ADMINISTRATION UNIX

INTRODUCTION UNIX

DÉFINITION

- Unix est un système d'exploitation permettant de contrôler un PC et ses différents périphériques. Unix se distingue par les caractéristiques suivantes :
- **multi-utilisateurs** (qui peut être utilisé simultanément par plusieurs personnes)
- **multi-tâches** (un utilisateur peut exécuter plusieurs programmes en même temps)

UNIX

- Sous UNIX, tout est fichier
  - fichiers (!)
  - répertoires
  - devices
  - liens
  - Pipes….

UNIX

- Noyau
  - Gère les ressources physiques
  - Conçu pour un matériel spécifique
  - S'occupe des opérations critiques du systèmes d'exploitation (control des processus, support I/O…)
    - Permet aux processus d'accéder au matériel de manière ordonnée
    - Gestion de la mémoire virtuelle
  - L'interface avec les programmes utilisateurs est définie par un ensemble de procédures (au niveau du noyau ou via des bibliothèques)
  - Le noyau est constitué d'un ensemble de procédures écrites en langage C

UNIX

- Interpréteur de commandes Shell
  - Interface de type « ligne de commande »
  - Performante et peut être utilisée au clavier
  - Permet l'exécution des programmes via des commandes
  - Permet l'écriture des programmes (appelés scripts) via des commandes

UNIX

- Utilitaires
  - Des langages de programmation
  - Des utilitaires pour le développement et maintenance des logiciels
  - Outils bureautiques
  - Des éditeurs de texte
  - Des outils pour le web
  - Système de messagerie complet

UNIX - DISTRIBUTIONS

- **« Versions » différentes de Linux**
  - **Payantes** (RHEL), semi-payantes (Mandriva) ou gratuites (presque toutes)
  - Pour **l'expert** (Debian), le **débutant** (Kubuntu), le **maniaque** (LFS), le **patient** (Gentoo), le **nostalgique** (Yggdrasil, Slackware)
  - **Orientée bureautique** (Ubuntu), appliance (Damn Small), serveur (Trustix) ou généraliste (SuSE, Fedora)
  - Plus de 350 **distributions** sur le «marché»

UNIX – LANCEMENT DU NOYAU

- **Lancement du système : boot et chargement du noyau**
  - Au boot le BIOS exécute le MBR (Master Boot Record) situé sur le premier secteur du support bootable choisi (disque, CD, clef USB, …)
- Le MBR (Master Boot Record) :
  - scanne le disque pour trouver LA partition bootable (flag)
  - lance le boot loader (chargeur de démarrage) du secteur de boot (premier secteur) de la partition bootable
- Le bootloader :
  - charge le noyau en mémoire et l'exécute
  - charge le ramdisk initrd.img en mémoire
- 2 bootloader possibles:
  - Lilo (Linux Loader)
  - Grub (Grand Unified Bootloader)

UNIX – PROCESSUS INIT

- **Lancement du système : boot -> init**
  - Une fois le noyau chargé en mémoire, il lance le premier processus : */bin/init*
  - *init* est le père de tous les autres processus qui seront créés par l'appel system *fork()*

UNIX – SERVICES & DÉMONS

- **Lancement du système : boot -> init -> modules/services**
  - Après le chargement du noyau, le script correspondant à *sysinit* dans fichier *inittab* est chargé
  - Ce script d'initialisation est chargé de 2 tâches fondamentales :
    - charger les modules dans le noyau (gestion des périphériques)
    - démarrer les services en exécutant les processus «Deferred Auxiliary Executive Monitor» (daemons) correspondant

RAPPEL
COMMANDES DE BASES

GESTION DES FICHIERS - BASE

- La commande **pwd**
  - Affiche le nom du répertoire courant
- La commande **ls**
  - Affiche les fichiers et sous-répertoire d'un répertoire
  - **Syntaxe** : ls [option] [nom_de_répertoire]
    - Options utiles
      - l : affiche les principaux attributs des fichiers et répertoires

