

ADMINISTRATION DE L'INFRASTRUCTURE INFORMATIQUE

Pr. Najat TISSIR

tissir.najat@gmail.com

Filière: Génie Informatique

CHAPITRE 2: GESTION DE FICHIERS ET RÉPERTOIRES

1

Système de fichiers

2

Opérations sur les répertoires

3

Opérations sur les fichiers

4

Sauvegarde d'une arborescence

OPÉRATIONS SUR LES FICHIERS

Créer un fichier: touch, cat

- Il faut avoir la permission d'écriture dans le répertoire qui désigne le fichier.
- La commande **touch** crée un fichier vide.

```
touch file
```

- La commande **cat** permet de créer et d'éditer un fichier.

```
cat > essai
```

```
Bonjour
```

```
Il fait beau
```

```
<ctrl-d>
```

OPÉRATIONS SUR LES FICHIERS

Afficher le contenu d'un fichier: cat

■ **Cat** fichier

```
najat@najat-VirtualBox:~$ cat >pres  
Bonjour  
najat@najat-VirtualBox:~$
```

OPÉRATIONS SUR LES FICHIERS

Afficher le contenu d'un fichier: more, less

- Si le fichier s'étend au delà d'une page, il est préférable d'opter pour la commande **more** ou **less**.

more fichier.txt : Affichage page par page

- Affiche le fichier écran par écran
 - ▷ **Flèche bas / Entrée** : Défiler ligne par ligne
 - ▷ **Espace** : Passer à la page suivante
 - ▷ **q** : Quitter

OPÉRATIONS SUR LES FICHIERS

Afficher le contenu d'un fichier: more, less

less fichier.txt : Affichage interactif avec défilement

- **less** est très puissante que **more** car elle permet de:
 - ▷ Remonter dans le fichier
 - ▷ Rechercher avec **/mot** (appuyer sur n pour le suivant)
 - ▷ Naviguer d'une manière plus fluide
- **Flèche bas / Entrée** : Descendre ligne par ligne
- **Flèche haut** : Remonter ligne par ligne
- **Espace** : Page suivante
- **b** : Page précédente
- **/mot** : Rechercher un mot
- **n** : Aller à la prochaine occurrence trouvée
- **q** : Quitter

