

# M104 Fonctionnement d'un système d'exploitation

## Open source Linux

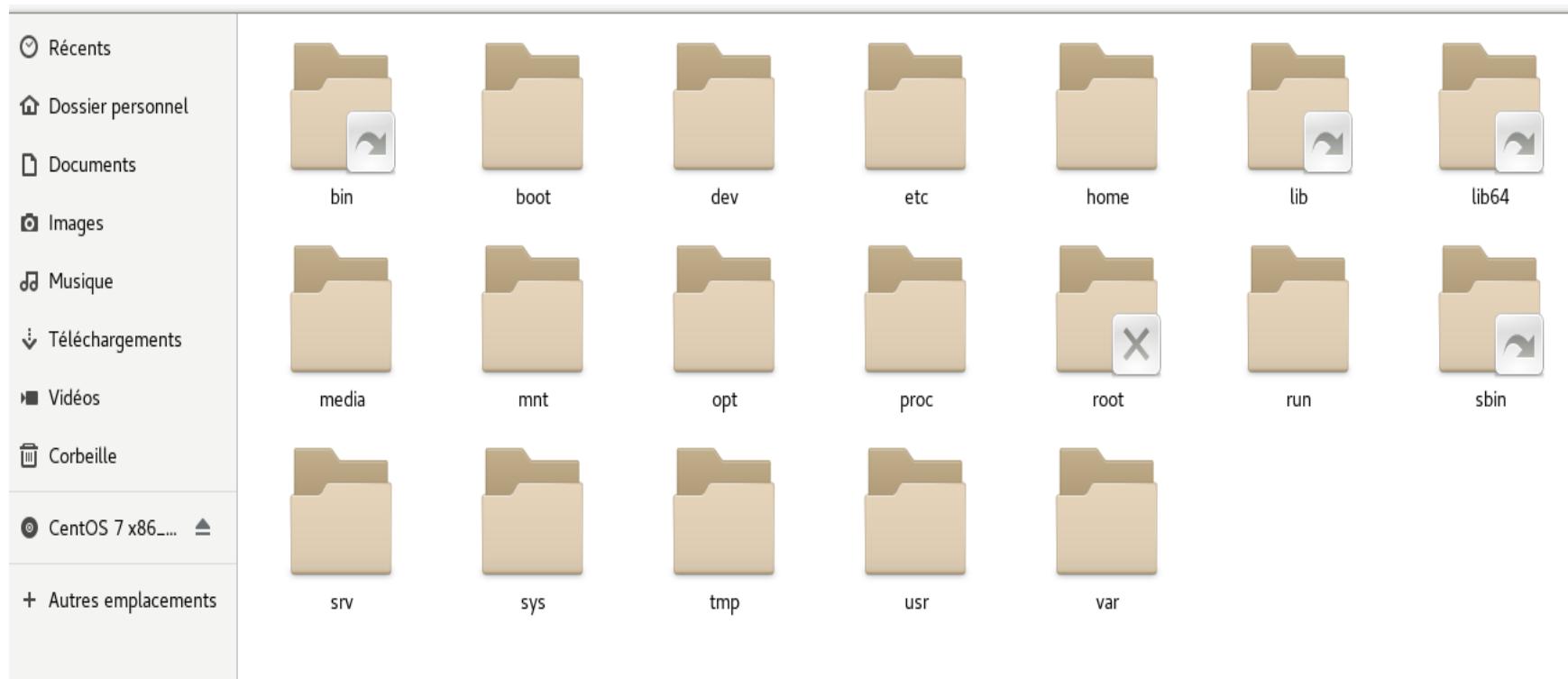


# Arborescence des fichiers sous linux

- Linux a besoin de deux partitions pour fonctionner correctement
  - Une partition Système
  - Une partition Swap (mémoire virtuelle)
- La partition système Linux est identifiée par /
- / est appelé racine de la partition linux

# Arborescence des fichiers sous linux

## La Racine du système d'exploitation Linux



# Arborescence des fichiers sous linux

- La partition système Linux contient plusieurs répertoires, comme :
  - **boot** : Ce répertoire contient des fichiers système nécessaires pour le démarrage de linux
  - **etc** : Ce répertoire contient des fichiers de configuration nécessaires pour l'administration des serveurs linux
  - **opt** : Ce répertoire contient les packages des programmes installés sur Linux

# Arborescence des fichiers sous linux

- **home** : Ce répertoire contient les documents personnels des utilisateurs
- **root** : Ce répertoire contient les documents personnels de l'administrateur root.
- **bin** : Ce répertoire contient les exécutables des commandes shell linux
- **dev** : Ce répertoire contient les fichiers spéciaux des périphériques
- **mnt** : ce répertoire contient les points de montages des périphériques

# Premières commandes

**whoami** : afficher l'utilisateur connecté sur le terminal en cours

**pwd** : afficher la position actuelle sur le disque dur

**uname -a** : afficher la version du noyau

**date** : afficher la date système

**su -** : se connecter en tant qu'administrateur (root)

**who** : afficher la liste des sessions ouvertes sur le serveur linux

**exit** : se déconnecter de la session actuelle

**cd** : se déplacer sur le disque dur

**ls** : afficher le contenu d'un répertoire

# Premières commandes

**shutdown -h now** : arrêter la machine linux immédiatement

**shutdown -t 60** : arrêter la machine linux apres 60 secondes

**shutdown -r now** : redémarrer la machine linux immédiatement

**shutdown -h 23:00** :Arrêter la machine linux à 23h

**shutdown -c** : Annuler un arrêt planifié

# Gestion des fichiers et des répertoires

**mkdir** : créer un nouveau répertoire

ex : mkdir /home/stagiaire/Bureau/linux

**touch** : créer un nouveau fichier

ex : touch /home/stagiaire/Bureau/cours.doc

**cp -R** : copier un répertoire vers une nouvelle destination

ex: cp -R /home/stagiaire/Bureau/linux /home/stagiaire

**cp** : copier un fichier vers une nouvelle destination

ex : cp /home/stagiaire/Bureau/cours.doc /home/stagiaire

# Gestion des fichiers et des répertoires

**mv** : déplacer un répertoire ou un fichier vers une nouvelle destination

ex: mv /home/stagiaire/Bureau/cours /home/stagiaire

**mv** : Renommer un répertoire ou un fichier

ex:

```
cd /home/stagiaire/Bureau  
mv cours cours2020  
mv exercices.txt exercices2020.txt
```

**rm -rf** : Supprimer un répertoire

ex : rm -rf /home/stagiaire/Bureau/linux

**rm -f** : supprimer un fichier

ex : rm -f /home/stagiaire/Bureau/cours.doc

# Gestion des fichiers et des répertoires

- Dans cette démo nous allons exécuter les opérations suivantes :
  - 1- création d'un dossier **cours** dans le Bureau de l'utilisateur stagiaire
  - 2- Création de deux dossier : **linux** et **systèmes** à l'intérieur du dossier cours
  - 3- Création d'un fichier nommé **exercices.txt** à l'intérieur du dossier **linux**
  - 4- copie du dossier **linux** à l'intérieur du dossier **systèmes**
  - 5- modification du nom du dossier **cours** par **modules**

# Gestion des fichiers et des répertoires

```
[stagiaire@localhost ~]$ cd /home/stagiaire/Bureau
[stagiaire@localhost Bureau]$ mkdir cours
[stagiaire@localhost Bureau]$ cd cours
[stagiaire@localhost cours]$ mkdir linux
[stagiaire@localhost cours]$ mkdir systemes
[stagiaire@localhost cours]$ cd linux
[stagiaire@localhost linux]$ touch exercices.txt
[stagiaire@localhost linux]$ cp -R /home/stagiaire/Bureau/cours/linux /home/stagiaire/Bureau/cours/systemes
[stagiaire@localhost linux]$ cd /home/stagiaire/Bureau
[stagiaire@localhost Bureau]$ mv cours modules
```

# **Edition des fichiers texte**

- Plusieurs outils de commandes peuvent être utilisés pour ouvrir et éditer les fichiers sous linux. Notons par exemple :

