

## CHAPITRE 6 : Gestion des Systèmes des fichiers Sous Linux

**Mohammed SABER**

Département Électronique, Informatique et Télécommunications  
Ecole Nationale des Sciences Appliquées "ENSA"  
Université Mohammed Premier OUJDA

Année Universitaire : 2018-2019

### Plan de chapitre

- 1 Introduction
- 2 Les principaux types de systèmes de fichiers
- 3 Les commandes de gestion des système de fichiers
- 4 Le montage d'un système de fichiers
- 5 Automatiser le montage des système de fichiers
- 6 Les quotas
- 7 Gestion des journaux (logs)

### Plan de chapitre

- 1 Introduction
- 2 Les principaux types de systèmes de fichiers
- 3 Les commandes de gestion des système de fichiers
- 4 Le montage d'un système de fichiers
- 5 Automatiser le montage des système de fichiers
- 6 Les quotas
- 7 Gestion des journaux (logs)

### Introduction

- Un système de fichiers est une structure dotée d'une organisation hiérarchique permettant de stocker des fichiers sur toute unité de type bloc ; en particulier les durs, les volumes RAID et LVM.
- L'arborescence de fichiers Linux peut être composée de plusieurs systèmes de fichiers stockés sur des unités de bloc distinctes.
- Chaque système de fichiers impose son propre format d'enregistrement des fichiers sur le périphérique de stockage.
- Linux supporte plusieurs systèmes de fichiers grâce à l'utilisation d'un système de fichiers virtuel (VFS) acceptant tous les appels système de base à la manipulation des fichiers sous Linux.



## Plan de chapitre

- 1 Introduction
- 2 Les principaux types de systèmes de fichiers
- 3 Les commandes de gestion des système de fichiers
- 4 Le montage d'un système de fichiers
- 5 Automatiser le montage des système de fichiers
- 6 Les quotas
- 7 Gestion des journaux (logs)

## Les principaux types de systèmes de fichiers

### Les systèmes de fichiers de MSDOS et de Windows

- Linux supporte tous les systèmes de fichiers de Microsoft.
- Il faut simplement noter que le module nécessaire à NTFS n'est généralement pas pris en charge par le noyau (anciennes version) qui vous est fourni. Cela implique de recompiler le noyau en cochant la case demandant le support de NTFS.
- VFAT (FAT 32), NTFS, ....

### Les systèmes de fichiers distribués

- **NFS** : Le système de fichiers distribués mis par la société SUN dans le domaine public. Il permet d'accéder, de façon transparente, à des disques d'ordinateurs distants.
- **SMB** : Le système de fichiers qui implémente le protocole SMB de Microsoft pour accéder à des disques partagés d'ordinateurs distants fonctionnant avec Windows.

### Autres systèmes de fichiers

**ISO 9660** : Le système de fichiers pour CD-ROM, conforme à la norme ISO

## Les principaux types de systèmes de fichiers

### Les premiers systèmes de fichiers de Linux

- **minix** : Le premier système de fichiers de Linux. Bien que limité, il est pratique pour les systèmes de fichiers sur disquette ou en RAM. On peut cependant le considérer comme obsolète.
- **ext** : La version améliorée de Minix.
- **ext2** : C'était jusqu'à il y a peu le standard des systèmes de fichiers sous Linux. Bien que son usage tende fortement à diminuer, il reste incontournable. Il est reconnu par toutes les distributions

### Les systèmes de fichiers journalisés

- Les systèmes de fichiers journalisés sont plus performants et plus sûrs.
- Ils sont plus performants de par leur organisation interne, par exemple l'organisation des répertoires, et plus sûrs en supprimant de nombreux risques d'incohérences en cas de problème.
- Ext3 (Linux), Ext4 (Linux), jfs (AIX IBM) ....