OPÉRATIONS SUR LES FICHIERS

Afficher le contenu d'un fichier: more, less

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mail List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
systemd-timesync:x:102:104:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:103:106:/nonexistent:/usr/sbin/nologin
syslog:x:104:110:/home/syslog:/usr/sbin/nologin
_apt:x:105:65534:/nonexistent:/usr/sbin/nologin
tss:x:106:111:TPM software stack,,:/var/lib/tpm:/bin/false
uuid:x:107:114:/run/uuid:/usr/sbin/nologin
tcpdump:x:108:115:/nonexistent:/usr/sbin/nologin
avahi-autoipd:x:109:116:Avahi autoip daemon,,:/var/lib/avahi-autoipd:/usr/sbin/nologin
usbmux:x:110:46:usbmux daemon,,:/var/lib/usbmux:/usr/sbin/nologin
rtkit:x:111:117:RealtimeKit,,:/proc:/usr/sbin/nologin
dnsmasq:x:112:65534:dnsmasq,,:/var/lib/misc:/usr/sbin/nologin
cups-pk-helper:x:113:120:user for cups-pk-helper service,,:/home/cups-pk-helper:/usr/sbin/nologin
speech-dispatcher:x:114:29:Speech Dispatcher,,:/run/speech-dispatcher:/bin/false
avahi:x:115:121:Avahi mDNS daemon,,:/var/run/avahi-daemon:/usr/sbin/nologin
kernoops:x:116:65534:Kernel Oops Tracking Daemon,,:/usr/sbin/nologin
saned:x:117:123:/var/lib/saned:/usr/sbin/nologin
nm-openvpn:x:118:124:NetworkManager OpenVPN,,:/var/lib/openvpn/chroot:/usr/sbin/nologin
hplip:x:119:7:HPLIP system user,,:/run/hplip:/bin/false
whoopsie:x:120:125:/nonexistent:/bin/false
colord:x:121:126:colord colour management daemon,,:/var/lib/colord:/usr/sbin/nologin
geoclue:x:122:127:/var/lib/geoclue:/usr/sbin/nologin
pulse:x:123:128:PulseAudio daemon,,:/var/run/pulse:/usr/sbin/nologin
gnome-initial-setup:x:124:65534:/run/gnome-initial-setup:/bin/false
gdm:x:125:130:Gnome Display Manager:/var/lib/gdm3:/bin/false
najat:x:1000:1000:najat,,:/home/najat:/bin/bash
/etc/passwd
```

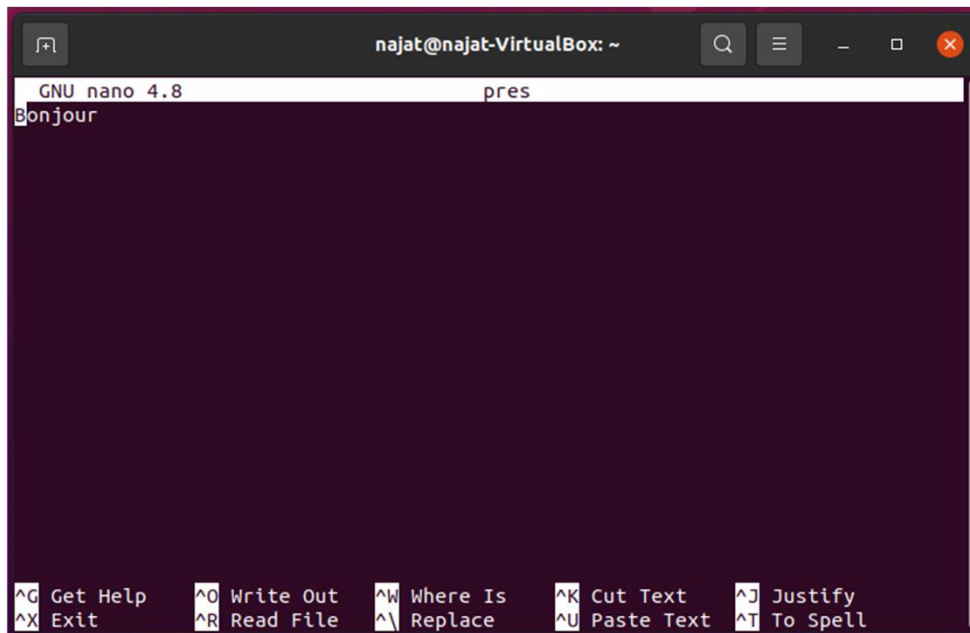
```
najat@najat-VirtualBox:~$ more /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mail List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
systemd-timesync:x:102:104:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:103:106:/nonexistent:/usr/sbin/nologin
syslog:x:104:110:/home/syslog:/usr/sbin/nologin
_apt:x:105:65534:/nonexistent:/usr/sbin/nologin
tss:x:106:111:TPM software stack,,:/var/lib/tpm:/bin/false
uuid:x:107:114:/run/uuid:/usr/sbin/nologin
tcpdump:x:108:115:/nonexistent:/usr/sbin/nologin
avahi-autoipd:x:109:116:Avahi autoip daemon,,:/var/lib/avahi-autoipd:/usr/sbin/nologin
usbmux:x:110:46:usbmux daemon,,:/var/lib/usbmux:/usr/sbin/nologin
rtkit:x:111:117:RealtimeKit,,:/proc:/usr/sbin/nologin
dnsmasq:x:112:65534:dnsmasq,,:/var/lib/misc:/usr/sbin/nologin
cups-pk-helper:x:113:120:user for cups-pk-helper service,,:/home/cups-pk-helper:/usr/sbin/nologin
speech-dispatcher:x:114:29:Speech Dispatcher,,:/run/speech-dispatcher:/bin/false
avahi:x:115:121:Avahi mDNS daemon,,:/var/run/avahi-daemon:/usr/sbin/nologin
kernoops:x:116:65534:Kernel Oops Tracking Daemon,,:/usr/sbin/nologin
saned:x:117:123:/var/lib/saned:/usr/sbin/nologin
nm-openvpn:x:118:124:NetworkManager OpenVPN,,:/var/lib/openvpn/chroot:/usr/sbin/nologin
hplip:x:119:7:HPLIP system user,,:/run/hplip:/bin/false
whoopsie:x:120:125:/nonexistent:/bin/false
colord:x:121:126:colord colour management daemon,,:/var/lib/colord:/usr/sbin/nologin
geoclue:x:122:127:/var/lib/geoclue:/usr/sbin/nologin
```


OPÉRATIONS SUR LES FICHIERS

Editer le contenu d'un fichier: nano

nano fichier.txt

- Elle permet d'éditer directement en tapant le texte
- Enregistrer : Ctrl + X, puis Y et Entrée
- Quitter sans enregistrer : Ctrl + X, puis N



The screenshot shows a terminal window titled 'najat@najat-VirtualBox: ~'. Inside, the GNU nano 4.8 editor is open, editing a file named 'pres'. The text 'Bonjour' is visible on the first line. The bottom status bar displays various keyboard shortcuts: ^G Get Help, ^O Write Out, ^W Where Is, ^K Cut Text, ^J Justify, ^X Exit, ^R Read File, ^_ Replace, ^U Paste Text, and ^T To Spell.

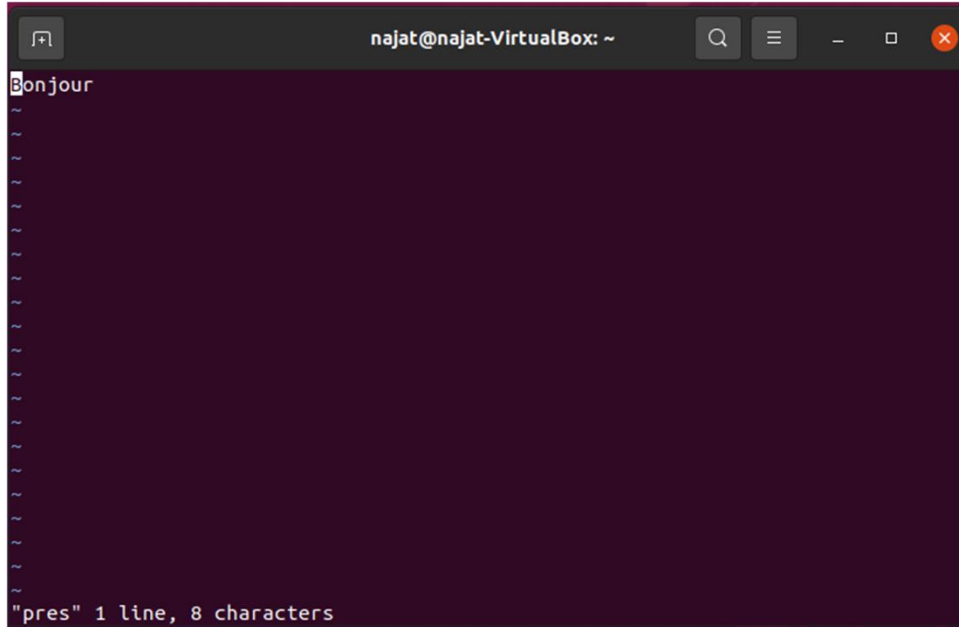
OPÉRATIONS SUR LES FICHIERS

Editer le contenu d'un fichier: vi / vim

- **vi** comprend deux modes: un mode commande et un mode insertion
- La touche **i** permet de passer en mode insertion (en bas de l'écran s'affiche la ligne INSERT)
- Pour sauvegarder le fichier appuyer d'abord sur la touche **Echap** pour sortir du mode INSERT puis tapez **:w**
- Puis en appuyant sur les touches **:wq** pour enregistrer et quitter l'éditeur
- Si vous souhaitez quitter sans enregistrer les modifications taper les touches: **:q!**

OPÉRATIONS SUR LES FICHIERS

Editer le contenu d'un fichier: vi / vim



The screenshot shows a terminal window titled "najat@najat-VirtualBox: ~". The terminal has a dark purple background. The text "Bonjour" is on the first line, with a white cursor at the end. Below it are several empty lines, each starting with a tilde (~). At the bottom, the status line reads "pres" 1 line, 8 characters. The window's title bar includes a search icon, a menu icon, and standard window controls (minimize, maximize, close).

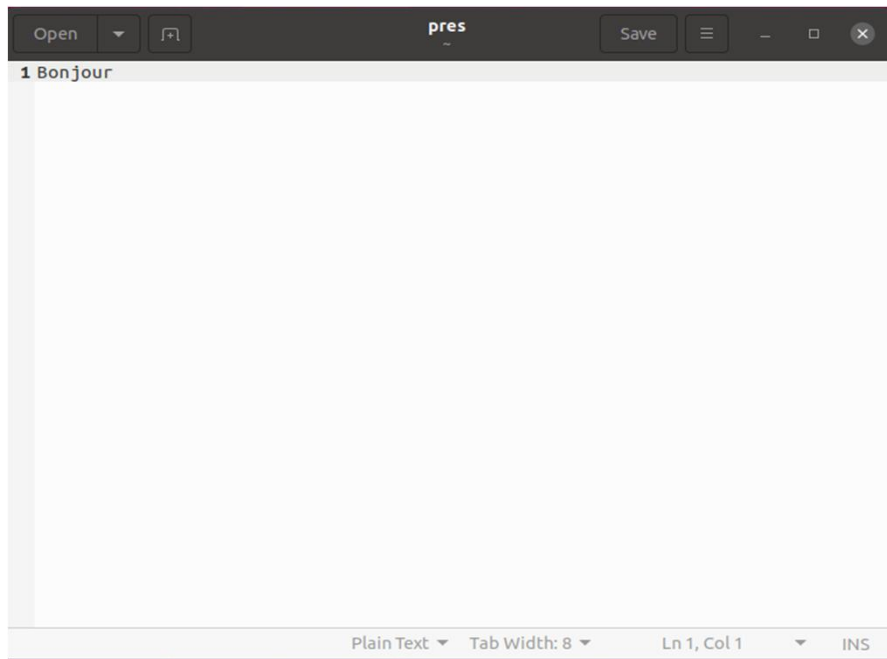
OPÉRATIONS SUR LES FICHIERS

Editer le contenu d'un fichier: depuis un environnement graphique

■ **gedit** fich.txt: Ubuntu/Gnome

■ **Kate** fich.txt: KDE

■ Exemple (gedit):



OPÉRATIONS SUR LES FICHIERS

Supprimer un fichier: rm

- Il faut avoir la permission d'écriture dans le répertoire qui désigne le fichier.
- La commande **rm** permet de supprimer un fichier.

rm file

- Les options de **rm** les plus utilisées sont :
 - ▷ **-r** supprime tout un répertoire ainsi que ses sous répertoires.
 - ▷ **-i** demande une confirmation avant de supprimer un fichier

OPÉRATIONS SUR LES FICHIERS

Supprimer un fichier: rm *

- La commande `rm *` permet de supprimer tous les fichiers dans le répertoire courant.

