

# Systeme de fichiers – Première partie

---

# Systeme de fichiers

---

## *Plan de la phase*

*Introduction*

*Les disques*

*Les systèmes de fichiers*

# Système de fichiers

---

## *Introduction*

- L'action de formater un disque, une clé ou tout support de données consiste uniquement à créer sur un support de mémoire secondaire l'organisation logique permettant d'y placer des données.
- Le mot formatage n'est quasiment jamais utilisé sous Linux, sauf pour expliquer le principe aux personnes provenant d'autres horizons. On parle de système de fichier qui est à la fois l'organisation logique des supports au niveau le plus bas comme au niveau de l'utilisateur.
- Outre l'organisation et le stockage des informations et des données sur les fichiers, le système de fichiers doit fournir à l'utilisateur une vision structurée de ses données, permettant de les distinguer, de les retrouver, de les traiter et de les manipuler.

# Système de fichier

## *Les disques*

### Vision physique d'un disque

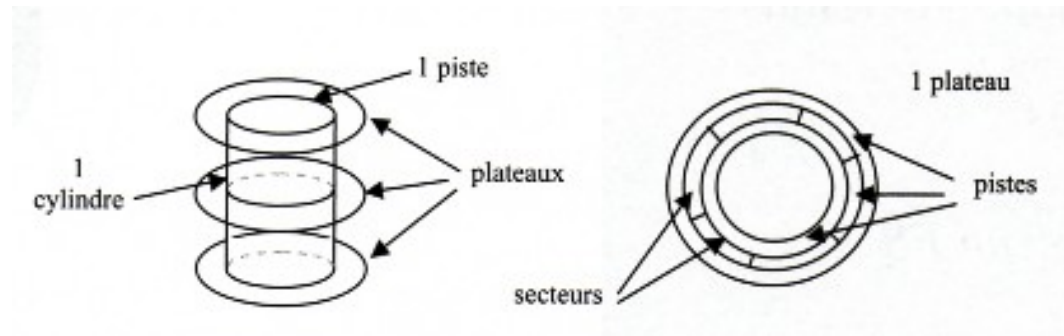


Figure 1

- Un disque est physiquement composé de plateaux ayant chacun deux faces.
- Sur chaque face, il y a un nombre de pistes concentriques.
- Chaque piste est composée d'un certain nombre de secteurs.
- Un cylindre correspond à l'ensemble des pistes (une par face) que le bras de lecture/écriture peut lire sans se déplacer. Un secteur a habituellement une taille de 512 octets.
- En résumé l'espace disque est fait d'un certain nombre de secteurs.

# Système de fichier

## *Les disques*

### Partitionnement, table des partitions, le MBR

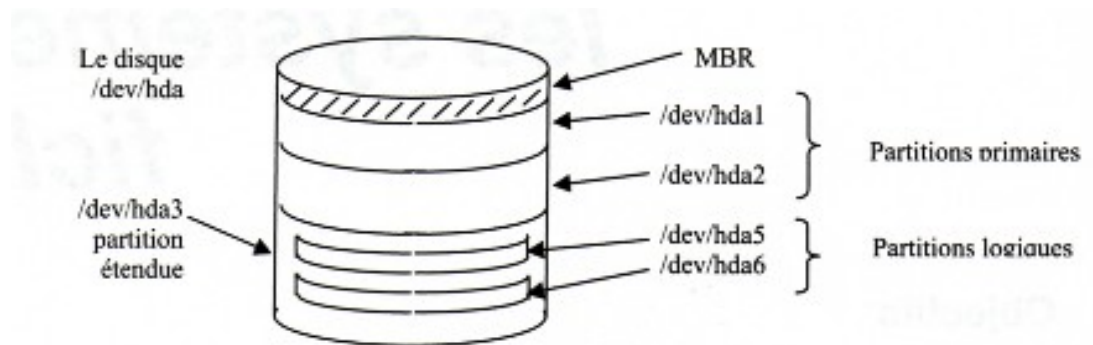


Figure 2

- Un disque est composé de partition, chacune peut être considérée comme un petit disque.
- A l'origine le nombre des partitions était limité à quatre. La table des partitions correspondante était stockée complètement dans le MBR (Master Boot Record), c'est à le premier secteur qui comprend également le code du chargeur primaire (primary loader).
- Maintenant, une des quatre partitions primaires peut avoir le type « étendue ». Dans ce cas, elle peut contenir une suite de partitions, appelées logiques.