## Plan de chapitre

- 1 Introduction
- 2 Les principaux types de systèmes de fichiers
- 3 Les commandes de gestion des système de fichiers
- 4 Le montage d'un système de fichiers
- 5 Automatiser le montage des système de fichiers
- 6 Les quotas
- 7 Gestion des journaux (logs)

## Les commandes de gestion des système de fichiers

- Les commandes de gestion de systèmes de fichiers sont nombreuses.
- Dans Linux, on trouve deux familles de commandes :
  - Les commandes génériques héritées du système UNIX, comme `mkfs` et `fsck` :
    - `mkfs` : interface commune pour plusieurs versions spécialisées : `mkfs.ext2`, `mkfs.ext3`, `mkfs.vfat`, ...
    - `mke2fs` : commande spécialisée dans le formatage de filesystem `ext2` ou `ext3`.
    - `tune2fs` : commande permettant la modification des paramètres d'un filesystem `ext2` ou `ext3` (Option `-j` : ajoute un journal à un fichier `ext2`).
  - Les commandes propres à un système de fichiers.

## Création d'un système de fichiers

- Parmi les commandes standard d'UNIX, la commande `mkfs` est bien sûr la plus importante.
- C'est elle qui crée un système de fichiers sur un disque.
- Elle comporte les trois paramètres fondamentaux d'un système de fichiers :
  - Le nom du disque où le système de fichiers doit être créé.
  - Le type du système de fichiers à créer, pour `mkfs`.
  - La taille du système de fichiers à créer, si l'on ne veut pas utiliser tout le disque.
- Création d'une partition : `fdisk`

```
root@Nom_machine : #fdisk -/dev/sda
Commande (m pour l'aide) : n
Action de commande : e étendue ou p partition primaire (1-4) : p
Numéro de partition (1-4) : 1
Premier cylindre (1-26, par défaut 1) : (Utilisation de la valeur par défaut 1)
Dernier cylindre ou +taille or +tailleM ou +tailleK (1-26, par défaut 26) : +100M
Commande (m pour l'aide) : w
La table de partitions a été altérée ! Synchronisation des disques.
Commande (m pour l'aide) : q
```

## Création d'un système de fichiers

- Vérification de la partition sous «`fdisk`».

```
Commande (m pour l'aide) : p
Périphérique Amorces Début Fin Blocs Id; Système
/dev/sda1              1    13 104391 83 Linux
```

- Initialisation du système de fichiers : Avec la commande «`mkfs`»

```
root@Nom_machine : #mkfs.ext3 -/dev/sda1
mke2fs 1.40.8 (09-Oct-2017)
Étiquette de système de fichiers=
Type de système d'exploitation : Linux
Taille de bloc=1024 (log=0)
Taille de fragment=1024 (log=0)
26104 i-noeuds, 104388 blocs
5219 blocs (5.00%) réservés pour le super utilisateur
Premier bloc de données=1
Nombre maximum de blocs du système de fichiers=67371008, 13 groupes de blocs
8192 blocs par groupe, 8192 fragments par groupe 2008 i-noeuds par groupe
Superblocs de secours stockés sur les blocs :8193, 24577,
```

## Gestion d'un système de fichiers

- La commande `e2label` affiche ou modifie le label d'un disque. Dans les distributions, comme RedHat ou Fedora, il faut penser à modifier le fichier `/etc/fstab` quand on modifie le label d'un disque.

```
root@Nom_machine : #e2label /dev/sda10
/boot
root@Nom_machine : #e2label /dev/sda10 administrateur
root@Nom_machine : #e2label /dev/sda10
administrateur
```

- Les commandes `du` et `df` permettent de gérer l'espace disque.

```
root@Nom_machine : #du -sh /home/admin
65M /home/admin
```