```
rm *
```

- `*` : joker (wildcard) qui signifie tous les fichiers du répertoire courant.
- Cette commande ne supprime pas les sous-répertoires (uniquement les fichiers).
- Attention ! Il n'y a pas de confirmation par défaut, donc les fichiers supprimés ne sont pas récupérables.

```
rm -rf *
```

- Cette commande permet de supprimer tout le contenu d'un répertoire, y compris les sous-répertoires, et sans confirmation
- **Soyez très prudent avec `rm -rf *`, surtout si vous êtes en mode root !**
- « `rm -r /*` » peut générer une catastrophe

OPÉRATIONS SUR LES FICHIERS

Chercher un /dans un fichier: locate, grep

- La commande **locate** permet d'afficher le nom complet de tout fichier ou répertoire correspondant à un critère de recherche donné.
- Recherche les fichiers et répertoires contenant la chaîne de caractères
- Exemple: **locate** touch
 - ▷ /usr/share/man/man1/fic.1.gz
 - ▷ /usr/X11R6/man/man4/mufic.4x.gz /usr/X11R6/man/man4/mufic.4x.gz
 - ▷ /bin/fic
- la commande **grep** recherche, dans un ou plusieurs fichiers, de toutes les lignes contenant une chaîne donnée de caractères
- Exemple: **grep** Bonjour file
 - ▷ Bonjour tout le monde
 - ▷ Cette commande recherche la chaîne de caractères Bonjour dans le fichier file

OPÉRATIONS SUR LES FICHIERS

Copier un fichier: cp

cp [OPTION]... SOURCE... DIRECTORY

■ Utiliser l'option **-i** pour le mode interactive.

■ Si le fichier copié existe alors il sera écrasé

■ Exemples:

- ▷ Copie le fichier « fic1.txt » vers « /home/najat »

```
cp fic1.txt /home/najat
```

- ▷ Copie tous les fichiers du répertoire « /home/hiba » vers « /home/najat »

```
cp /home/hiba/* /home/najat
```

- ▷ Copie le rep « /home/hiba » tout entier vers « /home/najat »

```
cp -r /home/hiba/ /home/najat
```

- ▷ Copie en conservant les droits et l'appartenance (-a)

```
cp -a /home/hiba/fic.txt /home/najat
```


OPÉRATIONS SUR LES FICHIERS

Copier un fichier: cp

■ La commande **cp** * copie tous les fichiers du répertoire courant vers un autre emplacement.

■ **cp** * ne copie que les fichiers, pas les sous-dossiers (sauf si vous utilisez **cp -r**).

■ Exemples:

▷ Copier tous les fichiers vers un dossier « backup » :

```
cp * backup/
```

▷ Copier tous les fichiers .txt vers un autre dossier

```
cp *.txt backup/
```

■ Si des fichiers ont le même nom dans <destination>, ils seront écrasés sans avertissement (sauf si vous utilisez cp -i).

OPÉRATIONS SUR LES FICHIERS

Copier un fichier: cp en préservant les attributs des fichiers lors de la copie

- Copier un fichier revient à créer un nouveau fichier identique à l'original
 - ▷ C'est donc l'utilisateur qui a effectué la copie qui devient le propriétaire
 - ▷ A moins d'utiliser l'option `-a`
 - ▷ Exemple :

Je suis connecté en tant que root

Je liste le répertoire de hiba

```
root@Nom:/home/hiba# ls -l
total 0
lrwxrwxrwx 1 hiba hiba 26 2008-08-21 18:04 Examples -> /usr/share/example-content
-rw-r--r-- 1 hiba hiba 0 2008-08-21 18:14 fic1.txt
```

Ce fichier appartient à hiba

```
root@Nom:/home/hiba# cp fic1.txt /home/najat/
```

Je le copie chez najat

```
root@Nom:/home/hiba# ls -l /home/najat/
```

```
total 4
-rw-r--r-- 1 root root 0 2008-08-21 18:14 fic1.txt
drwxr-xr-x 2 najat najat 4096 2008-08-21 17:59 rep1
drwxr-xr-x 2 najat najat 4096 2008-08-21 18:00 rep2
drwxr-xr-x 2 najat najat 4096 2008-08-15 14:41 vidéos
root@Nom:/home/hiba#
```

La copie m'appartient (root)

OPÉRATIONS SUR LES FICHIERS

Copier un fichier: cp en préservant les attributs des fichiers lors de la copie

- Avec l'option **-p**: on préserve les attributs suivants du fichier d'origine :
 - ▷ Date de modification et d'accès
 - ▷ Propriétaire et groupe
 - ▷ Permissions

- L'option **-a** (archive) est plus complète que **cp -p** (équivalente à **cp -rp --preserve=all**). Elle préserve tout ce que **-p** conserve, mais en plus :
 - ▷ Copies récursives des répertoires (comme -r)
 - ▷ Liens symboliques conservés
 - ▷ Attributs étendus

OPÉRATIONS SUR LES FICHIERS

Copier un fichier: cp

```
najat@najat-VirtualBox:~/temp$ ls -l
total 4
-rw-rw-r-- 1 najat najat 0 23:25 16 فبراير f1
-rw-rw-r-- 1 najat najat 0 23:25 16 فبراير f2
drwxrwxr-x 2 najat najat 4096 23:27 16 فبراير temp1
najat@najat-VirtualBox:~/temp$ cp f1 f2 temp1
najat@najat-VirtualBox:~/temp$ ls -l
total 4
-rw-rw-r-- 1 najat najat 0 23:25 16 فبراير f1
-rw-rw-r-- 1 najat najat 0 23:25 16 فبراير f2
drwxrwxr-x 2 najat najat 4096 23:27 16 فبراير temp1
najat@najat-VirtualBox:~/temp$ cp -i * temp1
cp: overwrite 'temp1/f1'? y
cp: overwrite 'temp1/f2'? y
cp: -r not specified; omitting directory 'temp1'
najat@najat-VirtualBox:~/temp$ ls -l temp1
total 0
-rw-rw-r-- 1 najat najat 0 23:28 16 فبراير f1
-rw-rw-r-- 1 najat najat 0 23:28 16 فبراير f2
najat@najat-VirtualBox:~/temp$
```

OPÉRATIONS SUR LES FICHIERS

Déplacer/Renommer un fichier: mv

■ La commande « **mv** » déplace ou renomme une source vers une destination.

■ Syntaxe :

