuter, les autres n'ont aucun droit.
- **644** indique un fichier que tous peuvent lire, mais où le propriétaire seul est autorisé à écrire.

VI. Droits d'accès et utilisateurs:

4. Commande « chown »:

Le changement de propriétaire d'un fichier ne peut être réalisé que par l'administrateur du système.

Cette opération se fait à l'aide de la commande suivante :

chown [utilisateur] [fichier]

[utilisateur] est le nom de l'utilisateur qui doit devenir propriétaire du fichier,
[fichier] est le fichier devant changer de propriétaire.

Exemple:

chown root tp.txt → permet de rendre l'administrateur propriétaire du fichier

tp.txt

```
[root@localhost id1]# ls -l | grep ^-  
-rw-rw-r--. 1 id1 id1 0 20 déc. 07:52 tp.txt  
[root@localhost id1]# chown root tp.txt  
[root@localhost id1]#  
[root@localhost id1]#  
[root@localhost id1]# ls -l | grep ^-  
-rw-rw-r--. 1 root id1 0 20 déc. 07:52 tp.txt
```

VI. Droits d'accès et utilisateurs:

5. Commande « chgrp »

- Le changement de groupe peut être réalisé par n'importe quel utilisateur, mais on ne peut donner un fichier qu'à l'un des groupes dont on est membre.
- Cette opération se fait à l'aide de la commande suivante :

chgrp [groupe] [fichier]

[groupe] : le nom du groupe qui doit être affecté au fichier,

[fichier] : le fichier devant changer de groupe.

Exemple:

chgrp group1 fich1 : rendre le groupe « group1 » propriétaire du fichier fich1

VI. Droits d'accès et utilisateurs:

6. Gestion des utilisateurs:

- ❑ Un utilisateur est caractérisé par une ligne dans le fichier `/etc/passwd`
- ❑ Une ligne est formée par les champs suivants :

login : passwd : uid : gid : comment : home : shell

- **login** : Il s'agit du nom de l'utilisateur
- **passwd** : Il apparaît dans ce fichier sous forme codée.
- **uid** : Il s'agit d'un numéro unique de l'utilisateur. Compris entre 0 et 65535.
- **gid** : Numéro de groupe. Chaque utilisateur appartient à un groupe principal. Il pourra également appartenir à des groupes secondaires. (séparés par des virgules)
- **comment** : Nom complet de l'utilisateur.
- **home** : Chemin du répertoire personnel de l'utilisateur
- **shell** : programme qui interagit avec l'utilisateur et qui permet de taper des commandes (`csh`, `sh`, `bash`, `tcsh` , ...)

Exemples:

root : x:0:0:root:/root:/bin/bash

user : x:500:500:user:/home/user:/bin/bash

VI. Droits d'accès et utilisateurs:

6. Gestion des utilisateurs:

1. Créer un utilisateur:

Useradd [-c commentaire] [-d répertoire] [-g groupe] [-G groupes] [-p passe] [-s shell] [-u uid] nom_login

-c : commentaire

-d : répertoire Personnel

-g : groupe-primaire

-G : groupe-secondaire :

-p : mot_passe

-s : shell

-u : uid

-m -k /etc/skel : pour créer le répertoire personnel et recopier le contenu de rép-skel dans le rép.personnel

VI. Droits d'accès et utilisateurs:

6. Gestion des utilisateurs:

1. Créer un utilisateur:

Exemple :

Créer l'utilisateur linux:

- Commentaire : " utilisateur linux "
- Son shell : /bin/sh
- Son répertoire personnel : /home/linux
- Nous recopions le contenu de /etc/skel dans son répertoire
- Il appartient aux groupes « dev » et « id »

Réponse :

```
root @localhost# useradd -c "utilisateur linux" -d /home/linux -G id,dev  
-s /bin/sh -m -k /etc/skel linux
```

VI. Droits d'accès et utilisateurs:

6. Gestion des utilisateurs:

2. Attribuer un mot de passe :

La commande est la suivante : **passwd [options] nom_login**

Options :

- d : pour supprimer le mot de passe
- l : pour verrouiller le compte et empêcher sa connexion
- u : pour déverrouiller

3. Supprimer le compte d'un utilisateur (non connecté)

La commande est la suivante : **userdel -r nom_login**

L'option **-r** supprime toute trace de l'utilisateur dans le fichier de configuration :
/etc/passwd y compris dans les groupes d'utilisateurs.