        Type droits liens propriétaire groupe taille date nom
      - a : affiche les fichiers et répertoires dont le nom commence par un point.
      - d : n'affiche pas le contenu du répertoire
      - R : affichage récursif

GESTION DES FICHIERS - BASE

- La commande **cd** (change directory)
  - Permet un déplacement sous le répertoire passé en argument
  - **Syntaxe** : cd [nom_répertoire]
  - Exemple
    - cd ..
    - cd nom_répertoire
    - cd - -

      Dans le cas d'un chemin relatif, le répertoire est recherché à partir du répertoire courant ou en utilisant la variable CDPATH.

GESTION DES FICHIERS - COPIE

- **Création d'un fichier vide**
  - **touch** nom_fichier
- **Affichage du contenu d'un fichier (texte)**
  - **cat nom_fichier [nom_fichier, …]**
  - **cat -n fichier** *:affiche le contenu du fichier avec numérotation des lignes*
- **Copier un fichier ou dossier**
  - **cp *fichier1 fichier2*** *(fichier2 sera écrasé par le contenu du fichier1)*
  - **cp *fichier1 [fichier2 fichier3 …] répertoire*** *(les 3 fichiers seront copiés dans le répertoire)*
  - **cp *-r répertoire1 répertoire2*** (copie le répertoire1 dans le répertoire2)

GESTION DES FICHIERS – DÉPLACEMENT

- **Renommer un fichier ou répertoire (mv**:move**)**
  - **mv *fichier1 fichier2*** *(fichier1 sera **renommé** par fichier2)*
  - **mv *dossier1 dossier2*** *(dossier1 sera **renommé** par dossier2 **si dossier2 n'est pas crée**)*

- **Déplacer un fichier ou répertoire (mv**:move**)**
  - **mv *fichier1 [fichier2 fichier3 …] répertoire*** (les fichiers seront déplacés vers répertoire)

  ***Ex:* mv /etc/passwd ~/Bureau** (le fichier passwd sera déplacé vers Bureau)
  - **mv *dossier1 dossier2*** (dossier1 sera **déplacé** vers dossier2 **si dossier2 est déjà crée**)

GESTION DES FICHIERS - SUPPRESSION

- **Supprimer un fichier**
  - `rm` ***fichier***
  - `rm` ***fichier1 fichier2***
- **Supprimer un répertoire vide**
  - `rmdir` ***rep1 rep2***
- **Supprimer un répertoire vide ou non vide**
  - `rm -r` ***rep*** (le répertoire sera supprimé quelque soit son contenu)
  - `rm -r` ***rep/**** (le contenu du répertoire qui sera supprimé seulement)
  - Options utiles
    - -i : affiche un message de confirmation de suppression
    - -f : pour forcer la suppression

GESTION DES FICHIERS – AFFICHAGE

- **head -n fichier** (permet d'afficher les **n <u>premières</u>** lignes)
  - ***Ex: head* /etc/passwd** (**sans option** il affiche les **10 premières** lignes du fichier /etc/passwd)
  - ***Ex: head* -6 /etc/passwd** (Affiche les 6 premières lignes du fichier /etc/passwd)

- **tail -n fichier** (permet d'afficher les **n <u>dernières</u>** lignes)
  - ***Ex: tail* /etc/passwd** (**sans option** il affiche les **10 dernières** lignes du fichier /etc/passwd)
  - ***Ex: tail* -8 /etc/passwd** (Affiche les 8 dernières lignes du fichier /etc/passwd)
  - ***Ex: tail* -12 /etc/passwd** (Affiche les 12 dernières lignes du fichier /etc/passwd

GESTION DES FICHIERS - COMMANDES

- **wc [option] fichier** (permet d'afficher le nombre: **de lignes, de mots, de caractères et d'octets** d'un fichier)

Options:

- -l : nombre de lignes.
- -w : nombre de mots.
- -c : nombre d'octets.
- -m : nombre de caractères.