# Système de fichier

## Les disques

### Caractéristiques d'une partition

En plus d'être primaire, étendue ou logique, une partition possède les caractéristiques suivantes :

- La plage de cylindre utilisée, et donc sa taille en secteurs.
- Un drapeau qui indique si c'est la partition active.
- Le type (*tag*) qui spécifie l'utilisation de la partition. Le type 83 indique un système de fichier Linux, 82 un espace de swap Linux et le type 5 indique une partition étendue, les partitions Windows NTFS ont le type 7, les partitions LVM ont le type 8e.

```
Disque /dev/sda: 8589 Mo, 8589934592 octets
255 têtes, 63 secteurs/piste, 1044 cylindres
Unités = cylindres de 16065 * 512 = 8225280 octets
Identifiant de disque : 0x3b393b38

Périphérique Amorce Début      Fin      Blocs    Id  Système
/dev/sda1    *           1        392     3148708+ 7  HPFS/NTFS
/dev/sda2           393       523     1052257+ b  W95 FAT32
/dev/sda3           524       588       522112+ 82 Linux swap / Solaris
/dev/sda4           589       980       3148740  5  Etendue
/dev/sda5           589       980     3148708+ 83  Linux
```

Figure 3

### Utilisation d'une partition

Une partition peut abriter :

- Un système de fichier, i.e une arborescence de fichiers.
- Un espace de swap, i.e une extension de la mémoire.
- Un espace disque dédié à une application.

# Système de fichier

## Les disques

### Commentaire de la figure 3

- Taille du disque (en octets) = (taille d'un secteur en octets) \* (nbr de têtes) \* (nbr de secteurs par piste) \* (nb cylindres)
- Unité (en octets) = (cylindre = (nbr de têtes) \* (nbr de secteurs par piste)) \* (taille d'un secteur en octets)
- Taille d'une partition (en octets) = Unité \* (Cylindre de Fin – Cylindre de Début + 1)

### Exemple de la figure :

- Taille du disque =  $512 * 255 * 63 * 1044 = 8587192320$  octets (pourquoi cette différence ??)
- Unité =  $255 * 63 * 512 = 8225280$  octets
- Taille de la partition `/dev/sda1` =  $8225280 * (392 - 1 + 1) = 3224309760$  (pourquoi cette différence ??)

### **Que signifie 3148708** dans :

Périphérique	Amorce	Début	Fin	Blocs	Id	Système
<code>/dev/sda1</code>	*	1	392	<b>3148708</b>	7	HPFS/NTFS

On a avec `sudo sfdisk -l` :

Disque `/dev/sda` : 1044 cylindres, 255 têtes, 63 secteurs/piste

Unités= cylindres de 8225280 octets, blocs de **1024 octets**, décompte à partir de 0

Donc :

Taille de la partition `/dev/sda1` =  $3148708 * 1024 = 3224276992$  o =  $3224276992 / 1024^3 = 3\text{Go}$ .

# Systeme de fichier

---

## *Les disques*

### **Nom des disques**

Sous Linux, les disques IDE (ou ATA ou PATA) sont nommés comme suit :

- /dev/hda : le disque maître géré par le premier contrôleur.
- /dev/hdb : le disque esclave géré par le premier contrôleur.
- /dev/hdc : le disque maître géré par le second contrôleur.
- /dev/hdd : le disque esclave géré par le second contrôleur.
- Etc ...

Les disque SCSI, SATA, SAS et USB sont nommés comme suit :

- /dev/sda : le premier disque.
- /dev/sdb : le second disque.
- Etc ...

### **Nom des partitions**

Pour un disque, les partitions sont nommées comme suit (par exemple pour le disque sda) :

- /dev/sda1 : la première partition primaire.
- /dev/sda2 : la seconde partition primaire.
- /dev/sda3 : la troisième partition primaire.
- /dev/sda4 : la quatrième partition primaire (la partition étendue).
- /dev/sda5 : la première partition logique (à l'intérieur de la partition étendue)

Toutes les partitions primaires ne sont pas forcément présentes. Une (et une seule) partition primaire peut jouer le rôle de partition étendue.



# Système de fichier

---

## *Les disques*

### Les commandes

- fdisk : l'outil de partitionnement des disques
- **sfdisk : commande de partitionnement scriptable.**
- parted : commande de partitionnement, permet de conserver les données
- cfdisk : commande de partitionnement en mode texte plein écran.
- partprobe : informe le système que les tables de partition ont été modifiée.