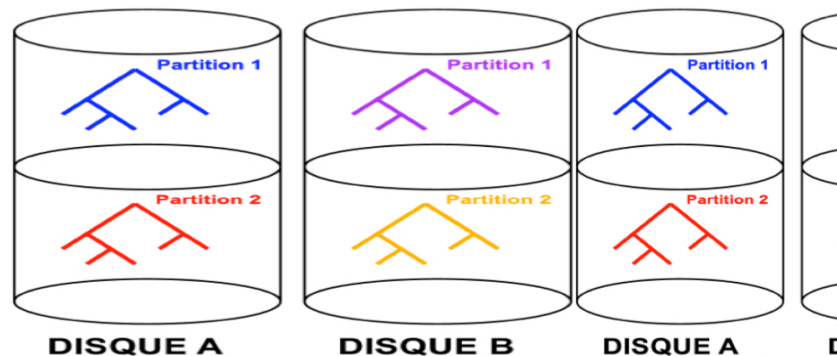
```
Fichier  Édition  Affichage  Terminal  Aide
$ df -h
Sys. de fich.      Tail. Occ. Disp. %Occ. Monté sur
/dev/sdc1          28G   17G   63% /
udev              2.0G   372K   2.0G   1% /dev
none              2.0G   1.2M   2.0G   1% /dev/shm
/dev/sdc3          202G   180G   22G   90% /media/Movies
/dev/sdb2          346G   310G   37G   90% /media/Stockage
/dev/sdb1          120G   86G   35G   71% /media/MP3
/dev/sda3          206G   139G   68G   68% /media/Download
/dev/sda2          200G   126G   75G   63% /media/Games
/dev/sda1           60G    40G   13G   66% /media/Custom
```

## Réparation d'un système de fichier

- En cas de problème de cohérence sur le système de fichier (qui peuvent arriver même en ext3), la commande `fsck` (File system Check Consistency) permet de réparer un système de fichier.
- `fsck` : interface commune à plusieurs version de `fsck` en fonction du filesystem : `fsck.ext2`, `fsck.ext3`, `fsck.msdos`, `fsck.vfat`.
- `e2fsck` : vérification des filesystem ext2 et ext3.
  - Syntaxe : `e2fsck [options] fichier-special`
  - option « -y » : répond « yes » à toutes les questions posées (utilisation non interactive).
  - option « -p » : répare automatiquement sans poser de questions.
- **Attention** : Ne jamais lancer `fsck` sur un filesystem monté : risque de corruption.

## Le montage d'un système de fichiers

- Dans le système Linux, les fichiers d'un disque, ne sont accessibles que si le disque a été monté.
- La greffe ou l'implémentation d'une partition sur une arborescence s'appelle le «**montage**» d'un système des fichiers.



## Plan de chapitre

- 1 Introduction
- 2 Les principaux types de systèmes de fichiers
- 3 Les commandes de gestion des système de fichiers
- 4 **Le montage d'un système de fichiers**
- 5 Automatiser le montage des système de fichiers
- 6 Les quotas
- 7 Gestion des journaux (logs)

## Le montage d'un système de fichiers

- Le montage d'un système de fichiers consiste à attacher la racine de l'arbre du système de fichiers à un répertoire d'un système de fichiers déjà actif. Cette opération, qui s'appelle le montage du système de fichiers, est réalisée par la commande `mount`.
- Le démontage, c'est à dire la suppression du lien entre le répertoire de montage et le système de fichiers est effectué par la commande `umount`.
- Les fichiers d'un système de fichiers ne sont accessibles, par les commandes usuelles (`cp`, `rm`, `mv`, `cat`, ...), que s'il est monté.

### Remarque

Si le répertoire de montage n'est pas vide au moment de l'exécution de la commande `mount`, les fichiers qu'il contient sont cachés jusqu'au démontage.

## Montage par la commande `mount`

### • Syntaxe :

```
root@hostname : #mount [param] [-t typefs] [-o options]
fichier-spécial point-de-montage
```

```
root@hostname : #mount /dev/sda1 /data
```

- Cette commande «**monte**» la partition `/dev/sda1` «**dans**» le répertoire `/data`.

### Attention

- Le point de montage doit exister avant de lancer la commande.
- Le contenu (éventuel) point de montage devient invisible après montage.
- On ne peut pas monter un filesystem si sa structure interne est incohérente (pour rendre cohérent le filesystem : `fsck`).