**gedit**

**cat**

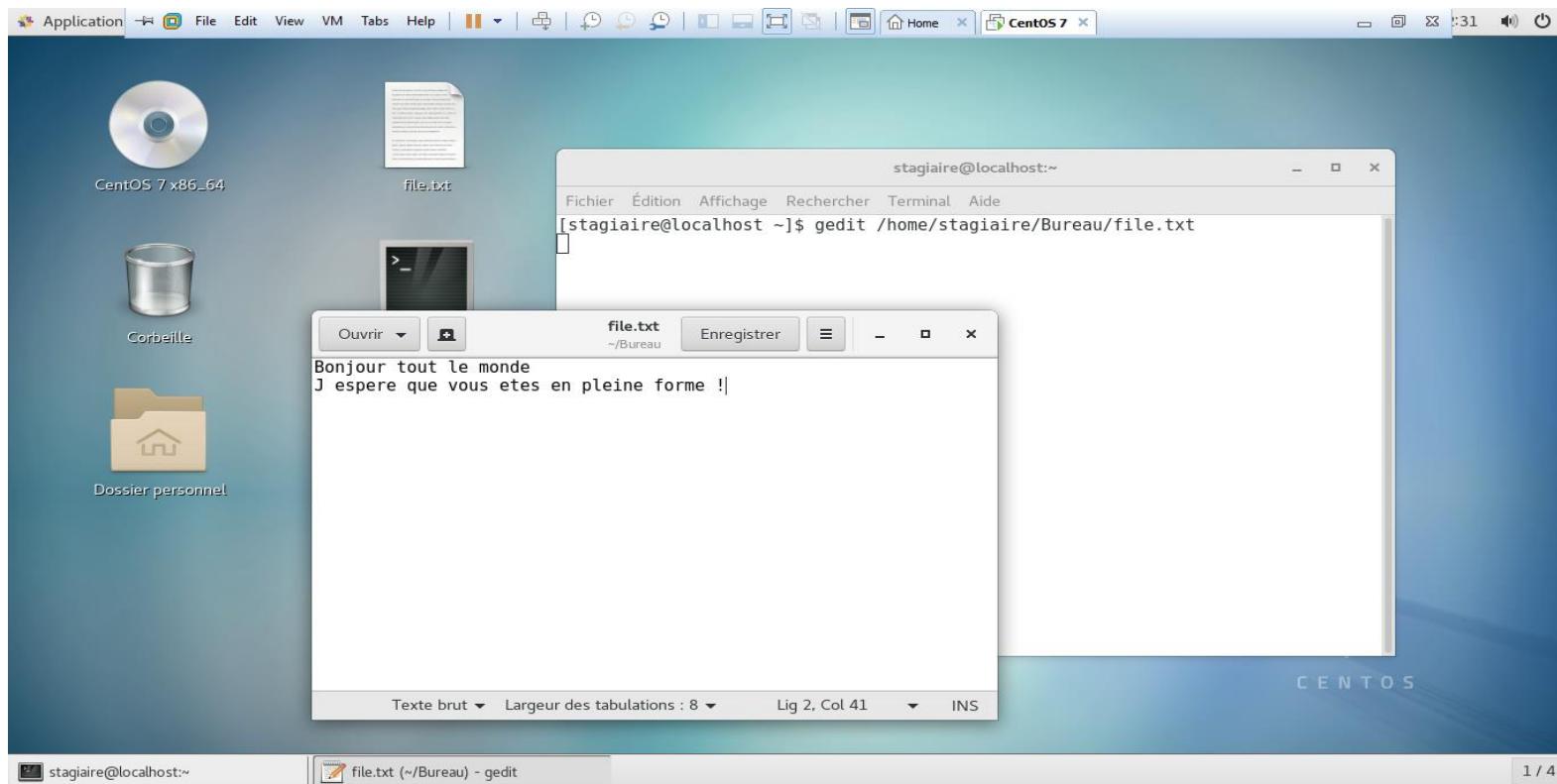
**more**

**less**

**vi**

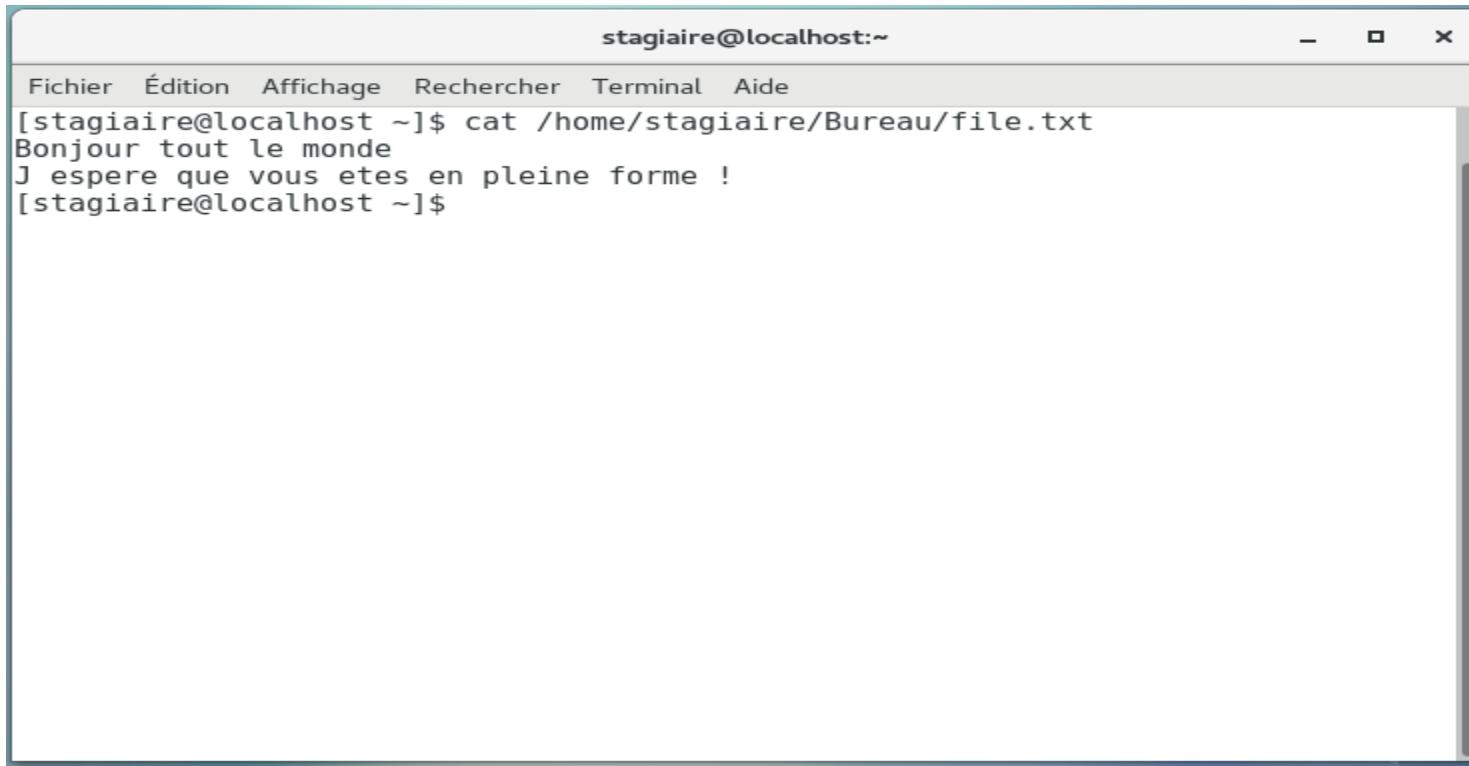
# Edition des fichiers texte

- La commande **gedit** permet d'ouvrir et éditer un fichier texte en mode graphique.



# Edition des fichiers texte

- La commande **cat** permet **d'ouvrir en lecture seule** une fichier texte dans le terminal shell

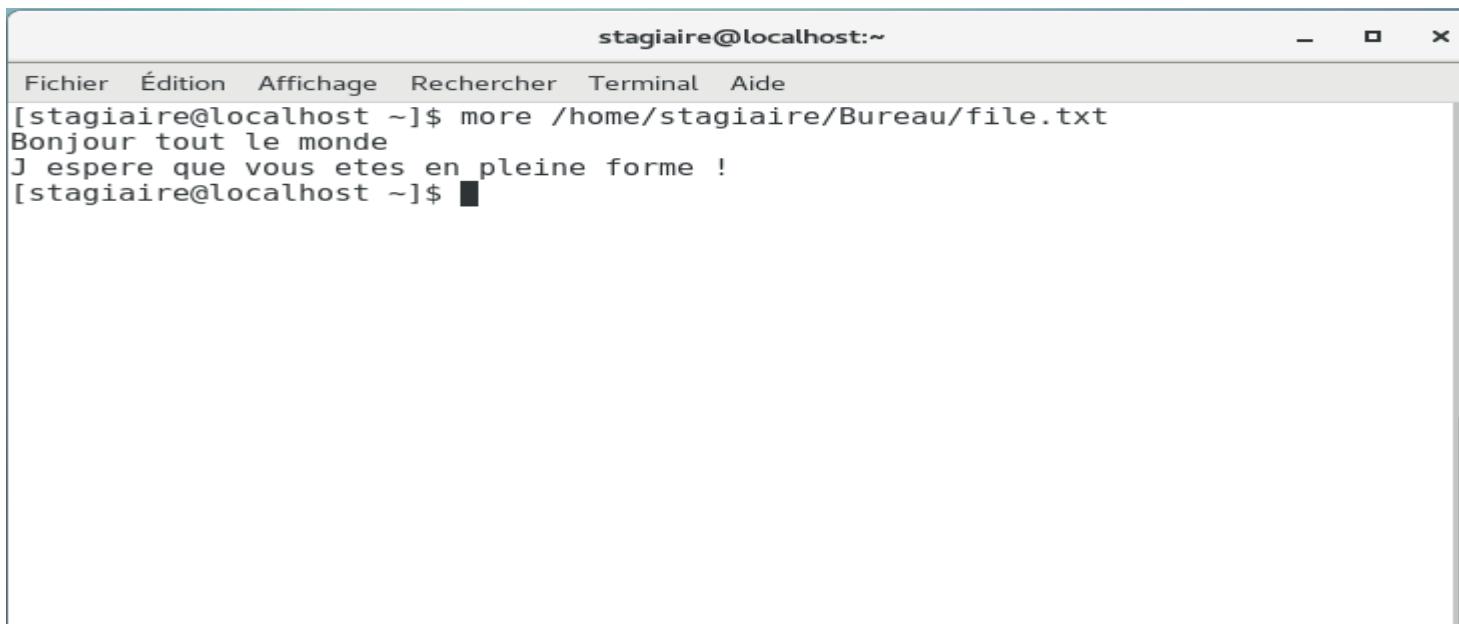


A screenshot of a terminal window titled "stagiaire@localhost:~". The window has a menu bar with "Fichier", "Édition", "Affichage", "Rechercher", "Terminal", and "Aide". The main area of the terminal shows the following text:

```
Fichier Édition Affichage Rechercher Terminal Aide
[stagiaire@localhost ~]$ cat /home/stagiaire/Bureau/file.txt
Bonjour tout le monde
J espere que vous etes en pleine forme !
[stagiaire@localhost ~]$
```

# Edition des fichiers texte

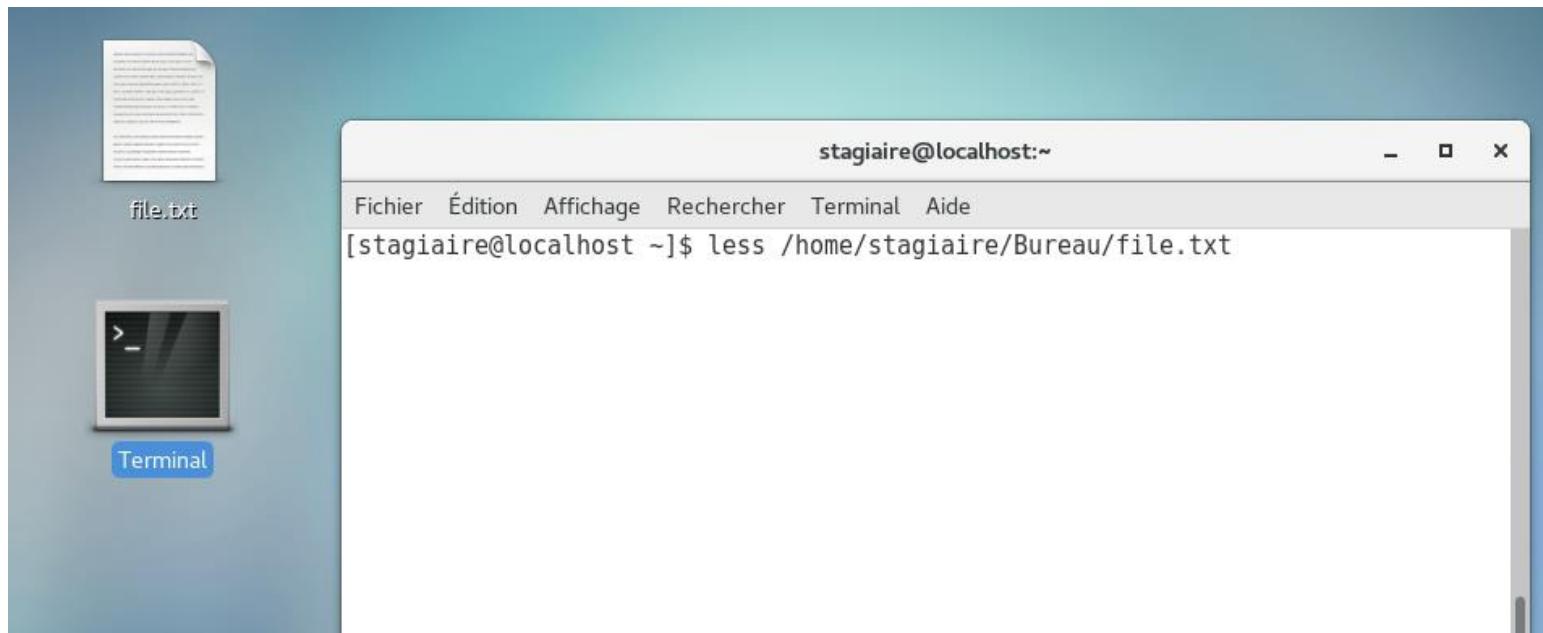
- La commande **more** permet d'ouvrir en lecture seule un fichier texte dans le terminal shell.



A screenshot of a terminal window titled "stagiaire@localhost:~". The window has standard window controls (minimize, maximize, close) at the top right. The menu bar includes "Fichier", "Édition", "Affichage", "Rechercher", "Terminal", and "Aide". The terminal prompt is "[stagiaire@localhost ~]\$". The user runs the command "more /home/stagiaire/Bureau/file.txt". The output shows two lines of text: "Bonjour tout le monde" and "J espere que vous etes en pleine forme !". The terminal prompt "[stagiaire@localhost ~]\$" appears again at the bottom, followed by a small black square cursor icon.