```
mv [OPTION]... SOURCE... DIRECTORY
```

■ Exemples:

▷ Déplace le fichier « fic1.txt » vers « /home/najat » sans le renommer

```
mv fic1.txt /home/najat
```

▷ Renomme le fichier « fic1.txt » en « fic2.txt »

```
mv fic1.txt fic2.txt
```

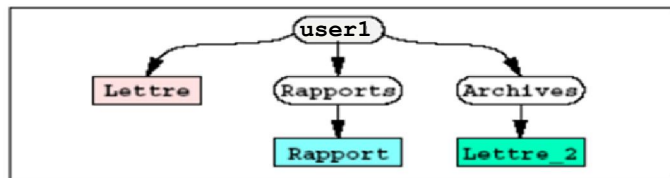
▷ Déplace tous les fichiers de « rep1 » vers « rep2 »

```
mv rep1/* rep2/
```

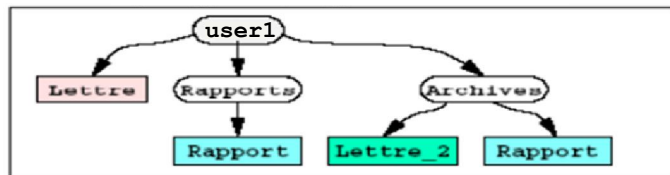
OPÉRATIONS SUR LES FICHIERS

Démonstration:

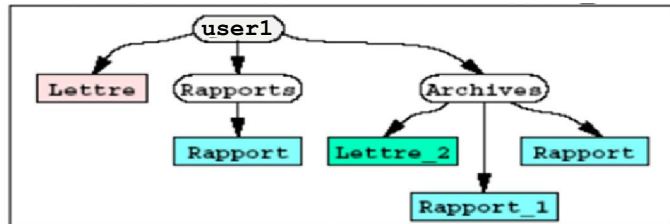
Etat initial



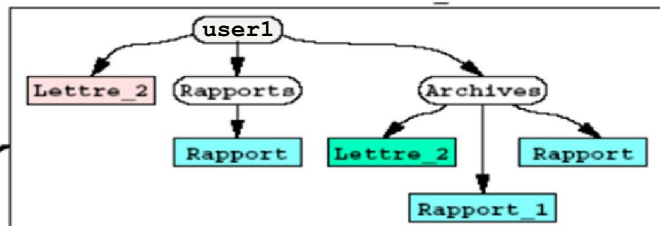
`cp Rapports/Rapport Archives`



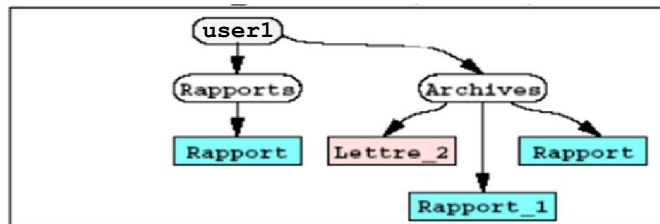
`cp Rapports/Rapport Archives/Rapport_1`



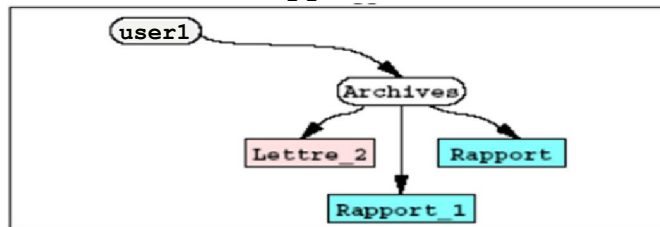
`mv Lettre Lettre_2`



`mv Lettre_2 Archives (écrasement)`



`rm -r Rapports`



OPÉRATIONS SUR LES FICHIERS

Inode

- Un « inode » est une structure de données concernant un fichier
- Contient des informations sur :
 - ▷ Les droits, le propriétaire et le groupe
 - ▷ Le périphérique qui le contient
 - ▷ Des données relatives au système de fichiers et à l'emplacement du fichier sur le support de stockage
- A chaque fichier, correspond un « inode »
 - ▷ Il est unique pour le périphérique de stockage qui contient le fichier
- Pour connaître l'inode d'un fichier, la commande « **ls** » avec l'option « **-i** ».

```
najat@Nom:~/Documents/essais$ ls -il /home/najat/Documents
total 12
8246  drwxr-xr-x 2 najat najat 4096 2008-08-25 15:08 essais
467165 drwxr-xr-x 3 najat najat 4096 2008-06-02 14:20 software
475969 drwxr-xr-x 3 najat najat 4096 2008-07-29 15:54 vmware-tools
```

inode

OPÉRATIONS SUR LES FICHIERS

Inode

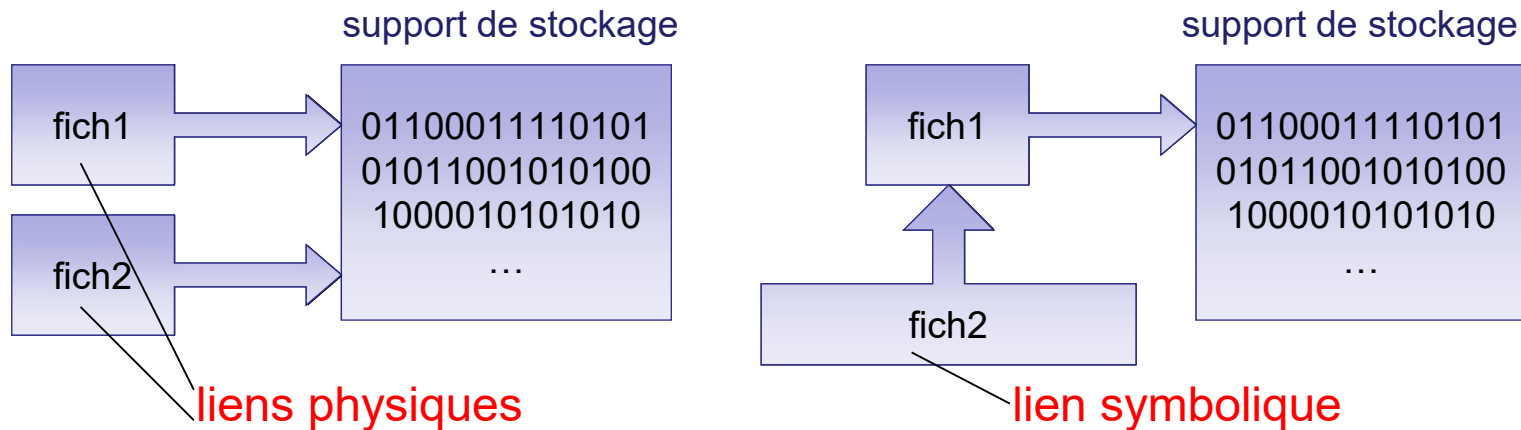
■ Les informations stockées dans un inode disque sont :

- ▷ utilisateur propriétaire
- ▷ groupe propriétaire
- ▷ type de fichier : fichier ordinaire, répertoire, etc.
- ▷ droits d'accès
- ▷ son nombre de liens physiques
- ▷ date de dernier accès
- ▷ date de dernière modification
- ▷ date de dernière modification de l'inode
- ▷ taille du fichier
- ▷ adresses des blocs-disque contenant le fichier.
- ▷ diverses informations parmi lesquelles des pointeurs sur le contenu du fichier

OPÉRATIONS SUR LES FICHIERS

Les liens

- Un lien est un type spécial de fichier qui fait référence à un autre fichier
- Axe central du fonctionnement de Linux, le lien permet :
 - De créer des raccourcis vers des fichiers existants
 - D'éviter de stocker plusieurs fois le même fichier dans des répertoires différents
- Un petit dessin :



OPÉRATIONS SUR LES FICHIERS

Les liens symboliques

- Le lien symbolique est une référence vers un fichier cible
 - ▷ Lorsque le fichier cible est effacé, le lien est rompu
 - ▷ Lorsque le lien est effacé, le fichier cible n'est pas effacé

■ Exemple :

