

Module : Administration Système Linux

Cycle Ingénieur : Génie Informatique
Niveau : GI2

Pr. Ahmad EL ALLAOUI

cyclegi2@gmail.com

2022/2023

Administration Systèmes Linux

Gestion des disques

Gestion des disques

- *partitions* -

- Le disque est “découpé” en partitions
 - commandes & applications
 - comptes utilisateurs
 - swap
 - fichiers temp
 - périphériques (disques, ...)
 - ...
- accès transparent

Le système de fichiers

- *partitions* -

- Tout appartient à /
- Chaque périphérique est monté sur un répertoire appartenant à /
- Comment faire : avec la commande mount
- mount /dev/periph /mnt/repertoire
- Puis on peut naviguer dans /mnt/repertoire

Partitions

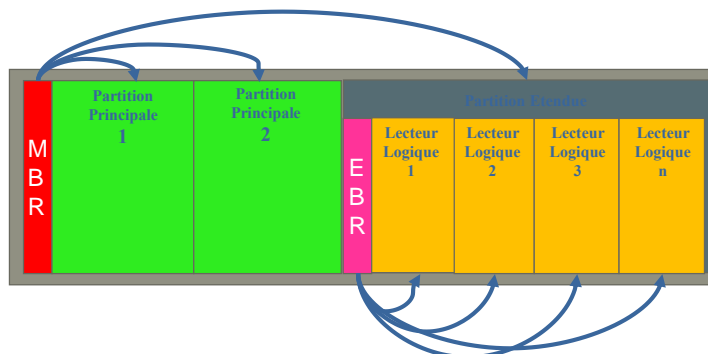
- Il consiste à créer des zones sur le disque dont les données ne seront pas mélangées.
- Permet d'installer des **systèmes d'exploitation** différents n'utilisant pas le même **système de fichiers**.
- Il y aura donc au minimum autant de partitions que de systèmes d'exploitation utilisant des systèmes de fichiers différents. Dans le cas d'un utilisateur d'un système d'exploitation unique, une seule partition de la taille du disque peut suffire, sauf si l'utilisateur désire en créer plusieurs pour faire par exemple plusieurs lecteurs dont les données sont séparées.
- Il y a trois sortes de partitions: la **partition principale**, la **partition étendue** et les **lecteurs logiques**.
- Un disque peut contenir jusqu'à **quatre partitions** principales (dont une seule peut être active), ou trois partitions principales et une partition étendue.
- Dans la partition étendue l'utilisateur peut créer des lecteurs logiques (c'est-à-dire "simuler" plusieurs disques durs de taille moindre).

Partitions

Pour les systèmes **DOS** (DOS, Windows 9x), seulement la partition principale est bootable, c'est donc la seule sur laquelle on peut démarrer le système d'exploitation.

On appelle **partitionnement** le processus qui consiste à écrire les **secteurs** qui constitueront la table de partition (qui contient les informations sur la partition: taille de celle-ci en terme de nombre de secteurs, position par rapport à la partition principale, types de partitions présentes, systèmes d'exploitation installés,...).

Lorsque la partition est créée, on lui donne un *nom de volume* qui va permettre de l'identifier facilement.

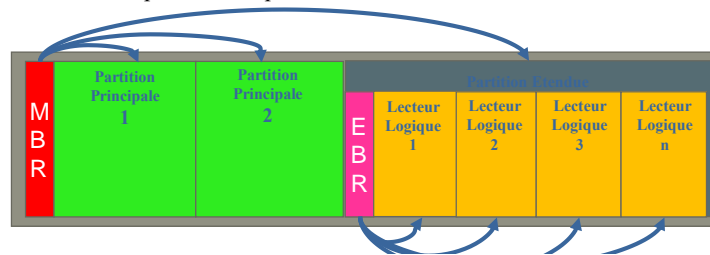


Partitions

Le **secteur de démarrage** (appelé **Master Boot Record** ou **MBR** en anglais) est le premier secteur d'un disque dur (cylindre 0, tête 0 et secteur 1), il contient la **table de partition principale** (en anglais *partition table*) et le code, appelé **boot loader**, qui, une fois chargé en mémoire, va permettre d'amorcer (*booter*) le système.

Ce programme, une fois en mémoire, va déterminer sur quelle partition le système va s'amorcer, et il va démarrer le programme (appelé *bootstrap*) qui va amorcer le système d'exploitation présent sur cette partition.

En plus, ce secteur du disque qui contient toutes les informations relatives au disque dur (fabricant, numéro de série, nombre d'octets par secteur, nombre de secteurs par cluster, nombre de secteurs,...). Ce secteur est donc le secteur le plus important du disque dur, il sert au setup du [BIOS](#) à reconnaître le disque dur. Ainsi, sans celui-ci votre disque dur est inutilisable, c'est donc une cible de prédilection pour les [virus](#).



Nommage des partitions

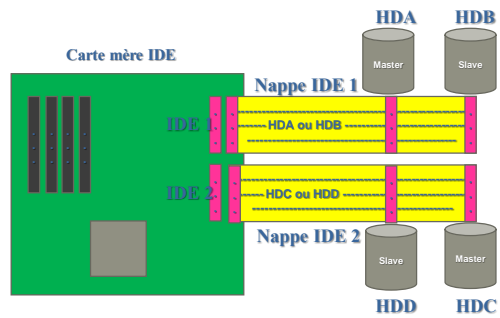
Windows

- Pas de différenciation entre PP/PE
- Pas de différenciation entre SATA/IDE
- Nombre de disque limité

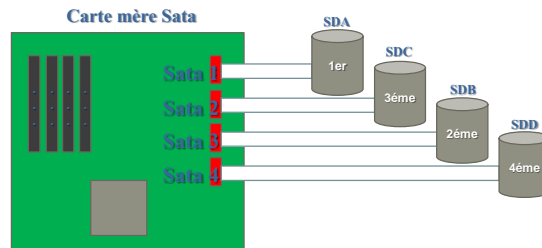
Linux

- Infinité de nom de disque/partition
- On peut reconnaître les différents types de disques et de partition

