



UNIVERSITÉ CADI AYYAD
FACULTÉ DES SCIENCES SEMLALIA
MARRAKECH



Système d'exploitation LINUX - Le système de fichiers -

DR. MOURDI YOUSSEF
YOUSSEF.MOURDI@CED.UCA.MA

Plan du chapitre

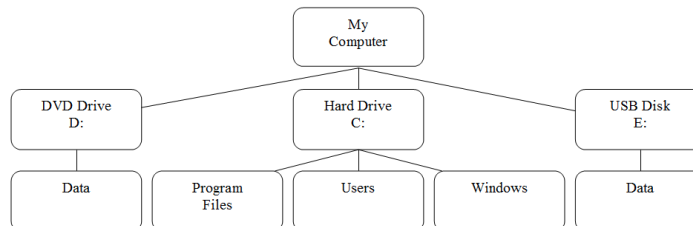
- l'organisation du système de fichiers Linux
- Notion de chemin absolu et relatif
- Les commandes de base de manipulation des fichiers et dossiers
- Les liens physiques et symboliques
- L'archivage et la compression

Introduction

- Lorsque on travail sur un système Linux on a souvent besoin de manipuler les fichiers et dossiers , certaines distributions possèdent une interface graphique permettant de réaliser cette tâche facilement mais il est important de savoir comment réaliser cette tâche en utilisant la ligne de commande (CLI)
- Le Shell offre plusieurs commandes permettant de manipuler et gérer les fichiers
- Dans ce chapitre on va voir les commandes de base permettant de lister, copier, déplacer... les fichiers
- On va également voir comment archiver et compresser les fichiers en utilisant le Shell Linux

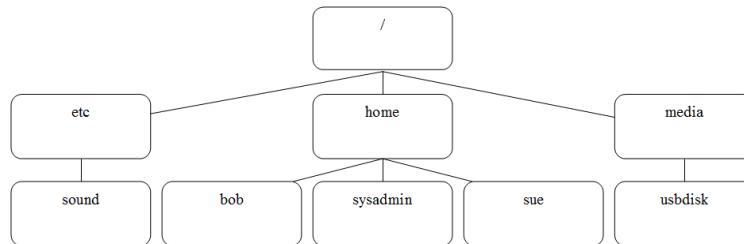
La hiérarchie du système de fichiers

- Les fichiers sont utilisés pour stocker des données (texte,image,musique...) alors que les dossiers permettent d'organiser les différents fichiers sous forme d'une arborescence
- Sous Windows cette arborescence a comme racine "mon ordinateur" et chaque périphérique de stockage(disque dur,clé USB,cd-rom..) se voit attribué une lettre (C :,A :,D :...) et figure dans « mon ordinateur »



La hiérarchie du système de fichiers

- Comme Windows, Linux possède un système de fichiers hiérarchique dans le quel le répertoire racine est appelé “racine” et est représenté par un “/”
- Les périphériques sont accessibles à travers un dossier (pas à travers une lettre comme pour Windows)



Le système de fichiers linux

- Sous Linux tout est considéré comme fichier (fichier texte, répertoire, périphérique...)
- Il existe trois types de fichiers
 - Fichiers ordinaires ; fichier texte, image..
 - Répertoire : c'est un fichier qui contient d'autres fichiers.
 - Fichiers spéciaux : périphériques...

Les inodes

- Sous linux les fichiers sont gérés à l'aide de structure appelée inode
- Les inodes (contraction de « index » et « node ») sont des structures de données contenant des informations (attributs) concernant les fichiers stockés dans des systèmes de fichiers de type Linux/Unix.
- À chaque fichier correspond un numéro d'inode dans le système de fichiers dans lequel il réside, unique au périphérique sur lequel il est situé.
- Les inodes peuvent, selon le système de fichiers, contenir aussi des informations concernant le fichier, tel que :
 - N° d'inode unique par disque
 - type de fichier
 - taille
 - Droits d'accès + Propriétaire
 - Dates relatives aux principales opérations réalisables sur le fichier.
 - Adresse des blocs de données occupés par le fichier sur le disque

Le système de fichiers linux :les principaux répertoires

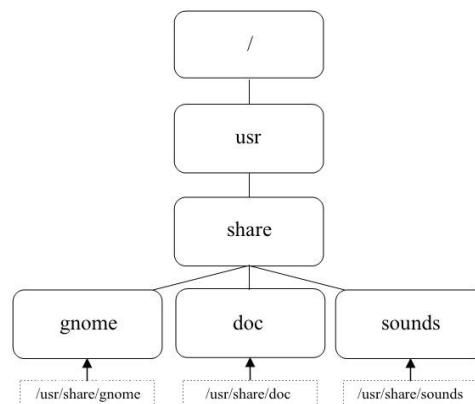
Répertoire	Description
/	Répertoire racine,point d'entrée au système de fichiers
/boot	Contient le noyau Linux et l'amorceur
/bin	Contient les exécutables de base:ls,cp,mv...
/dev	Contient des fichiers spéciaux représentant les périphérique de la machine
/etc	Contient les fichiers de configuration du système
/home	Contient les fichiers des utilisateurs(répertoire par utilisateur)
/lib	Contient les bibliothèques et les modules du noyau (/lib/modules)
/media	Point de montage des média (CD,USB...)
/root	Répertoire personnel de l'administrateur
/sbin	Contient les exécutables relatifs à l'administration du système
/tmp	Contient les fichiers temporaires
/usr	Contient les exécutables des programmes(/usr/bin et /usr/sbin) la documentation (usr/doc)
/var	Contient les fichiers relatifs à la maintenance du système

Notion de chemin

- On prend la figure précédente comme référence on remarque qu'il y a un répertoire nommé "sound" qui se trouve dans le répertoire "etc" qui se trouve lui-même dans le répertoire "racine"
- Pour accéder au répertoire "sound" on spécifie son chemin comme suit: /etc/sound où:
 - Le premier "/" représente le répertoire racine
 - Les autres "/" sont utilisés pour séparer les noms des répertoires
 - Ce type de chemin est appelé chemin absolu (tout chemin qui commence à partir de la racine est appelé chemin absolu)
- Le chemin donc permet de déterminer l'emplacement d'un répertoire ou d'un fichier

Notion de chemin

- Autres exemples de chemin



Le répertoire personnel de l'utilisateur

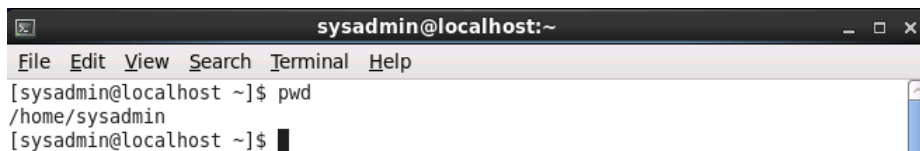
- Sous le répertoire racine "/" on trouve un répertoire appelé "home"
- Sous le répertoire "home" il y aura un répertoire pour chaque utilisateur dans le système, ce répertoire porte le nom de l'utilisateur et est appelé répertoire personnel de l'utilisateur