La commande *fdisk* permet de :

- Afficher la table de partitionnement (commande p).
  - Ajouter une nouvelle partition (commande n).
  - Détruire une partition (commande d).
  - Créer une table de partition vide, pour un nouveau disque (commande o).
  - Changer le type d'une partition d'une partition (commande t).
  - Mettre/enlever le drapeau partition active (commande a).
- 
- Pour sortir de l'utilitaire, on a le choix entre la commande w (write) qui valide les modifications et la commande q (quit) annule les changements.

# Système de fichier

## *Les systèmes de fichiers*

### Notion de FS, de montage

- L'arborescence globale, celle vue par les applications est composée d'un ou plusieurs système de fichiers.
- Chaque FS correspond à une arborescence de fichiers gérée comme un tout.
- Typiquement un FS est stockée dans une partition.
- Mais un FS peut être stocké dans un volume logique, un cdrom, une clé USB, en mémoire (live-cd), ou même dans un fichier ordinaire.
- Les fichiers d'un FS ne sont accessibles que si le FS est activé, on dit « monté ».
- Le montage d'un FS implique d'associer la racine du FS à un répertoire, dit répertoire de montage.
- Le démontage d'un FS rompt cette association.
- Le FS *root* est monté automatiquement par le noyau lors des première phase du démarrage.
- Les autre FS peuvent être montés ultérieurement, de manière automatique ou manuelle.

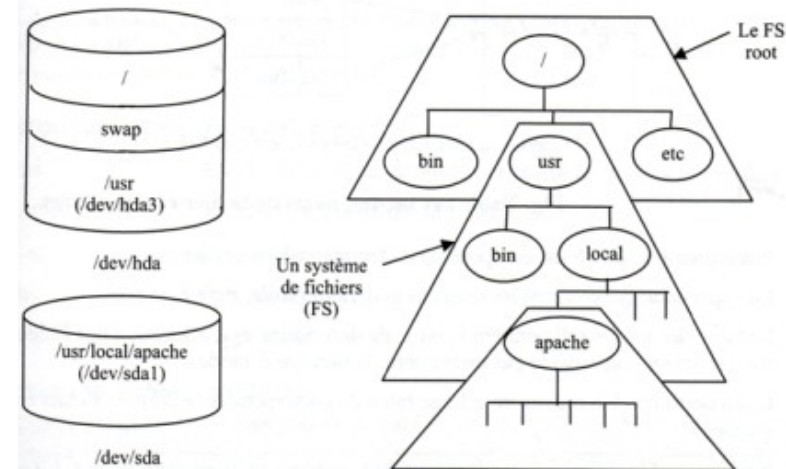


Figure 4

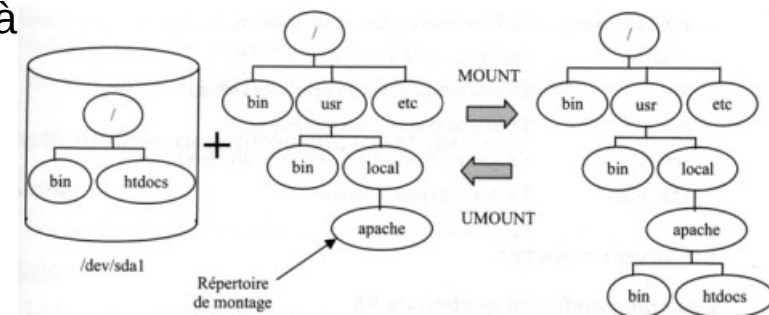


Figure 5

# Système de fichier

## Les systèmes de fichiers

### Tables systèmes , inodes

Physiquement, un FS est composé de différentes tables systèmes :

- Le super bloc qui contient les données générales (taille, monté ou non...).
- La table des inodes qui contient la table de description et d'allocation des fichiers. Chaque inode (fichier) est repérée par un numéro, le numéro d'inode.
- Les répertoires. Un répertoire est une table de correspondance nom de fichier numéro d'inode.

**Remarque** : Un FS est non seulement limité en blocs mais aussi en inodes. Le nombre d'inodes indique le nombre de fichiers que l'on peut créer dans un FS.