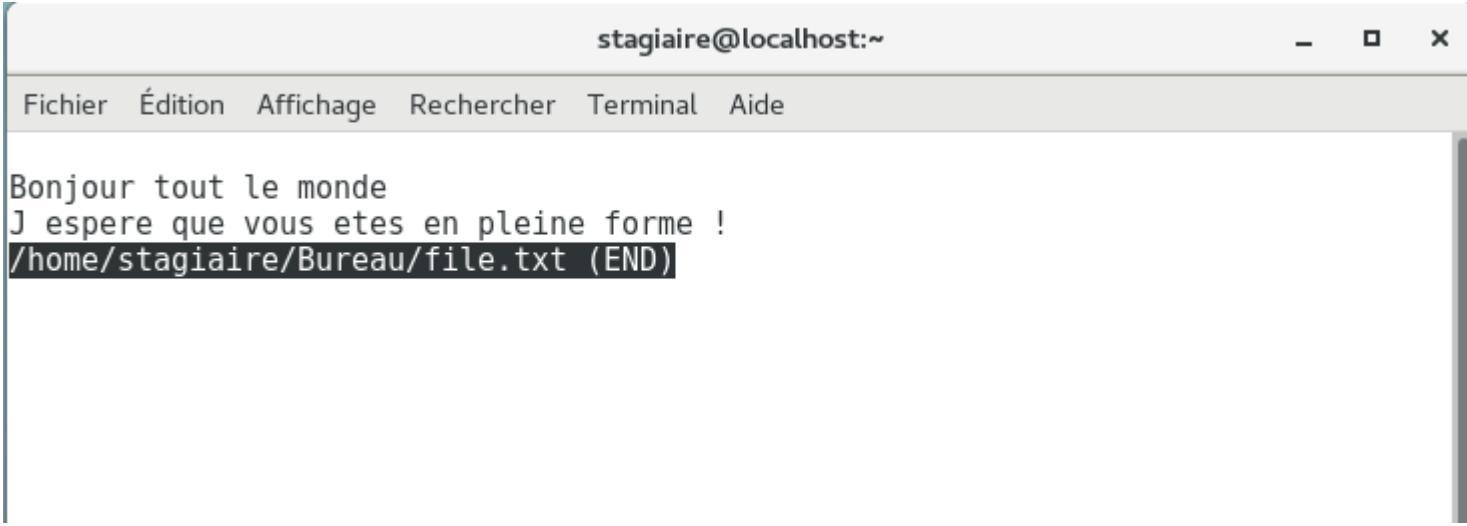
# Edition des fichiers texte

- La commande **less** permet d'ouvrir en lecture seule un fichier texte dans le terminal shell.



# Edition des fichiers texte

- Pour arrêter l'exécution de la commande **less** et revenir à l'invite de commande shell, il faut appuyer dans le clavier sur la touche 'q'

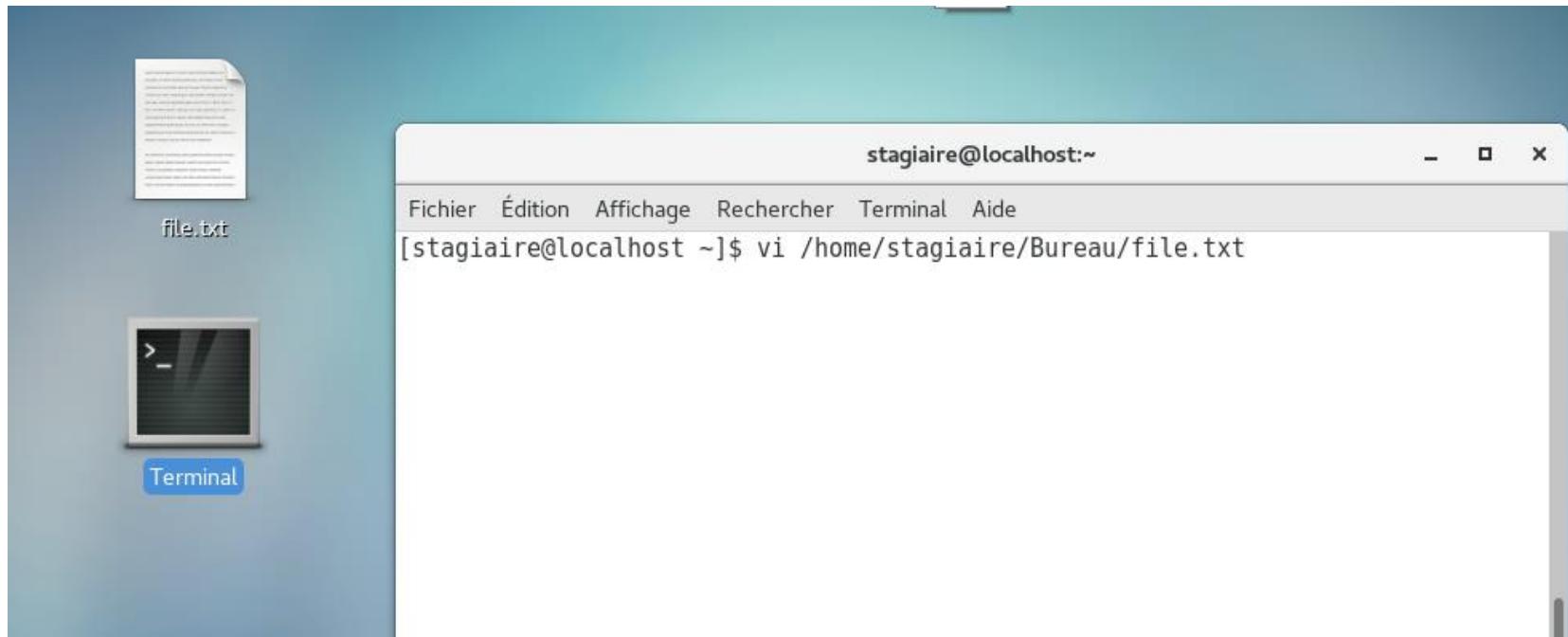


A screenshot of a terminal window titled "stagiaire@localhost:~". The window has a standard OS X-style title bar with minimize, maximize, and close buttons. Below the title bar is a menu bar with French labels: Fichier, Édition, Affichage, Rechercher, Terminal, and Aide. The main pane of the terminal displays the contents of a file named "file.txt":

```
Bonjour tout le monde
J espere que vous etes en pleine forme !
/home/stagiaire/Bureau/file.txt (END)
```

# Edition des fichiers texte

- La commande **vi** permet d'ouvrir un fichier texte à l'intérieur du terminal shell, **et offre également la possibilité de modifier le fichier**



# Edition des fichiers texte

- Pour commencer à modifier votre fichier avec la commande **vi**, il faut commencer par taper sur la touche ‘a’

The screenshot shows a terminal window with the title bar "stagiaire@localhost:~". The menu bar includes "Fichier", "Édition", "Affichage", "Rechercher", "Terminal", and "Aide". The main area displays the following text:

```
Bonjour tout le monde
J espere que vous etes en pleine forme !
Maintenant, je peux modifier le texte avec la commande vi
Pour pouvoir modifier votre texte, n'oubliez pas d'appuyer sur la touche 'a'
Amusez vous donc !
```

Below the text, there is a vertical scroll bar with several small blue tick marks. At the bottom of the window, there is a status bar with the text "-- INSERTION --" on the left, "5,19" in the center, and "Tout" on the right.

# Edition des fichiers texte

- Pour sauvegarder et fermer le fichier. Appuyer sur la touche ‘echap’ , puis saisir sur le clavier ‘:wq!’ puis taper sur la touche ‘entrée’



The screenshot shows a terminal window titled "stagiaire@localhost:~". The menu bar includes "Fichier", "Édition", "Affichage", "Rechercher", "Terminal", and "Aide". The main area displays the following text:

```
Bonjour tout le monde
J'espere que vous etes en pleine forme !
Maintenant, je peux modifier le texte avec la commande vi
Pour pouvoir modifier votre texte, n'oubliez pas d'appuyer sur la touche 'a'
Amusez vous donc !
```

A vertical scroll bar is visible on the left side of the terminal window. At the bottom of the screen, the command ":wq!" is typed into the terminal prompt.

# Edition des fichiers texte

- Pour fermer le fichier sans sauvegarder les modifications. Appuyer sur la touche ‘echap’ , puis saisir sur le clavier ‘:q!’ puis taper sur la touche ‘entrée’



The screenshot shows a terminal window titled "stagiaire@localhost:~". The menu bar includes "Fichier", "Édition", "Affichage", "Rechercher", "Terminal", and "Aide". The main text area contains the following French text:

```
Bonjour tout le monde
J'espere que vous etes en pleine forme !
Bon courage dans vos études !
```

At the bottom of the terminal window, the command `:q!` is typed into the input field, indicating the user's intention to quit without saving changes.

# La commande wc

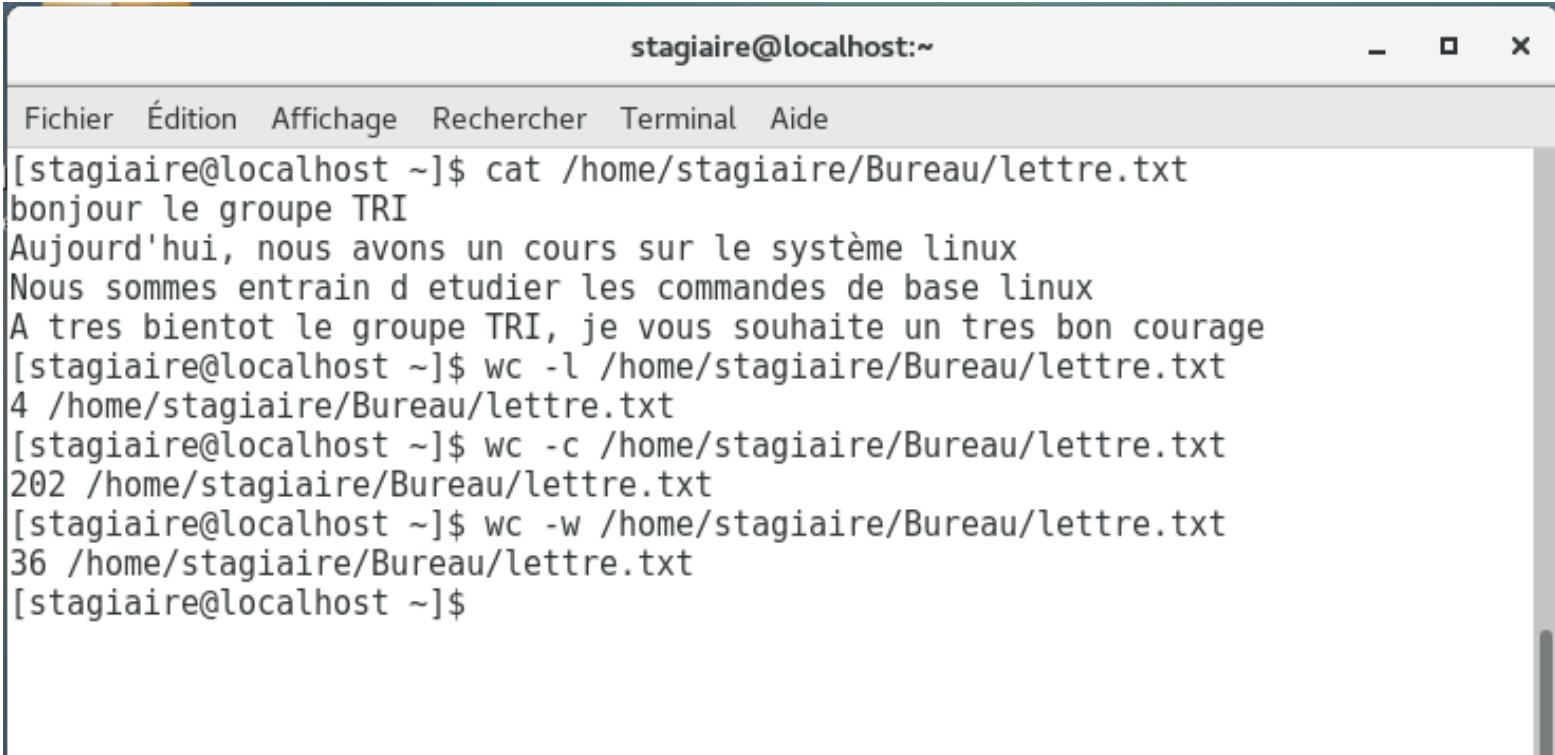
- La commande **wc** permet d'extraire des informations sur les fichiers , comme la taille, nombre de lignes ect...