Exemple : Un utilisateur appelé "bob" aura comme répertoire personnel "/home/bob"

- Lorsque vous ouvrez un Shell vous serez placé automatiquement dans votre répertoire personnel
- L'utilisateur a un contrôle total sur son répertoire personnel
- Vous pouvez représenter votre répertoire personnel par le symbole "~" dans l'invite de commande au lieu d'utiliser le chemin complet, Si votre répertoire personnel est /home/sysadmin vous pouvez l'écrire aussi ~
- Pour représenter le répertoire personnel d'un autre utilisateur on peut taper ~**user** (user est le nom de l'utilisateur), **Par exemple :** ~bob représente le répertoire personnel de bob /home/bob

Le répertoire courant

- Votre répertoire courant est le répertoire dans lequel vous travaillez actuellement dans un terminal
- Cet emplacement est affiché par le prompt
- Par défaut le répertoire courant est votre répertoire personnel mais on peut le changer
- Pour afficher votre emplacement (répertoire) courant on utilise la commande : **pwd**



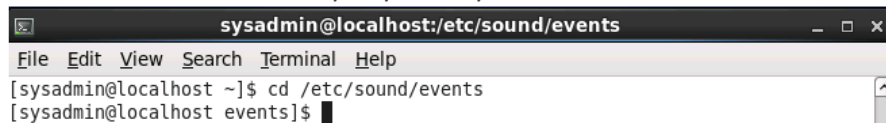
```

sysadmin@localhost:~
File Edit View Search Terminal Help
[sysadmin@localhost ~]$ pwd
/home/sysadmin
[sysadmin@localhost ~]$
  
```

Changement de répertoire (1)

- Pour changer votre répertoire courant utilisez la commande **cd** (change directory)

Cet exemple va changer le répertoire courant vers un répertoire appelé `/etc/sound/events`

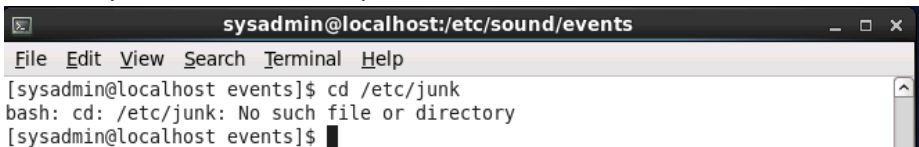


```

sysadmin@localhost:~$ cd /etc/sound/events
sysadmin@localhost events$

```

Si l'emplacement n'existe pas la commande cd va afficher une erreur



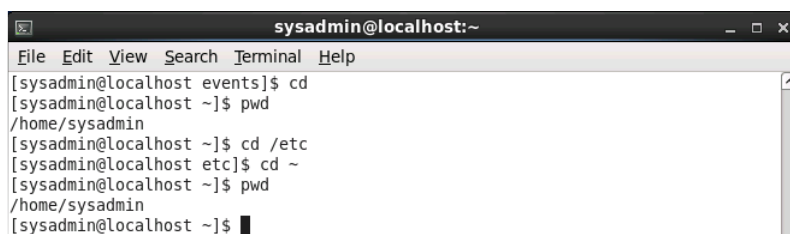
```

sysadmin@localhost:~$ cd /etc/junk
bash: cd: /etc/junk: No such file or directory
sysadmin@localhost events$

```

Changement de répertoire (2)

- Pour revenir à votre répertoire personnel on tape la commande cd sans arguments ou avec l'argument `~`



```

sysadmin@localhost:~$ cd /etc
sysadmin@localhost etc$ cd ~
sysadmin@localhost ~$ pwd
/home/sysadmin
sysadmin@localhost ~$ cd /etc
sysadmin@localhost etc$ cd ~
sysadmin@localhost ~$ pwd
/home/sysadmin
sysadmin@localhost ~$

```

Chemin absolu et chemin relatif (1)

- Rappelons que le chemin indique l'emplacement d'un fichier ou d'un répertoire dans le système de fichiers
- Lorsqu'on indique un chemin à partir de la racine "/" ce chemin est appelé chemin absolu

```
sysadmin@localhost:/$ cd /etc/perl/Net
sysadmin@localhost:/etc/perl/Net$
```

- Par contre un chemin relatif est un chemin qui ne commence pas à partir de la racine. Supposons qu'on se trouve dans **/etc/perl** pour se rendre à **/etc/perl/Net** on peut donner comme argument à la commande **cd** le répertoire **Net** au lieu de donner le chemin complet à partir de la racine

```
sysadmin@localhost:/etc/perl$ cd Net
sysadmin@localhost:/etc/perl/Net$
```

Chemin absolu et chemin relatif (2)

- Le chemin relatif utilise le répertoire courant comme référence par contre un chemin absolu utilise toujours la racine comme point de référence
- Pour construire les chemins relatifs on utilise souvent deux répertoires particuliers:
 - Le répertoire **".."** représente le répertoire parent
 - Le répertoire **"."** représente le répertoire courant

```
sysadmin@localhost:/etc/perl/Net$ pwd
/etc/perl/Net
sysadmin@localhost:/etc/perl/Net$ cd ..
sysadmin@localhost:/etc/perl$ pwd
/etc/perl
sysadmin@localhost:/etc/perl$
```

```
sysadmin@localhost:/usr/share/doc$ cd /etc/perl/Net
sysadmin@localhost:/etc/perl/Net$ cd ../../usr/share/doc
sysadmin@localhost:/usr/share/doc$ pwd
/usr/share/doc
sysadmin@localhost:/usr/share/doc$
```


Lister le contenu d'un répertoire (1)

- La commande **ls** (abréviation de list) permet de lister le contenu d'un répertoire ainsi que des informations sur ce contenu

```

sysadmin@localhost:~
File Edit View Search Terminal Help
[sysadmin@localhost ~]$ ls
Desktop Documents Downloads Music Pictures Public Templates Videos
[sysadmin@localhost ~]$

```

- ls** liste les fichiers et les affiche en utilisant un code de couleurs, chaque type de fichier est affiché avec une couleur spécifique

Lister le contenu d'un répertoire (2)