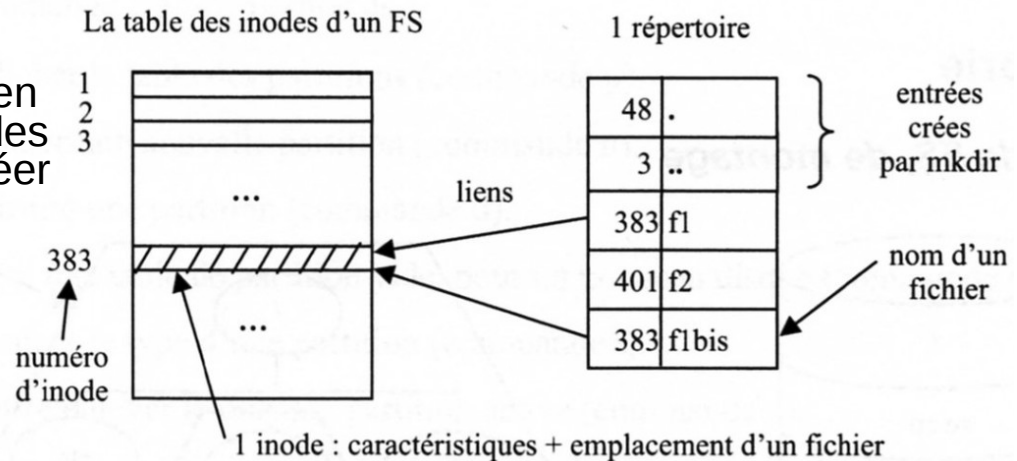


Figure 6

# Système de fichier

---

## *Les systèmes de fichiers*

### Les différents types de FS

Un système Linux peut gérer différents type de FS, chacun ayant des fonctionnalités différentes.

**ext2** : le « second extended filesystem » est considéré comme le système de fichier historique de Linux. Il est toujours utilisé voire conseillé dans certains cas. Il est rapide et nécessite moins d'écritures que les autres FS, donc il occasionne moins d'usure des supports de stockage, notamment les disque SSD (*pour Solid State Drive*), les clés USB ou les carte mémoires. Ces supports peuvent parfois ne supporter qu'un nombre restreint de cycles de lecture/écriture...

Les **fichiers** peuvent avoir jusqu'à une taille de **2 To** (2048 Go). Tandis qu'une partition peut atteindre **32 To**, voire 128 To, selon la taille des blocs. ext2 n'est **pas journalisé**.

**ext3** : le « third extended filesystem » est le successeur de ext2 depuis 1999. il est **journalisé**. Surtout, il est entièrement compatible avec ext2. Le journal est une extension de ext2. Il est possible d'utiliser un système de fichiers ext3 comme étant ext2, avec les mêmes commandes, les mêmes manipulation. Il est possible de transformer en quelques secondes un système ext2 en ext3, et vice versa. C'est le système de fichiers de choix pour Linux, et le plus utilisé pour sa souplesse.

- Taille des **fichiers** : **2 To**.
- Taille d'une **partition** : **32 To**.

**ext4** (2008) : est le successeur du système de fichiers ext3 et garde une compatibilité avec celui-ci. La fonctionnalité majeure de ext4 est l'allocation par *extent* qui permettent la pré-allocation d'une zone contiguë pour un fichier, pour minimiser la fragmentation. Une partition ext3 peut toujours être montée comme ext4 (compatibilité ascendante).

Taille **fichier** = **16 To** et Taille d'une **partition** = **1 Eo**

# Système de fichier

---

## *Les systèmes de fichiers*

### Les différents types de FS

#### reiserfs

reiserfs a été intégré à linux avant ext3. Sa force réside, outre dans son journal, dans l'organisation indexée de ses entrées des répertoires (les tables catalogues contenant les associations inodes/fichiers). Et la manipulation des fichiers de petits tailles. Ses performances sont exceptionnelles en présence de milliers de fichiers, de faible à moyen volume. Il est redimensionnable à chaud, il devient plus long sur des gros fichiers.

Les **fichiers** peuvent atteindre **8 To**, et les **partition 16 To**. Les nom de fichiers peuvent avoir 4032 caractères mais sont limités par Linux à 255 caractères.

reiserfs est moins utilisé malgré ses fortes qualités pour diverses raisons dont la principale est l'impossibilité de convertir un système de fichiers ext2/ext3 en reiserfs et vice versa et ce à cause de la forte base de machines installées en ext2/ext3.