Nommage IDE



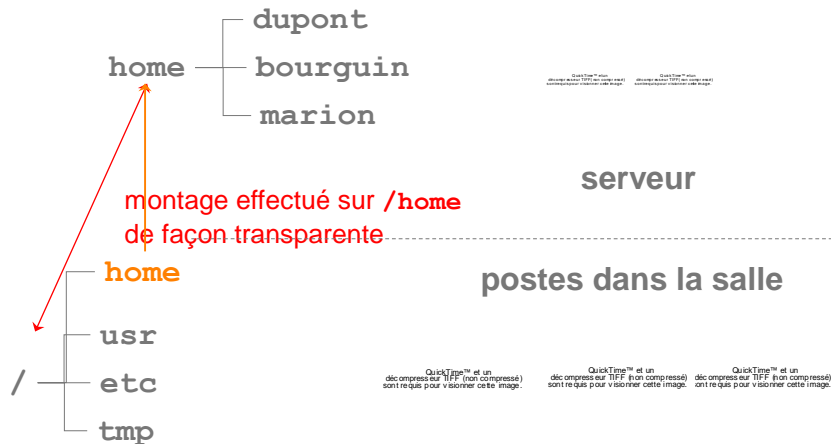
Nommage SATA



Le système de fichiers

- partitions -

- exemple: les comptes utilisateurs



Le système de fichiers

- partitions -

- tous les disques amovibles (cdrom, clé usb) dans : **/mnt**
- ex pour utiliser un CDROM:
 - Montage:
mount /mnt/cdrom
 - lire/écrire dans **/mnt/cdrom**
 - Démontage:
umount /mnt/cdrom
- idem pour clés usb

Plan de partitionnement

Objectifs :

- Concevoir le plan de partitionnement des disques d'un système Linux.
- L'allocation de systèmes de fichiers ou d'espaces de swap.
- L'adaptation de ce plan aux besoins auxquels le système devra répondre.

Plan de partitionnement

Partitions / et swap :

- Lorsque l'on fait une installation, il est nécessaire de créer au minimum deux partitions :
 - / (système de fichiers racine): partition contenant la distribution Linux.
 - Espace de swap : partition permettant au noyau de faire tourner plus de processus que la RAM seule ne pourrait en supporter.

Plan de partitionnement

Partitions Swap :

- La partition de swap n'a pas besoin d'un système de fichiers.
- Le noyau pourra y accéder en mode brut (raw mode).
- Cela permet d'éviter les surcoûts dus aux appels systèmes nécessaires à l'utilisation d'un système de fichiers.

Problèmes de vitesse des disques.

Usage :

Avant de décider votre plan de partitionnement, vous devez connaître les types d'applications qui tourneront sur votre système.

Serveur de mails

Serveur Web

Applications graphiques s'appuyant sur X-Window
et plus encore

Problèmes de vitesse des disques.

Usage :

Si votre système possède plusieurs disques, utilisez le plus rapide pour conserver la majorité de vos données.

/ Contient la plupart des utilitaires systèmes qui ne sont pas souvent utilisés. → **le disque le plus lent.**

/var/log contient beaucoup d'informations de log. → **Le disque le plus rapide.**

/usr : La prévision d'utiliser de nombreux clients démarrant de nombreuses applications graphiques pousse souvent à mettre cette partition sur un disque rapide.

Problèmes de vitesse des disques.

Exemple d'applications systèmes:

Pour distribuer les e-mails, **Sendmail** écrits dans deux queues de mail.

Généralement **/var/spool/mqueue** et **/var/spool/mail** ainsi qu'à d'autres endroits.

Apache utilise différents fichiers, deux fichiers de logs pour enregistrer et accéder aux pages actuelles.

Apache passe un temps non négligeable à écrire dans ces fichiers de logs.

Administration Systèmes Linux

FHS : Filesystem Hierarchy Standard

19

Introduction

Objectifs :

La disposition des fichiers dans l'arborescence des systèmes Unix n'est pas d'une compréhension **évidente** de prime abord.

Il est cependant indispensable de **connaître les règles** qui président à la **distribution des fichiers et répertoires** dans le système des fichiers afin de :

- Déterminer un plan de sauvegarde
- Gérer la sécurité
- Agir efficacement lors de la résolution des problèmes
- Installer des logiciels non disponibles sous formes de paquetage

Introduction

Objectifs :

Un document, le FHS (Filesystem Hierarchy Standard) disponible sur cette page : <http://www.pathname.com/fhs/>

A pour ambition de proposer la normalisation de l'organisation de système de fichiers pour les systèmes Unix.

La plupart des distributions Linux s'y conforme même s'il reste beaucoup de différences dans les détails d'implémentation.

Introduction

Sous Unix un fichier est :

- toujours désigné par un nom.
- possède un unique inode (certaines informations concernant le fichier).
- possède les fonctionnalités suivantes :
 - ouverture
 - fermeture.
 - lecture (consultation).
 - écriture (modification)

Un fichier peut être :

- ordinaire (on utilise parfois le terme "normal") (-)
- un répertoire (d)
- un lien symbolique (l)

Filesystem Hierarchy Standard

FHS:

La racine de l'arborescence Linux contient ces répertoires :

<u>bin</u>	programmes utilisateur essentiels (nécessaires au démarrage du système)
<u>boot</u>	fichiers nécessaires au chargement de Linux (<i>bootloader, initrd, noyau</i>)
<u>dev</u>	fichiers spéciaux offrant l'accès aux périphériques
<u>etc</u>	configuration du système et des services
<u>home</u>	répertoires principaux des utilisateurs
<u>lib</u>	librairies partagées essentielles (nécessaires au démarrage du système)
<u>mnt</u>	contient des points de montage temporaires (cdrom, floppy, etc.)
<u>proc et sys</u>	systèmes de fichiers virtuels permettant d'accéder aux structures internes du noyau (sys est nouveau depuis la version 2.6 de Linux)
<u>root</u>	répertoire principal de l'utilisateur <i>root</i>
<u>sbin</u>	exécutables système essentiels (nécessaires au démarrage du système)
<u>tmp</u>	répertoire pour le stockage de fichiers temporaires
<u>usr</u>	arborescence contenant la plupart des fichiers des applications
<u>var</u>	données vivantes du système et des applications

/etc /lib

/etc

- /etc/X11 : fichiers et répertoires de configuration du système et des applications
- /etc/rc* : configuration du système X11
- /etc/pam.d : configuration, par service, des *Pluggable Authentication Modules*

/etc

- /etc/httpd : fichiers de configuration du serveur Web *Apache*
- /etc/mail : fichiers de configuration du système de messagerie *Sendmail*
- /etc/ssh : fichiers de configuration et clés asymétriques d' *OpenSSH*

/lib

- /lib/modules : modules du noyau
- /lib/security : librairies PAM (*Pluggable Authentication Modules*)
- /lib/iptables : greffons *iptables*
- /lib/kbd : codages clavier et polices de la console

/usr

/usr

/usr/bin	la plupart des programmes utilisateur
/usr/games	jeux ...
/usr/include	en-têtes standards pour le développement
/usr/lib ressources	bibliothèques (.so pour <i>Shared Objects</i>) et autres partagées
/usr/libexec programmes	binaires exécutables appelés par d'autres
/usr/share forme	ressources partagées indépendantes de la plate- (pages de manuel, documentation, fichiers de données, etc.)
/usr/src	sources du système (noyau, paquetages, etc.)