```
najat@Nom:~/Documents$ ls -l
total 8
lrwxrwxrwx 1 najat najat 29 2008-08-25 14:23 ip -> /proc/sys/net/ipv4/ip_forward
drwxr-xr-x 3 najat najat 4096 2008-06-02 14:20 software
drwxr-xr-x 3 najat najat 4096 2008-07-29 15:54 vmware-tools
```

Nom du lien

Indique que c'est un lien

Emplacement du vrai fichier

- La commande « **ln** » avec l'option « **-s** » est utilisée pour créer un lien symbolique

```
najat@Nom:~/Documents$ ln -s /proc/sys/net/ipv4/ip_forward ip
```

Cible (Target)

Nom du lien (link name)

OPÉRATIONS SUR LES FICHIERS

Les liens symboliques

- On peut aussi utiliser la commande « **cp** » avec l'option **-s** pour créer un lien symbolique :

```
najat@najat-VirtualBox:~/temp$ ls -l
total 4
-rw-rw-r-- 1 najat najat    0 23:25 16 فېر اير f1
-rw-rw-r-- 1 najat najat    0 23:25 16 فېر اير f2
drwxrwxr-x 2 najat najat 4096 23:27 16 فېر اير temp1
najat@najat-VirtualBox:~/temp$ cp -s /bin/cp copier1
najat@najat-VirtualBox:~/temp$ ls -il
total 4
 632 lrwxrwxrwx 1 najat najat    7 23:38 16 فېر اير copier1 -> /bin/cp
 621 -rw-rw-r-- 1 najat najat    0 23:25 16 فېر اير f1
 627 -rw-rw-r-- 1 najat najat    0 23:25 16 فېر اير f2
25434 drwxrwxr-x 2 najat najat 4096 23:27 16 فېر اير temp1
najat@najat-VirtualBox:~/temp$
```

OPÉRATIONS SUR LES FICHIERS

Les liens physiques

- Un lien physique est associé à un emplacement sur le support de stockage
 - ▷ 2 liens peuvent être associés au même « inode »
 - ▷ Similaire à la notion de « pointeurs » du langage C
 - ▷ Deux liens physiques sont considérés comme 2 fichiers indépendants (2 i-nodes différents)
 - ▷ Même si leur contenu est au même emplacement sur le support
 - ▷ Le lien physique est vu comme un fichier régulier

- Créer un lien physique avec la commande « **ln** » :

```
najat@Nom:~/Documents$ ln /home/najat/Documents/rapport2007-2008.doc rap0708
najat@Nom:~/Documents/essais$ ls -il
total 176
470930 -rw-r--r-- 2 najat najat 84091 2008-08-25 14:48 rap0708
470930 -rw-r--r-- 2 najat najat 84091 2008-08-25 14:48 rapport-annee2007_2008.doc
```

nom du fichier

cible

L' « inode » est identique. Il s'agit bien de liens physiques

Nombre de liens vers cet inode. C'est un indice permettant de supposer qu'il s'agit d'un lien

OPÉRATIONS SUR LES FICHIERS

Les liens physiques

- L'utilisation la plus courante d'un lien physique concerne les fichiers exécutables.
- Cela permet au même code exécutable d'avoir plusieurs noms sans occuper d'espace disque supplémentaire.
- Les commandes **gzip**, **gunzip** et **zcat** sont sur Mandriva des noms désignant le même inode.
- Le code correspondant adapte son comportement selon le nom par lequel il a été appelé, et n'admet pas les mêmes options.

OPÉRATIONS SUR LES FICHIERS

Exercices:

- Où seront copiés les fichiers ? (chemin absolu)

```
hiba@Nom:~/Documents$ cp /usr/docs-ensa/* ensa/
```

▷ **Réponse : Dans « /home/hiba/Documents/ensa/ »**

- Que risque t-on d'effacer ?

```
root@Nom:/# rm -r ./*
```

▷ **Réponse : Tout le système de fichiers !!! (en tant que root)**

- Par quelle commande peut-on remplacer la suivante afin d'éviter d'avoir plusieurs copies des mêmes fichiers (et économiser l'espace disque)?