COMMANDES SUR LES FICHIERS : LES FILTRES (CUT)

- La commande cut permet d'afficher/sélectionner/filtrer des zones spécifiques d'un fichier.
- Sélection de **colonnes** et de champs

  Syntaxe: **cut –csélection_de_colonne [fichier …]**

- **Sélection_de_colonne** représente l'une des informations suivantes:
  - Une colonne seule (par exemple -c5)**=>5ème colonne.**
  - Une plage de colonnes. Exp: -c3-10 :**Les colonnes de 3 au 5** et -c8- :**de la 8ème à la dernière.**
  - Une liste de nombres séparés par des virgules (par exemple -c3,7,9),)**=>3ème ,7ème et 9ème colonne.**
  - Une combinaison des trois formes précédentes (par exemple -c1-3,7,20-).

COMMANDES SUR LES FICHIERS : LES FILTRES (CUT)

- Exemples:

cut -c3 /etc/passwd =>affiche la 3ème colonne du fichier /etc/passwd

cut -c5,7 /etc/passwd =>affiche la 5ème et la 7ème colonne du fichier /etc/passwd

cut -c5-7 /etc/passwd =>affiche de la 5ème à la 7ème colonne du fichier /etc/passwd

cut -c6- /etc/passwd =>affiche de la 6ème colonne jusqu'à la fin du fichier /etc/passwd

COMMANDES SUR LES FICHIERS : LES FILTRES (CUT)

- Sélection de **champs**

Syntaxe:

cut [-dx] –fsélection_de_champ [fichier ..]

- **x:** désigne le caractère de séparation de champs
- **Sélection_de_champ** suit les mêmes règles que la sélection des colonnes.

- **Exemple :** le séparateur de champs dans le fichier/etc/passwd est « **:** »

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

kmaster:x:500:500:kmaster:/home/kmaster:/bin/bash

COMMANDES SUR LES FICHIERS : LES FILTRES (CUT)

cut -d: -f3 /etc/passwd =>affiche le 3ème champ du fichier /etc/passwd

cut -d: -f5,7 /etc/passwd =>affiche le 5ème et le 7ème champ du fichier /etc/passwd

cut -d: -f5-7 /etc/passwd =>affiche de le 5ème à le 7ème champ du fichier /etc/passwd

cut -d: -f6- /etc/passwd =>affiche de le 6ème champ jusqu'à la fin du fichier /etc/passwd

COMMANDES SUR LES FICHIERS : RECHERCHE

- Modèles de noms de fichier
  - * : remplace zéro ou plusieurs caractères
  - ? : remplace un caractère unique quelconque
  - [] : représente une série ou une plage de caractère
- **Exemples**
  - **a*** : les fichiers dont le nom commence par **a**
  - ***a** : les fichiers dont le nom termine par **a**
  - **a??** : fichiers en trois lettres **exactement**, commençant par **a**
  - **[aA]*** : fichiers dont le nom commence par un **a** minuscule ou majuscule.
  - **[a-m]*** : fichiers dont le nom commence par une lettre de la plage a-m.
- **Attention** rm * :supprime tous les fichiers du répertoire courant.

COMMANDES SUR LES FICHIERS : RECHERCHE

- La commande find parcourt les répertoires et leurs sous-répertoires de manière récursive, à la recherche de fichiers.
- Syntaxe **find répertoire(s) critère_de_sélection option(s)**
  - Répertoire : répertoire à partir duquel la recherche doit commencer
  - Critère_de_sélection : critères de recherche à mettre en oeuvre
  - Option : que doit-il se passer si un fichier répond à ce critère
- Lors de la recherche, un répertoire ne peut être parcouru que si l'utilisateur dispose des autorisations de lecture et d'exécution sur ce répertoire