/var

/var

/var/tmp	fichiers temporaires préservés entre deux démarrages
/var/www	racine du serveur Web
/var/log	journaux du système et des applications
/var/spool	données en attente d'un traitement
/var/spool/mail	boîtes aux lettres des utilisateurs
/var/spool/cron	tâches planifiées par les utilisateurs
/var/spool/lpd	files pour le système d'impression

Les catégories

La FHS spécifie deux catégories indépendante de fichiers

A données partagées /
non partagées

A données variables /
non variables (statique)

Les données
partagées
peuvent être
partagées
entre les hôtes

Les données
non partagées
sont
spécifique à
une hôte
donné (comme
les fichiers de
configuration).

Les données
variables
peuvent être
modifiés

Les données
statiques ne
sont pas
modifiable
(sauf via les
fichier de
configuration)

Les inodes

Les inodes (contraction de « index » et « node », en français : *nœud d'index*) sont des **structures de données** contenant des informations concernant les fichiers stockés dans certains systèmes de fichiers (notamment de type **Linux/Unix**).

À chaque fichier correspond un numéro d'inode (**i-number**) 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 son créateur (ou propriétaire), son type d'accès (par exemple sous Unix : **lecture, écriture et exécution**), etc.

Les inodes

Les inodes contiennent notamment les **métadonnées** des systèmes de fichiers, et en particulier celles concernant les droits d'accès.

Les inodes sont créés lors de la création du système de fichiers.

La quantité d'inodes (généralement déterminée lors du formatage et dépendant de la taille de la partition) indique le nombre maximum de fichiers que le système de fichiers peut contenir.

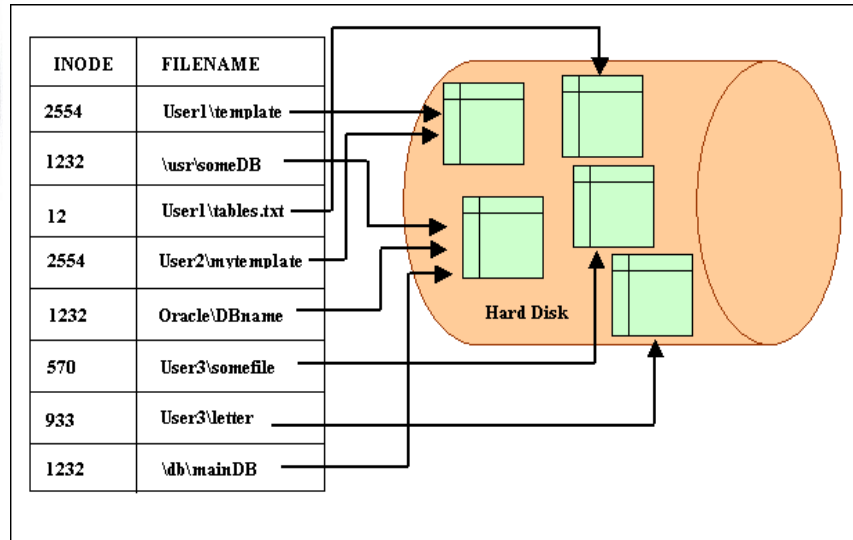
Les inodes

Le numéro d'inode est un entier unique pour le périphérique dans lequel il est stocké.

Le numéro d'inode d'un fichier **toto** peut être affiché avec la commande

ls -li toto

Les Inodes



Norme POSIX

La taille du fichier en octets

L'identifiant du groupe auquel appartient le fichier

Le standard POSIX s'est basé sur les systèmes de fichiers traditionnels d'Unix. Cette norme impose donc que les fichiers réguliers aient les attributs suivants :

Identifiant du périphérique contenant le fichier

Le mode du fichier qui détermine quel utilisateur peut lire, écrire et exécuter ce fichier

Un compteur indiquant le nombre de liens physiques sur cet inode.

L'identifiant du propriétaire du fichier

Le numéro d'inode qui identifie le fichier dans le système de fichier

Horodatage

Les Inodes

Les Commandes

ls -i

df -i

```
samir@localhost:/  
Fichier  Édition  Affichage  Terminal  Aide  
[root@localhost /]# ls -i fichier  
187 fichier  
[root@localhost /]# df -i /dev/sda1  
Sys. de fich.      Inodes    IUtil.  ILib. %IUtil. Monté sur  
/dev/sda1          512064   118828   393236   24% /  
[root@localhost /]# df -i /dev/sda1  
Sys. de fich.      1K-blocs   Occupé Disponible Capacité Monté sur  
/dev/sda1          8063408   4005864   3647944   53% /  
[root@localhost /]#
```

Les Inodes

La Commande stat

```
samir@localhost:/  
Fichier  Édition  Affichage  Terminal  Aide  
[root@localhost /]# stat fichier  
File: 'fichier'  
Size: 0          Blocks: 0          IO Block: 4096   fichier régulier vide  
Device: 801h/2049d Inode: 187         Links: 1  
Access: (0644/-rw-r--r--)  Uid: (   0/   root)   Gid: (   0/   root)  
Access: 2010-01-24 02:04:03.000000000 +0100  
Modify: 2010-01-24 02:04:03.000000000 +0100  
Change: 2010-01-24 02:04:03.000000000 +0100  
[root@localhost /]# stat --file-system fichier  
File: "fichier"  
ID: d6f613ca0ead93af Namelen: 255   Type: ext2/ext3  
Block size: 4096      Fundamental block size: 4096  
Blocks: Total: 2015852 Free: 1014386 Available: 911986  
Inodes: Total: 512064 Free: 393235  
[root@localhost /]#
```

Les Systèmes de fichier

Les systèmes de fichiers existent pour vous permettre de stocker, **lire et manipuler des données** sur un lecteur *block*.