## Montage par la commande `mount`

- Utilisée sans argument, elle affiche les systèmes de fichiers montés.

```
File Edit View Search Terminal Help
$ mount
/dev/sda3 on / type ext4 (rw)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
tmpfs on /dev/shm type tmpfs (rw,rootcontext="system_u:object_r:tmpfs_t:s0")
/dev/sda1 on /boot type ext2 (rw)
/dev/sda2 on /spare type ext4 (rw)
/dev/sda8 on /tmp type ext4 (rw)
/dev/sda5 on /usr type ext4 (rw)
/dev/sda6 on /var type ext4 (rw)
$
```

- En fait le contenu du fichier `/etc/fstab` où Linux mémorise les systèmes de fichiers actuellement montés.

## Montage par la commande `mount`

```
root@hostname : #mount [parametres] [-t typefs] [-o options]
Partition-à-montée point-de-montage
```

- Paramètres de la commande `mount` :
  - `-a` : monter tous les FS définis dans `/etc/fstab`
  - `-r` : montage en lecture seule
  - `-w` : montage en lecture/écriture
  - `-L` : utiliser le label du périphérique plutôt que son nom de fichier de périphérique
  - `-U` : utiliser l'UUID du périphérique plutôt que son nom de fichier de périphérique
  - `-t` : Types de filesystem (ext2, ext3, ext4fs, reiserfs, jfs, xfs, vfat, iso9660) (Mount détecte le type de **fs** si pas d'option **t**).
- Options de la commande `mount` :
  - `noauto` : pas de montage automatique au démarrage
  - `user` : montage possible par un utilisateur différent de root. (Utile pour les medias amovibles).
  - `users` : idem sauf que démontage possible par tout utilisateur.
  - `remount` : pour changer des options (rendre écriturable par exemple).
  - `ro` : accès en lecture seule.
  - `rw` : accès en lecture/écriture.
  - `owner` : idem user mais l'utilisateur doit être propriétaire du fichier de périphérique

## Démontage par la commande `umount`

- L'opération duale de `mount` est `umount` pour démonter un système des fichiers.
- Syntaxe :

```
root@hostname : #umount Partition-à-montée or
point-de-montage
```

```
root@hostname : #umount /data
```

- Cette commande «**démonte**» la partition `/dev/sda1` du répertoire `/data`.
- Dans certains cas, on ne peut pas démonter une partition :
  - Si une commande s'exécute dans la partition.
  - Si l'on a un fichier ouvert ou que l'on se trouve dans un répertoire de la partition.

```
root@hostname : #umount /data
umount : /data : périphérique occupé
umount : /data : périphérique occupé
```

## Démontage par la commande `umount`

### fuser

- Cette commande permet d'identifier les fichiers ouverts et quels processus y sont attachés.
- Il suffit d'arrêter le ou les processus qui possèdent des fichiers ouverts sur le système des fichiers pour pouvoir le démonter.

```
root@hostname : #fuser /data
/data/ : 7022c
```

### lsof (list of open files)

Cette commande permet d'identifier les fichiers ouverts et quels processus y sont attachés

```
root@hostname : #lsof /data
COMMAND PID USER  FD  TYPE DEVICE SIZE NODE NAME
Bash      7022 saber  cwd  DIR   7,0  1024   2  /data
```

## Plan de chapitre

- 1 Introduction
- 2 Les principaux types de systèmes de fichiers
- 3 Les commandes de gestion des système de fichiers
- 4 Le montage d'un système de fichiers
- 5 **Automatiser le montage des système de fichiers**
- 6 Les quotas
- 7 Gestion des journaux (logs)

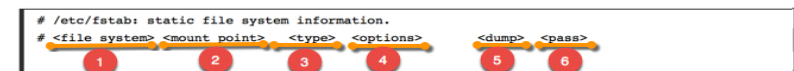
## Automatiser le montage des système de fichiers

