

La commande apt-get

- ❖ **apt-get update** : Mettre à jour le cache des paquets
- ❖ **apt-cache search votrepaquet** : Rechercher un paquet
- ❖ **apt-get install nompaquet** : installer un paquet
- ❖ **apt-get remove nompaquet** : désinstaller un paquet. Le paquet sera alors désinstallé de votre ordinateur. Toutefois, cela ne supprime pas les dépendances du paquet devenues inutiles. Pour demander à apt-get de supprimer **aussi** les dépendances inutiles, on utilise **autoremove**.
- ❖ **apt-get upgrade** : met à jour tous les paquets installés

Introduction aux système Linux

Créer/extrire une archive

gzip : C'est le plus connu et le plus utilisé ;

bzip2 : Il est un peu moins fréquemment utilisé. Il compresse mieux mais plus lentement que gzip.

Exemple :

gzip fichier compressé un fichier au format .gz

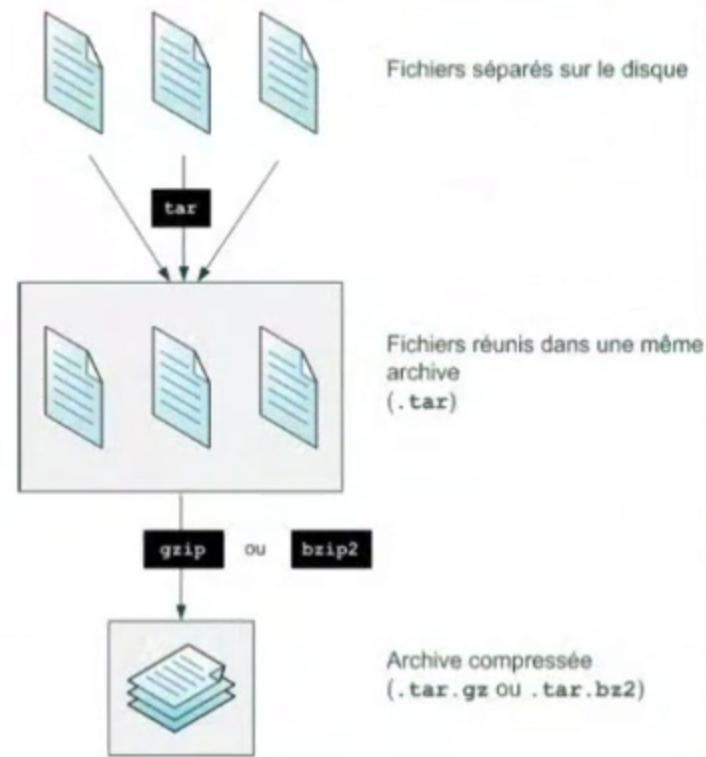
gunzip fichier décompressé un fichier au format .gz

-zcvf : Archiver et compresser en gzip

-jcvf : Archiver et compresser en bzip2

tar -zcvf projet.tar projet/* : crée et compressé une archive contenant le contenu du répertoire projet et nommée projet.tar.gz

tar -zxvf projet.tar.gz : extrait le contenu de l'archive compressée nommée projet.tar.gz



Créer/extraire une archive

tar [options] [archive-file] [fichier ou rép to be archived]

- c pour créer une archive
- x Extraire archive
- r Ajoute fichier à archive existante
- t Affiche la liste des fichiers
- v Permet d'obtenir une description du contenu archivé (le détail d'opération)
- f Spécifier un nom pour l'archive (assembler l'archive dans un fichier)
- p Respect des permissions

Exemples:

tar -cvf projet.tar projet/* : Crée une archive contenant le contenu du répertoire projet et nommée projet.tar

tar -xf projet.tar: Extrait le contenu d'archive nommée projet.tar

Les droits d'accès

- ❖ Lors de la création d'un fichier ou d'un répertoire et qu'on regarde ensuite leurs droits, on obtient généralement rw-r-r{ (644) pour un fichier et rwxr-xr-x (755) pour un répertoire.
- ❖ Ces valeurs sont contrôlées par un masque, lui-même modifiable par la commande **umask**. La commande prend comme paramètre une valeur octale qui sera soustraite aux droits d'accès maximum.
- ❖ Par défaut, tous les fichiers sont créés avec les droits 666 (rw-rw-rw) et les répertoires avec les droits 777 (rwxrwxrwx), puis le masque est appliqué
- ❖ Sur la plupart des Unix, le masque par défaut est 022, soit ----w--w-. Pour obtenir cette valeur, on tape **umask** sans paramètre.