VI. Droits d'accès et utilisateurs:

6. Gestion des utilisateurs:

4. Modifier le compte d'un utilisateur (non connecté)

- La commande est la suivante : **usermod** [options] nom_login
- Les options sont les mêmes que useradd

Exemple :

- **usermod -G stagiaire,prof user1**
- Ajoute **user1** dans les 2 groupes stagiaire et prof (qui doivent exister)

VI. Droits d'accès et utilisateurs:

7. Gestion des groupes:

- Un groupe est un ensemble d'utilisateurs.
- Chaque utilisateur doit faire partie au moins d'un groupe.
- la liste des groupes est donnée par le fichier /etc/group
- le fichier de déclaration des groupes /etc/group contient une ligne par groupe dans un format similaire à : **Groupe*:GID:utilisateurs**
 - Nom du groupe
 - mot de passe du groupe
 - numéro du groupe (gid)
 - liste des utilisateurs appartenant au groupe séparés par des virgules

Exemples:

id101:x:501:ali,omar

id102:x:502:ali,omar,mohammed,fatima

id103:x:503:u1,u2

VI. Droits d'accès et utilisateurs:

7. Gestion des groupes:

- Pour créer un nouveau groupe :

`groupadd -g GID nom_groupe`

- supprimer un groupe :

`groupdel nom_groupe` (le groupe est supprimé du fichier /etc/group).

- Modifier le nom d'un groupe :

`groupmod -n nouveau_nom ancien_nom_groupe`

- Pour lister tous les groupes d'un utilisateur :

`groups nom_login`

VII. Gestion des paquets et mises à jour

1. Introduction:

- **Un paquet** (en [anglais](#) *package*) est un fichier qui contient les informations et nécessaires à l'installation d'un [logiciel](#) ou d'un service.
- Il existe différents outils d'installation de packages et d'applications:
 - **rpm,**
 - **yum,**
 - **deb**
 - **apt,**
 - **Snap**
 - **...**

VII. Gestion des paquets et mises à jour

2. rpm

- **Mettre à jour un paquet**
 - `rpm -Uhv « nom paquet.rpm »`
- **Installer un paquet**
 - `rpm -ihv « nom paquet.rpm »`
- **Connaitre la date d'installation d'un paquet**
 - `rpm -qa --last | grep truc`
- **Désinstaller un paquet**
 - `rpm -e « nom paquet sans '.rpm' »`
- **Rechercher si le paquet truc est installé**
 - `rpm -q « nom complet du paquet, sans '.rpm' »`
- **Liste de tous les paquets installés**
 - `rpm -qa`

VII. Gestion des paquets et mises à jour

2. rpm

- **Force l'installation d'un paquet**
 - rpm -ivh --force « nom du package »
- **Lister les fichiers d'un paquet rpm**
 - rpm -ql « nom du package »
- **Connaître le paquet d'un fichier**
 - rpm -qf « nom_fichier »
- **Connaître le contenu d'un rpm**
 - rpm -pql « nom_paquet »
- **Avoir les informations sur un paquet installé**
 - rpm -qi « nom du package »

VII. Gestion des paquets et mises à jour

3. yum

- *yum* est un gestionnaire de paquets, utilisé par certaines distributions Linux comme CentOS, Fedora, RedHat, etc.
- Il permet de gérer l'installation et la mise à jour des logiciels installés sur le système d'exploitation.
- En résumé, ***yum*** peut :
 - **Lister les mises à jour disponibles**
 - **Lister les paquets installés**
 - **Rechercher un paquet particulier**
 - **Installer/Désinstaller un paquet**
 - **Mettre à jour le système d'exploitation**

VII. Gestion des paquets et mises à jour

3. yum

Afficher les paquets installés:

```
yum list
```

Installer un paquet:

```
yum install nom_paquet
```

Réinstaller un paquet:

```
yum reinstall nom_paquet
```

Afficher les infos sur un paquet:

```
yum info nom_paquet
```

Désinstaller un paquet:

```
yum remove nom_paquet
```

VII. Gestion des paquets et mises à jour

3. yum

Vérifier les mises à jour:

yum check-update

Exemple:

```
~]# yum check-update
Loaded plugins: product-id, search-disabled-repos, subscription-manager
dracut.x86_64                033-360.el7_2      rhel-7-server-rpms
dracut-config-rescue.x86_64  033-360.el7_2      rhel-7-server-rpms
kernel.x86_64               3.10.0-327.el7     rhel-7-server-rpms
rpm.x86_64                  4.11.3-17.el7      rhel-7-server-rpms
rpm-libs.x86_64             4.11.3-17.el7      rhel-7-server-rpms
rpm-python.x86_64           4.11.3-17.el7      rhel-7-server-rpms
yum.noarch                  3.4.3-132.el7      rhel-7-server-rpms
```

VII. Gestion des paquets et mises à jour

3. Yum

Mises à jour des paquets:

`yum update nom_paquet`

Mettre à jour des paquets liés à la sécurité

`yum update --security`