- Les disques qui contiennent des systèmes de fichiers doivent, en général, être montés à chaque démarrage du système et démontés à chaque arrêt, le montage étant toujours réalisé sur le même répertoire.
- Pour automatiser ces opérations, l'administrateur d'un système Linux doit modifier le fichier `/etc/fstab` qui contient la liste des disques à monter automatiquement.
- La prise en compte sera réalisée dès le prochain démarrage du système. C'est la seule opération à réaliser !
- C'est la commande `mount -a`, exécutée par les scripts de démarrage, qui prend en compte le contenu du fichier `/etc/fstab`. Ce fichier contient une ligne par système de fichiers.
- Le contenu du fichier `/etc/fstab` est un fichier texte, contient une ligne par système de fichiers.

```
# /etc/fstab: static file system information.
# <file system> <mount point> <type> <options> <dump> <pass>
# / was on /dev/sda5 during installation
UUID=33b870b8-a81e-4203-a4fd-7affa9f412fb / ext4 errors=remount-ro 0 1
# swap was on /dev/sda6 during installation
UUID=2c442228-1991-48c7-bad9-a80dfc8267cf none swap sw 0 0
```

## Automatiser le montage des système de fichiers

Chaque ligne composée par plusieurs champs :



- **Champ 1** : Nom du fichier de périphérique ;
- **Champ 2** : Point de montage ;
- **Champ 3** : Type de système de fichiers ;
- **Champ 4** : Options de montage (**auto** : montage automatique au démarrage ; **noauto** : ne pas monter automatiquement ; **user** : users sont autorisés à monter ce SF ; **ro** : lecture seul ; **rw** : L'attribut le plus fréquent est **rw** (read/write) qui donne un accès complet au système de fichiers) ;
- **Champ 5** : Utilisé pour les sauvegardes (par la commande `dump`) ;
  - 0 : pas de dump ;
  - 1 : dump de la partition ;
- **Champ 6** : Numéro d'ordre du disque lors de la vérification par la commande «`fsck`» ;
  - 0 : pas de `fsck` ;
  - 1 : root ;
  - 2 : les autres ;

## Plan de chapitre

- 1 Introduction
- 2 Les principaux types de systèmes de fichiers
- 3 Les commandes de gestion des système de fichiers
- 4 Le montage d'un système de fichiers
- 5 Automatiser le montage des système de fichiers
- 6 Les quotas**
- 7 Gestion des journaux (logs)

## Mise en œuvre des quotas

La démarche de l'administrateur pour installer des quotas sur un disque est la suivante :

- Monter le système de fichiers avec l'une des options de montage `usrquota` ou `grpquota` ou les deux. Les options `usrquota` et `grpquota` indiquent si les quotas s'appliquent aux utilisateurs ou aux groupes.

```
root@hostname : #mount -o usrquota /dev/hda9 /mnt/TP
root@hostname : #mount
.....
/dev/hda9 on /mnt/TP type ext4 (rw,usrquota)
```

- Créer les fichiers qui mémorisent les quotas des utilisateurs et des groupes. Ces fichiers doivent être créés dans le répertoire de montage du système de fichiers.
- Éditer les quotas des utilisateurs :

```
root@hostname : #edquota -u saber
```

- Éditer les quotas des groupes :

```
root@hostname : #edquota -u ensao
```

## Les quotas

- La mise en œuvre des quotas va permettre à l'administrateur de limiter le nombre de fichiers ou de blocs d'un utilisateur ou d'un groupe, sur un disque.
- Les quotas offrent un plus grand intérêt dans Linux qui est souvent utilisé comme serveur de fichiers (samba) ou comme de messagerie (Sendmail, Postfix, ...).
- Pour les fichiers aussi bien que les blocs, il existe deux limites :
  - La **limite hard** qui est infranchissable. Un utilisateur ou un groupe qui atteint sa limite hard de fichiers ne pourra pas en créer un de plus. L'éditeur de texte `vi` refusera ainsi d'exécuter la commande de sauvegarde : `w`.
  - La **limite soft** peut être franchie pendant un certain nombre de jours consécutifs. Si, au terme de ce laps de temps, l'utilisateur n'est pas redescendu en dessous de sa limite soft, le point atteint devient à son tour infranchissable, jusqu'au retour à la normale.