Introduction aux système Linux

Les droits d'accès

- Les fichiers possèdent un certain nombre d'attributs qui définissent les autorisations d'accès.

r autorisation à lire : read

w autorisation à écrire : write

x autorisation à l'exécution : execute

- Ces attributs sont groupés en 3 groupes (3 types utilisateurs)

	rwx	r-x	r--
User	u	Group	g
Other			o

- ◆ **chmod** : modifier les droits d'accès d'un fichier (ex: **chmod 754 toto.txt**)

u = user (propriétaire)

g = group (groupe)

o = other (autres)

+ signifie : « Ajouter le droit »

- signifie : « Supprimer le droit »

Exemples:

chmod +x toto.txt ; chmod g+w,o-w toto.txt

Droit	Chiffre
r	4
w	2
x	1

Le paramètre -R existe aussi pour chmod. Si vous affectez des droits sur un dossier avec -R, tous ses fichiers et sous-dossiers récupéreront le même droit.

Gestion des utilisateurs et des groupes

- ❖ Le fichier **/etc/passwd** contient les informations sur les utilisateurs décomposés en **7 champs** séparés par ":" c'est :
 - Le nom du compte de l'utilisateur
 - A l'origine contenait le mot de passe de l'utilisateur. Le x indique que les mots de passe sont dans /etc/shadow
 - L'entier qui identifie l'utilisateur pour le système d'exploitation (UID=User ID, identifiant utilisateur)
 - L'entier qui identifie le groupe de l'utilisateur (GID=Group ID, identifiant de groupe)
 - Le commentaire dans lequel on peut retrouver des informations sur l'utilisateur ou simplement son nom réel
 - Le répertoire de connexion qui est celui dans lequel il se trouve après s'être connecté au système
 - La commande qui est exécutée après connexion au système (c'est fréquemment un interpréteur de commandes)

Gestion des utilisateurs et des groupes

- ❖ Le fichier **/etc/passwd** contient les informations sur les utilisateurs décomposées en **7 champs** séparés par ":" c'est :
 - Le nom du compte de l'utilisateur
 - A l'origine contenait le mot de passe de l'utilisateur. Le x indique que les mots de passe sont dans /etc/shadow
 - L'entier qui identifie l'utilisateur pour le système d'exploitation (UID=User ID, identifiant utilisateur)
 - L'entier qui identifie le groupe de l'utilisateur (GID=Group ID, identifiant de groupe)
 - Le commentaire dans lequel on peut retrouver des informations sur l'utilisateur ou simplement son nom réel
 - Le répertoire de connexion qui est celui dans lequel il se trouve après s'être connecté au système
 - La commande qui est exécutée après connexion au système (c'est fréquemment un interpréteur de commandes)

Gestion des utilisateurs et des groupes

- ❖ Le fichier **/etc/shadow** contient le mot de passe chiffré, il est décomposé en **9 champs** séparés par « : »:
 - Le nom du compte de l'utilisateur
 - Le mot de passe chiffré.
 - Le nombre de jours depuis le dernier changement du mot de passe.
 - Le nombre de jours avant que le mot de passe ne puisse être changé .
 - Le nombre de jours après lesquels le mot de passe doit être change (99999 indique que l'utilisateur peut garder son mot de passe inchangé)
 - Le nombre de jours pour avertir l'utilisateur qu'un mot de passe ne va plus être valable
 - Le nombre de jours avant de désactiver le compte après expiration du mot de passe
 - Le nombre de jours pendant lesquels un compte a été désactivé
 - Un champ réservé pour une utilisation future possible

Gestion des utilisateurs et des groupes

Le fichier **/etc/group** contient la liste des utilisateurs appartenant aux différents groupes. Il se compose de **4 champs** séparés par ":" :

- Le nom du groupe
- Un champ de mot de passe de groupe (peu utilisé)
- L'entier qui identifie le groupe
- Les membres du groupe

Gestion des utilisateurs et des groupes

❖ **chown** : changer le propriétaire d'un fichier (l'utilisateur)

La commande chown, qui doit être utilisée **en tant que root**, attend deux paramètres au moins : Le nom du nouveau propriétaire et le nom du fichier à modifier. (exemple: chown ali rapport.txt)

❖ **chgrp** : changer le groupe propriétaire d'un fichier.