**wc -l** : afficher le nombre de lignes d'un fichier texte

**wc -c** : afficher la taille en octect d'un fichier texte

**wc -w** : afficher le nombre de mots dans un fichier texte

# La commande wc



The screenshot shows a terminal window titled "stagiaire@localhost:~". The window has a menu bar with "Fichier", "Édition", "Affichage", "Rechercher", "Terminal", and "Aide". The main area of the terminal displays the following text:

```
[stagiaire@localhost ~]$ cat /home/stagiaire/Bureau/lettre.txt
bonjour le groupe TRI
Aujourd'hui, nous avons un cours sur le système linux
Nous sommes entrain d etudier les commandes de base linux
A tres bientot le groupe TRI, je vous souhaite un tres bon courage
[stagiaire@localhost ~]$ wc -l /home/stagiaire/Bureau/lettre.txt
4 /home/stagiaire/Bureau/lettre.txt
[stagiaire@localhost ~]$ wc -c /home/stagiaire/Bureau/lettre.txt
202 /home/stagiaire/Bureau/lettre.txt
[stagiaire@localhost ~]$ wc -w /home/stagiaire/Bureau/lettre.txt
36 /home/stagiaire/Bureau/lettre.txt
[stagiaire@localhost ~]$
```

# La commande grep

- La commande **grep** est une commande qui permet de rechercher une chaîne de caractères (mot , phrase ...) dans un fichier texte
- **grep** affiche l'ensemble des lignes du fichier texte contenant la chaîne de caractères recherchée
- Exemple :

```
grep "TRI" /home/stagiaire/Bureau/lettre.txt
```

# La commande grep

```
stagiaire@localhost:~ - □ ×
Fichier Édition Affichage Rechercher Terminal Aide
[stagiaire@localhost ~]$ cat /home/stagiaire/Bureau/lettre.txt
bonjour le groupe TRI
Aujourd'hui, nous avons un cours sur le système linux
Nous sommes entrain d etudier la commande grep
A tres bientot le groupe TRI, je vous souhaite un tres bon courage
[stagiaire@localhost ~]$ grep "TRI" /home/stagiaire/Bureau/lettre.txt
bonjour le groupe TRI
A tres bientot le groupe TRI, je vous souhaite un tres bon courage
[stagiaire@localhost ~]$ █
```

# Expressions régulières

Voici quelques expressions régulières sous linux

- ^ : début de ligne
- \$ : fin de ligne
- . : un seul caractère quelconque
- \* : une chaîne de caractères quelconque
- [2az] : un caractère égal à 2 , a , ou bien z
- [0-9] : un chiffre allant de 0 jusqu'à 9
- [a-z] : une lettre alphabétique

# La commande grep

- Exercice :

A quoi sert les commandes suivantes :

- 1- **grep** “^a” fichier
- 2- **grep** “r\$” fichier
- 3- **grep** “[0-9]” fichier
- 4- **grep** “^user” fichier
- 5- **grep** “^r..[s2t]” fichier

# La commande grep

- Réponse :

- 1- Rechercher toutes les lignes commençant par a
- 2- Rechercher toutes les lignes se terminant par r
- 3- Rechercher toutes les lignes commençant par un chiffre
- 4- Rechercher toutes les lignes qui commencent par user
- 5- Rechercher toutes les lignes qui commencent par r et dont le Quatrième caractère est soit s , 2 ou bien t

# Les commandes head et tail

- La commande head permet de filtrer les premières lignes d'un fichier texte
- Utilisation :  
**head -n x fichier** : affichage des x premières lignes du fichier.

## Exemples :

```
head -n 4 /etc/group
```

```
stagiaire@localhost:~ - □ ×
Fichier Édition Affichage Rechercher Terminal Aide
[stagiaire@localhost ~]$ head -n 4 /etc/group
root:x:0:
bin:x:1:
daemon:x:2:
sys:x:3:
[stagiaire@localhost ~]$ █
```

# Les commandes head et tail

- La commande tail permet d'afficher des lignes qui se situent en bas d'un fichier texte
- Utilisation :  
**tail -n x fichier** : affichage des x dernières lignes du fichier

## Exemple :

**tail -n 6 /etc/group** : affichage des 6 dernières lignes du fichier /etc/group

# Les commandes head et tail



A screenshot of a terminal window titled "stagiaire@localhost:~". The window has standard Linux-style window controls (minimize, maximize, close) at the top right. A menu bar at the top includes "Fichier", "Édition", "Affichage", "Rechercher", "Terminal", and "Aide". The main area of the terminal shows the command "[stagiaire@localhost ~]\$ tail -n 6 /etc/group" followed by its output:

```
[stagiaire@localhost ~]$ tail -n 6 /etc/group
finances:x:1015:
commerciaux:x:1016:
formateurs:x:1017:
electriciens:x:1018:
user99:x:1019:
dhcpd:x:177:
[stagiaire@localhost ~]$
```

# La commande sort

- La commande sort permet de trier les lignes d'un fichier texte selon l'ordre alphanumérique
- **Utilisation:**  
`sort fichier`
- **Exemple:**  
`sort /etc/group`

# La commande sort

```
stagiaire@localhost:~ - □ ×
Fichier Édition Affichage Rechercher Terminal Aide
[stagiaire@localhost ~]$ cat /home/stagiaire/Bureau/lettre.txt
bonjour le groupe TRI
Aujourd'hui, nous avons un cours sur le système linux
Nous sommes entrain d etudier les commandes de base linux
A tres bientot le groupe TRI, je vous souhaite un tres bon courage
[stagiaire@localhost ~]$
[stagiaire@localhost ~]$
[stagiaire@localhost ~]$ sort /home/stagiaire/Bureau/lettre.txt
A tres bientot le groupe TRI, je vous souhaite un tres bon courage
Aujourd'hui, nous avons un cours sur le système linux
bonjour le groupe TRI
Nous sommes entrain d etudier les commandes de base linux
[stagiaire@localhost ~]$ █
```

# La commande cut

- La commande **cut** est un filtre qui permet d'afficher des zones spécifiques dans un fichier texte.
- Avec la commande **cut** , on spécifie le séparateur entre les zones du texte avec l'option -d, et les colonnes à afficher avec l'option -f
- Dans l'exemple suivant, nous allons travailler sur un fichier qui contient des informations de 4 étudiants. Les informations sont séparées par ":"

# La commande cut

```
stagiaire@localhost:~ - □ ×
Fichier Édition Affichage Rechercher Terminal Aide
[stagiaire@localhost ~]$ cat /home/stagiaire/Bureau/file.txt
Ismaili:ali:TRI:1992
Moktadir:imane:TDI:1990
ELfilali:leila:TMSIR:1995
FAIZ:othmane:TRI:1998
[stagiaire@localhost ~]$
[stagiaire@localhost ~]$
[stagiaire@localhost ~]$ cut -d":" -f2 /home/stagiaire/Bureau/file.txt
ali
imane
leila
othmane
[stagiaire@localhost ~]$ cut -d":" -f1,3 /home/stagiaire/Bureau/file.txt
Ismaili:TRI
Moktadir:TDI
ELfilali:TMSIR
FAIZ:TRI
[stagiaire@localhost ~]$ █
```

# Manuel des commandes

- Chaque commande possède un manuel d'aide
- Le manuel d'aide définit la commande et explique son rôle et le rôle de ses options
- Pour afficher le manuel d'une commande, utiliser la commande **man**
- Exemples :

**man mkdir**

**man wc**

# Les liens symboliques et physiques

- Les liens symboliques ou physiques sont des raccourcies permettant d'accéder plus rapidement à des fichiers ou répertoires sous linux à partir d'autres emplacements sur le disque dur.
- Pour **les fichiers**, il est possible de créer des liens symboliques ou des **liens physiques**.
- Pour **les dossiers**, il n'est possible de créer que des liens symboliques

# Les liens symboliques et physiques

- Un lien physique se comporte comme une copie, car il reste fonctionnel dans le cas où le fichier d'origine est supprimé.
- Un lien symbolique s'écrase dans le cas où le fichier ou le dossier d'origine est supprimé.
- Pour créer un lien physique ou symbolique, on utilise la commande **Ln**

# Les liens symboliques et physiques

## Utilisation :

- Lien symbolique

ln -s source destination

- Lien physique

ln source destination

## Exemple :

ln -s /home/stagiaire/cours/exercices.txt /home/stagiaire/Bureau/exercices.txt

ln /home/stagiaire/examens/linux.pdf /home/stagiaire/Bureau/linux.pdf

# Les liens symboliques et physiques

- Dans cette démonstration nous allons :
  - 1- créer un fichier commandes.txt sur le répertoire /home/stagiaire/cours/linux
  - 2- Ajouter un petit texte sur le fichier commandes.txt
  - 3- créer un lien physique du fichier commandes.txt sur le Bureau
  - 4- créer un lien symbolique du fichier commandes.txt sur le Bureau
  - 5- supprimer le fichier source commandes.txt
  - 6- vérifier l'ouverture du lien physique et du lien symbolique

# Les liens symboliques et physiques

```
[stagiaire@localhost ~]$ mkdir -p /home/stagiaire/cours/linux
[stagiaire@localhost ~]$ touch /home/stagiaire/cours/linux/commandes.txt
[stagiaire@localhost ~]$ vi /home/stagiaire/cours/linux/commandes.txt
[stagiaire@localhost ~]$ cat /home/stagiaire/cours/linux/commandes.txt
Bonjour tout le monde
ceci est le fichier commandes.txt
Je vous souhaite un tres bon courage dans votre cours
[stagiaire@localhost ~]$ ln /home/stagiaire/cours/linux/commandes.txt /home/stagiaire/Bureau/commandes-ph.txt
[stagiaire@localhost ~]$ ln -s /home/stagiaire/cours/linux/commandes.txt /home/stagiaire/Bureau/commandes-sym.txt
[stagiaire@localhost ~]$ cat /home/stagiaire/Bureau/commandes-ph.txt
Bonjour tout le monde
ceci est le fichier commandes.txt
Je vous souhaite un tres bon courage dans votre cours
[stagiaire@localhost ~]$ cat /home/stagiaire/Bureau/commandes-sym.txt
Bonjour tout le monde
ceci est le fichier commandes.txt
Je vous souhaite un tres bon courage dans votre cours
[stagiaire@localhost ~]$ rm -f /home/stagiaire/cours/linux/commandes.txt
[stagiaire@localhost ~]$ cat /home/stagiaire/Bureau/commandes-ph.txt
Bonjour tout le monde
ceci est le fichier commandes.txt
Je vous souhaite un tres bon courage dans votre cours
[stagiaire@localhost ~]$ cat /home/stagiaire/Bureau/commandes-sym.txt
cat: /home/stagiaire/Bureau/commandes-sym.txt: Aucun fichier ou dossier de ce type
[stagiaire@localhost ~]$
```