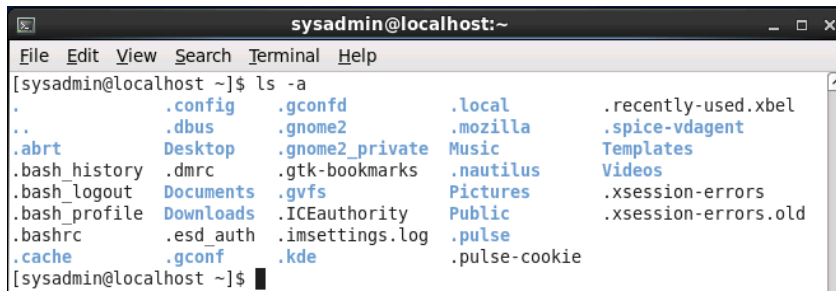
- Le tableau suivant donne la liste des types de fichiers et leur description

fichier	description	Couleur d'affichage avec ls
Fichier normal	Un fichier qui n'est pas spécial (fichier texte par exemple)	blanc
répertoire	Un fichier qui contient d'autres fichiers	bleu
exécutable	Fichier binaire exécutable	vert
Lien symbolique	Fichier qui pointe vers un autre fichier	Bleu clair (cyan)

- Il faut noter que ce comportement de ls est dû à l'option `-color` et a un alias (`ls='ls --color=auto'`)

Lister le contenu d'un répertoire (3): lister les fichiers cachés

- Un fichier caché est tout fichier ou répertoire dont le nom commence par point(.), la commande ls n'affiche pas les fichiers cachés par défaut
- Pour afficher les fichiers cachés avec ls on utilise l'option -a : **ls -a**



```

sysadmin@localhost:~$ ls -a
.          .config    .gconfd    .local      .recently-used.xbel
..         .dbus      .gnome2    .mozilla    .spice-vdagent
.abrt      Desktop    .gnome2_private Music        Templates
.bash_history .dmrc      .gtk-bookmarks .nautilus    Videos
.bash_logout Documents  .gvfs        Pictures      .xsession-errors
.bash_profile Downloads .ICEauthority Public         .xsession-errors.old
.bashrc    .esd_auth .imsettings.log .pulse
.cache     .gconf    .kde         .pulse-cookie
  
```

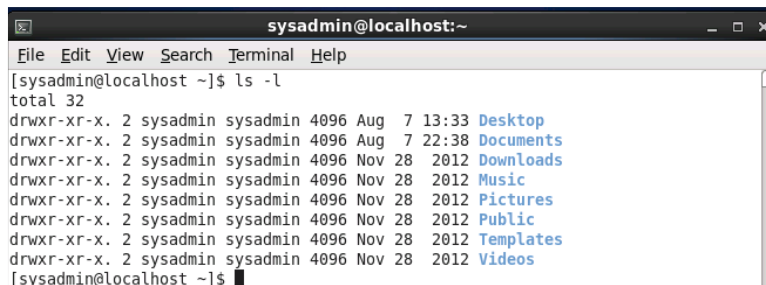
03/12/2020

DR. MOURDI YOUSSEF

19

Lister le contenu d'un répertoire (4): lister les fichiers et leur informations

- Chaque fichier possède des informations descriptive qui indiquent : le type, la taille, les permissions et le propriétaire du fichiers et d'autres informations utiles , ces informations sont appelés **metadata**
- Pour lister les fichiers avec leur metadata on utilise l'option -l de la commande ls : **ls -l**



```

sysadmin@localhost:~$ ls -l
total 32
drwxr-xr-x. 2 sysadmin sysadmin 4096 Aug  7 13:33 Desktop
drwxr-xr-x. 2 sysadmin sysadmin 4096 Aug  7 22:38 Documents
drwxr-xr-x. 2 sysadmin sysadmin 4096 Nov 28 2012 Downloads
drwxr-xr-x. 2 sysadmin sysadmin 4096 Nov 28 2012 Music
drwxr-xr-x. 2 sysadmin sysadmin 4096 Nov 28 2012 Pictures
drwxr-xr-x. 2 sysadmin sysadmin 4096 Nov 28 2012 Public
drwxr-xr-x. 2 sysadmin sysadmin 4096 Nov 28 2012 Templates
drwxr-xr-x. 2 sysadmin sysadmin 4096 Nov 28 2012 Videos
  
```

03/12/2020

DR. MOURDI YOUSSEF

20

Lister le contenu d'un répertoire (5): lister les fichiers et leur informations

- La figure suivante montre les différentes informations que fournit la commande `ls -l`

```

[selena@firenze "$] ls -l
total 19
-rw-r--r-- 1 selena selena 8749 May 5 23:31 ctrl-strategy-admin.tcl
-rw-r--r-- 1 selena selena 8709 May 5 23:29 ctrl-strategy-biz.tcl
drwxrwxr-x 3 selena selena 1024 May 1 20:30 public_html
[selena@firenze "$]
  
```

Labels pointing to the output columns:

- File Type (points to the first character)
- Permissions (points to the next nine characters)
- Owner (points to the user name)
- Group (points to the group name)
- File Size (points to the size in bytes)
- Last Modified (points to the date and time)
- File Name (points to the file name)

Lister le contenu d'un répertoire (5): lister les fichiers et leur informations

- Le tableau suivant donne une description des différents champs affichés par la commande `ls -l`

type	Le premier caractère indique le type du fichier : « - » fichier normal, « d » répertoire, « l » lien symbolique
Droits d'accès	Détermine les droits d'accès des utilisateurs au fichiers
liens	Nombre de liens sur le fichier 1 signifie que le fichier n'a aucun lien qui pointe vers lui, 2 (ou plus) signifiant qu'il existe un lien (ou plus) vers lui. Pour un répertoire en a au moins deux (. et ..)
propriétaire	Le nom de l'utilisateur auquel appartient le fichier
groupe	Nom du groupe propriétaire du fichier
taille	La taille en octets du fichier
Date de dernière modification	Date où le fichier a été modifié pour la dernière fois
nom	Nom du fichier

Lister le contenu d'un répertoire (6): lister les fichiers et leur informations

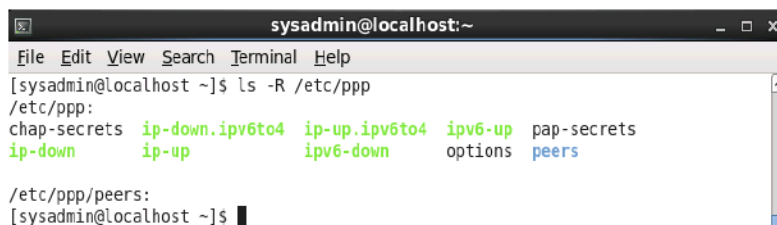
- Avec l'option `-l` de la commande `ls` la taille du fichier est affichée en octets se qui peut rendre sa lecture difficile pour l'utilisateur il serait donc mieux d'afficher cette taille en ko, Mo ou en Go
- Pour se faire il faut combiner l'option `-h` avec l'option `-l`

```
sysadmin@localhost:/$ ls -l /usr/bin/omshell
-rwxr-xr-x 1 root root 1561400 Oct  9 2012 /usr/bin/omshell
sysadmin@localhost:/$
```

```
sysadmin@localhost:/$ ls -lh /usr/bin/omshell
-rwxr-xr-x 1 root root 1.5M Oct  9 2012 /usr/bin/omshell
sysadmin@localhost:/$
```