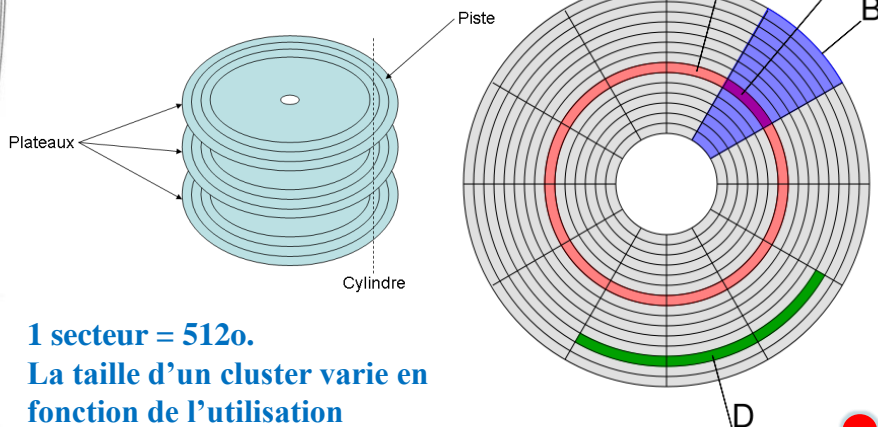
En fait, les systèmes de fichiers sont la seule manière efficace d'accéder au contenu des lecteurs *block*.

Un système de fichiers maintient la structure interne des données (*meta-data*) qui fait que vos données restent **organisées et accessibles**.

Exemple de lecteur *block* : *Disque dur, DVD, Disquette ...*

Structure d'un disque dur

- (A) Piste
- (B) Secteur géométrique
- (C) secteur d'une piste
- (D) cluster (la plus petite unité pour un SE)



La Journalisation

- Un journal est la partie d'un **système de fichiers journalisé** qui trace les opérations d'écriture tant qu'elles ne sont pas **terminées** et cela en vue de garantir l'intégrité des données en **cas d'arrêt brutal**.
- L'intérêt est de pouvoir plus facilement et plus rapidement récupérer les données en cas d'arrêt brutal du système d'exploitation (**coupure d'alimentation, plantage du système, etc.**), alors que les partitions n'ont pas été correctement synchronisées et démontées.

Les Types

ext2: Ancien et très stable.

Fonctionne pour des fichiers dont la taille est supérieurs à ~2-3K.

ext3: Extension pour la journalisation de ext2.

Journaliser un système de fichiers revient à ajouter une nouvelle structure de données appelée un journal.

reiserfs: Système de fichiers journalisé.

Plus performant pour les fichiers de petit taille (internet)

XFS: Système de fichiers journalisé; pour optimiser la manipulation des très gros fichiers, de plus de 9 ExaBytes (9'000'000'000 GigaBytes).

JFS: Un autre système de fichiers journalisé créé par IBM.

FAT32 et NTFS: Le système de fichiers Windows.

Les Types

Nom du système de fichiers	Taille maximale d'un fichier	Taille maximale d'une partition	Journalisée ou non ?	Gestion des droits d'accès?	Notes
ext2fs (Extended File System)	2 TiB	4 TiB	Non	Oui	Extended File System est le système de fichiers natif de Linux. En ses versions 1 et 2, on peut le considérer comme désuet, car il ne dispose pas de la journalisation. Ext2 peut tout de même s'avérer utile sur des disquettes 3½ et sur les autres périphériques dont l'espace de stockage est restreint, car aucun espace ne doit être réservé à un journal.
ext3fs	2 TiB	4 TiB	Oui	Oui	ext3 est essentiellement ext2 avec la gestion de la journalisation. Il est possible de passer une partition formatée en ext2 vers le système de fichiers ext3 (et vice versa) sans formatage.
ext4fs	16 TiB	1 EiB	Oui	Oui	ext4 est le successeur du système de fichiers ext3. Il est cependant considéré par ses propres concepteurs comme une solution intérimaire en attendant le vrai système de nouvelle génération que sera Btrfs
ReiserFS	8 TiB	16 TiB	Oui	Oui	Développé par Hans Reiser et la société Namesys, ReiserFS est reconnu particulièrement pour bien gérer les fichiers de moins de 4 ko. Un avantage du ReiserFS, par rapport à ext3, est qu'il ne nécessite pas une hiérarchisation aussi poussée: il s'avère intéressant pour le stockage de plusieurs fichiers temporaires provenant d'Internet. Par contre, ReiserFS n'est pas recommandé pour les ordinateurs portables, car le disque dur tourne en permanence, ce qui consomme beaucoup d'énergie.

Créer des partitions /systemes de fichiers

Quelques utilitaires:

fdisk

mkfs , mke2fs

dumpe2fs

tune2fs

fsck

mount , partprobe

/etc/fstab

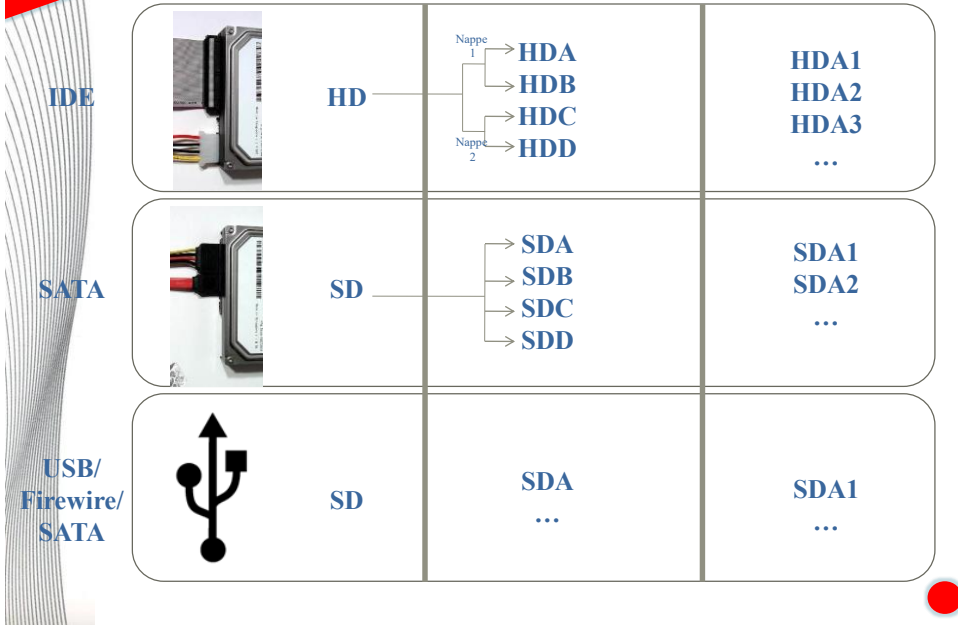
Nommage des partitions sous windows

- Nommage des partitions par des lettre de l'alphabet sous Windows
- Commence a C: → Z:
- Pas d'information sur le disque dur (disque dur 1, disque dur 2)
- Pas d'information sur le type de disque dur (IDE ou SATA)
- Pas s'information sur le type de partition (PP, LL)

Nommage des partitions sous Linux

- Si votre disque dur est **IDE**, son nom commencera par **hd**.
- Si votre disque est **SATA ou SCSI ou USB**, son nom commencera **sd**.
- Sur la **première nappe** :
 - Le disque maitre est **hda**
 - Le disque esclave est **hdb**
- Sur la **deuxième nappe** :
 - Le disque maitre est **hdc**
 - Le disque esclave est **hdd**
- Le **premier disque** sata est **sda**, le deuxième **sdb**, etc.
- Si votre disque est partitionné, on ajoute seulement **un numéro** : **sda1**, **hdb4**.
- La numérotation des partitions logiques **commence à 5**.

Nommage des partitions sous Linux



Exercice 1

```
arafet@arafet-server: ~  
Fichier Édition Affichage Terminal Onglets Aide  
arafet@arafet-server:~$ s  
[sudo] password for arafet:  
  
_____  
  
_____  
_____  
_____  
  
_____
```

Exercice 2

Je possède :

- Un Disque 1 de 250 Go IDE → Sur la nappe 1 (master)
- Un Disque 2 de 80 Go IDE → Sur la nappe 2 (master)
- Une Clé USB de 4 Go

Le Disque 1 contient : Deux P Principales de 100 Go chaque une
Deux L Logiques de 25 Go chaque un

Le Disque 2 contient : Une P Principal de 50 Go
Un L Logique de 30 Go

La Clé USB contient : Deux Partitions de 2 Go

Nommez les partitions

Créer des partitions : fdisk

Pourquoi créer différentes partitions ?
Comment créer une partition ?

Avec la commande **fdisk [disque]**

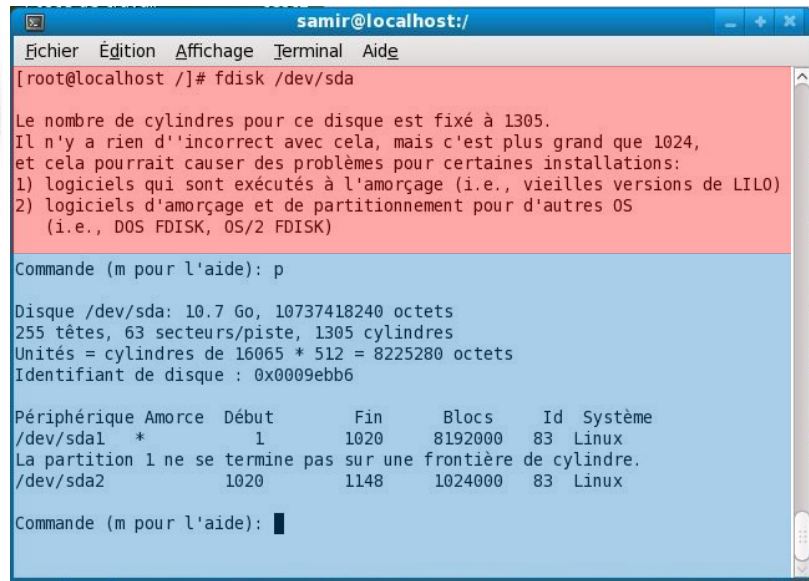
fdisk est un programme utilisé pour la création et la manipulation de tables de partitions.

fdisk -l → afficher la table de partitions.
Exemple : **fdisk /dev/sda**

Options:

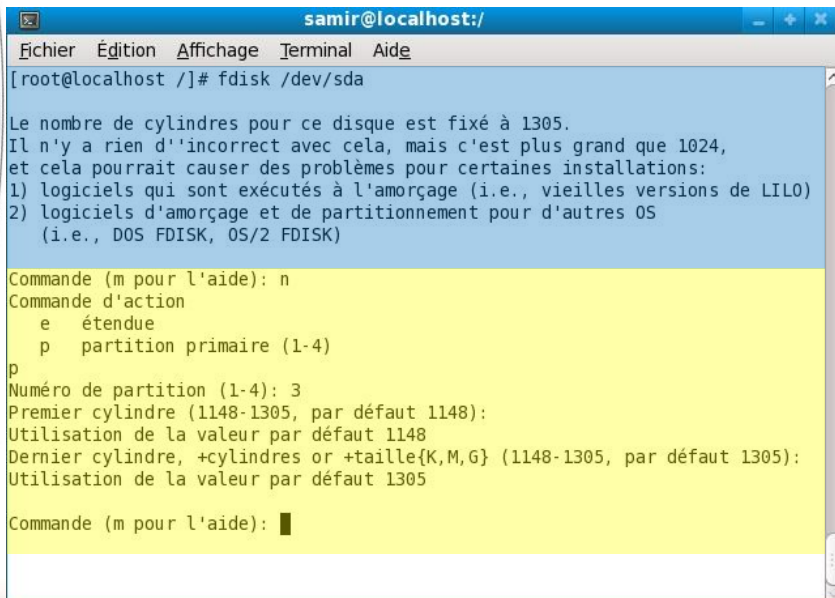
- n** : ----> pour ajouter une partition
- p** : ----> pour afficher la table de partitions
- m** : ----> pour afficher le help
- w** : ----> enregistrer les modifications