# La commande find

- La commande **find** permet de rechercher des fichiers ou des dossiers dans un système d'exploitation linux.
- Il est possible d'utiliser des critères de recherche comme le nom , la taille ect...
- Utilisation standard :

```
find domaine_de_recherche -type f ou d –name "nom de fichier ou du dossier recherché"
```

## Exemples :

```
find / -type f -name "exercices.doc"
```

```
find /home/stagiaire -type d -name "Tps_linux"
```

# La commande find

**Dans la démonstration suivante, nous allons:**

- Créer un répertoire selon le chemin suivant ,  
`/home/stagiaire/Bureau/cours/systemes/linux`
- Créer un fichier avec le chemin suivant:  
`/home/stagiaire/Bureau/cours/reseaux/routeurs.doc`
- Rechercher le dossier linux en utilisant la commande **find**
- Rechercher le fichier routeurs.doc en utilisant la commande **find**

# La commande find

```
stagiaire@localhost:~ - □ ×
Fichier Édition Affichage Rechercher Terminal Aide
[stagiaire@localhost ~]$ mkdir -p /home/stagiaire/Bureau/cours/systemes/linux
[stagiaire@localhost ~]$ cd /home/stagiaire/Bureau/cours
[stagiaire@localhost cours]$ mkdir reseaux
[stagiaire@localhost cours]$ cd reseaux
[stagiaire@localhost reseaux]$ touch routeurs.doc
[stagiaire@localhost reseaux]$ cd
[stagiaire@localhost ~]$ find /home/stagiaire -type d -name "linux"
/home/stagiaire/Bureau/cours/systemes/linux
[stagiaire@localhost ~]$ find /home/stagiaire -type f -name "routeurs.doc"
/home/stagiaire/Bureau/cours/reseaux/routeurs.doc
[stagiaire@localhost ~]$
```

# La commande find

- Autres options de la commande find :

**-name** : nom de l'élément recherché

**-type** : type de l'élément

**-size** : taille de l'élément recherché

**-ctime** : date de création de l'élément recherché

**-atime** : date du dernier accès à l'élément

**-mtime** : date de la dernière modification de l'élément recherché

# La commande find

- A l'aide de la commande find , recherchez :
  - 1- Un dossier s'appelant cours
  - 2- Tous les fichiers avec l'extension mp3
  - 3- Tous les fichiers avec une taille égale à 150 koctets
  - 4- Tous les fichiers avec une taille supérieur à 150 koctets
  - 5- Tous les dossiers dont le dernier accès s'est fait dans les dernières 24 heures

# La commande find

## Réponses:

- 1- find / -type d -name "cours"
- 2- find / -type f -name "\*.mp3"
- 3- find / -type f -size 150k
- 4- find / -type f -size +150k
- 5- find / -type d -atime -1

# Gestion des utilisateurs et des groupes

La liste des commandes suivantes représente des outils en ligne de commandes permettant à un administrateur linux de gérer les comptes utilisateurs et les groupes dans une machine linux

**useradd** : ajouter un nouvel utilisateur

ex : useradd amine

**userdel -r** : supprimer un utilisateur existant

ex : userdel -r ahmed

**groupadd** : ajouter un nouveau groupe d'utilisateurs

ex: groupadd TRI

**groupdel** : supprimer un groupe d'utilisateurs existant

ex : groupdel TRI

# Gestion des utilisateurs et des groupes

**passwd** : Créer et modifier le mot de passe d'un utilisateur :

ex : passwd amine

**usermod -g** :Rajouter un utilisateur existant à un groupe existant :

ex : usermod -g TRI amine

**usermod -G** : Rajouter un utilisateur existant à plusieurs groupes existants :

usermod -G techniciens,stagiaires amine

(TRI est un groupe primaire pour amine , car il est le premier groupe qui va accueillir l'utilisateur amine)

(techniciens et stagiaires sont des groupes secondaires)

# Gestion des utilisateurs et des groupes

**usermod -l** :Modifier le nom d'un utilisateur :

ex : usermod -l amine12 amine

( usermod -l <nouveau-nom> <ancien-nom>

**groups** : Afficher les noms des groupes d'un utilisateur:

ex : groups amine

# Gestion des utilisateurs et des groupes

- Dans la démonstration suivante nous allons :
  - Créer un utilisateur nommé **ahmed**
  - Affecter un mot de passe à **ahmed**
  - Créer trois groupes : **TRI – Techniciens – informaticiens**
  - Affecter ahmed à son groupe principal **TRI**
  - Affecter ahmed aux groupes secondaires  
**techniciens et informaticiens**
  - Afficher les groupes de ahmed

# Gestion des utilisateurs et des groupes

```
root@localhost:~ - □ ×
Fichier Édition Affichage Rechercher Terminal Aide
[stagiaire@localhost ~]$ su -
Mot de passe :
Dernière connexion : mercredi 28 mars 2018 à 12:00:15 CEST sur pts/0
[root@localhost ~]# useradd ahmed
[root@localhost ~]# passwd ahmed
Changement de mot de passe pour l'utilisateur ahmed.
Nouveau mot de passe :
MOT DE PASSE INCORRECT : Le mot de passe comporte moins de 8 caractères
Retapez le nouveau mot de passe :
passwd : mise à jour réussie de tous les jetons d'authentification.
[root@localhost ~]# groupadd TRI
[root@localhost ~]# groupadd techniciens
[root@localhost ~]# groupadd informaticiens
[root@localhost ~]# usermod -g TRI ahmed
[root@localhost ~]# usermod -G techniciens,informaticiens ahmed
[root@localhost ~]# groups ahmed
ahmed : TRI techniciens informaticiens
[root@localhost ~]#
```

# Gestion des utilisateurs et des groupes

- Remarques :

1- On peut consulter la liste des utilisateurs créés sur le fichier : **/etc/passwd**

2- La liste des groupes créés est enregistrée dans le fichier : **/etc/group**

3- Chaque utilisateur possède un numéro d'identification **uid** (user identifier)

4- Chaque groupe possède un numéro d'identification **gid** (group identifier)

# Gestion des utilisateurs et des groupes

- Dans cette démonstration nous allons :
  - 1- créer un utilisateur nommé **ali**
  - 2- créer un groupe nommé **techniciens**
  - 3- affecter ali à son groupe principal **techniciens**
  - 4- vérifier l'**uid** de ali dans le fichier /etc/passwd
  - 5- vérifier le **gid** du groupe techniciens dans le fichier /etc/group

# Gestion des utilisateurs et des groupes

```
[stagiaire@localhost ~]$ su -  
Mot de passe :  
Dernière connexion : jeudi 29 mars 2018 à 01:10:58 CEST sur pts/0  
[root@localhost ~]# useradd ali  
[root@localhost ~]# groupadd techniciens  
[root@localhost ~]# usermod -g techniciens ali  
[root@localhost ~]# tail -n 1 /etc/passwd  
ali:x:1002:1024::/home/ali:/bin/bash  
[root@localhost ~]# tail -n 1 /etc/group  
techniciens:x:1024:  
[root@localhost ~]# █
```

Nous remarquons que :  
uid de l'utilisateur ali est : 1002  
gid du groupe techniciens est : 1024

# Gestion des utilisateurs et des groupes

- Question :

En une seule ligne de commandes , créez un utilisateur avec les propriétés suivantes (On suppose que les groupes sont déjà créés ) :

Nom de login	:	user800
uid	:	1020
groupe primaire	:	TRI
groupes secondaires	:	techniciens , informaticiens
dossier personnel	:	/home/user800_2030
date d'expiration	:	2030-12-31

# Gestion des utilisateurs et des groupes

Réponse :

```
useradd -u 1020 -g TRI -G techniciens,informaticiens -e 2030-12-31 -d /home/user800_2030 user800
```

# Les droits d'accès aux fichiers et dossiers

- La commande ls -l permet d'afficher le contenu d'un répertoire avec plusieurs informations détaillées :

**Les droits d'accès – Le nombre des liens physiques – le propriétaire – le groupe propriétaire – la taille en octets – la date de la dernière modification**

- les principaux droits d'accès sont :

**r = Lecture**

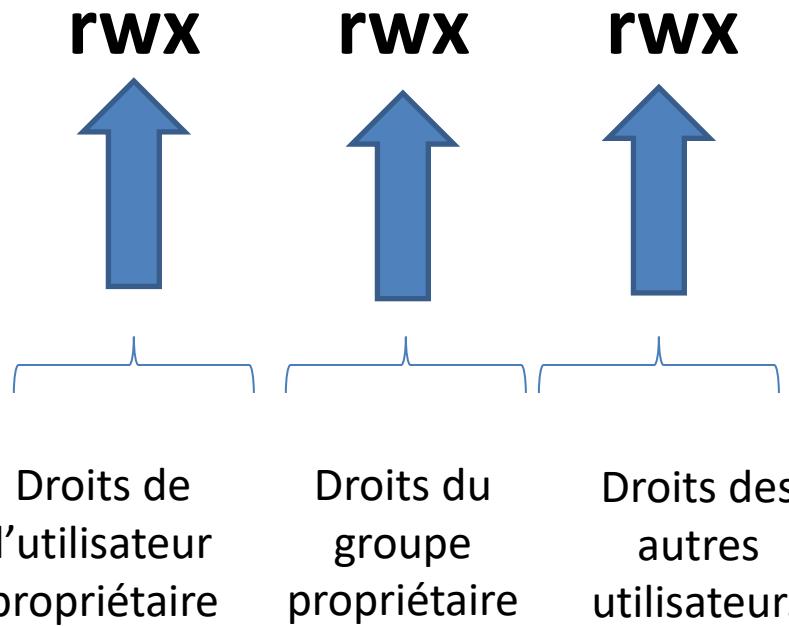
**w = Ecriture**

**x = Accessibilité (dossiers) / Exécution (fichiers exécutables)**

# Les droits d'accès aux fichiers et dossiers

```
[stagiaire@localhost ~]$ ls -l /etc
total 1408
drwxr-xr-x.  3 root root          101 30 oct.  13:15 abrt
-rw-r--r--.  1 root root          16 30 oct.  13:35 adjtime
-rw-r--r--.  1 root root        1518 7 juin   2013 aliases
-rw-r--r--.  1 root root      12288 30 oct. 13:41 aliases.db
drwxr-xr-x.  2 root root          51 30 oct.  13:16 alsa
drwxr-xr-x.  2 root root        4096 30 oct. 13:30 alternatives
-rw-----.  1 root root          541 3 août   2017 anacrontab
-rw-r--r--.  1 root root          55 1 mars   2017 asound.conf
-rw-r--r--.  1 root root           1 5 nov.   2016 at.deny
drwxr-x---.  3 root root          43 30 oct. 13:16 audisp
drwxr-x---.  3 root root          83 30 oct. 13:41 audit
-rw-r--r--.  1 root root      14392 4 août   2017 autofs.conf
-rw-----.  1 root root          232 4 août   2017 autofs_ldap_auth.conf
-rw-r--r--.  1 root root          795 4 août   2017 auto.master
drwxr-xr-x.  2 root root             6 4 août   2017 auto.master.d
-rw-r--r--.  1 root root          524 4 août   2017 auto.misc
-rw xr-xr-x.  1 root root        1260 4 août   2017 auto.net
-rw xr-xr-x.  1 root root          687 4 août   2017 auto.smb
```

# Les droits d'accès aux fichiers et dossiers



# Les droits d'accès aux fichiers et dossiers

- Le **root** et le **propriétaire** peuvent modifier les droits d'accès grâce d'un fichier ou un dossier grâce à la commande : **chmod**
- On peut utiliser la commande **chmod** avec deux méthodes :
  - 1- Méthode octale
  - 2- Méthode symbolique

# Les droits d'accès aux fichiers et dossiers

## 1- Méthode octale

Dans la méthode octale nous utilisons la représentation octale des nouveaux droits à affecter au fichier ou le dossier en question.