COMMANDES SUR LES FICHIERS : RECHERCHE

Options des sélection des fichiers et répertoires

- name : recherche par nom de fichier
- type : recherche par type de fichier
- user : recherche par propriétaire
- group : recherche par l'appartenance à un groupe
- size : recherche par taille de fichier
- atime : recherche par date de dernier accès
- mtime : recherche par date de dernière modification
- ctime : recherche par date de création
- perm : recherche par autorisations d'accès

COMMANDES SUR LES FICHIERS : RECHERCHE

Option de commande

- Les options de commande seront traitées pour chaque fichier trouvé. En dehors de l'option **–print** qui montre le chemin d'accès au fichier, on peut mettre en place l'option **–exec et -ok**

COMMANDES SUR LES FICHIERS : RECHERCHE

Option de commande

Traitement sans confirmation avec –exec

La spécification d'une commande derrière –exec est soumise à certaines règles:

- L'option –exec doit être la dernière de la commande find
- La commande placée derrière –exec doit être terminée par le paramètre ;. Comme c'est un caractère spécial, il doit être masqué par un \
- Pour accéder à un des fichiers trouvés, dans le cadre de la commande placée derrière –exec, utilisez le raccourci {}

Exemples

- find . –user student –exec ls –l {}\; (affiche le format long de la commande ls pour les fichiers trouvés)
- find . –type f –atime +13 –exec rm {} \; (supprime les fichiers auxquels vous n'avez plus accédé depuis plus de 2 semaines)

COMMANDES SUR LES FICHIERS : RECHERCHE

Option de commande

Traitement après confirmation avec –ok

- Les mêmes explications s'appliquent également à l'option –ok, mais celle-ci, demande une confirmation d'exécution de la commande pour chaque fichier trouvé.
- Ce n'est qu'en cas de réponse affirmative (avec y), que la commande placée derrière –ok sera exécutée.

Exemple :

find . –ok rm {} \;

COMMANDES SUR LES FICHIERS : RECHERCHE

Combinaison de critères pour recherches complexes

- Les critères de sélection peuvent être combinés. Un fichier valide doit répondre simultanément à l'ensemble des critères définis.
- La commande find permet également des combinaisons logiques de ses options:

  - ! : négation logique des options
  - a : liason par ET logique des options de recherche (and)
  - o : liaison par OU logique des options de recherche (or)

COMMANDES SUR LES FICHIERS : RECHERCHE

Combinaison de critères pour recherches complexes

- Si vous utilisez plusieurs opérateurs logiques pour combiner des options de sélection. Il faut tenir:
- Compte de l'ordre des signes !, -a, -o. **! > -a > -o**
- Pour spécifier une combinaison d'options basée sur un lien ET ou OU, il faut placer l'ensemble de l'expression entre parenthèse. Les parenthèses doivent être verrouiller.
- Exemple
- find . \ (-type d –o –name "*ham" \) –print //recherche les fichiers de type repertoire ou se terminant par ham

COMMANDES SUR LES FICHIERS : TRI

- Tri par lignes
- La commande **sort** effectue un tri par lignes en ordre croissant.

- Syntaxe: sort **[options] [pos1] [pos2] …] [fichier …]**

  - Par défaut le tri porte sur chaque ligne prise en intégralité
  - Avec les options pos1 et pos2, vous sélectionnerez les champs à utiliser comme critères de tri. Sort numérote le champs en commençant par 0.
  - Le second critère ne sera mis en œuvre que si le premier critère ne suffit pas à départager les lignes

COMMANDES SUR LES FICHIERS : TRI

- Les options utiles:

  - d : tri ascii
  - n : tri numérique
  - b : ignore les espaces placés en début de champs
  - f : aucune différence n'est faite entre majuscules et minuscules
  - r : inverse l'ordre de tri, donc tri décroissant
  - tc : le caractère « c » est défini comme caractère de séparation entre les champs d'une ligne

COMMANDES SUR LES FICHIERS : LIENS

- Lien sur un fichier ou répertoire (**raccourci**).