```
najat@Nom:~$ cp -r /usr/docs-ensa ./
```

▷ **Réponse :**

```
najat@Nom:~$ ln -s /usr/docs-ensa docs-ensa
```

- Comment copier des fichiers en conservant les droits des fichiers sources?

▷ **Réponse : Utiliser l'option -a de la commande « cp »**

1

Système de fichiers

2

Opérations sur les répertoires

3

Opérations sur les fichiers

4

Sauvegarde d'une arborescence

SAUVEGARDE D'UNE ARBORESCENCE

Avec tar (le plus couramment utilisé)

■ Syntaxe: **tar -cvf** sauvegarde.tar /chemin/arborescence

- ▷ -c : Création d'une archive.
- ▷ -v : Affiche les fichiers en cours de traitement (optionnel).
- ▷ -f : Spécifie le nom de l'archive.

■ Avec compression (gzip ou bzip2) :

- ▷ **tar -czvf** sauvegarde.tar.gz /chemin/arborescence # Compression avec gzip
- ▷ **tar -cjvf** sauvegarde.tar.bz2 /chemin/arborescence # Compression avec bzip2

SAUVEGARDE D'UNE ARBORESCENCE

Avec cpio (utile pour les sauvegardes système)

■ Syntaxe: `find /chemin/arborescence -print | cpio -ov > sauvegarde.cpio`

- ▷ **find** : Liste tous les fichiers de l'arborescence.
- ▷ **cpio -o** : Crée une archive.
- ▷ **-v** : Affiche les fichiers traités.
- ▷ **-print**: Affiche (ou envoie en sortie) la liste des fichiers et répertoires trouvés.

■ Avec compression:

- ▷ `find /chemin/arborescence -print | cpio -ov | gzip > sauvegarde.cpio.gz`

SAUVEGARDE D'UNE ARBORESCENCE

Avec rsync (idéal pour sauvegardes incrémentielles)

■ Syntaxe: `rsync -av /chemin/arborescence /destination`

- ▷ **-a** : Mode archive (préserve les permissions, les liens, etc.).
- ▷ **-v** : Mode verbeux (affiche les fichiers en cours de copie).

■ Avec compression et transfert distant:

- ▷ `rsync -avz /chemin/arborescence user@serveur:/destination`
- ▷ **-z** : Compression des fichiers avant envoi.

SAUVEGARDE D'UNE ARBORESCENCE

Avec dd (copie bas niveau, utile pour sauvegarder un disque ou partition)

■ Syntaxe: **dd if=/dev/sdX of=sauvegarde.img bs=4M status=progress**

- ▷ **if=/dev/sdX** : Source (remplacez sdX par votre partition).
- ▷ **of=sauvegarde.img** : Destination du fichier image.
- ▷ **bs=4M** : Taille des blocs pour optimiser la vitesse.
- ▷ **status=progress** : Affiche la progression.

SAUVEGARDE D'UNE ARBORESCENCE

Exercices:

- Vous devez sauvegarder le répertoire **/home/user/docs** dans un fichier compressé **backup.tar.gz** situé dans **/backup**.

Réponse:

```
tar -czvf /backup/backup.tar.gz /home/user/docs
```

- Vous souhaitez synchroniser le contenu du répertoire **/data** vers **/backup/data** en conservant les permissions et en copiant uniquement les fichiers modifiés.

Réponse:

```
rsync -avz /data/ /backup/data/
```

- Vous devez sauvegarder uniquement les fichiers **.txt** du répertoire **/home/user/documents** dans un fichier sauvegarde.cpio.

Réponse:

```
find /home/user/documents -name "*.txt" | cpio -ov > sauvegarde.cpio
```

CHAPITRE 3: PROCESSUS DE BOOT

1

Introduction

2

Étape 1 : BIOS/UEFI & Bootloader

3

Étape 2 : Chargement du noyau

4

Étape 3 : Initialisation du système

5

Étape 4 : Lancement des services

6

Étape 5: Activation de l'interface utilisateur

7

Étape 6 : Arrêt ou redémarrage

INTRODUCTION

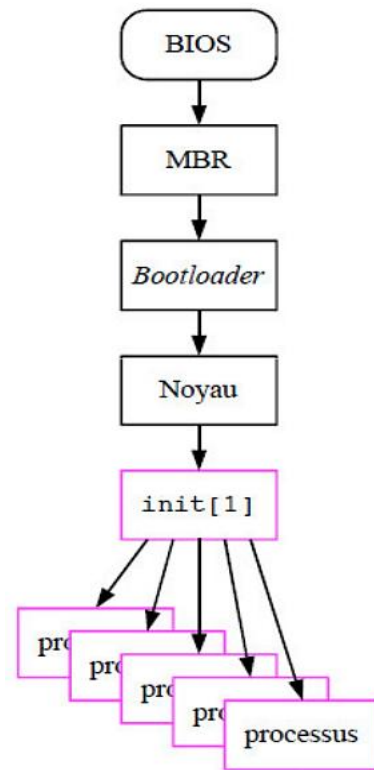
Processus démarrage / d'arrêt :

■ Le démarrage d'un système Linux ou Windows suit un enchaînement de phases bien définies :

1. Initialisation du **BIOS/UEFI**
2. Chargement du Bootloader (**GRUB, LILO, Windows Boot Manager**)
3. Initialisation du noyau (**Kernel**)
4. Démarrage des services et de l'environnement utilisateur

■ Lorsqu'un utilisateur arrête le système, plusieurs étapes sont effectuées :

1. Fermeture des sessions utilisateur
2. Arrêt des services
3. Dé-montage des systèmes de fichiers
4. Extinction du matériel



INTRODUCTION

SysVinit et Systemd

- **SysVinit** et **Systemd** sont deux versions du système d'Initialisation (ou init systems) utilisés pour gérer le démarrage, l'arrêt et l'exécution des services sous Linux.
- SysVinit : Système d'initialisation traditionnel basé sur des scripts Shell et des runlevels.
- Systemd : Système plus moderne, basé sur des unités (units) et des targets, offrant une gestion plus rapide et avancée des services.
- La transition de SysVinit vers Systemd a commencé autour de 2010-2014:
 - ▷ Debian (8 "Jessie", 2015) → Migration officielle vers Systemd.
 - ▷ Red Hat (RHEL 7, 2014) → Abandon de SysVinit pour Systemd.
 - ▷ Ubuntu (15.04, 2015) → Remplacement d'Upstart par Systemd.

INTRODUCTION

SysVinit et Systemd

■ SysVinit était limité :

- ▷ Il exécutait les services de manière séquentielle (un après l'autre)
- ▷ Mauvaise gestion des dépendances entre services.
- ▷ Pas de supervision native des processus.

■ Systemd apporte des améliorations :

- ▷ Démarrage parallèle → Plus rapide
- ▷ Gestion avancée des services et dépendances automatiques.
- ▷ Supervision améliorée avec systemctl et journalctl.
- ▷ Centralisation des logs et journalisation plus efficace.

ÉTAPE 1 : BIOS/UEFI & BOOTLOADER

- Lorsque l'ordinateur est allumé, le **BIOS** ou l'**UEFI** prend le contrôle.
- Il effectue un **POST** pour vérifier l'état des composants matériels (RAM, processeur, disque dur, clavier...).
- Le BIOS exécute le **MBR** (Master Boot Record) situé sur le premier secteur (512 octets) du support bootable choisi (HDD, SSD, clé USB, CD-ROM, réseau...).
- Le **MBR**:
 - ▷ Scanne le disque pour trouver la partition bootable (flag)
 - ▷ Lance le bootloader du secteur de boot (premier secteur) de la partition bootable
- Le chargeur de démarrage (GRUB2) charge le noyau Linux (**vmlinuz**) en mémoire et initialise l'image du disque virtuel (**initramfs**).
- 2 bootloader possibles pour linux:
 - ▷ **Lilo**: Linux Loader
 - ▷ **Grub**: Grand Unified Bootloader
- Sous Windows, il s'agit du **Windows Boot Manager** (winload.exe).

ÉTAPE 2 : CHARGEMENT DU NOYAU

Commun à SysVinit et systemd

- Le noyau du système d'exploitation est décompressé et chargé en mémoire.
- Il détecte le matériel et initialise les pilotes nécessaires (USB, GPU, disque...).
- Il monte le système de fichiers racine (/ sous Linux, C:\ sous Windows).
- Sur Linux, il active le processus **init** ou **systemd (PID 1)**, tandis que sous Windows, le processus **ntoskrnl.exe** est lancé.

ÉTAPE 3 : INITIALISATION DU SYSTÈME

SysVinit

- Le noyau après le chargement des périphérique, lance **/sbin/init (PID 1)**, Seul et unique processus lancé par le Kernel,
- **Init** a pour tâche :
 - ▷ lancer chacun des processus;
 - ▷ les sessions de login;
 - ▷ Il doit aussi collecter les zombies (processus orphelins);
 - ▷ gérer l'arrêt du système;
 - ▷ prend en charge la suite des opérations;
 - ▷ lit le fichier de configuration **/etc/inittab**
- Par la suite le Kernel n'interviendra plus que pour répondre aux appels système.

ÉTAPE 3 : INITIALISATION DU SYSTÈME

SysVinit

- Init lit `/etc/inittab` pour déterminer le niveau d'exécution (runlevel)
- Le fichier **`/etc/inittab`** est un fichier texte composé de lignes de commentaires (commençant par `#`) et d'entrées constituées de 4 champs délimités par des deux points, elles sont de la forme :