### Exemple :

**rw- rw- rw-** => **rwX r-- ---**

chmod 740 fichier/dossier

chmod 740 /home/stagiaire/Bureau/file.txt

# Les droits d'accès aux fichiers et dossiers

---

```
[stagiaire@localhost ~]$ touch /home/stagiaire/Bureau/file.txt
[stagiaire@localhost ~]$ ls -l /home/stagiaire/Bureau
total 12
-rw-rw-r--. 1 stagiaire stagiaire    0 29 mars  20:00 file.txt
-rwxr--r--. 1 stagiaire stagiaire 10410  9 août  2017 org.gnome.Terminal.desktop
[stagiaire@localhost ~]$ chmod 740 /home/stagiaire/Bureau/file.txt
[stagiaire@localhost ~]$ ls -l /home/stagiaire/Bureau
total 12
-rwxr-----. 1 stagiaire stagiaire    0 29 mars  20:00 file.txt
-rwxr--r--. 1 stagiaire stagiaire 10410  9 août  2017 org.gnome.Terminal.desktop
[stagiaire@localhost ~]$ █
```

# Les droits d'accès aux fichiers et dossiers

## 2- Méthode symbolique :

Dans la méthode symbolique nous utilisons les symboles ci-dessous pour exprimer les modifications de droits à configurer.

u= le propriétaire

g= le groupe propriétaire

o= les autres utilisateurs

a= tout le monde

### Exemple :

**rw- rw- rw-** => **rwx r-- ---**

**chmod u+x,g-w,o-rw /home/stagiaire/Bureau/cours.doc**

# Les droits d'accès aux fichiers et dossiers

```
[stagiaire@localhost ~]$ touch /home/stagiaire/Bureau/file.txt
[stagiaire@localhost ~]$ ls -l /home/stagiaire/Bureau
total 12
-rw-rw-r--. 1 stagiaire stagiaire    0 29 mars  20:11 file.txt
-rwxr--r--. 1 stagiaire stagiaire 10410  9 août  2017 org.gnome.Terminal.desktop
[stagiaire@localhost ~]$ chmod 666 /home/stagiaire/Bureau/file.txt
[stagiaire@localhost ~]$ ls -l /home/stagiaire/Bureau
total 12
-rw-rw-rw-. 1 stagiaire stagiaire    0 29 mars  20:11 file.txt
-rwxr--r--. 1 stagiaire stagiaire 10410  9 août  2017 org.gnome.Terminal.desktop
[stagiaire@localhost ~]$ chmod u+x,g-w,o-rw /home/stagiaire/Bureau/file.txt
[stagiaire@localhost ~]$ ls -l /home/stagiaire/Bureau
total 12
-rwxr-----. 1 stagiaire stagiaire    0 29 mars  20:11 file.txt
-rwxr--r--. 1 stagiaire stagiaire 10410  9 août  2017 org.gnome.Terminal.desktop
[stagiaire@localhost ~]$
```

# Les droits d'accès aux fichiers et dossiers

- Autres exemples:

chmod **u=r** fichier/dossier

chmod **u=r,g=rw** fichier/dossier

chmod **u=rw,g=rw,o=r** fichier/dossier

Quel est l'effet de chacune de ces trois commandes ?

# La commande chown

- L'administrateur **root** peut modifier le propriétaire d'un fichier ou un répertoire en utilisant la commande **chown**
- Utilisation:  
**chown** <nom du nouveau propriétaire> <chemin du fichier / dossier >
- Exemple :  
**chown ahmed /home/stagiaire/Bureau/cours.doc**

# La commande chown

```
[stagiaire@localhost ~]$ touch /home/stagiaire/Bureau/cours.doc
[stagiaire@localhost ~]$ ls -l /home/stagiaire/Bureau
total 12
-rw-rw-r--. 1 stagiaire stagiaire    0 29 mars  20:29 cours.doc
-rwxr--r--. 1 stagiaire stagiaire 10410  9 août  2017 org.gnome.Terminal.desktop
[stagiaire@localhost ~]$ su -
Mot de passe :
Dernière connexion : jeudi 29 mars 2018 à 20:27:22 CEST sur pts/2
[root@localhost ~]# useradd ahmed
Création de la boîte à lettres: Le fichier existe
[root@localhost ~]# chown ahmed /home/stagiaire/Bureau/cours.doc
[root@localhost ~]# ls -l /home/stagiaire/Bureau
total 12
-rw-rw-r--. 1 ahmed      stagiaire    0 29 mars  20:29 cours.doc
-rwxr--r--. 1 stagiaire  stagiaire 10410  9 août  2017 org.gnome.Terminal.desktop
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]# █
```

# La commande chgrp

- L'administrateur **root** peut modifier le propriétaire d'un fichier ou un répertoire en utilisant la commande **chgrp**
- **Utilisation:**

**chgrp** <nom du groupe propriétaire> <chemin du fichier / dossier >

- **Exemple :**

**chgrp** TRI /home/stagiaire/Bureau/cours.doc

# La commande chgrp

```
[stagiaire@localhost ~]$ touch /home/stagiaire/Bureau/cours.doc
[stagiaire@localhost ~]$ ls -l /home/stagiaire/Bureau
total 12
-rw-rw-r--. 1 stagiaire stagiaire    0 29 mars  20:40 cours.doc
-rwxr--r--. 1 stagiaire stagiaire 10410  9 août  2017 org.gnome.Terminal.desktop
[stagiaire@localhost ~]$ su -
Mot de passe :
Dernière connexion : jeudi 29 mars 2018 à 20:39:15 CEST sur pts/2
[root@localhost ~]# groupadd TRI
[root@localhost ~]# chgrp TRI /home/stagiaire/Bureau/cours.doc
[root@localhost ~]# ls -l /home/stagiaire/Bureau
total 12
-rw-rw-r--. 1 stagiaire TRI          0 29 mars  20:40 cours.doc
-rwxr--r--. 1 stagiaire stagiaire 10410  9 août  2017 org.gnome.Terminal.desktop
[root@localhost ~]#
```

# La commande umask

- Exercice

Sur votre bureau créer deux dossiers et deux fichiers , les noms seront respectivement dossier1 ,dossier2, file1 et file2

- 1) Quels sont les droits d'accès à dossier1 , dossier2 et file1 et file2 ?
- 2) Appliquer la commande : umask 026
- 3) Créer sur votre bureau deux dossiers et deux fichiers nommés respectivement dossier3, dossier4 , file3 et file4
- 4) Comparer les droits d'accès avant et après l'application de umask 026
- 5) Quel est donc le rôle de la commande **umask** ?

# La commande umask

- umask modifie les droits d'accces par défaut pour les fichiers et les dossiers nouvellement créés; sa valeur par défaut est 022.

exemple :

umask 026

- Les droits par défaut d'un nouveau dossier sont :  
777 - (valeur de umask)  
Les droits par défaut d'un nouveau fichier sont :  
666 - (valeur de umask)

# La commande umask

- Dans cette démonstration nous allons :
- 1- Modifier la valeur de **umask** à **026**
  - 2- Créer un nouveau dossier nommé **cours**
  - 3- Vérifier des droits d'accès au dossier **cours**
  - 4- Créer un nouveau fichier nommé **file.txt**
  - 5- Vérifier les droits d'accès au fichier **file.txt**

# La commande umask

---

```
[stagiaire@localhost ~]$ umask 026
[stagiaire@localhost ~]$ mkdir /home/stagiaire/Bureau/cours
[stagiaire@localhost ~]$ ls -l /home/stagiaire/Bureau
total 12
drwxr-x--x. 2 stagiaire stagiaire    6 30 mars 10:24 cours
-rwxr--r--. 1 stagiaire stagiaire 10410  9 août 2017 org.gnome.Terminal.desktop
[stagiaire@localhost ~]$ touch /home/stagiaire/Bureau/file.txt
[stagiaire@localhost ~]$ ls -l /home/stagiaire/Bureau
total 12
drwxr-x--x. 2 stagiaire stagiaire    6 30 mars 10:24 cours
-rw-r----- 1 stagiaire stagiaire     0 30 mars 10:25 file.txt
-rwxr--r--. 1 stagiaire stagiaire 10410  9 août 2017 org.gnome.Terminal.desktop
[stagiaire@localhost ~]$ █
```

# Les redirections

- Une redirection permet de sauvegarder le résultat d'une commande sur un fichier texte.

## Utilisation :

Il est possible d'effectuer une redirection en utilisant le signe **>** ou **>>**

## exemples :

ls /etc > /home/stagiaire/Bureau/file.txt

pwd > /home/stagiaire/Bureau/file.txt

wc -l /etc/group >> /home/stagiaire/Bureau/file.txt

# Les redirections

- Remarques :
  - > : écrase le contenu du fichier texte avant d'appliquer la redirection
  - >> : effectue la redirection sans écraser le contenu du fichier texte.

# Les redirections

- Dans cette démonstration nous allons :
  - 1- créer un fichier texte file.txt sur le bureau
  - 2- rediriger le résultat de la commande **wc -l /etc/passwd** vers le fichier file.txt
  - 3- rediriger le résultat de la commande **pwd** vers le fichier file.txt
  - 4- rediriger le résultat de la commande **whoami** vers le fichier file.txt
  - 5- rediriger le résultat de la commande **ls -l /home/stagiaire** vers le fichier file.txt

# Les redirections

---

```
[stagiaire@localhost ~]$ touch /home/stagiaire/Bureau/file.txt
[stagiaire@localhost ~]$ wc -l /etc/passwd > /home/stagiaire/Bureau/file.txt
[stagiaire@localhost ~]$ cat /home/stagiaire/Bureau/file.txt
43 /etc/passwd
[stagiaire@localhost ~]$ pwd  >> /home/stagiaire/Bureau/file.txt
[stagiaire@localhost ~]$ whoami >> /home/stagiaire/Bureau/file.txt
[stagiaire@localhost ~]$ cat /home/stagiaire/Bureau/file.txt
43 /etc/passwd
/home/stagiaire
stagiaire
[stagiaire@localhost ~]$ ls -l /home/stagiaire > /home/stagiaire/Bureau/file.txt
[stagiaire@localhost ~]$ cat /home/stagiaire/Bureau/file.txt
total 516
drwxrwxrwx. 2 stagiaire stagiaire    98 28 mars  18:36 Bureau
-rw-rw-r--. 1 stagiaire stagiaire 525484 27 mars  19:50 dhcp-4.2.5-58.el7.centos.x86_64.rpm
drwxr-xr-x. 2 stagiaire stagiaire      6 30 oct. 13:46 Documents
drwxr-xr-x. 2 stagiaire stagiaire   210 27 mars  12:48 Images
drwxr-xr-x. 2 stagiaire stagiaire      6 30 oct. 13:46 Modèles
drwxr-xr-x. 3 stagiaire stagiaire     83 18 déc. 18:09 Musique
drwxr-xr-x. 2 stagiaire stagiaire      6 30 oct. 13:46 Public
drwxr-xr-x. 2 stagiaire stagiaire     69 27 mars  19:50 Téléchargements
drwxr-xr-x. 2 stagiaire stagiaire      6 30 oct. 13:46 Vidéos
[stagiaire@localhost ~]$ █
```

# Les tubes sous linux (les pipes)

- Un pipe est un moyen logiciel qui permet d'exécuter deux ou plusieurs commandes linux en superposition :

**Commande\_1 | commande\_2**

- Lorsqu'on superpose deux commandes linux avec un **pipe**, le résultat de la commande\_1 sera un paramètre de la commande\_2

# Les tubes sous linux (les pipes)

- Exemples :