#### xfs

Xfs est le plus ancien des systèmes de fichier journalisés sous linux, datant de 1993. possède un système de journalisation très performant et des mécanisme avancés comme la défragmentation en ligne (à chaud et au fur et à mesure des écriture), la capacité d'effectuer des snapshots (figer l'état d'un filesystem à un instant donnée pour le restaurer plus tard), le dimensionnement à chaud...

- Taille **fichier** théorique : **8 Eo** (Exaoctet), 1 Eo = 1024 Po (Petaoctet) donc 1048576 To, ou environ 1000 milliards de DVD !
- La **partition** : pour un contrôleur 64 bits = **16 Eo** et pour un contrôleur 32 bits = **16 Go**.

# Systeme de fichier

---

## *Les systemes de fichiers*

### Les différents types de FS

#### BTRFS

Le système de fichiers btrfs a gagné en fonctionnalité et en stabilité au fil des dernières versions du noyau Linux. **Appelé à remplacer ext3 et ext4** dans un avenir proche, cumulant les fonctionnalités d'un système de **fichier moderne et la flexibilité d'un gestionnaire de volumes logique** comme LVM, btrfs suscite un intérêt croissant pour ceux qui souffraient du manque de flexibilité des solutions actuelles. Sachant que le stockage a été amené à subir de très importantes évolutions ces dernières années, les nouvelles fonctionnalités de btrfs sont donc très attendues. (*Gnu/Linux Magazine N° 149, page 52*)

- Taille maximale de fichier : 16 Eio
- Taille maximale de volume : 16 Eio

#### Fonctionnalités :

- Migration de Extx vers Btrfs
  - Dimensionnement à chaud
  - Gestion de volume
  - Compression à la volée
  - Snapshots
- 
- TP possible : *Gnu/Linux Magazine N° 149, page 52*

# Systeme de fichier

---

## *Les systemes de fichiers*

### Les différents types de FS

#### ZFS

#### Déscription à faire

- Taille maximale de fichier : à faire
- Taille maximale de volume : à faire

#### Fonctionnalités :

- A faire

TP : <http://www.unixgarden.com/index.php/gnu-linux-magazine/zfs-sous-gnulinux>

# Système de fichier

---

## *Les systèmes de fichiers*

### Les différents types de FS

#### Système de fichiers distribué

Les gestionnaires de fichiers « orientés Cluster » permettent la gestion d'un système de fichiers par différents nœuds d'une grappe de serveurs. Quatre stratégies permettent le partage de fichiers dans un cluster :

1. La première consiste à s'appuyer sur le volume physique pour gérer les verrous. Chaque serveur a un accès direct au disque physique via un GNBD (*Global Network Block Device, cf tp sauvegarde*),
2. La deuxième approche consiste à proposer des serveurs logiques en écoute du réseau pour synchroniser les accès aux disques. Chaque serveur possède un disque avec une partie des répliquas de l'ensemble.
3. La troisième approche est un mélange des deux premières. Des serveurs partagent des verrous et tous partagent les mêmes volumes physiques.
4. Enfin, la dernière est un serveur de fichiers type NFS, permettant d'exposer un disque distant.

Notez que l'utilisation de ces architectures peut avoir des influences sur les bonnes pratiques de développement. Les programmes qui fonctionnent convenablement sur un disque local peuvent présenter des erreurs lors de l'utilisation d'un disque partagé.



# Système de fichier

---

## *Les systèmes de fichiers*

### Les différents types de FS

#### Système de fichiers distribué

- MooseFS :
  - <http://www.moosefs.org/reference-guide.html>
  - LM N°137
- GlusterFS :
  - LM N°133
- Hadoop FS :
  - <http://hadoop.apache.org/>
- OCFS et OCFS2 :
  - <https://oss.oracle.com/projects/ocfs2/>

# TP

# Système de fichier

---

## *Les systèmes de fichiers*

### Les différents types de FS

#### Vfat

Vfat (*Virtual Allocation Table*) est un terme générique regroupant les diverses versions de FAT supportant les noms longs sous Windows. Ces systèmes de fichiers sont conservés et continuent d'être utilisés pour des raisons à la fois historiques et pratiques. La plupart des supports amovibles, disques externes, clés USB et lecteurs MP3 utilisent un système de fichier de ce type.

Les raisons sont :

- Un système de fichiers adapté aux petits volumes.
- Un système de fichier simple à implémenter, idéal pour lecteurs multimédias.
- Une compatibilité entre diverses plateformes (Windows, Linux, BSD, MacOS, etc.).