```
id:runlevels:action:process [arguments]
```

- **Id**: Identifiant unique qui comprend de 1 à 4 caractères alphanumériques qui identifie l'entrée dans le fichier `/etc/inittab`.
- **runlevels**: Définit la liste des niveaux d'exécution pour lesquelles cette ligne s'applique.
- **action**: Méthode ou manière d'exécuter la commande spécifiée dans le champ suivant.
- **process**: Chemin de la commande à lancer avec ses paramètres pour les niveaux d'exécution définis précédemment.

ÉTAPE 3 : INITIALISATION DU SYSTÈME

SysVinit: Les niveaux de fonctionnement (runlevels)

■ Il existe huit runlevels définis sous Linux dont quatre réservés (0, 1, 6, S ou s)

NIVEAU	DESCRIPTION
0	Halt - Arrêt de la machine.
1	Single user- Mode mono-utilisateur ou maintenance.
2	Multi-utilisateurs sans le support NFS.
3	Multi-utilisateurs.
4	Libre.
5	X11 (Multi-utilisateurs avec graphique).
6	Reboot - redémarrage de la machine.
S ou s	Mode mono-utilisateur dans lequel seule la partition racine est montée.

■ Exemple: **id:3:initdefault:**

▸ Signifie que le système démarrera en Mode multi-utilisateurs complet.

ÉTAPE 3 : INITIALISATION DU SYSTÈME

SysVinit: Les niveaux de fonctionnement (runlevels)

■ Un certain nombre de commandes exécutées par Init se trouvent dans **/etc/rc.d** :

▷ **ls -l /etc/rc.d/**

```
drwxr-xr-x. 2 root root 4096 24 sept. 19:34 init.d
-rwxr-xr-x. 1 root root 2625 27 sept. 2010 rc
drwxr-xr-x. 2 root root 4096 24 sept. 20:24 rc0.d
drwxr-xr-x. 2 root root 4096 24 sept. 20:24 rc1.d
drwxr-xr-x. 2 root root 4096 24 sept. 20:24 rc2.d
drwxr-xr-x. 2 root root 4096 24 sept. 20:24 rc3.d
drwxr-xr-x. 2 root root 4096 24 sept. 20:24 rc4.d
drwxr-xr-x. 2 root root 4096 28 sept. 15:07 rc5.d
drwxr-xr-x. 2 root root 4096 24 sept. 20:24 rc6.d
-rwxr-xr-x. 1 root root 220 27 sept. 2010 rc.local
-rwxr-xr-x. 1 root root 19021 27 sept. 2010 rc.sysinit
```

ÉTAPE 3 : INITIALISATION DU SYSTÈME

SysVinit: Les niveaux de fonctionnement (runlevels)

- Un certain nombre de commandes exécutées par Init se trouvent dans **/etc/rc.d** :
 - ▶ Le programme Init exécute le script **/etc/rc.d/rc.sysinit** avant tous les autres. Il s'occupe de tout ce qui doit être fait sur le système au moment de son initialisation.
 - ▶ Le répertoire **/etc/rc.d/init.d** contient les scripts Shell permettant de lancer tous les services sur le système.
 - ▶ Chaque script est utilisé pour démarrer ou arrêter un service particulier.
 - ▶ La commande **/etc/rc.d/rc** est lancée pour chaque niveau d'exécution avec son numéro en argument
 - ▶ Exemple :

```
l2:2:wait:/etc/rc.d/rc 2
```

- ▶ Signifie que Init doit lancer le script **/etc/rc.d/rc** en lui passant 2 en paramètre à chaque fois qu'on entre dans le niveau 2 et qu'il doit attendre la terminaison de ce script avant de poursuivre

ÉTAPE 3 : INITIALISATION DU SYSTÈME

SysVinit: Les niveaux de fonctionnement (runlevels)

■ Un certain nombre de commandes exécutées par Init se trouvent dans **/etc/rc.d** :

- ▷ les répertoires **rc0.d à rc6.d** ne contiennent que des liens symboliques, avec des noms très précis.
- ▷ Ces liens pointent vers des scripts réels dans /etc/init.d/.
- ▷ Les noms des liens commencent par "S" (démarrer) ou "K" (arrêter), suivis d'un ordre numérique.
- ▷ **Exemple: Imaginons que nous ayons un script de service réel** /etc/init.d/httpd (serveur Apache).
- ▷ Dans /etc/rc3.d/, nous pouvons trouver un lien symbolique comme ceci :

```
ls -l /etc/rc3.d/
```

```
lrwxrwxrwx 1 root root 14 Jan 10 10:00 S20httpd -> ../init.d/httpd
```

S20httpd est un
lien symbolique
vers /etc/init.d/httpd

- ▷ Le préfixe "S" (Start) indique que le service doit être démarré à ce niveau d'exécution (rc3.d correspond au mode multi-utilisateur sans interface graphique).
- ▷ Le numéro "20" indique l'ordre de lancement

ÉTAPE 3 : INITIALISATION DU SYSTÈME

SysVinit: Les niveaux de fonctionnement (runlevels)

- Un certain nombre de commandes exécutées par Init se trouvent dans **/etc/rc.d** :
 - ▷ **rc.local** est appelé en dernier, c'est-à-dire après tous les autres scripts
 - ▷ Ce script peut être utilisé par l'administrateur pour exécuter certaines tâches une seule fois à la fin du niveau d'exécution correspondant.
 - ▷ Exemple : après chaque démarrage du serveur on voudra démarrer la base de données.

ÉTAPE 3 : INITIALISATION DU SYSTÈME

SysVinit: action

- Parmi les plus importantes directives pour le troisième champ « action », on trouve:
- **Respawn**: Le processus sera relancé s'il se termine. Utilisé principalement avec mingetty pour assurer la gestion des terminaux texte .
- **Once**: Le processus n'est exécuté qu'une seule fois.