Créer des partitions : fdisk



```
samir@localhost:/  
Fichier Édition Affichage Terminal Aide  
[root@localhost ~]# fdisk /dev/sda  
  
Le nombre de cylindres pour ce disque est fixé à 1305.  
Il n'y a rien d'incorrect avec cela, mais c'est plus grand que 1024,  
et cela pourrait causer des problèmes pour certaines installations:  
1) logiciels qui sont exécutés à l'amorçage (i.e., vieilles versions de LILO)  
2) logiciels d'amorçage et de partitionnement pour d'autres OS  
   (i.e., DOS FDISK, OS/2 FDISK)  
  
Commande (m pour l'aide): p  
  
Disque /dev/sda: 10.7 Go, 10737418240 octets  
255 têtes, 63 secteurs/piste, 1305 cylindres  
Unités = cylindres de 16065 * 512 = 8225280 octets  
Identifiant de disque : 0x0009ebb6  
  
Périphérique Amorç Début      Fin      Blocs    Id Système  
/dev/sda1  *      1        1020    8192000  83  Linux  
La partition 1 ne se termine pas sur une frontière de cylindre.  
/dev/sda2      1020    1148    1024000  83  Linux  
  
Commande (m pour l'aide):
```

Créer des partitions: fdisk



```
samir@localhost:/  
Fichier Édition Affichage Terminal Aide  
[root@localhost ~]# fdisk /dev/sda  
  
Le nombre de cylindres pour ce disque est fixé à 1305.  
Il n'y a rien d'incorrect avec cela, mais c'est plus grand que 1024,  
et cela pourrait causer des problèmes pour certaines installations:  
1) logiciels qui sont exécutés à l'amorçage (i.e., vieilles versions de LILO)  
2) logiciels d'amorçage et de partitionnement pour d'autres OS  
   (i.e., DOS FDISK, OS/2 FDISK)  
  
Commande (m pour l'aide): n  
Commande d'action  
  e  étendue  
  p  partition primaire (1-4)  
p  
Numéro de partition (1-4): 3  
Premier cylindre (1148-1305, par défaut 1148):  
Utilisation de la valeur par défaut 1148  
Dernier cylindre, +cylindres or +taille{K,M,G} (1148-1305, par défaut 1305):  
Utilisation de la valeur par défaut 1305  
  
Commande (m pour l'aide):
```



```

Samir@localhost:/home/Samir
Fichier Édition Affichage Terminal Aide
[root@localhost Samir]# fdisk /dev/sda

Le nombre de cylindres pour ce disque est fixé à 1305.
Il n'y a rien d'incorrect avec cela, mais c'est plus grand que 1024,
et cela pourrait causer des problèmes pour certaines installations:
1) logiciels qui sont exécutés à l'amorçage (i.e., vieilles versions de LILO)
2) logiciels d'amorçage et de partitionnement pour d'autres OS
   (i.e., DOS FDISK, OS/2 FDISK)

Commande (m pour l'aide): p

Disque /dev/sda: 10.7 Go, 10737418240 octets
255 têtes, 63 secteurs/piste, 1305 cylindres
Unités = cylindres de 16065 * 512 = 8225280 octets
Identifiant de disque : 0x0001dbc8

Périphérique Amorç Début Fin Blocs Id Système
/dev/sda1 * 1 510 4096000 83 Linux
La partition 1 ne se termine pas sur une frontière de cylindre.
/dev/sda2 510 638 1024000 82 Linux swap / Solaris
La partition 2 ne se termine pas sur une frontière de cylindre.

Commande (m pour l'aide): n
Commande d'action
e étendue
p partition primaire (1-4)
p
Numéro de partition (1-4): 3
Premier cylindre (638-1305, par défaut 638):
Utilisation de la valeur par défaut 638
Dernier cylindre, +cylindres or +taille[K,M,G] (638-1305, par défaut 1305):
Utilisation de la valeur par défaut 1305

Commande (m pour l'aide): p

Disque /dev/sda: 10.7 Go, 10737418240 octets
255 têtes, 63 secteurs/piste, 1305 cylindres
Unités = cylindres de 16065 * 512 = 8225280 octets
Identifiant de disque : 0x0001dbc8

Périphérique Amorç Début Fin Blocs Id Système
/dev/sda1 * 1 510 4096000 83 Linux
La partition 1 ne se termine pas sur une frontière de cylindre.
/dev/sda2 510 638 1024000 82 Linux swap / Solaris
La partition 2 ne se termine pas sur une frontière de cylindre.
/dev/sda3 638 1305 5362381 83 Linux

```

Pour enregistrer les
modification il faut
appuyer sur « w »
avant de quitter fdisk

Créer des partitions : parted

partprobe:

- Informe le noyau des nouvelles modifications faites sur la table de partitions
- A invoqué après chaque modification faites par la commande **fdisk**

Création d'un Systèmes de fichiers

Pour créer un système de fichiers sur une partition, il faut utiliser **mkfs**.

mkfs est utilisé pour formater un système de fichiers Linux sur un périphérique, généralement une partition de disque dur.

mkfs [options] -t [fstype] lecteur [blocksize]

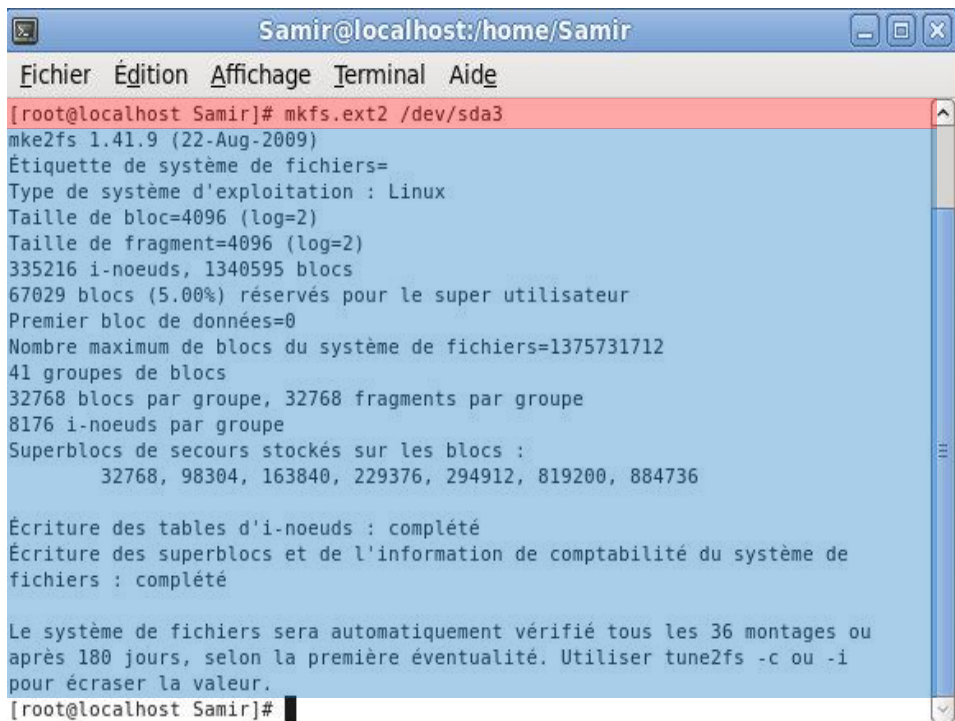
Principales options:

-t: fstype: Type de système de fichiers.

-c: Contrôle des *bad blocks* du lecteur avant la construction d'un système de fichiers.

mkfs.ext3 /dev/hda1

mkfs -t ext3 /dev/hda1 → les 2 commandes sont équivalentes



```
Samir@localhost:/home/Samir
Fichier Édition Affichage Terminal Aide
[root@localhost Samir]# mkfs.ext2 /dev/sda3
mke2fs 1.41.9 (22-Aug-2009)
Étiquette de système de fichiers=
Type de système d'exploitation : Linux
Taille de bloc=4096 (log=2)
Taille de fragment=4096 (log=2)
335216 i-noeuds, 1340595 blocs
67029 blocs (5.00%) réservés pour le super utilisateur
Premier bloc de données=0
Nombre maximum de blocs du système de fichiers=1375731712
41 groupes de blocs
32768 blocs par groupe, 32768 fragments par groupe
8176 i-noeuds par groupe
Superblocs de secours stockés sur les blocs :
    32768, 98304, 163840, 229376, 294912, 819200, 884736

Écriture des tables d'i-noeuds : complété
Écriture des superblocs et de l'information de comptabilité du système de
fichiers : complété

Le système de fichiers sera automatiquement vérifié tous les 36 montages ou
après 180 jours, selon la première éventualité. Utiliser tune2fs -c ou -i
pour écraser la valeur.
[root@localhost Samir]#
```

Creation d'un Systeme de fichiers étendu

Pour créer un système de fichiers étendu (ext2, ext3) sur une partition, il faut utiliser **mke2fs**.

mke2fs [options] device [blocksize]

Principales options:

- b: Specify the block size type: Type de système de fichiers.
- c: Contrôle si le lecteur contient des blocs défectueux avant de construire le système de fichiers.
- j: Crée le système de fichiers avec un journal ext3.
- L: Donne un label au système de fichiers.

Contrôle du systèmes de fichiers

L'utilitaire **dumpe2fs** permet d'afficher des informations du système de fichiers d'une partition:

Type de système de fichiers,
Nombre d'inodes,
Date de création du système de fichiers,
Date du dernier montage,
Date de dernière écriture,
.....etc

Exemple : **dumpe2fs -b /dev/sda1**

→ Affiche la liste des blocs défectueux

Modification d'un Système de fichier

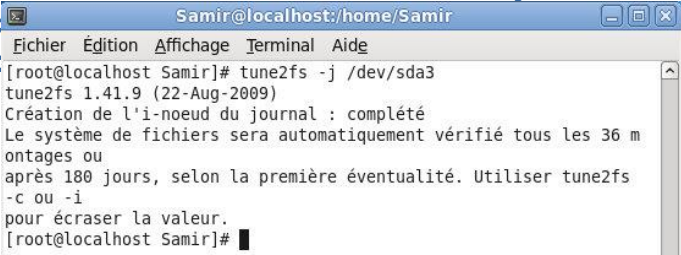
Pour modifier un système de fichiers étendu, il faut utiliser **tune2fs**.

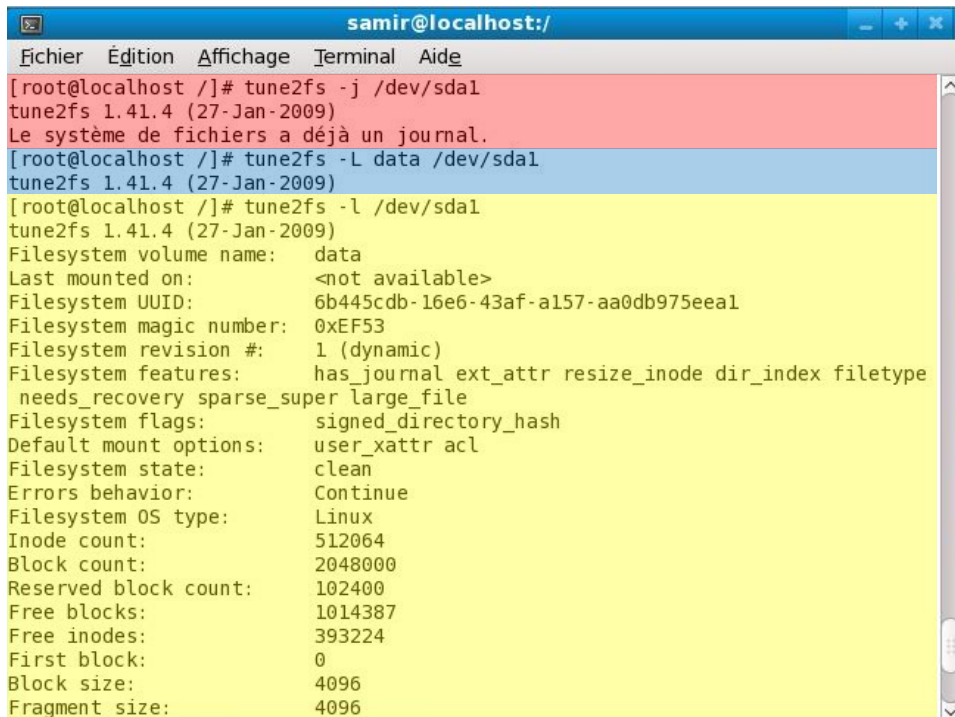
tune2fs [options] device

Principales options:

-l: Liste le contenu du *superblock* d'un système de fichiers.