  Syntaxe: **ln [-s] source lien**

  - **source:** le fichier ou le dossier sur lequel on va créer un lien.
  - **lien:** c'est le raccourci qui pointe sur **source**

Il y a 2 types de liens:

- **Lien symbolique ou logique** (raccourci):

Ex: ln -s fichier1 fichier2 (fichier2 est un raccourci vers fichier1 comme sous windows)

Ex: ln -s fichier1 fichier2 (fichier2 est un raccourci vers fichier1 comme sous windows)

Ex: ln -s rep1 rep2 (rep2 est un raccourci vers rep1 comme sous windows)

COMMANDES SUR LES FICHIERS : LIENS

- **Lien physique ou matériel:**

Ex: ln fichier1 fichier2 (fichier2 est une autre copie de fichier1)

NB:

- On peut accéder au fichier source à partir de son lien symbolique ou physique.
- On ne peut pas créer un lien physique vers un répertoire.
- Si on supprime le fichier source son lien physique reste et si on modifie l'un des deux, la modification va être effectuée dans les deux (lien physique et source)

COMMANDES SUR LES FICHIERS :LES FILTRES (GREP)

- Recherches de lignes dans un fichier (grep)

  La commande **grep** permet d'extraire des lignes particulières d'un fichier ou d'un flux de données au sein d'un tube

  Syntaxe: *grep [option] modèle_de_critères [fichier1 …]*

- Les cinq options du *grep* les plus utiles sont :
  - *-i* : pour ne pas faire de différence entre majuscules/minuscules,
  - *-v* : toutes les lignes ne contenant pas le critère,
  - *-n* : pour avoir les numéros de ligne contenant le critère,
  - *-l* : pour lister seulement les fichiers et non pas les lignes contenant la chaîne recherchée,
  - -c : pour afficher le nombre de lignes contenant le critère

COMMANDES SUR LES FICHIERS :LES FILTRES (GREP)

- Le **modèle de critères** peut contenir les caractères spéciaux suivants :

  - **[...]** : Plage de caractères.
  - **.** : Un caractère unique quelconque (y compris un espace)
  - ***** : Signe de répétition. Agit sur le caractère placé devant le * (**exemple:a*** signifie a répété zéro ou plusieurs fois)
  - **$** : fin de ligne
  - **|** : ou
  - **^** : début de ligne
  - **\{nombre\}** : Nombre exacte.

COMMANDES SUR LES FICHIERS :LES FILTRES (GREP)

- Afficher les lignes commençant par **t** Dans le fichier fichier1 :

grep "^t" fichier1

- recherche de **STOP** minuscule et majuscule avec le numéro de la ligne

grep -n -i " stop" fichier1

- Chercher dans le fichier /etc/passwd les lignes qui ne commencent pas par **t**

grep -v "^t" /etc/passwd

- Chercher dans le fichier /etc/passwd les lignes qui commencent par **a,b,…. ou j**

grep "^[a-j]" /etc/passwd

- Chercher dans le fichier /etc/passwd les lignes qui se terminent pas par **b**

grep "b$" /etc/passwd

COMMANDES UTILES

- **Gestion des fichiers**
  - **ls**: affiche le contenu d'un répertoire
  - **file** : affiche le type du contenu du fichier
    - L'exécution de la commande file nécessite le droit de lecture du fichier.
  - **touch** *nom_fichier*: pour la création d'un fichier vide dans le répertoire courant
  - **more** *nom_fichier*: visualiser le contenu d'un fichier page par page
  - **rm** *nom_fichier*: supprime un fichier
  - **mv** *nom_fichier nouveau_nom*: déplace et ou renomme un fichier
  - **cp** *nom-fichier répertoire-d'accueil/autre-nom*: pour copier un fichier
  - **sort** *nom_fichier*: trie les lignes d'un fichier texte

GESTION DES RÉPERTOIRES

- Créer un répertoire

  mkdir nom_rép [nom_rép …]

  Le répertoire est non vide :

  - **.** : référence au répertoire courant
  - **..** : référence au répertoire du niveau supérieur