❖ chown peut aussi changer le groupe propriétaire d'un fichier

ex: **Chown user1:groupe1 rapport.txt** . Il suffit de séparer par un symbole deux points (« : ») le nom du nouvel utilisateur (à gauche) et le nom du nouveau groupe (à droite).

Exercice :

- 1) Crée deux groupes d'utilisateurs : mid1 et mid2
- 2) Crée et affecté les deux utilisateurs Ali et ahmed au groupe mid1
- 3) Crée et affecté les deux utilisateurs user1 et user2 au groupe mid2
- 4) Affecter les utilisateurs ahmed et ali au groupe mid2 et supprimer le groupe mid1
- 5) Supprimer tous les utilisateur qui appartiennent au roupe mid2.

Manipulation de répertoires

cd : permet le positionnement sur un répertoire. syntaxe : **cd** [répertoire]
cd tout seul vous repositionne dans votre home.

Chemin relatif: Un chemin relatif est un chemin qui dépend du dossier dans lequel vous vous trouvez.

Chemin absolu: Contrairement aux chemins relatifs, les chemins absolus fonctionnent quel que soit le dossier dans lequel on se trouve. il commence toujours par la racine

mkdir : Permet la création d'un ou plusieurs répertoires.

mkdir -p: Permet la création des répertoire intermédiaires.

rmdir : Permet la suppression d'un ou plusieurs répertoires

Manipulation de fichiers

❖ Visualiser le contenu d'un fichier ?

➤ **cat** [option] [chemin vers le fichier] affiche le contenu d'un fichier

Exemples :

\$ **cat** fichier1 : Affiche le contenu du fichier1

\$ **cat** fichier1 > fichier2 : Ecrit le contenu du fichier1 dans fichier2

➤ **less** : Afficher le fichier page par page

➤ **head**: Afficher le début du fichier. Ex: **head -n 3 fichier1**: affiche les 3 premières lignes

➤ **tail**: afficher la fin du fichier. Ex: **tail -n 3 fichier1**: affiche les 3 dernières lignes(-f!)

Exemple:

1) Afficher le contenu du fichier /var/log/syslog. Utiliser les deux commandes (cat, less).

Manipulation de fichiers

- ❖ Obtenir des statistiques sur le contenu d'un fichier ?

wc [option] [chemin vers le fichier] affiche le nombre de mots / lignes / caractères d'un fichier

Exemples :

\$ wc -l toto : Affiche le nombre de lignes du fichier toto

\$ wc -c toto affiche le nombre de caractères du fichier toto

\$ ls | wc -l

: Affiche le nombre de fichiers dans le répertoire courant.

..

Manipulation de fichiers

- ❖ Crée un fichier

touch: Crée un fichier. La commande touch est à la base faite pour modifier la date de dernière modification d'un fichier.

- ❖ Copie un fichier

cp [option] [chemin vers fichier source] [chemin vers fichier destination]:

copie un fichier source en le renommant si le chemin du fichier destination contient un nom de fichier

Exemples :

\$ **cp fich1 /tmp/** copie le fich1 dans /tmp

\$ **cp toto /tmp/fich1** copie le fichier local toto dans /tmp en le nommant fich1

L'option **-R** : nous permettre de copier un dossier, ainsi que tous les sous-dossiers et fichiers qu'il contient.

Manipulation de fichiers

- ❖ Déplacer ou renommé un fichier

mv [option] [chemin vers fichier source] [chemin vers fichier destination]

Exemples :

\$ **mv toto /tmp/** : Déplace le fichier local toto dans /tmp (toujours nomme toto)

\$ **mv toto /tmp/fich1**: Déplace le fichier local toto dans /tmp en le nommant fich1

\$ **mv -i toto /tmp** déplace le fichier toto dans /tmp en prévenant l'utilisateur s'il existe déjà un fichier /tmp/toto

- ❖ Supprimer un fichier ?

rm [option] [chemin vers fichier] supprime un fichier

Exemples :

\$ **rm toto** : supprime le fichier toto I

\$ **rm -i toto**: supprime le fichier toto en demandant confirmation à l'utilisateur

\$ **rm -f toto***: Forcer la suppression des fichiers dont le nom commence par toto.