Lister le contenu d'un répertoire (7): la récursivité

- Parfois on souhaiterait afficher le contenu d'un répertoire et le contenu de ses sous répertoire , on appel ça un affichage récursif.
- Pour lister le contenu d'un répertoire d'une façon récursive on utilise l'option `-R` de la commande `ls` : **`ls -R`**



```
sysadmin@localhost:~
File Edit View Search Terminal Help
[sysadmin@localhost ~]$ ls -R /etc/ppp
/etc/ppp:
chap-secrets  ip-down.ipv6to4  ip-up.ipv6to4  ipv6-up  pap-secrets
ip-down       ip-up            ipv6-down      options  peers

/etc/ppp/peers:
[sysadmin@localhost ~]$
```

Lister le contenu d'un répertoire (8) : lister par ordre de (taille, date...)

- Par défaut la commande ls affiche les fichiers par ordre alphabétique , il serait intéressant d'afficher les fichiers classés suivants d'autres critères
- Pour classer les fichiers affichés par décroissant de taille on utilise l'option -S de la commande ls

```
sysadmin@localhost:~$ ls /etc/ssh
moduli          ssh_host_dsa_key.pub  ssh_host_rsa_key
ssh_config       ssh_host_ecdsa_key    ssh_host_rsa_key.pub
ssh_host_dsa_key ssh_host_ecdsa_key.pub ssh_import_id
sysadmin@localhost:~$ ls -S /etc/ssh
moduli          ssh_host_dsa_key    ssh_host_ecdsa_key
ssh_config       ssh_host_dsa_key.pub ssh_host_ecdsa_key.pub
ssh_host_rsa_key ssh_host_rsa_key.pub
ssh_config       ssh_import_id
sysadmin@localhost:~$
```

- Ici les mêmes fichiers sont affichés avec un ordre différent on ne peut pas dire que ces fichiers sont classés par taille car la taille n'est pas affichée.
- Il serait donc mieux de combiner l'option -S avec l'option -l : ls -lS

Lister le contenu d'un répertoire (9): lister par ordre de (taille,date...)

- On peut aussi afficher les fichiers avec ls suivant l'ordre décroissant de la date de la dernière modification (plus récent au plus ancien), pour se faire on utilise l'option -lt de la commande ls

```
sysadmin@localhost:~$ ls -lt /etc/ssh
total 172
-rw-r--r-- 1 root root 668 Jul 3 05:56 ssh_host_dsa_key
-rw-r--r-- 1 root root 607 Jul 3 05:56 ssh_host_dsa_key.pub
-rw-r--r-- 1 root root 227 Jul 3 05:56 ssh_host_ecdsa_key
-rw-r--r-- 1 root root 179 Jul 3 05:56 ssh_host_ecdsa_key.pub
-rw-r--r-- 1 root root 1675 Jul 3 05:56 ssh_host_rsa_key
-rw-r--r-- 1 root root 399 Jul 3 05:56 ssh_host_rsa_key.pub
-rw-r--r-- 1 root root 2489 Jul 3 05:56 sshd_config
-rw-r--r-- 1 root root 136156 Sep 6 2012 moduli
-rw-r--r-- 1 root root 1669 Sep 6 2012 ssh_config
-rw-r--r-- 1 root root 302 May 13 2012 ssh_import_id
sysadmin@localhost:~$
```

Lister le contenu d'un répertoire (10): lister par ordre de (taille, date...)

- On peut aussi faire un affichage inverse (croissant) de la taille ou de la date de modification en utilisant l'option `-r` avec les options `-S` et `-t`
 - `ls -lrt`
 - `ls -lrS`

```
sysadmin@localhost:~$ ls -lrt /etc/ssh
total 172
-rw-r--r-- 1 root root 302 May 13 2012 ssh_import_id
-rw-r--r-- 1 root root 1669 Sep 6 2012 ssh_config
-rw-r--r-- 1 root root 136156 Sep 6 2012 moduli
-rw-r--r-- 1 root root 2489 Jul 3 05:56 sshd_config
-rw-r--r-- 1 root root 399 Jul 3 05:56 ssh_host_rsa_key.pub
-rw----- 1 root root 1675 Jul 3 05:56 ssh_host_rsa_key
-rw-r--r-- 1 root root 179 Jul 3 05:56 ssh_host_ecdsa_key.pub
-rw----- 1 root root 227 Jul 3 05:56 ssh_host_ecdsa_key
-rw-r--r-- 1 root root 607 Jul 3 05:56 ssh_host_dsa_key.pub
-rw----- 1 root root 668 Jul 3 05:56 ssh_host_dsa_key
sysadmin@localhost:~$
```

03/12/2020

DR. MOURDI YOUSSEF

27

Lister le contenu d'un répertoire (11) : autres options de la commande ls

- Comme on a vu `ls` permet de lister le contenu d'un dossier passé comme argument, que se passera-t-il si on fait passer à la commande `ls` un fichier comme argument?

```
sysadmin@localhost:~$ ls /etc/enscript.cfg
/etc/enscript.cfg
sysadmin@localhost:~$
```

- Tout simplement le nom du fichier est affiché
- Cela s'avère utile si on veut afficher des informations sur un fichier donné en utilisant l'option `-l` de `ls`

```
sysadmin@localhost:~$ ls -l /etc/enscript.cfg
-rw-r--r-- 1 root root 4843 Nov 11 2010 /etc/enscript.cfg
sysadmin@localhost:~$
```

03/12/2020

DR. MOURDI YOUSSEF

28

Lister le contenu d'un répertoire (12) : autres options de la commande ls

- Parfois on aimerait afficher le nom et les informations sur un dossier plutôt que d'afficher son contenu il faut alors utiliser l'option -d de la commande ls

```

sysadmin@localhost:~$ ls /etc/xdg
autostart  menus  systemd  user-dirs.conf  user-dirs.defaults
[sysadmin@localhost ~]$ ls -l /etc/xdg
total 16
drwxr-xr-x. 2 root root 4096 29 janv. 14:09 autostart
drwxr-xr-x. 6 root root 4096 29 janv. 13:57 menus
drwxr-xr-x. 2 root root 18 29 janv. 13:59 systemd
-rw-r--r--. 1 root root 414 10 juin 2014 user-dirs.conf
-rw-r--r--. 1 root root 418 10 juin 2014 user-dirs.defaults
[sysadmin@localhost ~]$ ls -dl /etc/xdg
drwxr-xr-x. 5 root root 99 29 janv. 13:59 /etc/xdg
[sysadmin@localhost ~]$
  