1- ls /etc | grep "passwd"

2- ls /home | wc -l

3- grep "bonjour" lettre.txt | wc -l

( Sur votre clavier appuyer simultanément sur les touches alt gr et 6 pour écrire ‘|’ )

# Les tubes sous linux (les pipes)

- Exercice :

1- Donner le nombre des utilisateurs dont le nom commence par la lettre ‘u’

```
grep “^u” /etc/passwd | wc -l
```

2- Calculer le nombre des dossiers sur le répertoire /etc

```
ls -l /etc | grep “^d” | wc -l
```

3- donner une ligne d'instruction qui vous permettra d'afficher depuis la ligne 4 jusqu'à la ligne 10 du fichier /etc/passwd

```
head -n 10 /etc/passwd | tail -n 7
```

# Les tubes sous linux (les pipes)

---

```
[stagiaire@localhost ~]$ grep "^u" /etc/passwd | wc -l
1
[stagiaire@localhost ~]$ ls -l /etc | grep "^d" | wc -l
135
[stagiaire@localhost ~]$ head -n 10 /etc/passwd | tail -n 7
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
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
[stagiaire@localhost ~]$ █
```

# La compression des fichiers sous linux

- On peut utiliser les 2 outils en lignes de commandes suivants pour compresser les fichiers sous linux :
  - gzip
  - bzip2

## Compression :

gzip	fichier
bzip2	fichier

## Décompression :

gunzip	fichier.gz
bunzip2	fichier.bz2

# La compression des fichiers sous linux

- Dans la démonstration suivante nous allons :
  - 1- créer un fichier **file.txt** sur le bureau
  - 2- copier le contenu du fichier /etc/group dans le fichier **file.txt**
  - 3- afficher la taille du fichier **file.txt**
  - 4- compresser le fichier **file.txt** avec l'outil gzip
  - 5- afficher la taille du fichier **file.txt.gz** compressé
  - 6- décompresser le fichier **file.txt.gz**
  - 7- compresser le fichier **file.txt** avec l'outil bzip2
  - 8- afficher la taille du fichier **file.bz2**

# La compression des fichiers sous linux

```
stagiaire@localhost:~ - □ ×
Fichier Édition Affichage Rechercher Terminal Aide
[stagiaire@localhost ~]$ touch /home/stagiaire/Bureau/file.txt
[stagiaire@localhost ~]$ cp /etc/group /home/stagiaire/Bureau/file.txt
[stagiaire@localhost ~]$ wc -c /home/stagiaire/Bureau/file.txt
1300 /home/stagiaire/Bureau/file.txt
[stagiaire@localhost ~]$ gzip /home/stagiaire/Bureau/file.txt
[stagiaire@localhost ~]$ ls /home/stagiaire/Bureau
file.txt.gz org.gnome.Terminal.desktop
[stagiaire@localhost ~]$ wc -c /home/stagiaire/Bureau/file.txt.gz
680 /home/stagiaire/Bureau/file.txt.gz
[stagiaire@localhost ~]$ gunzip /home/stagiaire/Bureau/file.txt.gz
[stagiaire@localhost ~]$ ls /home/stagiaire/Bureau
file.txt org.gnome.Terminal.desktop
[stagiaire@localhost ~]$ bzip2 /home/stagiaire/Bureau/file.txt
[stagiaire@localhost ~]$ ls /home/stagiaire/Bureau
file.txt.bz2 org.gnome.Terminal.desktop
[stagiaire@localhost ~]$ wc -c /home/stagiaire/Bureau/file.txt.bz2
631 /home/stagiaire/Bureau/file.txt.bz2
[stagiaire@localhost ~]$ bunzip2 /home/stagiaire/Bureau/file.txt.bz2
[stagiaire@localhost ~]$ ls /home/stagiaire/Bureau
file.txt org.gnome.Terminal.desktop
[stagiaire@localhost ~]$ █
```

# L'archivage avec la commande tar

- Grace à la commande **tar** , Il est possible d'archiver un ensemble de fichiers et de dossiers sous linux dans une archive
- Utilisation :

**tar -cvf <nom de l'archive> < chemin des répertoires et fichiers à archiver >**

- Exemple :

**tar -cvf cours.tar cours\_linux cours\_réseaux exercices\_linux.doc**

# L'archivage avec la commande tar

On peut utiliser d'autres options avec la commande tar

- **tar -xvf cours.tar** : permet d'extraire le contenu de l'archive cours.tar
- **tar -tvf cours.tar** : permet de consulter le contenu de l'archive cours.tar , sans extraction
- **tar -czf** : permet de créer une archive compressé ( ayant une taille réduite)

# L'archivage avec la commande tar

Dans la démonstration suivante nous allons :

- Créer deux répertoires **cours\_linux** et **cours\_reseaux** sur le bureau
- Créer un fichier **exercices\_linux.doc** sur le bureau
- Archiver ces trois éléments dans une archive nommée **cours.tar**
- Consulter le contenu de l'archive cours.tar
- Extraire le contenu de l'archive cours.tar

# L'archivage avec la commande tar

```
[stagiaire@localhost ~]$ cd /home/stagiaire/Bureau/
[stagiaire@localhost Bureau]$ mkdir cours_linux
[stagiaire@localhost Bureau]$ mkdir cours_réseaux
[stagiaire@localhost Bureau]$ touch exercices_linux.doc
[stagiaire@localhost Bureau]$ tar -cvf cours.tar cours_linux cours_réseaux exercices_linux.doc
cours_linux/
cours_réseaux/
exercices_linux.doc
[stagiaire@localhost Bureau]$ ls
cours_linux  cours_réseaux  cours.tar  exercices_linux.doc  org.gnome.Terminal.desktop
[stagiaire@localhost Bureau]$ tar -tvf cours.tar
drwxrwxr-x stagiaire/stagiaire 0 2018-03-28 15:38 cours_linux/
drwxrwxr-x stagiaire/stagiaire 0 2018-03-28 15:38 cours_réseaux/
-rw-rw-r-- stagiaire/stagiaire 0 2018-03-28 15:38 exercices_linux.doc
[stagiaire@localhost Bureau]$ rm -rf cours_linux
[stagiaire@localhost Bureau]$ rm -rf cours_réseaux
[stagiaire@localhost Bureau]$ rm -f exercices_linux.doc
[stagiaire@localhost Bureau]$ ls
cours.tar  org.gnome.Terminal.desktop
[stagiaire@localhost Bureau]$ tar -xvf cours.tar
cours_linux/
cours_réseaux/
exercices_linux.doc
[stagiaire@localhost Bureau]$ ls
cours_linux  cours_réseaux  cours.tar  exercices_linux.doc  org.gnome.Terminal.desktop
[stagiaire@localhost Bureau]$ █
```

# Les processus sous linux

- Un processus est un programme en cours d'exécution sur votre machine linux.
- Pour afficher les processus en cours d'exécution sur linux , on peut utiliser la commande **ps -aux**
- La commande **ps -aux** affiche un tableau qui recense plusieurs informations sur les processus

# Les processus sous linux

```
[stagiaire@localhost ~]$ ps -aux
```

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	1	0.9	0.5	128164	5796	?	Ss	11:55	0:02	/usr/lib/system
root	2	0.0	0.0	0	0	?	S	11:55	0:00	[kthreadd]
root	3	0.0	0.0	0	0	?	S	11:55	0:00	[ksoftirqd/0]
root	4	0.0	0.0	0	0	?	R	11:55	0:00	[kworker/0:0]
root	5	0.0	0.0	0	0	?	S<	11:55	0:00	[kworker/0:0H]
root	6	0.0	0.0	0	0	?	S	11:55	0:00	[kworker/u2:0]
root	7	0.0	0.0	0	0	?	S	11:55	0:00	[migration/0]
root	8	0.0	0.0	0	0	?	S	11:55	0:00	[rcu_bh]
root	9	0.1	0.0	0	0	?	R	11:55	0:00	[rcu_sched]
root	10	0.0	0.0	0	0	?	S	11:55	0:00	[watchdog/0]
root	12	0.0	0.0	0	0	?	S	11:55	0:00	[kdevtmpfs]
root	13	0.0	0.0	0	0	?	S<	11:55	0:00	[netns]
root	14	0.0	0.0	0	0	?	S	11:55	0:00	[khungtaskd]
root	15	0.0	0.0	0	0	?	S<	11:55	0:00	[writeback]
root	16	0.0	0.0	0	0	?	S<	11:55	0:00	[kintegrityd]
root	17	0.0	0.0	0	0	?	S<	11:55	0:00	[bioset]
root	18	0.0	0.0	0	0	?	S<	11:55	0:00	[kblockd]

# Les processus sous linux

**User** : Le nom de la session source du déclenchement du processus

**PID** : numéro d'identification du processus

**%MEM**: charge mémoire du processus

**%CPU**: charge processeur du processus

**START** : heure de démarrage du processus

**Command** : nom du processus

# Les processus sous linux

- La commande **top** est un autre moyen qui permet de superviser en temps réel les processus déclenché sur votre machine linux

```
%Cpu(s): 18,4 us, 2,0 sy, 0,0 ni, 77,6 id, 1,7 wa, 0,0 hi, 0,3 si, 0,0 st
KiB Mem : 1016552 total, 83260 free, 704416 used, 228876 buff/cache
KiB Swap: 2097148 total, 2091700 free, 5448 used. 126968 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1729	stagiai+	20	0	1923880	213800	34496	R	16,2	21,0	0:22.03	gnome-shell
1103	root	20	0	289336	29348	5364	R	3,0	2,9	0:04.41	X
2538	stagiai+	20	0	709436	22996	14552	S	0,7	2,3	0:00.68	gnome-terminal-
515	root	20	0	48104	4336	1224	S	0,3	0,4	0:00.30	systemd-udevd
1518	stagiai+	20	0	36080	2432	916	S	0,3	0,2	0:00.26	dbus-daemon
1963	stagiai+	20	0	386604	11364	6384	S	0,3	1,1	0:09.21	vmtoolsd
1	root	20	0	128164	5148	3028	S	0,0	0,5	0:02.51	systemd
2	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kthreadd
3	root	20	0	0	0	0	S	0,0	0,0	0:00.10	ksoftirqd/0
4	root	20	0	0	0	0	S	0,0	0,0	0:00.13	kworker/0:0
5	root	0	-20	0	0	0	S	0,0	0,0	0:00.00	kworker/0:0H
6	root	20	0	0	0	0	S	0,0	0,0	0:00.01	kworker/u2:0
7	root	rt	0	0	0	0	S	0,0	0,0	0:00.00	migration/0
8	root	20	0	0	0	0	S	0,0	0,0	0:00.00	rcu_bh

# Les processus sous linux

- La commande **kill -9** permet d' arrêter l'activité d'un processus.
- Pour arrêter un processus , il faut tout d'abord Connaitre son pid grace à la commande ps –aux

Exemple :

**kill -9 168**

# Les processus sous linux