- Suppression d'un répertoire vide rmdir nom_rép [nom_rép …]
  - Il n'est pas possible de supprimer le répertoire courant
  - Pour la suppression d'un répertoire non vide, la commande rm avec l'option –r est plus pratique

PROTECTION FICHIER

- Le système Linux est un système multi-utilisateurs où l'accès aux fichiers est contrôlé par des droits.
  - La commande ***ls –l*** permet de les afficher.
- Pour contrôler l'accès à un fichier, le système UNIX divise les utilisateurs en quatre catégories:
  - **u** : le propriétaire (user)
  - **g** : le groupe (group)
  - **o** : les autres (others)
  - **a** : user, group et other (all)
- ainsi que quatre types de droits
  - **r** : lecture (read)
  - **w** : écriture (write)
  - **x** : exécution
  - **-** : aucun droit

COMMANDES UTILES

- **Gestion des répertoires**
  - **mkdir** *nom-de-répertoire*: Création d'un répertoire
  - **rmdir** n*om-de-répertoire*: Suppression d'un répertoire vide
  - **mv** *répertoire répertoire_d'accueil*: déplacement d'un répertoire
  - **mv** *répertoire nouveau-nom*: Changement de nom d'un répertoire
  - **pwd**: Affiche le nom du répertoire courant.
  - **cd** *répertoire*. Permet de se placer dans un répertoire donné.

EDITEUR DE TEXTE - VI

- **Editeur de texte Vi**

  vi <fichier>Ouvre un fichier.

- On se trouve alors en mode commande. Si le fichier n'existe pas, vi le crée.
- **Deux modes d'édition**
  - **Mode insertion**: les caractères tapés s'insèrent dans le fichier à éditer
  - **Mode commande**: Les caractères tapés sont considérés comme des commandesd'édition de texte

EDITEUR DE TEXTE - VI

- **Editeur de texte Vi**
  - **:q** quitte vi
  - **:q !** quitte vi sans sauver les changements
  - **:wq! ou :x** sauve si possible et quitte vi
  - **:w** sauve sous le nom de fichier courant
  - **:w !** sauve sans vérifier les droits d'écriture
  - **:w >>fich** ajoute le texte à la fin de fich
  - **i** pour passer en mode insertion
  - **<Echap>** pour passer en mode commande
  - **x** pour supprimer un caractère
  - **dd** pour supprimer une ligne

REDIRECTION DES ENTRÉES-SORTIES

- Redirection en sortie

  Redirection du résultat d'une commande vers un fichier au lien de l'écran

  Commande **>** *fichier*

  Ex: **ls –l >fichier_résultat** (le résultat de ls –l va être enregistré dans le fichier)

  - Si le fichier existe déjà son contenu sera perdu

    Sinon il faut utiliser deux fois le caractère supérieur

    Commande **>>** *fichier*

    - Si le fichier n'existe pas, il est créé automatiquement

REDIRECTION DES ENTRÉES-SORTIES

- Redirection de l'entrée

  Permet à une commande d'utiliser comme entrée, le contenu d'un fichier à la place d'une lecture clavier

  Commande **<** *fichier*

Ex: **cat <fichier** (cat va être appliquée sur fichier comme **cat fichier**)

- Canaux standard
  - Stdin 0 (représente l'entrée d'une commande **<**)
  - Stdtout 1 (représente la sortie du résultat d'une commande **>**)
  - Stderr 2 (ex : **commande 2> /dev/null** redirection des erreurs d'une commande sur la poubelle)