VFAT souffre cependant de défauts inhérents à sa conception :

- Ensembles d'informations stockées au sein d'une table unique, y compris le nom du fichier et chaque adresse et longueur des blocs (appelés clusters) composant les données du fichier.
- De ce fait, FAT tente de regrouper les données d'un fichier sur le plus de clusters contigus du support. En cas de nombreuses écritures (ajout, suppression, etc.), le système se retrouve fortement fragmenté.
- Toujours de ce fait, plus le support a une taille importante, plus FAT est lent, car il doit vérifier toute la table FAT pour trouver des clusters disponibles.
- La gestion des noms longs est considérée comme une bidouille par de nombreuses personnes, car FAT doit continuer à assurer une compatibilité (encore aujourd'hui) avec les noms courts en 8.3.
- Contrairement aux systèmes de fichiers Unix, Linux ou Windows récents, FAT ne gère aucun attribut étendu, notamment aucune notion des droits et des propriétaires.
- Taille **fichier 4 Go**

Linux gère parfaitement VFAT. Mais son utilisation sur des supports partagés entre Windows et Linux a de moins en moins de raison d'être, car ntfs3g permet d'utiliser NTFS de manière native.

# Systeme de fichier

---

## *Les systemes de fichiers*

### Les differents types de FS

#### NTFS

NTFS a été lancé en 1993 avec le système multi-utilisateur Windows NT.

La technologie FAT, ne permettait que difficilement d'offrir un système de fichiers multi-utilisateurs :

- pas de zone indiquant le propriétaire du fichier ;
- pas de date du dernier accès en lecture ;
- pas de droits d'accès de groupe.

Le pilote libre NTFS-3G, en version stable 1.0 depuis le 21 février 2007, permet une écriture fiable et la création de fichiers sur les partitions NTFS.

- Taille des **fichiers** : **en pratique 16 To, en théorie 16 Eo.**
- Taille d'une **partition** : **en pratique 256 To, en théorie 16 Eo.**

Pour plus d'information, voir cours Windows.

# Systeme de fichier

---

## *Les systemes de fichiers*

### **Les commandes de gestion de FS**

- `mkfs` : créer un FS (concrètement formate une partition).
- `fck` : Vérifie/répare un FS.
- `mount` : Monte un FS. L'option `-a` (all) monte l'ensemble des FS décrits dans `/etc/fstab`. Cette commande est activée par les scripts lancés automatiquement au démarrage.
- `umount` : Démonte un FS.
- `df` : Liste les FS montés, la place libre par FS.
- `du` : La taille occupée par une arborescence.
- `ls` : Liste des fichiers ouverts, permet de connaître les application accédant à un FS
- `e2label` : Met une étiquette (label) à un FS. Cette étiquette permet d'identifier le FS.

### **Les commandes de gestion de fichiers**

- `ls` : Affiche les attributs d'un fichier (nombre de liens, n° d'inode...).
- `stat` : Affiche l'ensemble des attributs d'un fichiers.
- `ln` : Crée un lien.
- `mv` : Déplace un lien.
- `rm` : Supprime un lien

# Système de fichier

## *Les systèmes de fichiers*

### Le fichier /etc/fstab

```

# /etc/fstab: static file system information.
#
# Use 'vol_id --uuid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point>   <type>  <options>          <dump>  <pass>
proc                /proc                proc     defaults           0        0
# / was on /dev/sda5 during installation
UUID=4e648317-e76c-447a-9344-b27bf5c1cf5b /                    ext3      relatime,errors=remount-ro 0
1
# swap was on /dev/sda3 during installation
UUID=261c16f3-1297-4fc5-9b75-bcfc0a748d40 none                swap      sw                0        0
/dev/scd0           /media/cdrom0        udf,iso9660 user,noauto,exec,utf8 0
/dev/fd0            /media/floppy0       auto      rw,user,noauto,exec,utf8 0
~

```

Figure 7

Chaque ligne du fichier fstab est composée de différents champs :

- La partition qui abrite le FS (son nom ou son UUID ou son étiquette).
- Le répertoire de montage.
- Son type.
- Les options de montage, defaults indiquant les options par défaut. i.e montage automatique au démarrage en lecture/écriture Inversement noauto, ro indiquent un montage en lecture seule.
- Les deux dernier champs n'ayant pas de rapport avec le montage.