```
root      2615  0.0  0.0 125332   896 ?          Ss  12:01  0:00 /usr/sbin/anacron -s
root      2643  0.0  0.0     0   0 ?          S    12:03  0:00 [kworker/0:2]
root      2751  0.0  0.0     0   0 ?          S    12:11  0:00 [kworker/0:3]
root      2808  0.0  0.0     0   0 ?          S    12:16  0:00 [kworker/0:0]
stagiai+ 2848  6.3 14.0 2054616 143240 ?          Sl  12:17  0:07 /usr/lib64/firefox/firefox
stagiai+ 2891  0.0  0.1 154564  1404 ?          S    12:17  0:00 /usr/libexec/gconfd-2
root      2949  0.0  0.0     0   0 ?          S    12:17  0:00 [kworker/u2:2]
stagiai+ 2966  1.0  5.3 1732968 53952 ?          Sl  12:17  0:01 /usr/lib64/firefox/plugin-container -greomni /usr/lib64/firefox/omni.ja -appomni /us
stagiai+ 3029  0.9  2.0 709164  21056 ?          Sl  12:18  0:00 /usr/libexec/gnome-terminal-server
stagiai+ 3035  0.0  0.0   8488   708 ?          S    12:18  0:00 gnome-pty-helper
stagiai+ 3036  0.1  0.2 116308  2844 pts/0          Ss  12:18  0:00 bash
root      3054  0.0  0.6 341496  6496 ?          Sl  12:18  0:00 /usr/sbin/abrt-dbus -t133
root      3086  0.0  0.0 107904   608 ?          S    12:18  0:00 sleep 60
root      3107  0.1  0.1 113128  1476 ?          SN  12:19  0:00 /bin/bash /bin/run-parts /etc/cron.daily
root      3128  0.4  0.0     0   0 ?          R    12:19  0:00 [kworker/0:1]
root      3134  0.0  0.1 113128  1376 ?          SN  12:19  0:00 /bin/sh /etc/cron.daily/mlocate
root      3135  0.0  0.0 113492   968 ?          SN  12:19  0:00 awk -v progrname=/etc/cron.daily/mlocate progrname { ???? print progrname ":\n" ?????
root      3140  1.3  0.0 107932   864 ?          RN  12:19  0:00 /usr/bin/updatedb -f sysfs bdev proc cgroup tmpfs debugfs secu
stagiai+ 3148  0.0  0.1 153148  1848 pts/0          R+  12:19  0:00 ps -aux
[stagiaire@localhost ~]$ kill -9 2848
[stagiaire@localhost ~]$ █
```

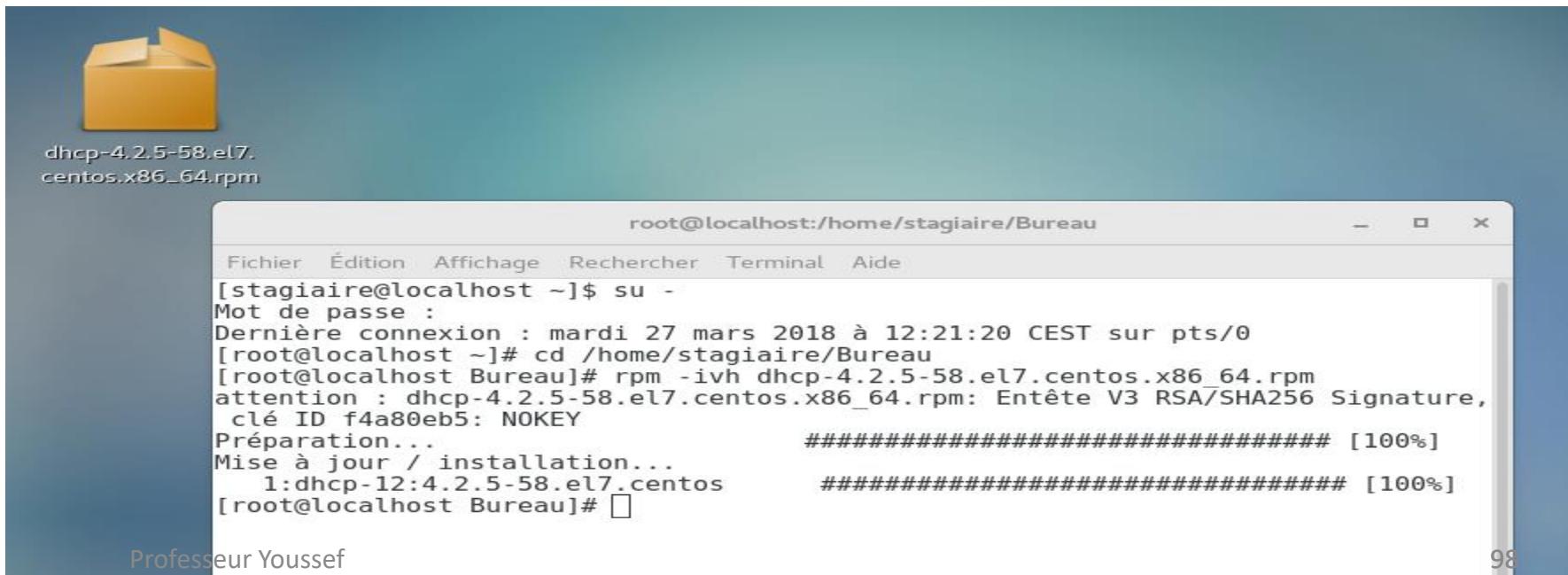
L'exécution de la commande **ps -aux** nous permet de remarquer que dans cette machine, 2848 est le pid de l'application firefox  
Pour pouvoir fermer l'application firefox , on va procéder à utiliser la Commande : **kill -9 2848**

# Installation des packages RPM

- Un package rpm est un fichier qui permet d'installer une nouvelle application, service ... sur votre machine linux
- Les packages rpm constituent un moyen efficace et sécurisé pour installer de nouveaux programmes sur certaines distributions linux comme : RedHat, CentOs, Fedora ...
- Un package rpm est un fichier avec l'extension .rpm
- Les fichiers rpm sont téléchargeables depuis plusieurs sites web sur internet comme : <http://fr.rpmfind.net>

# Installation des packages RPM

Dans cette démo, nous allons installer un package rpm permettant d'ajouter la fonctionnalité de serveur DHCP sur notre machine linux



A screenshot of a Linux desktop environment. On the desktop, there is a yellow folder icon labeled "dhcp-4.2.5-58.el7.centos.x86\_64.rpm". Below the icon, the file name is displayed. In the bottom right corner, there is a terminal window titled "root@localhost:/home/stagiaire/Bureau". The terminal window shows the following command being run:

```
root@localhost ~]$ su -  
Mot de passe :  
Dernière connexion : mardi 27 mars 2018 à 12:21:20 CEST sur pts/0  
[root@localhost ~]# cd /home/stagiaire/Bureau  
[root@localhost Bureau]# rpm -ivh dhcp-4.2.5-58.el7.centos.x86_64.rpm  
attention : dhcp-4.2.5-58.el7.centos.x86_64.rpm: Entête V3 RSA/SHA256 Signature,  
clé ID f4a80eb5: NOKEY  
Préparation... #### [100%]  
Mise à jour / installation... 1:dhcp-12:4.2.5-58.el7.centos #### [100%]  
[root@localhost Bureau]#
```

The terminal window has a standard Linux-style menu bar at the top.

# Installation des packages RPM

- On peut utiliser les commandes suivantes pour gérer les packages rpm sur une machine linux

**rpm -ivh** : installer package rpm

**rpm -Uvh** : mettre à jour un package rpm

**rpm -e** : désinstaller un package rpm

**rpm -q** : vérifier l'existence d'un package sur le machine

**rpm -qa** : afficher la liste de tous les packages installés sur la machine linux

# Installation des packages RPM

```
root@localhost:~ - □ X
Fichier Édition Affichage Rechercher Terminal Aide
[stagiaire@localhost ~]$ rpm -q dhcp
dhcp-4.2.5-58.el7.centos.x86_64
[stagiaire@localhost ~]$ su -
Mot de passe :
Dernière connexion : mardi 27 mars 2018 à 19:51:49 CEST sur pts/0
[root@localhost ~]# rpm -e dhcp
[root@localhost ~]# rpm -q dhcp
le paquet dhcp n'est pas installé
[root@localhost ~]#
```

# Installation des packages RPM

- Il est possible d'installer les packages rpm en ligne , si le serveur est connecté à internet, grâce à la commande **yum install**

```
[stagiaire@localhost ~]$ su -
Mot de passe :
Dernière connexion : mardi 27 mars 2018 à 20:03:47 CEST sur pts/0
[root@localhost ~]# yum install dhcp
Modules complémentaires chargés : fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: mirrors.coreix.net
 * extras: mirrors.coreix.net
 * updates: mirrors.coreix.net
Résolution des dépendances
--> Lancement de la transaction de test
-->> Le paquet dhcp.x86_64 12:4.2.5-58.el7.centos.3 sera installé
-->> Résolution des dépendances terminée

Dépendances résolues
=====
Package           Architecture      Version       Dépôt      Taille
=====
Installation :
  dhcp            x86_64          12:4.2.5-58.el7.centos.3      updates      514 k

Résumé de la transaction
=====
Installation 1 Paquet

Taille totale des téléchargements : 514 k
Taille d'installation : 1.4 M
Is this ok [y/d/N]: y
Downloading packages:
dhcp-4.2.5-58.el7.centos.3.x86_64.rpm | 514 kB  00:00:02
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
```

# Le montage des périphériques

- La commande **mount** permet de monter les périphériques ( clé USB, CD , DVD ...) **manuellement** sur votre machine linux
- Le montage manuelle des périphérique s'effectue à l'aide des fichiers spéciaux de périphériques localisés sur le répertoire /dev
- Avec la commande mount, on peut créer un point de montage d'un périphérique sur le système de fichiers de la machine linux

# Le montage des périphériques

- Utilisation de mount :

**mount <fichier spécial du périphérique> <chemin du point de montage>**

Exemple :

**mount /dev/sdb1 /mnt/USB**

On peut spécifier le système de fichiers du périphérique avec l'option –t

Exemple:

**mount -t vfat /dev/sdb1 /mnt/USB**

# Le montage des périphériques

- Dans la démonstration suivante, nous allons :
    - 1- Créer un dossier **mnt\_usb** sur le Bureau , qui sera le point de montage de la clé USB
    - 2- insérer une clé USB sur notre machine linux
    - 3- localiser le fichier spécial de la clé USB sur le répertoire **/dev**
    - 4- montrer manuellement le contenu de la clé USB sur le dossier **mnt\_usb** créé sur le Bureau
- Le dossier **mnt-usb** va contenir les fichiers et les dossiers qui sont enregistrés sur la clé USB
- 5- Afficher le contenu du dossier **mnt\_usb**
  - 6- Démonter manuellement la clé USB

# Le montage des périphériques

```
[stagiaire@localhost ~]$ cd /home/stagiaire/Bureau
```

```
[stagiaire@localhost Bureau]$ mkdir mnt-usb
```

```
[stagiaire@localhost Bureau]$ su -
```

Mot de passe :

Dernière connexion : mercredi 28 mars 2018 à 13:13:00 CEST sur pts/1

```
[root@localhost ~]# mount /dev/sdb1 /home/stagiaire/Bureau/mnt-usb
```

```
[root@localhost ~]# ls /home/stagiaire/Bureau/mnt-usb
```

4-PE Dév OS Web Mob.docx

AVIS AUX STAGIAIRES.docx

System Volume Information

5.1.2.12 Packet Tracer - Determining the DR and BDR Instructions.pdf

Passage-du-MCD-au-MLD.pdf

tp6.pdf

```
[root@localhost ~]# umount /home/stagiaire/Bureau/mnt-usb
```

```
[root@localhost ~]# ls /home/stagiaire/Bureau/mnt-usb
```

```
[root@localhost ~]#
```

```
[root@localhost ~]#
```

# Le montage des périphériques

## Autres exemples:

Pour monter manuellement un cd-rom sur votre machine linux, on peut utiliser la commande suivante:

```
mount -t iso9660 /dev/cdrom /mnt/mnt-cd
```

**iso9660**: c'est le système de fichiers des cd-rom

**/dev/cdrom** : il s'agit du fichier spécial du lecteur cd-rom

**mnt-cd** : un point de montage du cd-rom

**YouTube :**

<https://www.youtube.com/professeuryoussef1>