```

- L'option -i de ls permet d'afficher le numéro d'inode d'un fichier

Copier les fichiers (1)

- La commande cp est utilisé pour copier les fichiers , il faut alors spécifier la source et la destination
- La syntaxe de la commande cp est la suivante: **cp [source] [destination]**
 - La source représente le fichier qu'on souhaite copier
 - La destination est l'emplacement où on souhaite copier le fichier

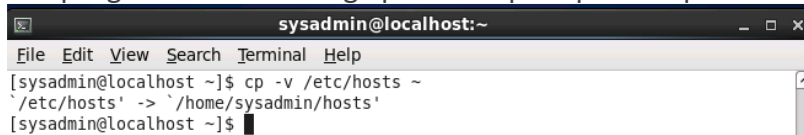
```

sysadmin@localhost:~$ cp /etc/hosts ~
[sysadmin@localhost ~]$ ls
Desktop  Downloads  Music      Public     Videos
Documents hosts      Pictures  Templates
[sysadmin@localhost ~]$
  
```

- Cette commande permet de copier le fichier /etc/hosts dans le repertoire personnel de l'utilisateur

Copier les fichiers (2)

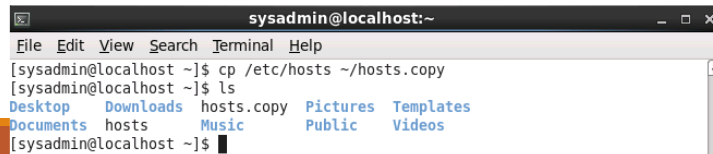
- La commande `cp` ne génère pas de sortie s'elle réussit , l'option `-v` force la commande `cp` a générée un message pour indiquer que la copie a réussit



```

sysadmin@localhost:~
File Edit View Search Terminal Help
[sysadmin@localhost ~]$ cp -v /etc/hosts ~
'/etc/hosts' -> '/home/sysadmin/hosts'
[sysadmin@localhost ~]$
  
```

- Lorsque la destination est un répertoire le fichier copie porte le même nom que le fichier original
- Pour modifier le nom du fichier copie il faut indiquer un nouveau nom dans le chemin de destination



```

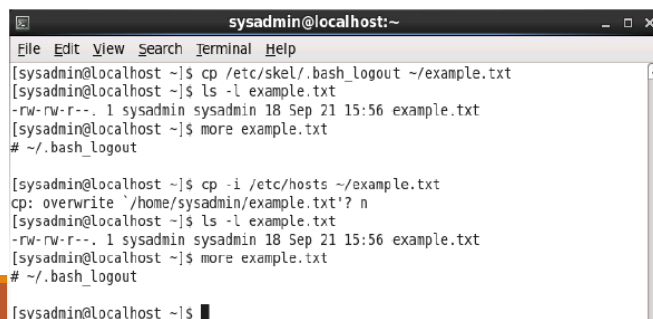
sysadmin@localhost:~
File Edit View Search Terminal Help
[sysadmin@localhost ~]$ cp /etc/hosts ~/hosts.copy
[sysadmin@localhost ~]$ ls
Desktop  Downloads  hosts.copy  Pictures  Templates
Documents hosts      Music      Public    Videos
[sysadmin@localhost ~]$
  
```

03/12/2020

31

Copier les fichiers (3) : éviter l'écrasement des données

- La commande `cp` peut être destructive si le fichier de destination existe déjà , dans ce cas le contenu du fichier de destination est remplacé par le contenu du fichier source
- L'option `-i` permet de demander confirmation de l'utilisateur (répondre par `n`(no) ou `y`(yes)) avant de faire une copie qui risque d'écraser un fichier existant



```

sysadmin@localhost:~
File Edit View Search Terminal Help
[sysadmin@localhost ~]$ cp /etc/skel/.bash_logout ~/example.txt
[sysadmin@localhost ~]$ ls -l example.txt
-rw-rw-r--. 1 sysadmin sysadmin 18 Sep 21 15:56 example.txt
[sysadmin@localhost ~]$ more example.txt
# ~/.bash_logout

[sysadmin@localhost ~]$ cp -i /etc/hosts ~/example.txt
cp: overwrite '/home/sysadmin/example.txt'? n
[sysadmin@localhost ~]$ ls -l example.txt
-rw-rw-r--. 1 sysadmin sysadmin 18 Sep 21 15:56 example.txt
[sysadmin@localhost ~]$ more example.txt
# ~/.bash_logout

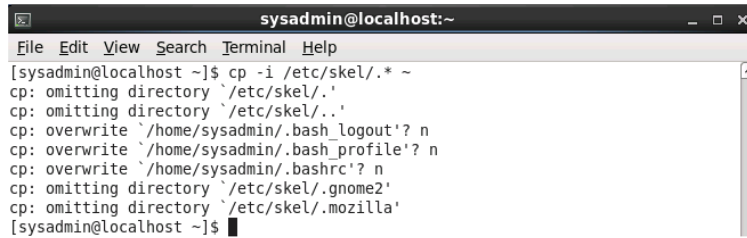
[sysadmin@localhost ~]$
  
```

03/12/2020

32

Copier les fichiers (4) : éviter l'écrasement des données

- L'option `-i` oblige l'utilisateur à répondre à la question "voulez vous écraser le fichier" par `n` ou `y`, on peut automatiser cette réponse par `n` (éviter l'écrasement) en utilisant l'option `-n` de la commande `cp`



```

sysadmin@localhost:~
File Edit View Search Terminal Help
[sysadmin@localhost ~]$ cp -i /etc/skel/* ~
cp: omitting directory '/etc/skel/.'
cp: omitting directory '/etc/skel/..'
cp: overwrite '/home/sysadmin/.bash_logout'? n
cp: overwrite '/home/sysadmin/.bash_profile'? n
cp: overwrite '/home/sysadmin/.bashrc'? n
cp: omitting directory '/etc/skel/.gnome2'
cp: omitting directory '/etc/skel/.mozilla'
[sysadmin@localhost ~]$
  
```

- Dans cet exemple on doit répondre à la question 3 fois, mais si on a 100 fichiers qui vont être écrasés on doit répondre 100 fois à la question de confirmation !!! d'où l'utilité de l'option **`-n`**

Copier les répertoires

- La commande `cp` ne copie pas les dossiers par défaut, mais permet de faire ça en l'utilisant avec l'option **`-r`**
- Avec cette option (`-r`) la commande `cp` copie le répertoire et tout son contenu