## Mise en œuvre des quotas

- L'administrateur peut aussi, via la commande `edquota -t`, modifier les valeurs par défaut des quotas, et en particulier, le nombre de jours où la **limite soft** peut être dépassée.
- Activer les quotas. La commande `quotaon` réalise cette opération. Désactiver les quotas. La commande `quotaoff` réalise cette opération.
- La commande `quota` :
  - Informations sur le quota d'un utilisateur ;
  - Peut être exécuter par tous utilisateur ;
  - Un utilisateur ne peut visualiser que son quota ;
- La commande `repquota` :
  - Informations sur les quotas de tous les utilisateurs et groupes ;
  - Uniquement **root** peut l'exécuter ;
  - D'autres information voir `man quota`.



## Plan de chapitre

- 1 Introduction
- 2 Les principaux types de systèmes de fichiers
- 3 Les commandes de gestion des système de fichiers
- 4 Le montage d'un système de fichiers
- 5 Automatiser le montage des système de fichiers
- 6 Les quotas
- 7 Gestion des journaux (logs)

## Examen des journaux

Chaque évènement (1 évènement par ligne) enregistré dans un journal contient :

- Date et heure ;
- Nom machine à l'origine du message ;
- Service ou utilisateur qui génère le message ;
- Texte du message ;

```
Nov 1 22 :09 :02 localhost rsyslogd : [origin software="rsyslogd"
swVersion="2.0.2" xpid=" 2206"
xinfo="http ://www.rsyslog.com"] [xconfigInfoudpReception="No"udpPort="514"
tcpReception="No" tcpPort="0"] restart
Nov 1 22 :45 :50 localhost gconfd (franck3194) : Sortie
Nov 1 22 :45 :50 localhost shutdown[9195] : shutting down for system halt
Nov 1 22 :45 :50 localhost NetworkManager : <info> Deactivating device eth1.
Nov 1 22 :45 :50 localhost NetworkManager : <info> eth1 : canceled DHCP
transaction,dhclient pid 3456
Nov 1 22 :45 :51 localhost kernel : ipw2200 : Failed to send ASSOCIATE :
Already sending a command.
```

## Gestion des journaux (logs)

- Journaux ou fichiers de log : fichiers textes ;
- Localisés dans le répertoire `/var/log` ;
- Dans certains cas, des sous-répertoires dédiés à des services ou applications (`/var/log/cups`, `/var/log/mail`, `/var/log/httpd`, ...);
- Localisation paramétrable dans le fichier de configuration du service ;
- Sur un serveur, intérêt à ce que `/var` soit dans une partition distincte de `/` ;
- Intérêt de la rotation des fichiers de log pour limiter l'espace utilisé ;
- `boot.log` : messages relatifs à la séquence de `boot` ;
- `cron` : messages relatifs à l'utilitaire `cron` ;
- `maillog` : messages relatifs au système de courrier ;
- `messages` : la presque totalité des messages qui ne sont pas dirigés dans un fichier spécifique ;
- `auth` ou `secure` : messages relatifs à l'authentification des utilisateurs sur le système (locale ou distante avec `telnet` ou `ssh`) ;
- `dmesg` : messages envoyés par le noyau (affiche tous les messages du noyau au cours de la vie du système) ;
- `/var/log/dmesg` : uniquement les messages générés par le noyau durant la séquence de `boot` ;

## Commandes utiles pour examiner les journaux

- `less` ;
- `head` ;
- `more` ;
- `tail -f` : affichage dynamique des nouvelles lignes qui apparaissent dans le journal. Pratique pour suivre l'évolution « en ligne » d'un journal **Crtl + C** pour quitter ;
- `grep` ;
- Plus de détail voir les pages `man` pour chaque commande.



**QUESTIONS ?**