```
x:5:once:/etc/X11/prefdm nodaemon
```
- **Wait**: Identique à la directive précédente mais ici, Init attend que le processus soit terminé avant de passer à la ligne suivante
- **Boot**: Le processus est exécuté au démarrage du système, le champ runlevels est ignoré.
- **Powerfail**: La commande est exécutée lorsque l'alimentation est sur le point d'être interrompue.

ÉTAPE 3 : INITIALISATION DU SYSTÈME

Systemd

- **/sbin/init** est remplacé par **systemd**.
- Il utilise des unités systemd (services, cibles, sockets, etc.) stockées dans **/lib/systemd/system/** et **/etc/systemd/system/**
- Les runlevels sont remplacés par des targets (*.target) :
 - ▷ **poweroff.target** (équivalent à runlevel 0)
 - ▷ **rescue.target** (équivalent à runlevel 1)
 - ▷ **multi-user.target** (équivalent à runlevel 3)
 - ▷ **graphical.target** (équivalent à runlevel 5)
 - ▷ **reboot.target** (équivalent à runlevel 6)
- Les services sont démarrés en parallèle
- Les targets permettent de démarrer des groupes de services plus spécifiques qu'un simple runlevel.

ÉTAPE 3 : INITIALISATION DU SYSTÈME

Systemd

- Voir le niveau de fonctionnement actuel :
 - ▷ **systemctl get-default**
- Changer temporairement le mode de fonctionnement :
 - ▷ **systemctl isolate rescue.target**
- Définir un mode de fonctionnement par défaut :
 - ▷ **systemctl set-default graphical.target**

ÉTAPE 3 : INITIALISATION DU SYSTÈME

Systemd: /lib/systemd/system/

- Le répertoire ***/lib/systemd/system/*** contient les fichiers unitaires de systemd, qui définissent les services, les cibles (targets), les sockets et autres unités du système.
 - Ces fichiers sont généralement fournis par les paquets logiciels installés et ne doivent pas être modifiés directement.
- ▷ **Services (*.service)** : Définit les services système (ex : ssh.service, networkd.service).
 - ▷ **Cibles (*.target)** : Remplace les runlevels de SysVinit (ex : multi-user.target, graphical.target).
 - ▷ **Sockets (*.socket)** : Gère les services activés par socket (ex : cups.socket).
 - ▷ **Timers (*.timer)** : Définit les tâches planifiées (ex : logrotate.timer).

ÉTAPE 3 : INITIALISATION DU SYSTÈME

Systemd: /etc/systemd/system/

- Le répertoire ***/etc/systemd/system/*** contient les fichiers unitaires personnalisés de systemd.
- Contrairement à ***/lib/systemd/system/***, ce répertoire est destiné aux modifications locales et a une priorité plus élevée, ce qui signifie que les fichiers ici écrasent ceux de ***/lib/systemd/system/*** s'ils portent le même nom.
 - **Services personnalisés (*.service)** : Définition de services créés ou modifiés localement.
 - **Cibles personnalisées (*.target)** : Personnalisation des niveaux de fonctionnement.
 - **Liens symboliques vers des unités activées** : Lorsque tu actives un service avec `systemctl enable`, un lien symbolique est créé ici.

ÉTAPE 4 : LANCEMENT DES SERVICES

Sysvinit

- Les scripts de démarrage (/etc/init.d/) sont exécutés séquentiellement, ce qui ralentit le démarrage.
- Pas de gestion avancée des dépendances.

ÉTAPE 4 : LANCEMENT DES SERVICES

Systemd

- Démarrage parallèle des services pour accélérer le boot.
- Gestion dynamique des services avec dépendances explicites (via After=, Requires=).
- Possibilité de démarrer des services à la demande grâce aux sockets.

ÉTAPE 5 : ACTIVATION DE L'INTERFACE UTILISATEUR

Sysvinit

- Lance getty pour afficher l'invite de connexion en mode texte.
- Si runlevel 5, il démarre un gestionnaire d'affichage graphique (GDM, SDDM, etc.).

ÉTAPE 5 : ACTIVATION DE L'INTERFACE UTILISATEUR

Systemd

- Démarre **multi-user.target** pour un environnement en ligne de commande.
- Lance **graphical.target** si une interface graphique est requise.

ÉTAPE 6 : ARRÊT OU REDÉMARRAGE

■ Pour éviter toute mauvaise surprise comme la perte de données. Il faut exécuter un certain nombre de tâches avant de couper le courant :

- ▷ Prévenir les utilisateurs connectés au système de l'arrêt imminent de la machine.
- ▷ Arrêter tous les services.
- ▷ Vider les buffers sur le disque et démonter les systèmes de fichiers.
- ▷ Sans cela, les données se trouvant dans le cache et non encore écrites sur les unités de stockage seront perdues.

ÉTAPE 6 : ARRÊT OU REDÉMARRAGE

Sysvinit: Shutdown

- La commande shutdown permet d'arrêter, de redémarrer et de passer le système en mode maintenance.
- Elle offre la possibilité de programmer cette opération à une date précise et d'en informer les utilisateurs.
- Si l'arrêt du système est prévu dans moins de cinq minutes, la commande shutdown empêchera tout utilisateur, autre que root de se connecter.
- Sa syntaxe est la suivante: `/sbin/shutdown [-t sec] [-arkhncfF] heure [messages]`
 - ▷ L'heure peut être spécifiée de plusieurs manières :
 - ▷ **hh:mm** heure à laquelle l'opération est programmée
 - ▷ **[+]m** nombre de minutes avant que l'opération soit effectuée
 - ▷ **now** l'opération doit être immédiate (alias de +0)

ÉTAPE 6 : ARRÊT OU REDÉMARRAGE

Sysvinit: Shutdown

■ Les options à retenir sont :

- ▷ **-h (halt)** arrêter le système.
- ▷ **-r (reboot)** redémarrer le système.
- ▷ **-c (cancel)** annuler l'opération d'arrêt ou de redémarrage programmée.
- ▷ **-f** effectuer un redémarrage rapide sans vérification des systèmes de fichiers.
- ▷ **-F** forcer la vérification des systèmes de fichiers au prochain démarrage

■ Exemple la commande suivante permet de mettre le système en mode maintenance dans exactement 10 minutes:

shutdown -r +10 "Passage en mode maintenance dans 10 minutes"

ÉTAPE 6 : ARRÊT OU REDÉMARRAGE

Sysvinit

- Utilisation de **telinit 0** (arrêt) ou **telinit 6** (redémarrage).
- Exécution des scripts **/etc/rc0.d/** ou **/etc/rc6.d/** séquentiellement.

ÉTAPE 6 : ARRÊT OU REDÉMARRAGE

Systemd

■ Commandes simplifiées :

```
systemctl poweroff  # Éteindre  
systemctl reboot    # Redémarrer
```

■ Fermeture parallèle des services, ce qui accélère l'arrêt.

■ Exemple la commande suivante permet de mettre le système en mode maintenance dans exactement 10 minutes:

```
systemctl rescue --timer=10min
```

- ▷ **systemctl rescue**: Bascule le système en mode maintenance (rescue.target), où seul un shell de récupération est actif.
- ▷ **--timer=10min**: Planifie l'exécution de la commande dans 10 minutes.