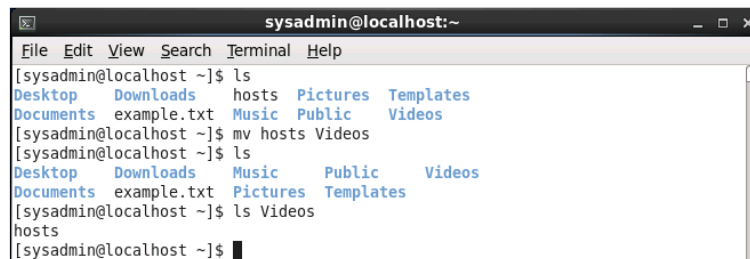
```

[sysadmin@localhost rep]$ ls ~/rep
test1.txt  test2.txt
[sysadmin@localhost rep]$ cp -r ~/rep ~/Documents
[sysadmin@localhost rep]$ ls ~/Documents/rep
test1.txt  test2.txt
[sysadmin@localhost rep]$
  
```

Déplacement de fichiers (1)

- Pour déplacer un fichier on utilise la commande `mv` , la syntaxe de la commande `mv` est la même que celle de la commande

`mv [source] [destination]`



```

sysadmin@localhost:~
File Edit View Search Terminal Help
[sysadmin@localhost ~]$ ls
Desktop Downloads hosts Pictures Templates
Documents example.txt Music Public Videos
[sysadmin@localhost ~]$ mv hosts Videos
[sysadmin@localhost ~]$ ls
Desktop Downloads Music Public Videos
Documents example.txt Pictures Templates
[sysadmin@localhost ~]$ ls Videos
hosts
[sysadmin@localhost ~]$
  
```

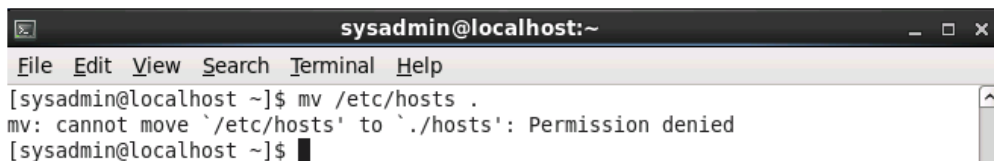
03/12/2020

DR. MOURDI YOUSSEF

35

Déplacement de fichiers (2)

- Lorsque on déplace un fichier on le supprime de son emplacement d'origine et on le met dans un nouvel emplacement, mais dans Linux pour supprimer un fichier d'un emplacement il faut disposer de certaines permissions, c'est pour cette raison que la commande **`mv`** refuse parfois de déplacer des fichiers sur lesquels on a pas assez de droits



```

sysadmin@localhost:~
File Edit View Search Terminal Help
[sysadmin@localhost ~]$ mv /etc/hosts .
mv: cannot move '/etc/hosts' to './hosts': Permission denied
[sysadmin@localhost ~]$
  
```

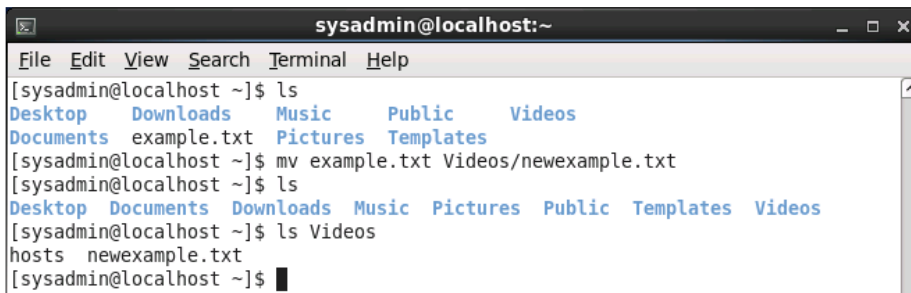
03/12/2020

DR. MOURDI YOUSSEF

36

Déplacement de fichiers (3): Déplacer et renommer

- Lorsque on fournit comme argument de destination un nom de répertoire la commande `mv` déplace le fichier source vers la destination et le fichier déplacé garde le même nom , alors que si on fournit en plus un nom de fichier dans la destination, le fichier source sera déplacé et renommé



```

sysadmin@localhost:~
File Edit View Search Terminal Help
[sysadmin@localhost ~]$ ls
Desktop  Downloads  Music      Public     Videos
Documents example.txt Pictures    Templates
[sysadmin@localhost ~]$ mv example.txt Videos/newexample.txt
[sysadmin@localhost ~]$ ls
Desktop Documents Downloads Music Pictures Public Templates Videos
[sysadmin@localhost ~]$ ls Videos
hosts newexample.txt
[sysadmin@localhost ~]$
  
```

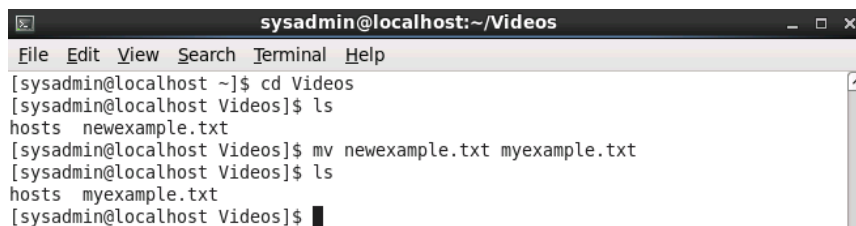
03/12/2020

DR. MOURDI YOUSSEF

37

Déplacement de fichiers(4) : renommer les fichiers

- La commande **`mv`** est aussi utilisée pour renommer un fichier dans ce cas le répertoire source et destination sont les mêmes alors que le nom du fichier source sera différent du nom de fichier destination



```

sysadmin@localhost:~/Videos
File Edit View Search Terminal Help
[sysadmin@localhost ~]$ cd Videos
[sysadmin@localhost Videos]$ ls
hosts newexample.txt
[sysadmin@localhost Videos]$ mv newexample.txt myexample.txt
[sysadmin@localhost Videos]$ ls
hosts myexample.txt
[sysadmin@localhost Videos]$
  
```

- Dans cet exemple on a renommé le fichier **`newexample.txt`** en **`myexample.txt`**

03/12/2020

DR. MOURDI YOUSSEF

38

Déplacement de fichiers(5) : autres options de la commande mv

- Comme pour la commande **cp** la commande **mv** fournit les options suivantes.

option	Description
-i	Déplacement interactif (demande de confirmation si le fichier sera écrasé)
-n	Ne pas écraser les fichiers de destination
-v	Affiche le résultat du déplacement

- L'option **-r** n'existe pas pour la commande **mv** car elle déplace par défaut les dossiers.