\$ **rm -r projet** : efface récursivement le contenu du répertoire projet

Introduction aux système Linux

Manipulation de fichiers

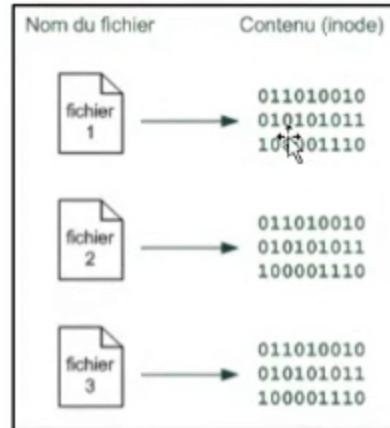
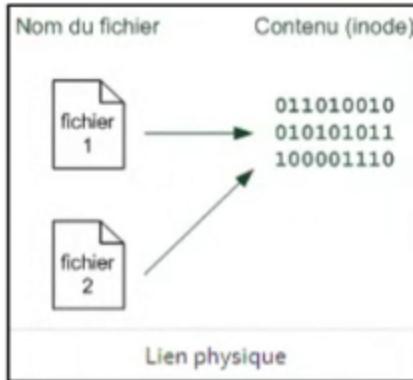
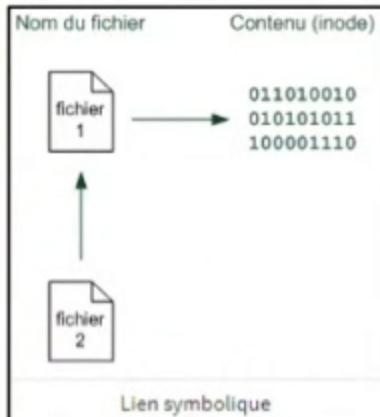
❖ Crée un lien vers un fichier ?

ln [options] fichier source lien création d'un lien. Un lien est un type spécial de fichier qui permet à plusieurs noms de fichiers de faire référence au même fichier sur le disque.

Exemples :

\$ **ln /home/yousef/toto.pdf /home/yousef/projet/toto.pdf**: le fichier toto.pdf du répertoire yousef du répertoire projet est un lien vers le fichier toto.pdf du répertoire yousef (lien physique).

\$ **ln -s /home/yousef/toto.pdf /home/toto/projet/toto.pdf** ! idem avec un lien symbolique



Manipulation de fichiers

- ❖ Retrouver un fichier ?

find [options]: Effectue une recherche a partir des informations données en option

Exemples :

\$ **find -name toto:** cherche, dans le répertoire courant et ses sous-repertoires, un fichier nomme toto

\$ **find /tmp/ -name sys***

\$ **find -size +10M**

\$ **find /tmp/ -type d** cherche uniquement les sous-repertoires du repertoire /tmp

\$ **find -name "*.jpg" -delete** : chercher des fichiers et les supprimer

NB : Avec **-exec**, vous pouvez appeler une commande qui effectuera une action sur chacun des fichiers trouvés.

\$ **find /tmp -type d -exec ls '{}'** : Affiche le contenu des sous-repertoires du repertoire /tmp

Manipulation de fichiers

- ❖ Rechercher un motif dans un fichier ?

grep [options] expression régulière fichier1 ... ! Effectue une recherche a partir d'un motif fourni dans une expression régulière donnée

Exemples :

\$ grep "listeria" /home/Cath/cours/* cherche, dans les fichiers du répertoire cours, des fichiers contenant le motif listeria

\$ grep -n "listeria" /home/Cath/cours/* idem, mais en affichant le numéro de ligne

\$ grep -c "listeria" /home/Cath/cours/* idem, mais en donnant le nombre d'occurences du motif