REDIRECTION DES ENTRÉES-SORTIES

- Redirection des erreurs vers un fichier:

commande 2> fichier

Ex: ls -l /etc 2>fichier_erreurs

- Redirection du résulat et des erreurs vers mêm fichier:

commande **>** fichier **2>&1**

Ex: ls -l /etc > fichier 2>&1

REDIRECTION DES ENTRÉES-SORTIES

- Redirection en sortie

  Récupération du résultat d'une commande ailleurs qu'à l'écran

  Commande > fichier

  Si le fichier existe déjà son contenu sera perdu

  Sinon il faut utiliser deux fois le caractère supérieur

  Commande >> fichier

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

REDIRECTION DES ENTRÉES-SORTIES

COMMANDE CAT

- La commande *cat* est une commande multi-usage qui permet d'afficher, de créer, de copier et de concaténer des fichiers.
- Exemples:
  - Lecture sur clavier et écriture sur écran
    - cat
  - Copie d'un fichier
    - cat f1 > f2
    - cat <f1 >f2
  - Concaténation des fichiers
    - cat f1 f2 f3 > f123
  - Ajouter le contenu d'un fichier à la fin d'un autre fichier
    - cat f1 >> f2
  - Création d'un fichier par saisie au clavier
    - cat >f1
  - Création d'un fichier avec condition de saisie
    - cat <<finsaisie >f1

CHAÎNER LES COMMANDES (PIPE)

- Le tube (pipe en anglais) sert à connecter la sortie d'une commande à l'entrée d'une autre commande sans passer par un fichier temporaire

- Exemple sans utiliser les tubes
  - Afficher les répertoires et fichiers du répertoire courant dont la protection est « rwxr-x--- »
    - Il faut commencer par afficher les droits de tous les répertoires et fichiers du répertoire courant en redirigeant la sortie vers un fichier
      - ls –l > temp
    - Utiliser la commande grep pour rechercher les lignes avec « rwxr-x--- »
      - grep rwxr-x--- temp
    - Supprimer le fichier temporaire
      - rm temp
- Exemple en utilisant les tubes
    - Ls –l | grep rwxr-x---

CHAÎNER LES COMMANDES (PIPE)

- Autres exemples:
  - Connaitre le nombre d'utilisateurs connectés
    - Who | wc –l
  - Le nombre de répertoires et fichier dans le répertoire courant
    - Ls | wc –l
  - Trier la liste des fichiers
    - ls -al | sort
  - Retrouver le nombre de fichiers contenant « ensias»
    - Ls –l | grep ensias| wc -l

COMPRESSION AVEC GZIP ET ZIP

Les fichiers comprimés utilisent moins d'espace disque.

- comprimer les fichiers Linux à l'aide de l'instrument Gzip ou Zip.
- Par convention, les fichiers comprimés se voient attribuer l'extension .gz.

Exemple

- **gzip filename.ext**
- **gunzip filename.ext.gz-**

COMPRESSION AVEC GZIP ET ZIP

- Pour comprimer un fichier à l'aide de **zip**:
- **zip -r filename.zip files**
- filename représente le fichier que vous créez, et files représente les fichiers que vous voulez placer dans le nouveau fichier
- Pour extraire le contenu d'un fichier zip, entrez :
- **unzip filename.zip**
- Vous pouvez compresser plusieurs fichiers en même temps avec zip ou gzip.
- **gzip filename.gz file1 file2 file3**

ARCHIVAGE AVEC TAR

- Les fichiers tar placent plusieurs fichiers ou le contenu d'un répertoire ou de plusieurs répertoires dans un seul fichier.
- Généralement, les fichiers tar terminent par l'extension .tar.
- Pour créer un fichier tar, : **tar -cvf filename.tar files/directories**
- Pour afficher la liste du contenu d'un fichier tar : **tar -tvf foo.tar**
- Pour extraire le contenu d'un fichier tar: **tar -xvf foo.tar**

Cette commande n'élimine pas le fichier .tar, mais elle place des copies du contenu de .tar dans le répertoire dans lequel vous travaillez actuellement.

ARCHIVAGE AVEC TAR

- La commande **tar** ne compresse pas automatiquement les fichiers. Pour compresser les fichiers tar : **tar -czvf foo.tar**
- Les fichiers tar compressés se voient attribuer l'extension .tgz et sont comprimés avec gzip.
- Pour décompresser un fichier tar : **tar -xzvf foo.tgz**