Création de fichiers

- Pour créer un fichier vide on utilise la commande **touch** comme le montre l'exemple

```

sysadmin@localhost:~
File Edit View Search Terminal Help
[sysadmin@localhost ~]$ ls
Desktop Documents Downloads Music Pictures Public Templates Videos
[sysadmin@localhost ~]$ touch sample
[sysadmin@localhost ~]$ ls -l sample
-rw-rw-r--. 1 sysadmin sysadmin 0 Sep 21 18:38 sample
[sysadmin@localhost ~]$

```

- On remarque que la taille du fichier crée est 0 octets , la commande touch n'ajoute pas des données au fichier crée

Suppression de fichiers

- Pour supprimer un fichier on utilise la commande **rm**
- Le fichier est supprimé sans poser de question à l'utilisateur ce qui peut causer problème et supprimer des fichiers qu'on ne souhaite pas supprimer surtout que cette opération est irréversible. Pour résoudre ce problème on utilise l'option -i de rm

The first terminal window shows the user listing files with 'ls', then deleting a file named 'sample' with 'rm sample', and then listing files again to confirm its removal.

```

sysadmin@localhost:~
File Edit View Search Terminal Help
[sysadmin@localhost ~]$ ls
Desktop Downloads Pictures sample Videos
Documents Music Public Templates
[sysadmin@localhost ~]$ rm sample
[sysadmin@localhost ~]$ ls
Desktop Documents Downloads Music Pictures Public Templates Videos
[sysadmin@localhost ~]$
  
```

The second terminal window shows the user creating three files with 'touch', then deleting them one by one with 'rm -i', which prompts for confirmation ('y' for yes, 'n' for no).

```

sysadmin@localhost:~
File Edit View Search Terminal Help
[sysadmin@localhost ~]$ touch sample.txt example.txt test.txt
[sysadmin@localhost ~]$ ls
Desktop Downloads Music Public Templates Videos
Documents example.txt Pictures sample.txt test.txt
[sysadmin@localhost ~]$ rm -i *.txt
rm: remove regular empty file `example.txt'? y
rm: remove regular empty file `sample.txt'? n
rm: remove regular empty file `test.txt'? y
[sysadmin@localhost ~]$ ls
Desktop Downloads Pictures sample.txt Videos
Documents Music Public Templates
[sysadmin@localhost ~]$
  
```

03/12/2020

DR. MOURDI YOUSSEF

41

Suppression de dossiers

- Par défaut la commande rm ne permet pas de supprimer un dossier
- Pour supprimer un dossier avec rm il faut utiliser l'option **-r** de la commande rm

The first terminal window shows the user attempting to delete a directory named 'Videos' with 'rm Videos', which results in an error message: 'rm: cannot remove `Videos': Is a directory'.

```

sysadmin@localhost:~
File Edit View Search Terminal Help
[sysadmin@localhost ~]$ rm Videos
rm: cannot remove `Videos': Is a directory
[sysadmin@localhost ~]$
  
```

The second terminal window shows the user successfully deleting the 'Videos' directory and its contents with 'rm -r Videos', followed by a 'ls' command to confirm its removal.

```

sysadmin@localhost:~
File Edit View Search Terminal Help
[sysadmin@localhost ~]$ ls
Desktop Downloads Pictures sample.txt Videos
Documents Music Public Templates
[sysadmin@localhost ~]$ rm -r Videos
[sysadmin@localhost ~]$ ls
Desktop Documents Downloads Music Pictures Public sample.txt Templates
[sysadmin@localhost ~]$
  
```

- Avec rm -r le dossier et tous son contenu sera supprimé il faut penser à utiliser l'option -i aussi
- On peut également supprimer un dossier avec la commande rmdir à condition que ce dernier soit vide

03/12/2020

DR. MOURDI YOUSSEF

42

Création de dossier

- Pour créer un nouveau dossier on utilise la commande **mkdir**

```

sysadmin@localhost:~
File Edit View Search Terminal Help
[sysadmin@localhost ~]$ ls
Desktop Documents Downloads Music Pictures Public sample.txt Templates
[sysadmin@localhost ~]$ mkdir test
[sysadmin@localhost ~]$ ls
Desktop Downloads Pictures sample.txt test
Documents Music Public Templates
[sysadmin@localhost ~]$

```

- l'option **-p** de mkdir permet de créer le répertoire parent s'il n'existe pas , cette option est utile pour créer une arborescence

```

[sysadmin@localhost Bureau]$ cd ./rep/
[sysadmin@localhost rep]$ mkdir ./parent/fils
mkdir: impossible de créer le répertoire « ./parent/fils »: Aucun fichier ou dossier de ce type
[sysadmin@localhost rep]$ mkdir -p ./parent/fils
[sysadmin@localhost rep]$ ls .
file1.txt parent
[sysadmin@localhost rep]$ ls ./parent/
fils

```

```

[sysadmin@localhost rep]$ mkdir -p ./rep1/rep2/rep3/rep4
[sysadmin@localhost rep]$

```

03/12/2020

43

Les liens

- Sous Linux on peut accéder à un fichier à partir de plusieurs emplacement et on fournissant plusieurs noms , pour se faire on utilise les liens.
- Un lien est un fichier particulier qui pointe vers un autre fichier, sous Linux il existe deux types de liens
 - Lien dur (physique)** : lien vers le N° d'inode d'un fichier,
 - Lien symbolique (logique)** : lien vers le chemin d'un fichier,

03/12/2020

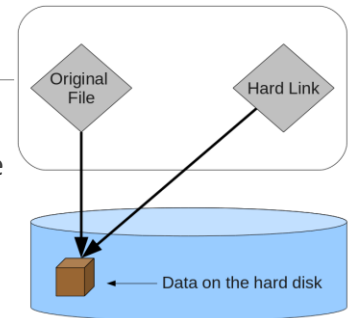
DR. MOURDI YOUSSEF

44

Les liens(2) : lien physique

- Permet de donner plusieurs noms au même fichier
- Le fichier et son lien physique ont le même numéro d'inode
- Ne peut pas être utilisé pour *les répertoires*
- Tous les liens doivent être dans la même partition
- Toutes modifications sur le fichier affectent le lien et vice versa
- Un fichier est complètement supprimé quand TOUS ses liens physiques sont supprimés (≠ raccourcis)
- Pour créer un lien physique on utilise la commande **ln** :

```
ln <fichier_original> <nom_lien>
```



03/12/2020

DR. MOURDI YOUSSEF

45

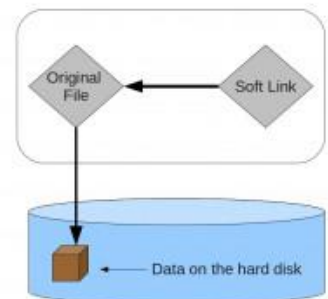
Les liens(3) : lien symbolique

- Crée un raccourci vers un fichier donné
- Si le fichier original est supprimé le lien est cassé
- Le fichier et le lien peuvent être dans des partitions différentes
- On peut aussi créer des liens symboliques pour les répertoires
- Le fichier original et le lien symbolique n'ont pas le même numéro d'inode
- Pour créer un lien symbolique on utilise la commande ln avec l'option **-s**

```
ln -s <fichier_original> <nom_lien>
```