Gestion des utilisateurs et des groupes

- ❖ Les trois fichiers utilisés dans la gestion des utilisateurs et des groupes sont les fichiers /etc/passwd, /etc/shadow et /etc/group
- ❖ **sudo cmd** : Devenir root un instant, (exécuter la commande tant que super utilisateur)
- ❖ **sudo su** : Devenir root et le rester (\$ -> #). (utiliser exit pour quitter le mode root)
- ❖ **adduser nom** : ajouter un utilisateur nom (exécuter en mode root).
- ❖ **deluser** : supprimer un utilisateur.
- ❖ **passwd utilisateur1** : Changer le mot de passe d'un utilisateur.
- ❖ **addgroup** : Crée un groupe. **delgroup** : supprimer un groupe
- ❖ **usermod**: Permet d'éditer un utilisateur. On a plusieurs paramètres
Exemple: **usermod -g group1 user1** (mettre user1 dans group1).

Introduction aux système Linux

Manipulation de répertoires

ls : Affiche le contenu du répertoire courant (ou du répertoire demandé).

Syntaxe : **ls** [options] [repertoire1] [repertoire2] cette commande accepte un très grand nombre d'options pouvant être combinées.

ls -a : liste toutes les entrées y compris les entrées cachées.

ls -l : liste les entrées et affiche toutes les informations (liste détaillée)

ls -f : liste les entrées et ajoute le type d'élément.

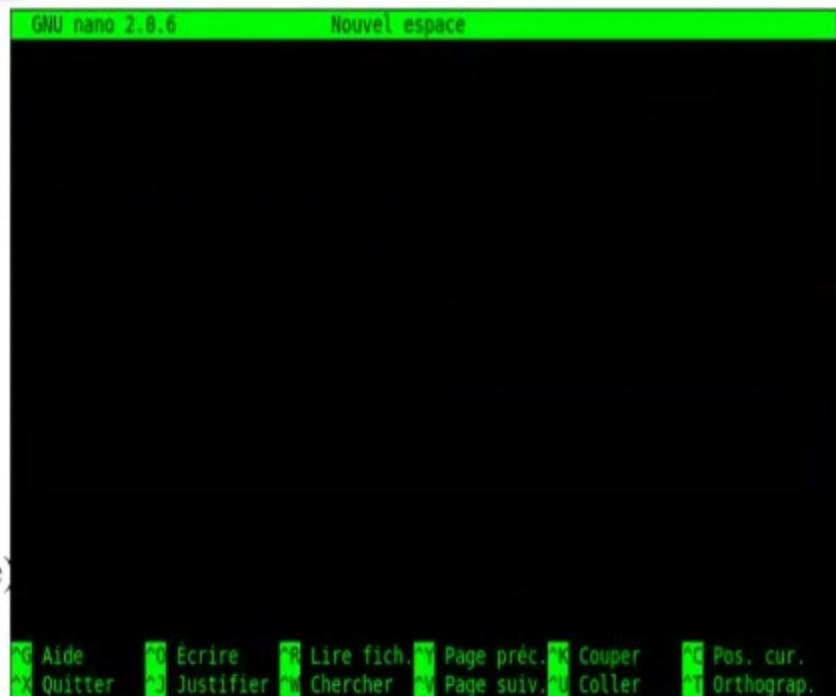
ls -h : afficher la taille en Ko, Mo, Go...

ls -t : trier par date de dernière modification

ls -color : liste les entrées avec une couleur différente suivant le type (fichier exécutable, texte, répertoire, liens ...)

L'éditeur du texte Nano

- Ctrl + G : afficher l'aide
- Ctrl + K : couper la ligne de texte (et la mettre dans le presse-papier)
- Ctrl + U : coller la ligne de texte que vous venez de couper
- Ctrl + C : afficher à quel endroit du fichier votre curseur est positionné (numéro de ligne...)
- Ctrl + W : rechercher dans le fichier
- Ctrl + O : enregistrer le fichier (écrire)
- Ctrl + X : quitter Nano
- **nano salut.txt** : ouvrir le fichier salut .txt pour le modifier



- Vous pouvez vous déplacer dans le fichier avec les flèches du clavier ainsi qu'avec les touches PageUp et PageDown pour avancer de page en page (les raccourcis Ctrl + Y et Ctrl + V)