-L: Donne un label au volume d'un système de fichiers.

-j:
A terminal window titled 'Samir@localhost:/home/Samir' showing the command 'tune2fs -j /dev/sda3' and its output. The output indicates that the journal creation is complete and that the filesystem will be automatically verified every 36 mountages or after 180 days. It also shows the command to force a write of the superblock using '-c' or '-i'.

A terminal window titled 'samir@localhost:/' showing the command 'tune2fs -l /dev/sda1' and its output. The output displays various filesystem statistics and features for the 'data' volume. The text is highlighted in yellow in the original image.

```
[root@localhost /]# tune2fs -j /dev/sda1
tune2fs 1.41.4 (27-Jan-2009)
Le système de fichiers a déjà un journal.
[root@localhost /]# tune2fs -L data /dev/sda1
tune2fs 1.41.4 (27-Jan-2009)
[root@localhost /]# tune2fs -l /dev/sda1
tune2fs 1.41.4 (27-Jan-2009)
Filesystem volume name:   data
Last mounted on:         <not available>
Filesystem UUID:         6b445cdb-16e6-43af-a157-aa0db975eea1
Filesystem magic number: 0xEF53
Filesystem revision #:    1 (dynamic)
Filesystem features:      has_journal ext_attr resize_inode dir_index filetype
                          needs_recovery sparse_super large_file
Filesystem flags:         signed_directory_hash
Default mount options:    user_xattr acl
Filesystem state:         clean
Errors behavior:          Continue
Filesystem OS type:       Linux
Inode count:              512064
Block count:              2048000
Reserved block count:     102400
Free blocks:              1014387
Free inodes:              393224
First block:              0
Block size:               4096
Fragment size:            4096
```

fscck

Si le système de fichier est endommagé ou corrompu, l'utilitaire **fscck** est utilisé pour vérifier et corriger le système

L'option **-i** permet de corriger les fichier corrompu

Remarque: pour des raisons de sécurité des données présentes sur le disque, il est fortement déconseillé de lancer un **fscck** sur une partition montée

```
Samir@localhost:/home/Samir
Fichier Édition Affichage Terminal Aide
[root@localhost Samir]# fscck /dev/sda3
fscck de util-linux-ng 2.16
e2fscck 1.41.9 (22-Aug-2009)
/dev/sda3 : propre, 11/335216 fichiers, 23671/1340595 blocs
[root@localhost Samir]#
```

```
samir@localhost:/
Fichier Édition Affichage Terminal Aide
[root@localhost /]# fscck /dev/sda1
fscck 1.41.4 (27-Jan-2009)
e2fscck 1.41.4 (27-Jan-2009)
/dev/sda1 est monté.

AVERTISSEMENT !!! L'exécution d'e2fscck sur un système de fichiers monté
peut causer des dommages SEVERES au système de fichiers.

Souhaitez-vous réellement continuer (o/n)? non

vérification stoppée.
[root@localhost /]#
```

monter et démonter un système de fichiers

manière temporaire

La commande **mount** permet de monter un système de fichiers (partition, lecteur de disquette, DVD ...).

La commande **mount -a** permet de monter tous les systèmes de fichiers déclarés dans le fichier **/etc/fstab**.

La syntaxe : **mount -t** système_de_fichier **-O** option point_de_montage

Exemple montage : **mount -t vfat /dev/hda1 -o ro /Dos**
mount /dev/hda1 /Dos
mount /dev/cdrom /cdrom

Exemple démontage : **umount /Dos**

On peut préciser l'option **-o** suivie **ro** pour monter un périphérique en lecture seule, tel qu'un CD-ROM ou une disquette protégée en écriture par exemple.

```

Samir@localhost:/home/Samir
Fichier Édition Affichage Terminal Aide
[root@localhost Samir]# mount /dev/sda3 /home
[root@localhost Samir]# mount
/dev/sda1 on / type ext2 (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")
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
gvfs-fuse-daemon on /home/Samir/.gvfs type fuse.gvfs-fuse-daemon (rw,nosuid,nodev,user=Samir)
/dev/sr0 on /media/Fedora 12 i386 DVD type iso9660 (ro,nosuid,nodev,uhelper=devkit,uid=500,gid=500,iocharset=utf
8,mode=0400,dmode=0500)
/dev/sda3 on /home type ext3 (rw)
[root@localhost Samir]# umount /home
[root@localhost Samir]# mount
/dev/sda1 on / type ext2 (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")
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
gvfs-fuse-daemon on /home/Samir/.gvfs type fuse.gvfs-fuse-daemon (rw,nosuid,nodev,user=Samir)
/dev/sr0 on /media/Fedora 12 i386 DVD type iso9660 (ro,nosuid,nodev,uhelper=devkit,uid=500,gid=500,iocharset=utf
8,mode=0400,dmode=0500)
[root@localhost Samir]#

```

Montage permanent /etc/fstab

Le fichier **/etc/fstab** contient tous les fichiers systèmes et les informations relatives qui seront utilisées pour monter un lecteur au démarrage du système

Exemple de fichier fstab :

#device	mount point	filesystem	options	dump	fsck
/dev/hda1	/	ext3	defaults	1	1
LABEL=/home	/home	reiserfs	defaults	0	0
/dev/hdb5	/windows	vfat	uid=500,umask=0	0	0
/dev/hdc	/mnt/cdrom	iso9660	users,noauto	0	0

L'emplacement physique du système de fichiers

Le point de montage
le répertoire doit déjà exister

Le type de système de fichiers

(ro, rw, noauto, user, exec ...)

1 = Sauvegarde de la partition

Ordre de vérification
0 = ne pas vérifier

Les option de montage de /etc/fstab

ro pour monter le système de fichiers en **lecture seule**,

rw pour monter le système de fichiers en **lecture-écriture**,

noauto pour que le système de fichiers ne soit pas monté au démarrage (option contraire : **auto**),

user pour qu'un simple utilisateur puisse monter et démonter le système de fichiers et pas seulement le root (option contraire : **nouser**),

uid, **gid** et **umask** pour définir des permissions pour l'ensemble du système de fichiers (pour les systèmes défectueux comme fat ou ntfs),

defaults pour les options par défaut (notamment rw, auto et nouser),
et enfin **sw** pour les systèmes de *swap*.