- Pour afficher la cible d'un lien symbolique

```
readlink <nom_lien>
```



03/12/2020

DR. MOURDI YOUSSEF

46

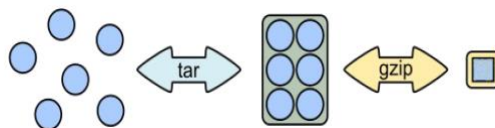
Autres commandes

- **cat <nom_fichier>**
 - permet d'afficher le contenu d'un fichier texte sur le terminal
- **file <nom_fichier>**
 - D'afficher le type d'un fichier (texte,image...)

```
[sysadmin@localhost rep]$ ls
fich1.doc  fich2.doc  fich3.doc  file1.txt  parent  repl
[sysadmin@localhost rep]$ file repl fich2.doc
repl:      directory
fich2.doc: ASCII text
[sysadmin@localhost rep]$ cat fich2.doc
Test de fichier non vide
[sysadmin@localhost rep]$ █
```

Archivage et compression (1)

- L'archivage et la compression sont deux opérations qui permettent de stocker et de transmettre des fichiers efficacement
 - **Archivage** : permet de combiner plusieurs fichiers pour créer un seul fichier afin de le stocker et de le déplacer plus facilement
 - **Compression** : rend un fichier plus petit en éliminant les informations redondantes
 - On peut archiver plusieurs fichiers en une seule archive puis compresser l'archive obtenue comme on peut compresser un seul fichier



Archivage et compression(2) : compression de fichiers

- Linux fournit plusieurs outils permettant la compression des fichiers les plus utilisés sont:
 - gzip
 - Bzip2

Archivage et compression(2) : compression de fichiers – gzip et gunzip -

- Syntaxe **gzip [options] fichier**
- compresse le fichier donné en argument

```
bob:tmp $ ls -l access_log*
-rw-r--r-- 1 sean sean 372063 Oct 11 21:24 access_log
bob:tmp $ gzip access_log
bob:tmp $ ls -l access_log*
-rw-r--r-- 1 sean sean 26080 Oct 11 21:24 access_log.gz
```

- Ici on a compressé avec **gzip** un fichier appelé **access_log** dont la taille était 372063 octets
- Après compression le fichier original est remplacé par un fichier compressé **access_log.gz** dont la taille est seulement 26080 octets

Archivage et compression(2) : compression de fichiers – gzip et gunzip -

- L'option **-l** de gzip permet de donner des informations sur le fichier compressé notamment le ratio de compression (taille gagnée après compression)

```
bob:tmp $ gzip -l access_log.gz
      compressed      uncompressed   ratio uncompressed_name
          26080           372063  93.0% access_log
```

- Dans cet exemple le ratio est : 93%
- Pour compresser tous les fichiers se trouvant dans une arborescence (dossier et ses sous-dossiers) on utilise l'option **-r** de **gzip**
- La commande **gunzip** est l'opposé de **gzip**, elle permet de décompresser un fichier compressé avec **gzip**
- Le fichier décompressé retrouve sa taille et son nom d'origine

```
bob:tmp $ gunzip access_log.gz
bob:tmp $ ls -l access_log*
-rw-r--r-- 1 sean sean 372063 Oct 11 21:24 access_log
```

Archivage et compression(3) : compression de fichiers - bzip2, bunzip2 -

- bzip2 permet de mieux compresser que gzip
- Syntaxe

bzip2 [options] fichier

bunzip2 [options] fichiers.bz2

```
[root@localhost ~]# ls -lh services
-rw-r--r-- 1 root root 354K mar 26 09:35 services
[root@localhost ~]# bzip2 services
[root@localhost ~]# ls -lh services.bz2
-rw-r--r-- 1 root root 91K mar 26 09:35 services.bz2
[root@localhost ~]# bunzip2 services.bz2
[root@localhost ~]# ls -lh services
-rw-r--r-- 1 root root 354K mar 26 09:35 services
[root@localhost ~]#
```

Archivage et compression (7) : archivage de fichiers

- Si on veut envoyer plusieurs fichiers à une personne on peut compresser chacun des fichiers et on aura une taille plus petite que d'envoyer les fichiers non compressés , mais le problème est que on doit manipuler plusieurs fichiers au même temps
- La solution à ce problème est l'archivage : cette opération permet de grouper un ensemble de fichiers en un seul fichier ce dernier peut être désarchivé pour retrouver les fichiers originaux
- La commande qui permet d'archiver des fichiers sous Linux est **tar**, et elle fonctionne en 4 modes:
 - Création : permet de créer une archive à partir de plusieurs fichiers (*l'option -c*)
 - Extraction : extraire(faire sortir) un ou plusieurs fichiers de l'archive (*l'option -x*)
 - Lister : afficher le contenu d'une archive sans l'extraire (*l'option -t*)
 - Mise à jour : modifier les fichiers archives ou ajouter de nouveaux fichiers a l'archive (*l'option -u*)

Archivage et compression (7) : archivage de fichiers

- Exemple1 :


```
$ tar -cf access_logs.tar access_log*
$ ls -l access_logs.tar
-rw-rw-r-- 1 542720 Oct 12 21:42 access_logs.tar
```
- Dans cet exemple on a créé une archive qui va comporter tous les fichiers dont le nom commence par **access_log** ,l'option **-c** permet de créer une archive alors que l'option **-f** permet d'indiquer le nom du fichier **.tar** qu'on souhaite créer.

Archivage et compression (7) : archivage de fichiers

- Exemple 2
 - `$ tar -czf access_logs.tar.gz access_log*`
 - `$ ls -l access_logs.tar.gz`
`-rw-rw-r-- 1 46229 Oct 12 21:50 access_logs.tar.gz`
 - `$ gzip -l access_logs.tar.gz`

compressed	uncompressed	ratio	uncompressed_name
46229	542720	91.5%	access_logs.tar
- Dans cet exemple on a ajouté comme option **-z** qui permet de créer une archive compressée avec **gzip**
- Comme on peut le voir la taille de l'archive est réduite après compression
- Le fichier obtenu porte par convention l'extension **tar.gz**
- Si on veut faire une compression avec **bzip2** on utilise alors l'option **-j** au lieu de **-z** et par convention le fichier archive compressé avec bzip2 a comme extension tar.bz2
- `tar -cjf file.tar.bz2 access_log*`